TELKOMNIKA, Vol.15, No.4, December 2017, pp. 1955~1970
ISSN: 1693-6930, accredited **A** by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v15i4.6326

Hierarchy Process Mining from Multi-source Logs

Riyanarto Sarno*, Yutika Amelia Effendi
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo, Jalan Raya ITS, Surabaya, Jawa Timur 60111, (031) 5994251, Indonesia
*e-mail: riyanarto@if.its.ac.id

Abstract

Nowadays, large-scale business processes is growing rapidly; in this regards process mining is required to discover and enhance business processes in different departments of an organization. A process mining algorithm can generally discover the process model of an organization without considering the detailed process models of the departments, and the relationship among departments. The exchange of messages among departments can produce asynchronous activities among department process models. The event logs from departments can be considered as multi-source logs, which cause difficulties in mining the process model. Discovering process models from multi-source logs is still in the state of the art, therefore this paper proposes a hierarchy high-to-low process mining approach to discover the process model from a complex multi-source and heterogeneous event logs collected from distributed departments. The proposed method involves three steps; i.e. firstly a high level process model is developed; secondly a separate low level process model is discovered from multi-source logs; finally the Petri net refinement operation is used to integrate the discovered process models. The refinement operation replaced the abctract transitions of a high level process model with the corresponding low level process models. Multi-source event logs from several departments of a yarn manufacturing were used in the computational study, and the results showed that the proposed method combined with the modified time-based heuristics miner could discover a correct parallel process business model containing XOR, AND, and OR relations.

Keywords*: process mining, process discovery, multi source log, petri net, refinement operation, time-based interval, time-based heuristics miner, double timestamp event log*

1. Introduction

In case of business process, optimized business processes is important to reduce costs. A business process modeling is required in analyzing the varieties of business process [1]. Several large-scale information systems such as Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), and Workflow Management System (WFMS), store information of business processes in the event log which are used for decision making in the workflow of business process [2]. Event logs contain the information for all executed activities of a workflow and one of data which can be analyzed in order to get a complete process model [3]. They give the complete information about when and who is actor which performed which activity, this matter contains very important information in terms of execution of business processes [4]. To analyze the business processes based on real executions from event log data, process mining technique is generally used [5].

Currently, the world of industry begins to give interest in the development of mining process in regulating the flow of its business processes. Enterprise information systems are constructed from many components so that they are increasingly complex [6]. For example ERP system, such as SAP and Odoo, there are dozens of event logs in different department of organizations for process mining [7]. Therefore the mining process is appropriate to be used for analyzing cross-organizational workflows and to obtain the process model from event logs from several departments in an organization. A process mining algorithm can discover the process model of an organization without considering the detailed process models of the departments and the relationship among departments. The exchange of messages among departments can produce asynchronous activities among department process models. The event logs from departments can be considered as multi-source logs, which cause difficulties in mining the process model. Discovering process models from multi-source logs is still in the state of the art, therefore this research proposes a hierarchy high-to-low process mining approach to discover

Received August 5, 2017; Revised October 18, 2017; Accepted November 20, 2017

the process model from a complex multi-source and heterogeneous event logs collected from distributed departments. The proposed method contains three steps: firstly a high level process model is developed; secondly a separate low level process model is discovered from multi-source logs; then the Petri net refinement operation is used to integrate the discovered process models.

We explore the distributed process mining from heterogeneous logs using a low-to-high process mining approach in [8]. This work first separately obtain the process models of each organization, and then integrate these models using four coordination patterns to obtain the business process model. This work assumed that those distributed servers are the same, i.e. they are functionally equal with each other [9]. After we obtain the integrated process model, we use modified heuristics miner algorithm to discover process model. In this paper, we use Modified Time-Based Heuristics Miner (MTBHM) algorithm [10]. This modified algorithm is similar to the original Heuristics Miner; whereas the difference is in how to mine the parallel activities; the use of direct and indirect activities. Heuristics miner uses direct activities, but MTBHM algorithm uses both of them, direct and indirect activities [11]. Our experiment results show that the MTBHM algorithm can discover parallel relation XOR, OR and AND, so that we can produce complete business process model. Next, we present related work in section II. We introduce high-to-low process mining and modified time-based heuristics miner in section III. In section IV introduces a yarn manufacturing business process as a typical case to illustrate our high-to-low process mining and modified time-based heuristics miner approaches. Finally, we concludes the paper in section V.

2. Related Works

In the related works, we review works that relate to our proposed method in business process mining.

2.1. Business Process Model

A set of linked activities which is produced for specific service is the definition of the business process [10]. Business process has information about where and when the activities are executed, input and output of activity, initial condition before activity is executed and final condition after activity is executed [12]. The characteristics of the business process itself are as follows:

1. Have a specific purpose
2. Have a specific input
3. Have a specific output
4. Utilize resource
5. Have an activity that can be executed in a certain order
6. It can involve more than one organization

Business process model can represents business process correctly. There are a lot of ways to represent business process model, such as UML, Causal Net, BPEL, BPMN, EPC, PNML, etc. Each type of model has different characteristic, such as Petri Net uses token to connect the activity in the business process model, while in Causal Net, the activity can be connected directly.

2.2. Event Log

In the case of process discovery, there is no prior process model. To discover the process model, we use an event log as the beginning point of business process model analysis. Event log is a basic resource that helps provide information about business process activities. Event log will contain many information, it depends on each organizational information [10]. Generally, event log is divided into three main parts i.e. Case, Trace, and Activity.

a. Case and Trace

A case is a record of events related to a single executed process instance. Case can be described as the production process of one stuff. Whereas trace records sequence of events that belong to the same case [10]. For example, there is an event log N:

$$L = \begin{bmatrix} \langle apple, ball, cup\rangle^{45}, \langle doll, ball, cup\rangle^{42}, \\ \langle apple, ball, doll\rangle^{38}, \langle doll, apple, cup\rangle^{22} \end{bmatrix}$$

We get the data about traces and cases in the event log L:
1. Contains 4 traces .i.e. (apple, ball, cup), (doll, ball, cup), (apple, ball, doll), (doll, apple, cup)
2. Contains 147 cases .i.e. (apple, ball, cup) is executed 45 time, (doll, ball, cup) is executed 42 time, (apple, ball, doll) is executed 38 time, and (doll, apple, cup) is executed 22 time.

b. Activity

Activity is part of event log which presents sub of production process of a product. For example, there are four activities in the event log L= {apple, ball, cup, doll}.

2.3. Mining Parallel Activity using Modified Time-based Heuristics Miner Algorithm

The MTBHM algorithm uses double timestamp event log and involves the direct and indirect activities contained in the event log. This algorithm can also discover the process models which contain parallel relations XOR, AND, and OR [10].

2.4. Parallel Relation XOR

XOR is one of types of parallel relation. XOR is divided into XOR-split and XOR-join. If there are three activities in the event log and only one of them will be executed at the same time, so this relation is categorized as XOR [10]. All of discovery algorithms can model the parallel relation XOR [3].

2.5. Parallel Relation AND

AND is one of types of parallel relation. AND is divided into AND-split and AND-join. If there are three activities in the event log and three of them will be executed all at the same time, so this relation is categorized as AND [10]. All of discovery algorithms can model the parallel relation AND [3].

2.6. Parallel Relation OR

OR is one of types of parallel relation. OR is divided into OR-split and OR-join. If there are three activities in the event log and two of them will be executed all at the same time, so this relation is categorized as OR [10]. Some discovery algorithms are not able to discover the OR correctly. Some of them discover this parallel relation as AND-split and the others discover it as XOR-split [8].

3. Research Method

In this section, we explain about the proposed method, including framework for high-to-low process mining, an introduction example for the event log, and formal definition of the event logs.

3.1. Framework for High-to-Low Process Mining

a. Recording Event Logs. While a workflow system runs on several distributed servers, each server can record the event logs for each activity and store them into a database of event logs. Such event logs collected from multi-source servers are used for our high-to-low process mining. An example of event logs will be presented in the following subsection.

b. Developing High Level Process Model. Using the collected event logs, our proposed method can discover the high level process model of the workflow. The results of this step are in extended form of Petri nets with abstract transitions.

c. Discovering Low Level Process Model from Event Logs. Using the collected event logs, our low level process mining method aims to discover the detailed process model for each abstract procedure in the high level. The obtained low level process models are in the standard form of Petri nets without any abstract transitions.

d. Integrating Process Model based on Petri Net Refinement Operation. After obtaining both the high level process model and the low level process model from the distributed event logs, the Petri net refinement operation is used to refine the abstract transitions with its corresponding low level models to obtain the integrated model of the whole workflow system.

3.2. An Introduction Example for the Event Log

During the execution of business process, information of each activity and abstract procedure are recorded. We have the following explanations for the example event log: (1) There are two events in the segment which records information about one activity A_3 and one abstract procedure PA_1 (recorded as T1 in the event log); (2) The event log of one activity records the case ID, activity, time stamp, operator, input message and output message. For example, the operator of A_3 is the Consigner, the start time of activity A_3 is 09:13 April 14, and the end time is 09:44 April 14. The input message record of A_3 is empty, which means that the execution of A_3 does not need any message from other partners, and its output message is pm1; (3) there are some differences between the event log of one activity and that of an abstract procedure. Obviously, the log of an abstract procedure also records the case ID, activity, time stamp, operator, input message, and output message. In addition, the messages read and written during its execution are also recorded. For example, the messages read of PA_1 are pm_3, pm_6, and pm_7, and its write messages are pm_2, pm_5, pm_8, which means that during its execution PA_1 receives messages pm_3, pm_6, and pm_7 from other partners and sends messages pm_2, pm_5, and pm_8 to others.

3.3. Formal Definition of the Event Logs

We present the formal definitions of the event logs used in this proposed method:

a. Event log of an activity

Alog=(CaseID, A_i, t_s, t_e, operator, Input Message, Output Message), where (1) CaseID indicates the case which A_i runs in; (2) A_i is the name of activity; (3) t_s is the start running time of activity A_i; (4) t_e is the end running time of activity A_i; (5) operator is the operator ID of A_i; (6) InputMessage is the input message set to execute A_i; (7) OutputMessage is the output message set when finishing A_i.

b. Event log of an abstract procedure

Plog=(CaseID, PA_i, t_s, t_e, operator, InputMessage, Output Message, Read Message, Write Message), where (1) CaseID indicates the case which PA_i runs in; (2) PA_i is the name of abstract procedure; (3) t_s is the start running time of activity PA_i; (4) t_e is the end running time of activity PA_i; (5) operator is the operator ID of PA_i; (6) InputMessage is the input message set to execute PA_i; (7) Output Message is the output message set when finishing PA_i; (8) Read Message is the read message set during the execution of PA_i; and (9) Write Message is the write message set during the execution of PA_i.

In the following, both activity and abstract procedure are called by a joint name as the assignment, which is formalized as ASLog=(CaseID, AS_i, t_s, t_e, Operator, Required Message, Sent Message). It is noting that (1) for an activity, the Required Message and the SentMessage are same as its Input Message and Output Message; and (2) for an abstract procedure, we have Required Message=Input Message ∪ Read Message and Sent Message=Output Message ∪ Write Message. For the rest of this paper, we use the term assignment synonymously with activity and abstract procedure. In order to obtain complete and correct process models for business process of the organization, the followings are the steps to be performed:

3.3.1. Step 1. Develop High Level Process Model

High level process model is mainly composed of two functional components. First, to obtain assignment dependency relations and second, to take these relations as inputs to construct the final high level process model. High Level Process Model: $\sum HPM$ is a kind of Petri nets extended with abstract transitions, i.e. there are two kinds of transitions; to represent the normal activities and to represent the abstract procedures. To differ from the normal transitions, a double rectangle is used to represent an abstract transition. For example, a high level process model in Petri net form is presented in Figure 1, sub 1 means an abstract transition.

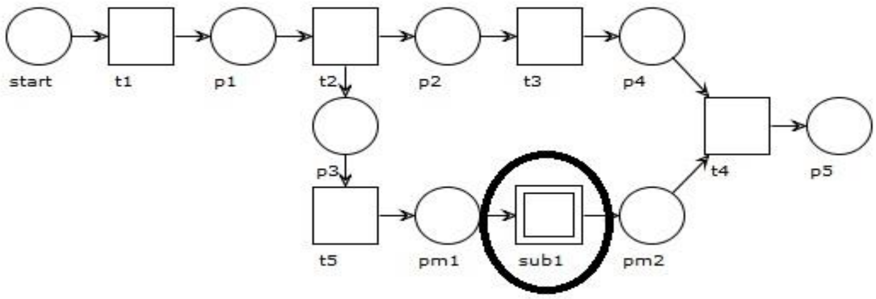

Figure 1. An Example of High level Process Model

3.3.2. Step 2. Discover Low Level Process Model from Event Logs

To refine abstract procedures in a high level process model, we need the complete low level process models. This step is to mine the low level process model from its corresponding event logs. Low Level Process Model: $\sum LPM$ defined is different from a $\sum HPM$ as it does not contain any abstract transition. Therefore, its firing rule is the same as a standard form of Petri net. For example, a low level process model in Petri net form is shown in Figure 2.

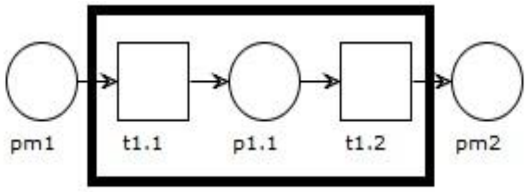

Figure 2. An Example of Low level Process Model

3.3.3. Step 3. Integrate Process Model based on Petri Net Refinement Operation

Process mining technology is used to separately discover the high level models and low level models. With the refinement operation, one abstract transition in the high level process model can be refined by its corresponding low level model. Refinement Operation: The refinement operation aims to refine the abstract transition by a $\sum LPM$. The structure of a $\sum LPM$ will replace the abstract transition and other parts in the original $\sum HPM$. The $\sum HPM$ after refinement is shown in Figure 3. Obviously, the $\sum HPM$ becomes a standard form after refining sub1 with the $\sum LPM$.

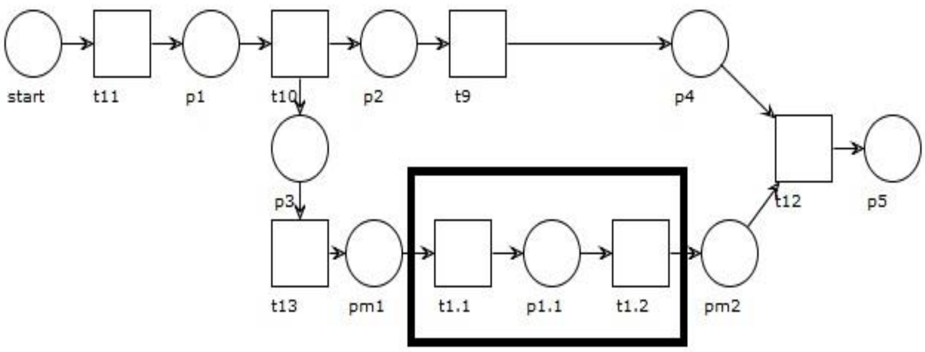

Figure 3. An Example Model after Refinement

3.3.4. Step 4. Mine using Modified Time-based Heuristics Miner Algorithm

There are three main steps to get a process model from an event log using MTBHM algorithm [10].

a. Mining Dependency Graph

Mining Dependency Graph will produce matrix dependency measure which is used to make dependency graph [13]. In Equation (1), we can determine the mining dependency graph:

$$A =>_w B = \left(\frac{|A>_w B|-|B>_w A|}{|A>_w B|+|B>_w A|+2|A||wB|+1}\right) \quad (1)$$

where the symbols mean:

$A =>_w B$: dependency from activity A to B

$|A >_w B|$: frequency of activity A and B which follow each other directly

$|B >_w A|$: frequency of activity B and A which follow each other directly

From Equation 1, we get the dependency measure matrix which will be used to make the dependency graph. However, there are three thresholds which relate to the dependency measure according to Weijters [14]-[15] and Cnudde [3]:

1. RBT (Relative to Best Threshold)

We can obtain the value of RBT using Equation (2).

$$RBT = Avg\ PDM - \left(\frac{SD\ PDM}{2}\right) \quad (2)$$

2. POT (Positive Observations Threshold)

This threshold aims to obtain the minimum dependency of all activities in the event log.

3. DT (Dependency Measure)

This threshold aims to consider the edge which will be chosen and put into the process model. To determine the dependency threshold, we use the formula in Equation (3):

$$RBT = Avg\ PDM - SD\ PDM \quad (3)$$

b. Checking Short Loops

In process mining, there is a condition that one activity is executed multiple times in the event log. This condition is well known as a loop. Short loop is divided into two types; LOL (length of one loop) and LTL (length of two loop). The formula which is used to calculate the LOL is in Equation (4).

$$A =>_w A = \left(\frac{|A>_w A|}{\max\{|A>_w X||\ X \in e\}}\right) \quad (4)$$

where the symbols mean:

$A =>_w A$: dependency of LOL

$|A >_w A|$: frequency of activity A and A which follow each other directly

$\max\{|A >_w X||\ X \in e\}$: frequency of activity A and X which follow each other directly, where A is activity in the event log

Meanwhile, using Equation (5) we can calculate the LTL.

$$A =>_{2w} B = \left(\frac{|A \gg_w B|+|B \gg_w A|}{|A \gg_w B|+|B \gg_w A|+1}\right) \quad (5)$$

where the symbols mean:

$A =>_{2w} B$: dependency of LTL

$|A \gg_w B|$: frequency of activity ABA

$|B \gg_w A|$: frequency of activity BAB

c. Mining Parallel Activities

Mining Parallel aims to calculate the parallel relations in the event log. This method uses a double timestamp event log and formula in Equation (6).

$$A =>_w (B \wedge C) = \left(\frac{|B \geqslant_w C|+|C \geqslant_w B|+2|B||wC|}{|A \geqslant_w B|+|A \geqslant_w C|+|B \gg>not_w C|+|C \gg>not_w B|+1}\right) \quad (6)$$

where the symbols refer to:
$A =>_w B \wedge C$: parallel measure of activity B and C, where the split is in activity A
$|A \geqslant_w B|$: frequency of activity A and B which follow each other directly
$|C \geqslant_w B|$: frequency of activity C and B which follow each other directly
$|A \geqslant_w B|$: frequency of activity A and B which follow each other directly
$|A \geqslant_w C|$: frequency of activity A and C which follow each other directly
$|B||wC|$: parallel relation of activity B and C in the event log (counted per case ID)
$|B >>> not_w C|$: frequency of activity B and C which do not follow each other directly
$|C >>> not_w B|$: frequency of activity C and B which do not follow each other directly
After we apply the formula in Equation (6), we get the all relations both sequences and parallel, then we group the parallel relations into AND, OR or XOR.

$$Avg\ PDM = \frac{\sum_{i=1}^{n_e} e_i}{n_e} \quad (7)$$

$$Avg\ PM = \frac{\sum_{i=1}^{n_{PM}} PM_i}{n_{PM}} \quad (8)$$

The classification of XOR, OR, and AND are:

a. XOR

$$If\ Avg\ PM \leq Minimum\ PDM\ then\ XOR \quad (9)$$

b. OR

$$If\ Minimum\ PDM \leq Avg\ PM \leq Avg\ PDM\ then\ OR \quad (10)$$

c. AND

$$If\ Avg\ PDM \leq Avg\ PM\ then\ AND \quad (11)$$

where the symbols mean:
PDM : positive dependency measure
$Avg\ PDM$: average of PDM value
$Minimum\ PDM$: minimum value of PDM value
e_i : arc weight from dependency measure
n_e : total number of all edges
PM : parallel measure
$Avg\ PM$: average of PM value
n_{PM} : total number of PM value

4. Results and Analysis

A yarn manufacturing business process is used as a case to illustrate our high-to-low process mining approaches. Figure 4 shows high level process model of yarn manufacturing business process. Step 1 Table 1 shows part of running logs of the high level architecture that involves one running case, *Case*$_1$. According to Table I, RequiredMessages and SentMessages of each assignment can be obtained directly. Taking these running logs as input, the Pre-Set, Post-set, ReceivedMessage and SentMessage of each assignment are shown in Table 2. After executing Table 2 as input, the high level process model of yarn manufacturing business process is shown in Figure 5.

The result of high level mining is a high level process model with abstract transitions that are represented by transition A_i (i=1, 2, …, 9) and abstract transitions T_j (j=1,2,3). The detailed process about these three abstract procedures cannot be obtained at this stage. The meanings of message places are explained in Table 4.

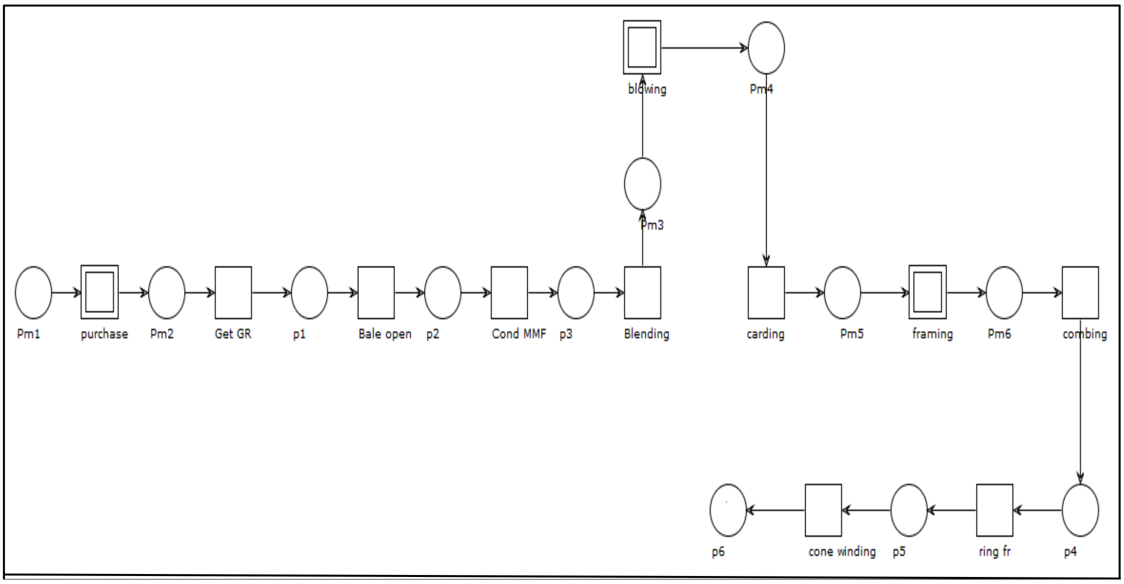

Figure 4. High-level Process Model of Yarn Manufacturing

Table 1. Part of the Running Logs of the High Level Process Model

Assignment	Activity	Pre-Set	Post-Set	Required Message	Sent Message
T1	Purchase	{}	{t1}	{Pm1,Pm2}	{Pm3,Pm4}
t1	J	{T1}	{t2}	{Pm3}	{Pm2}
t2	K	{t1}	{t3}	{}	{}
t3	L	{t2}	{t4}	{}	{}
t4	M	{t3}	{T2}	{}	{Pm5}
T2	Blowing	{t4}	{t5}	{Pm5}	{Pm6,Pm7}
t5	T	{T2}	{T3}	{Pm7}	{Pm8}
T3	Framing	{t5}	{t6}	{Pm6,Pm8,Pm11}	{Pm9,Pm10}
t6	AA	{T3}	{t7}	{Pm10,Pm9}	{}
t7	AB	{t6}	{t8}	()	{Pm11}
t8	AC	{t7}	{}	{}	{}

Step 2. Part of the running logs of the purchase department, the blowing department and the framing department are shown in Tables 5-7.

a. First, we consider Table 8, the Pre-Set, Post-set, ReceivedMessage and SentMessage of each assignment, shown in Table 8. Then, by executing Table 8, we can obtain the low level process model for the purchase department (T1), shown in Figure 6.
b. Second, we consider Table 9, the Pre-Set, Post-set, ReceivedMessage and SentMessage of each assignment, shown in Table 9. Then, by executing Table 9, we can obtain the low level process model for the blowing department (T2), shown in Figure 7.
c. Third, we consider Table 10, the Pre-Set, Post-set, ReceivedMessage and SentMessage of each assignment, shown in Table 10. Then, by executing Table 10, we can obtain the low level process model for the framing department (T3), shown in Figure 8.

Step 3. The low level process models in Figure 6, Figure 7 and Figure 8 are correspond with the three abstract procedures in Figure 5. Then the abstract transitions T1, T2 and T3 can be refined by the models in Figure 6, Figure 7 and Figure 8. The refined yarn manufacturing business process model is shown in Figure 9.

Step 4. Process Mining using Modified Time-based Heuristics Miner Algorithm. We use the event log in Table 3 for our experiments. The information of the event log consist of the case ID, the activities, start time and end time and the organisator.

Table 2. Pre-Set and Post-Set of Each Assignment in the High Level Process Model

Case ID	Activity	Time Stamp	End Time	Organisator	Operator	Required Message	Sent Message
PP1	T1	20/06/2014 08.32	20/06/2014 13.42	Purchase Dept	Operator 2	{Pm1,Pm2}	{Pm3,Pm4}
PP1	t1	20/06/2014 13.42	20/06/2014 23.41	Spinning Dept	Operator 1	{Pm3}	{Pm2}
PP1	t2	20/06/2014 23.41	21/06/2014 08.16	Spinning Dept	Operator 1	{}	{}
PP1	t3	21/06/2014 08.16	21/06/2014 10.46	Spinning Dept	Operator 1	{}	{}
PP1	t4	21/06/2014 10.46	21/06/2014 16.57	Spinning Dept	Operator 1	{}	{Pm5}
PP1	T2	21/06/2014 16.57	6/21/2014 19:15:56	Blowing Dept	Operator 3	{Pm5}	{Pm6,Pm7}
PP1	t5	21/06/2014 19.15	22/06/2014 10.51	Spinning Dept	Operator 1	{Pm7}	{Pm8}
PP1	T3	22/06/2014 10.51	22/06/2014 23.42	Framing Dept	Operator 4	{Pm6,Pm8,Pm11}	{Pm9,Pm10}
PP1	t6	22/06/2014 23.42	23/06/2014 04.48	Spinning Dept	Operator 1	{Pm10,Pm9}	{}
PP1	t7	23/06/2014 04.48	23/06/2014 16.44	Spinning Dept	Operator 1	{}	{Pm11}
PP1	t8	23/06/2014 16.44	23/06/2014 23.26	Spinning Dept	Operator 1	{}	{}

Table 3. Double Timestamp Event Log

Case ID	Activity	Time Stamp	End Time	Organisator
PP1	A	6/20/2014 8:32	6/20/2014 8:45	Purchase Dep
PP1	B	6/20/2014 8:45	6/20/2014 11:20	Purchase Dep
PP1	C	6/20/2014 11:20	6/21/2014 9:09	Purchase Dep
PP1	D	6/21/2014 9:09	6/21/2014 10:46	Purchase Dep
PP1	E	6/21/2014 10:46	6/21/2014 11:20	Purchase Dep
PP1	F	6/21/2014 11:20	6/20/2014 14:48	Purchase Dep
PP1	G	6/20/2014 14:48	6/20/2014 16:02	Purchase Dep
PP1	H	6/20/2014 16:02	6/20/2014 17:42	Purchase Dep
PP1	I	6/20/2014 17:42	6/20/2014 19:42	Purchase Dep
PP1	J	6/20/2014 19:42	6/20/2014 23:41	Spinning Dep
PP1	K	6/20/2014 23:41	6/21/2014 8:16	Spinning Dep
PP1	L	6/21/2014 8:16	6/21/2014 10:46	Spinning Dep
PP1	M	6/21/2014 10:46	6/21/2014 16:57	Spinning Dep
PP1	N	6/21/2014 16:57	6/21/2014 18:09	Blowing Dep
PP1	O	6/21/2014 18:09	6/21/2014 18:20	Blowing Dep
PP1	S	6/21/2014 18:20	6/21/2014 19:15	Blowing Dep
PP1	T	6/21/2014 19:15	6/22/2014 10:51	Spinning Dep
PP1	U	6/22/2014 10:51	6/22/2014 15:08	Framing Dep
PP1	V	6/22/2014 15:08	6/22/2014 19:25	Framing Dep
PP1	W	6/22/2014 19:25	6/22/2014 23:42	Framing Dep
PP1	X	6/22/2014 23:42	6/23/2014 1:42	Framing Dep
PP1	Y	6/23/2014 1:42	6/23/2014 2:16	Framing Dep
PP1	Z	6/23/2014 2:16	6/23/2014 3:17	Framing Dep
PP1	AA	6/23/2014 3:17	6/23/2014 4:48	Spinning Dep
PP1	AB	6/23/2014 4:48	6/23/2014 16:44	Spinning Dep
PP1	AC	6/23/2014 16:44	6/23/2014 23:26	Spinning Dep
PP6	A	7/8/2014 0:31	7/8/2014 7:27	Purchase Dep
PP6	D	7/8/2014 7:27	7/8/2014 7:58	Purchase Dep
PP6	E	7/8/2014 7:58	7/8/2014 8:27	Purchase Dep
PP6	F	7/8/2014 8:27	7/8/2014 8:58	Purchase Dep
PP6	G	7/8/2014 8:58	7/8/2014 9:27	Purchase Dep
PP6	B	7/8/2014 9:27	7/8/2014 9:58	Purchase Dep
PP6	C	7/8/2014 9:58	7/8/2014 10:27	Purchase Dep
PP6	H	7/8/2014 10:27	7/8/2014 10:58	Purchase Dep
PP6	I	7/8/2014 10:58	7/8/2014 11:27	Purchase Dep
PP6	J	7/8/2014 11:27	7/8/2014 12:06	Spinning Dep
PP6	K	7/8/2014 12:06	7/8/2014 20:02	Spinning Dep
PP6	L	7/8/2014 20:02	7/8/2014 22:50	Spinning Dep
PP6	M	7/8/2014 22:50	7/8/2014 23:25	Spinning Dep
PP6	P	7/8/2014 23:25	7/9/2014 2:13	Blowing Dep
PP6	O	7/9/2014 2:13	7/9/2014 3:55	Blowing Dep
PP6	R	7/9/2014 3:55	7/9/2014 5:37	Blowing Dep
PP6	S	7/9/2014 5:37	7/9/2014 7:20	Blowing Dep
PP6	T	7/9/2014 7:20	7/9/2014 14:38	Spinning Dep
PP6	X	7/9/2014 14:38	7/9/2014 21:02	Framing Dep
PP6	Y	7/9/2014 21:02	7/10/2014 3:18	Framing Dep
PP6	Z	7/10/2014 3:18	7/10/2014 7:07	Framing Dep
PP6	U	7/10/2014 7:07	7/10/2014 13:01	Framing Dep
PP6	V	7/10/2014 13:01	7/10/2014 16:03	Framing Dep
PP6	W	7/10/2014 16:03	7/10/2014 19:05	Framing Dep
PP6	AA	7/10/2014 19:05	7/10/2014 22:09	Spinning Dep
PP6	AB	7/10/2014 22:09	7/11/2014 7:55	Spinning Dep
PP6	AC	7/11/2014 7:55	7/11/2014 14:30	Spinning Dep

Table 4. Meaning of Each Message

Message	Meaning	Message	Meaning
Pm1	Accept Booking	Pm3	Good Preparation notice
Pm2	Goods Arrival Notice	Pm4	Deliver Goods
Pm5	Deliver Clumps of Cotton Fiber	Pm9	Complete Lap Former for Spinning Process Notice
Pm6	Processed Rolls Lap Notice		
Pm7	Deliver Web Silver	Pm10	Deliver Bobbin Roving
Pm8	Deliver Sliver Can	Pm11	Bobbin Roving Verification notice

Table 5. Part of Running Logs of the Purchase Department

Case ID	Activity	Operator	Required Message	Sent Message
PP1	A1.1	Operator 2	{Pm1}	{}
PP1	A1.2	Operator 2	{}	{}
PP1	A1.3	Operator 2	{}	{Pm3}
PP1	A1.4	Operator 2	{}	{}
PP1	A1.5	Operator 2	{}	{}
PP1	A1.6	Operator 2	{}	{}
PP1	A1.7	Operator 2	{}	{}
PP1	A1.8	Operator 2	{Pm2}	{}
PP1	A1.9	Operator 2	{}	{Pm4}

Table 6. Part of Running Logs of the Blowing Department

Case ID	Activity	Operator	Required Message	Sent Message
PP1	A2.1	Operator 3	{Pm5}	{}
PP1	A2.2	Operator 3	{}	{}
PP1	A2.3	Operator 3	{}	{}
PP1	A2.4	Operator 3	{}	{}
PP1	A2.5	Operator 3	{}	{Pm6}
PP1	A2.6	Operator 3	{}	{Pm7}

Table 7. Part of Running Logs of the Framing Department

Case ID	Activity	Operator	Required Message	Sent Message
PP1	A3.1	Operator 4	{Pm8}	{}
PP1	A3.2	Operator 4	{}	{}
PP1	A3.3	Operator 4	{Pm6}	{Pm9}
PP1	A3.4	Operator 4	{}	{}
PP1	A3.5	Operator 4	{}	{}
PP1	A3.6	Operator 4	{Pm11}	{Pm10}

Table 8. Pre-Set and Post-Set of Each Assignment in the Purchase Department

Assignment	Meaning	Pre-Set	Post-Set	Required Message	Sent Message
A2.1	Opposing Spike	{}	{A2.2}	{Pm5}	{}
A2.2	Air Current Blowing	{A2.1}	{A2.3}	{}	{}
A2.3	Open Clumps of Paper	{A2.2}	{A2.4}	{}	{}
A2.4	Cleaning Fiber with Dirt	{A2.3}	{A2.5}	{}	{}
A2.5	Cleaning/ Separation Fibers with Dirt and Material to Make Rolls of Cloth	{A2.4}	{A2.6}	{}	{Pm6}
A2.6	Striking Cotton	{A2.5}	{}	{}	{Pm7}

Table 9. Pre-Set and Post-Set of Each Assignment in the Blowing Department

Assignment	Meaning	Pre-Set	Post-Set	Required Message	Sent Message
A3.1	Drawing Breaker	{}	{A3.2}	{Pm8}	{}
A3.2	Drawing Finisher	{A3.1}	{A3.3}	{}	{}
A3.3	Lap Former	{A3.2}	{A3.4}	{Pm6}	{Pm9}
A3.4	Drafting	{A3.3}	{A3.5}	{}	{}
A3.5	Twisting	{A3.4}	{A3.6}	{}	{}
A3.6	Winding	{A3.5}	{}	{Pm11}	{Pm10}

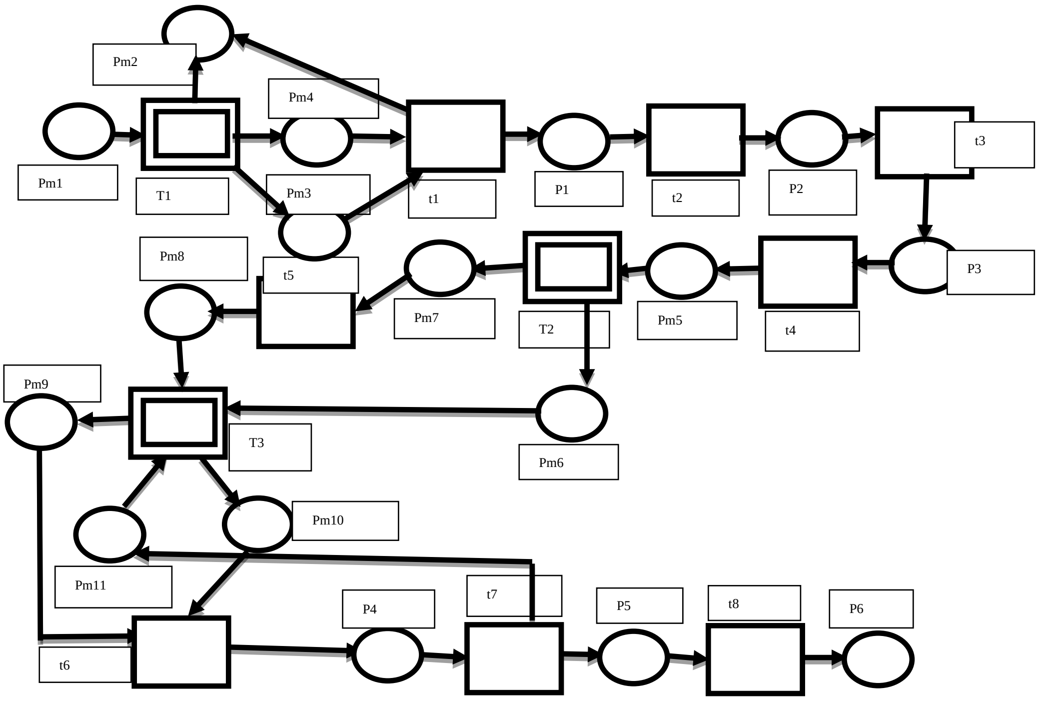

Figure 5. High-Level Model Mined for Yarn Manufacturing Business Process

a. Mining Dependency Graph

The event log in Table 10 generates the matrix in Table 11. We use Equation 1 to obtain the dependency measure. In Table 12, we obtain the dependency measure matrix. Next, we need to determine the thresholds using Equation (2) and (3). We get the value of RBT is 0.6645, POT is 3 and DT is 0.578 with average of PDM is 0.751 and SD PDM is 0.173.

Table 10. Pre-Set and Post-Set of Each Assignment in the Framing Department

Assignment	Meaning	Pre-Set	Post-Set	Required Message	Sent Message
A1.1	Sending PO Number	{}	{A1.2}	{Pm1}	{}
A1.2	Producing Good Orders Sup1	{A1.1}	{A1.3}	{}	{}
A1.3	Determining PPh and Giving Permission	{A1.2}	{A1.4}	{}	{Pm3}
A1.4	Paying PPh	{A1.3}	{A1.5}	{}	{}
A1.5	Producing Good Orders Sup2	{A1.4}	{A1.6}	{}	{}
A1.6	Packaging Good Orders	{A1.5}	{A1.7}	{}	{}
A1.7	Packaging Good Orders and Getting PPh Confirm	{A1.6}	{A1.8}	{}	{}
A1.8	Sending Good Orders	{A1.7}	{A1.9}	{Pm2}	{}
A1.9	Receiving Good Receive	{A1.8}	{}	{}	{Pm4}

Table 11. Direct Successor of Frequency Matrix

No	Value of DM from activity 1 to activity 2	Value of DM	No	Value of DM from activity 1 to activity 2	Value of DM
1	$A =>_w B$	0.75	21	$M =>_w N$	0.75
2	$A =>_w D$	0.8	22	$M =>_w P$	0.8
3	$B =>_w C$	0.83	23	$N =>_w O$	0.75
4	$C =>_w D$	0.66	24	$O =>_w S$	0.75
5	$C =>_w E$	0.5	25	$P =>_w Q$	0.8
6	$C =>_w H$	0.66	26	$Q =>_w R$	0.8
7	$D =>_w B$	0.5	27	$R =>_w S$	0.8
8	$D =>_w E$	0.75	28	$S =>_w T$	0.875
9	$D =>_w F$	0.66	29	$T =>_w U$	0.8
10	$E =>_w F$	0.37	30	$T =>_w X$	0.75

Table 11. Direct Successor of Frequency Matrix

No	Value of DM from activity 1 to activity 2	Value of DM	No	Value of DM from activity 1 to activity 2	Value of DM
11	$E =>_w G$	0.66	31	$U =>_w V$	0.83
12	$F =>_w E$	-0.37	32	$V =>_w W$	0.83
13	$F =>_w G$	0.8	33	$W =>_w X$	0.66
14	$G =>_w B$	0.5	34	$W =>_w AA$	0.75
15	$G =>_w I$	0.83	35	$X =>_w Y$	0.83
16	$H =>_w I$	0.875	36	$Y =>_w Z$	0.83
17	$I =>_w J$	0.875	37	$Z =>_w U$	0.5
18	$J =>_w K$	0.875	38	$Z =>_w AA$	0.8
19	$K =>_w L$	0.875	39	$AA =>_w AB$	0.875
20	$L =>_w M$	0.875	40	$AB =>_w AC$	0.875

Table 12. Dependency Measure Matrix

No	Frequency of activity 1 directly followed by activity 2	Frequency number	No	Frequency of activity 1 directly followed by activity 2	Frequency number
1	$\|A >_w B\|$	3	21	$\|M >_w N\|$	3
2	$\|A >_w D\|$	4	22	$\|M >_w P\|$	4
3	$\|B >_w C\|$	5	23	$\|N >_w O\|$	3
4	$\|C >_w D\|$	2	24	$\|O >_w S\|$	3
5	$\|C >_w E\|$	1	25	$\|P >_w Q\|$	4
6	$\|C >_w H\|$	2	26	$\|Q >_w R\|$	4
7	$\|D >_w B\|$	1	27	$\|R >_w S\|$	4
8	$\|D >_w E\|$	3	28	$\|S >_w T\|$	7
9	$\|D >_w F\|$	2	29	$\|T >_w U\|$	4
10	$\|E >_w F\|$	5	30	$\|T >_w X\|$	3
11	$\|E >_w G\|$	2	31	$\|U >_w V\|$	5
12	$\|F >_w E\|$	2	32	$\|V >_w W\|$	5
13	$\|F >_w G\|$	4	33	$\|W >_w X\|$	2
14	$\|G >_w B\|$	1	34	$\|W >_w AA\|$	3
15	$\|G >_w I\|$	5	35	$\|X >_w Y\|$	5
16	$\|H >_w I\|$	7	36	$\|Y >_w Z\|$	5
17	$\|I >_w J\|$	7	37	$\|Z >_w U\|$	1
18	$\|J >_w K\|$	7	38	$\|Z >_w AA\|$	4
19	$\|K >_w L\|$	7	39	$\|AA >_w AB\|$	7
20	$\|L >_w M\|$	7	40	$\|AB >_w AC\|$	7

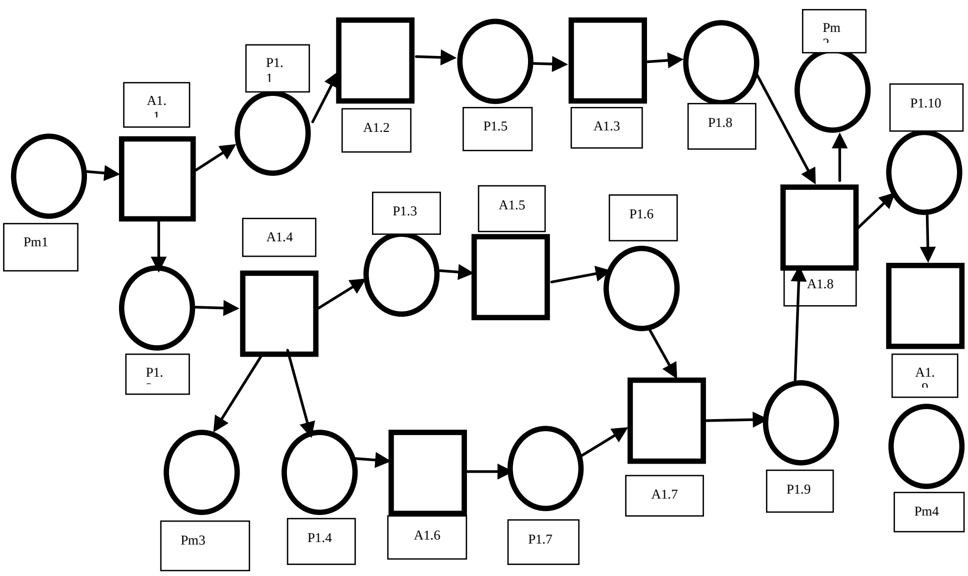

Figure 6. Process Model Mined for the Purchase Department

4.1. Checking Short Loop

Equation 4 and Equation 5 use to calculate matrix of short loops. However, this model does not have short loop because all of value in frequency short loop are zero.

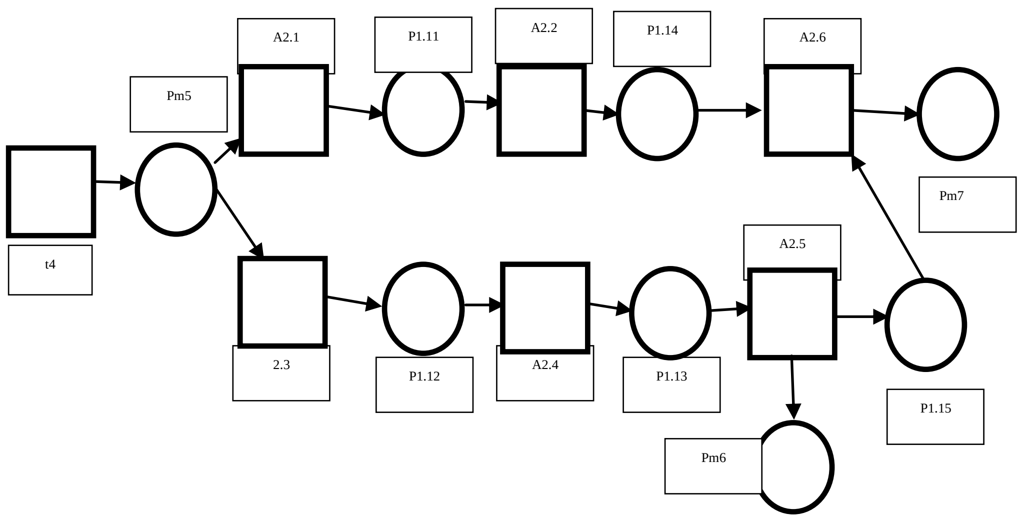

Figure 7. Process Model Mined for the Blowing Department

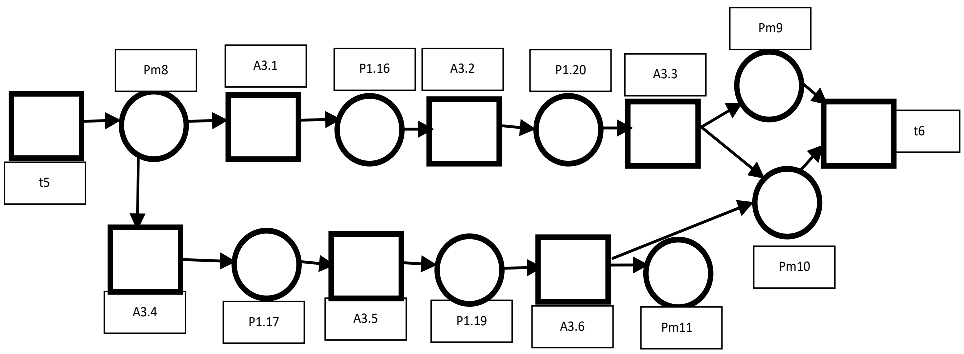

Figure 8. Process Model Mined for the Framing Department

4.2. Mining Parallel Activity

Causal matrix is the first thing to be done in this step. The causal matrix is shown in Table 13. We obtain the parallel activities, which are $a \rightarrow w\ b \wedge d$, $h \rightarrow w\ c \wedge g$, $d \rightarrow w\ e \wedge f$, $g \rightarrow w\ e \wedge f$, $m \rightarrow w\ p \wedge n$, $s \rightarrow w\ o \wedge r$, $t \rightarrow w\ u \wedge x$, $aa \rightarrow w\ x \wedge z$.

The parallel measure is calculated using Equation 6. $a \rightarrow w\ b \wedge d$ = 0.714, $h \rightarrow w\ c \wedge g$ = 0.5882, $d \rightarrow w\ e \wedge f$ = 0.833, $g \rightarrow w\ e \wedge f$ = 0.73846, $m \rightarrow w\ p \wedge n$ = 0.1333, $s \rightarrow w\ o \wedge r$ = 0.1333, $t \rightarrow w\ u \wedge x$ = 0.5833, $aa \rightarrow w\ x \wedge z$ = 0.53846.

Then, we need to check if the parallel measurement can be averaged using Equation 8 and 9. Then, we determine parallel relation XOR, OR, and AND using Equation 10, 11, 12. Using Equation 8 and 9, we obtain the average dependency measure is 0.751 and the minimum dependency measure is 0.37.

a. The average parallel measure $a \rightarrow w\ b \wedge d$ (0.714) and $h \rightarrow w\ c \wedge g$ (0.5882) is 0.651. This model uses OR.
b. The average parallel measure $d \rightarrow w\ e \wedge f$ (0.833) and $g \rightarrow w\ e \wedge f$ (0.73846) is 0.786. This model uses AND.
c. The average parallel measure $m \rightarrow w\ p \wedge n$ (0.1333) and $s \rightarrow w\ o \wedge r$ (0.1333) is 0.133. This model uses XOR.

d. The average parallel measure $t \rightarrow w\ u \wedge x$ (0.5833) and $aa \rightarrow w\ x \wedge z$ (0.53846) is 0.561. This model uses OR.

Hence, we can get the final model from Modified Time-based Heuristics Miner in Figure 10. The meanings of each activity code are: A=Sending PO Number, B=Producing Good Orders Sup1, C=Packaging Good Orders, D= Determining PPh and Giving Permission, E=Paying PPh, F=Producing Good Orders Sup2, G=Packaging Good Orders and Getting PPh Confirm, H=Sending Good Orders, I=Receiving Good Receive, J=Getting Good Receive, K=Bale Opening, L=Conditioning of MMP Fiber, M=Blending, N=Opposing Spike, O= Cleaning Fiber with Dirt, P= Air Current Blowing, Q= Open Clumps of Paper, R=Cleaning/ Separation Fibers with Dirt and Material to Make Rolls of Cloth, S=Striking cotton, T=Carding, U=Drawing Breaker, V=Drawing Finisher, W=Lap Former, X=Drafting, Y=Twisting, Z=Winding, AA=Combing, AB=Ring framing, AC=Cone winding.

Table 13. Causal Matrix

INPUT	ACTIVITY	OUTPUT	INPUT	ACTIVITY	OUTPUT
{}	A	B, D	N	O	S
A	B	C	P	Q	R
A	D	E, F	Q	R	S
B	C	H	S	T	U, X
D	E, F	G	T	U	V
E, F	G	H	U	V	W
H	I	J	T	X	Y
I	J	K	X	Y	Z
J	K	L	V	W	AA
K	L	M	Y	Z	AA
L	M	N, P	AA	AB	AC
M	N	O	AB	AC	{}
M	P	Q			

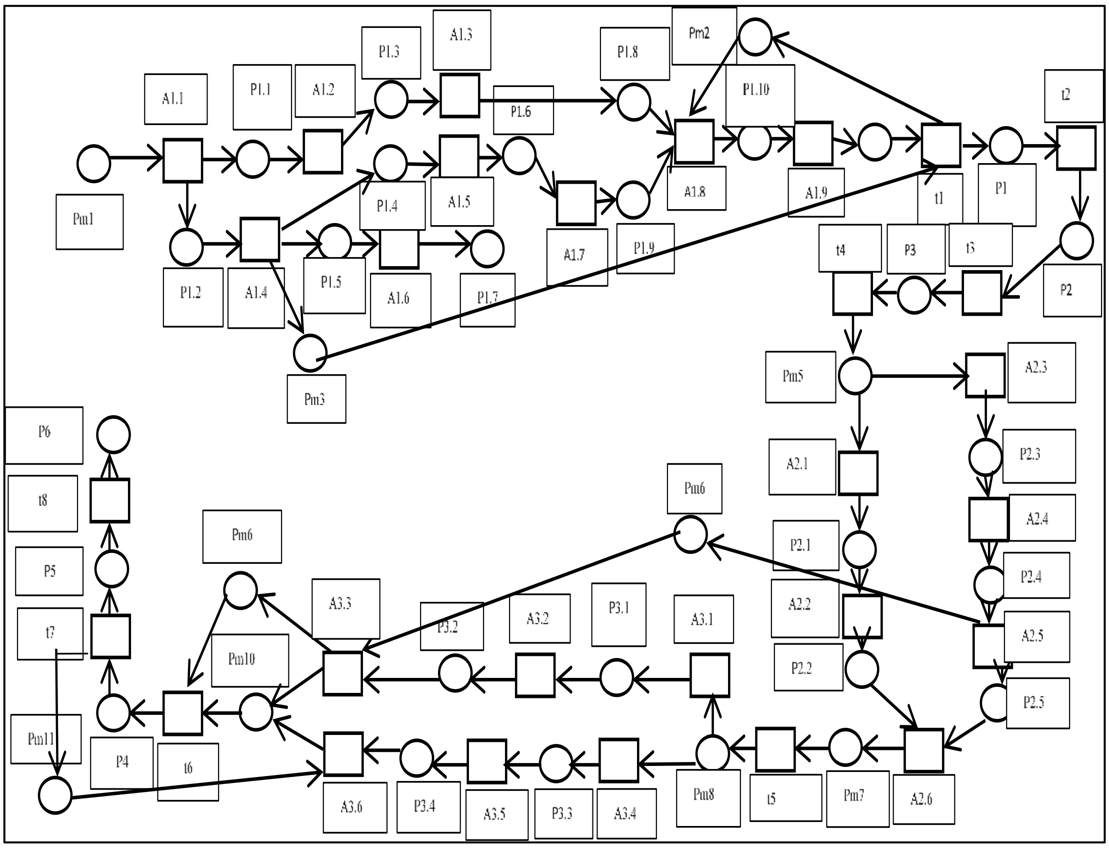

Figure 9. Refinement Model for the Yarn Manufacturing Business Process

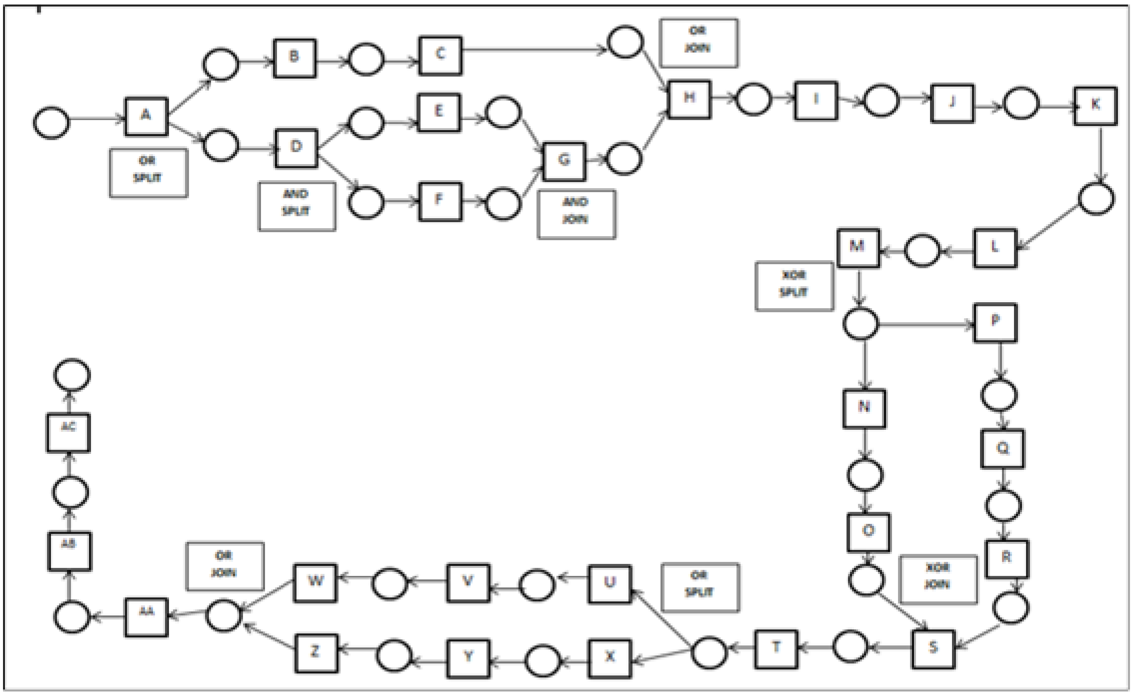

Figure 10. Final Model from Modified time-based Heuristics Miner

4. Conclusion

We have proposed a hierarchy process mining to discover the process model from a complex multi-source and heterogeneous event logs collected from distributed departments of a yarn manufacturing. The method developed a high level process model from multi-source logs, then discovered separately the low level process models from the event logs of the corresponding departments.

The Modified Time-based Heuristics Miner was employed to discover the process model containing sequence relations and parallel relations (XOR, AND, and OR). Further the Petri net refinement operation was used to integrate the high level process model with the corresponding low level process models. The refinement operation replaced the abstract transitions of a high level process model with the corresponding low level process models. Finally, a correct process model in the form of Petri net without any abstract transitions was discovered.

References

[1] Olatunde T. Baruwa, Miquel A. Piera. Identifying FMS repetitive patterns for efficient search-based scheduling algorithm: A colored Petri net approach. *Journal of Manufacturing Systems.* 2015; 35: 120-135.
[2] A. Burattin, A. Sperduti. *Heuristics Miner for Time Intervals.* ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. 2010; 41-46.
[3] S. De Cnudde, J. Claes, G. Poels. Improving the Quality of the Heuristicss Miner in Prom 6.2. *Expert System with Application*. 2014; 41: 7678-7690. http://dx.doi.org/10.1016/j.eswa.2014.05.055.
[4] A. Sanaa, S. B. Abid, A. Boulila, C. Messaoud, M. Boussaid, N. B. Fadhel. Modelling hydrochory effects on the Tunisian island populations of Pancratium maritimum L. using colored Petri Nets. *BioSystems.* 2015; 129: 19-24. http://dx.doi.org/10.1016/j.biosystems.2015.02.001
[5] R. Sarno, W. A. Wibowo, Kartini, F. Haryadita, Y. A. Effendi. Determining Model Using Non-Linear Heuristics Miner and Control-Flow Pattern. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. 2016; 14(1): 349-359. http://dx.doi.org/10.12928/telkomnika.v14i1.3257
[6] N. Y. Setiawan, R. Sarno. Multi-Criteria Decision Making for Selecting Semantic Web Service Considering Variability and Complexity Trade-Off. *Journal of Theoretical and Applied Information Technology*. 2016; 86 (2): 316-326.

[7] R. Sarno, H. Ginardi, E. W. Pamungkas, D. Sunaryono. *Clustering of ERP Business Process Fragments*. Proceedings IEEE International conference on computer, control, informatics, and its applications. 2013; 319-324.
[8] R. Sarno, E.W. Pamungkas, D. Sunaryono, Sarwosri. Business process composition based on Meta models. *International Seminar on Intelligent Technology and Its Applications (ISITIA)*. 2015; 315 – 318. http://doi.org/10.1109/isitia.2015.7219998
[9] R. Sarno, K. Sungkono. Hidden Markov Model for Process Mining of Parallel Business Processes. *International Review on Computers and Software (IRECOS)*. 2016; 11 (4): 290-300. http://dx.doi.org/10.15866/irecos.v11i4.8700
[10] R. Sarno, Y. A. Effendi, F. Haryadita. Modified Time-Based Heuristics Miner for Parallel Business Processes. *International Review on Computers and Software (IRECOS)*. 2016; 11 (3): 249-260. http://dx.doi.org/10.15866/irecos.v11i3.8717
[11] V. R. L. Shen, H. Y. Lai, A. F. Lai. The implementation of a smartphone-based fall detection system using a high-level fuzzy Petri Net. *Applied Soft Computing.* 2015; 26: 390-400. http://dx.doi.org/10.1016/j.asoc.2014.10.028
[12] R. A. Sutrisnowati, H. Bae, L. Dongha, K. Minsoo. *Process Model Discovery based on Activity Lifespan*. International Conference on Technology Innovation and Industrial Management Seoul. 2014; 137-156.
[13] W. M. P. van der Aalst. Process mining: discovery, conformance and enhancement of business processes. *Springer Science and Business Media*, 2011. http://dx.doi.org/10.1007/978-3-642-19345-3
[14] A. J. M. M. Weijters, J. T. S. Ribeiro. *Flexible Heuristics Miner (FHM).* IEEE Symposium on Computational Intelligence and Data Mining (CIDM). 2011. http://dx.doi.org/10.1109/cidm.2011.5949453
[15] A. J. M. M. Weijters, W. M. P. van der Aalst, A. A. de Medeiros. Process mining with the Heuristicss-miner algorithm. *Technische Universiteit Eindhoven, Tech. Rep. WP.* 2006; 166: 1-34.