

Time-based α^+ miner for modelling business processes using temporal pattern

Yutika Amelia Effendi¹, Riyanarto Sarno²

¹Information System Study Program, Department of Engineering, Universitas Airlangga, Indonesia

²Department of Informatics, Institut Teknologi Sepuluh Nopember, Indonesia

Article Info

Article history:

Received Mar 19, 2019

Revised Jun 30, 2019

Accepted Jul 18, 2019

Keywords:

Alpha+ miner

Business process model

Process discovery

Process mining

Temporal pattern

ABSTRACT

Business processes are implemented in an organization. When a business process is run, it generates event log. One type of event log is double timestamp event log. Double timestamp has the start and complete time of each activity executed in the business process and has a close relationship with temporal pattern. In this paper, seven types of temporal pattern between activities were presented as extended version of relations used in the double timestamp event log. Since the event log was not always executed in sequential way, therefore using temporal pattern, event log was divided into several small groups to mine the business process both sequential and parallel. Both temporal pattern and Time-based α^+ Miner algorithm were used to mine process model, determined sequential and parallel relations and then evaluated the process model using fitness value. This paper was focused on the advantages of temporal pattern implemented in Time-based α^+ Miner algorithm to mine business process. The results also clearly stated that the proposed method could present better result rather than that of original α^+ Miner algorithm.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Information System Study Program,

Department of Engineering,

Universitas Airlangga,

68 Dharmawangsa Dalam Selatan St., Airlangga, Gubeng,

Surabaya, Jawa Timur 60286, (031) 5033869, Indonesia.

Email: yutika.effendi@vokasi.unair.ac.id

1. INTRODUCTION

An organization has information systems which are used to collect, store and process data to gain information on certain things [1, 2]. They can also support management and decision making of business processes [3]. Systems in the organization will generate event log which gives knowledge about the activities of business processes run recently or few years ago [4, 5]. Because of the huge amount of event log, they cannot analyze directly. Therefore process mining technique, which is a combination of data mining and machine learning, is present to be able to handle problems in the event log [4, 6, 7]. In process mining, the technique related to mine the event log to get process models describing the real business processes is called process discovery [6]. Process models act as guidance for analyzing and verifying the performance of the existing business processes [8].

In process discovery, event log is one of the main input needed [9]. Data which must be contained in the event log are activities and timestamp [10]. Activities define the name of events or jobs which are executed in business processes [4]. To get the real time of business processes, we can use timestamp [11]. This research uses double timestamp event log because it has start and complete time. Double timestamp

event log has a close relationship with the temporal pattern, which is a pattern of event log that occurs from each activity performed [12, 13]. In reality, to obtain the business processes, event log is not always executed in sequential way from start to end [13]. It can be done by executing the activities in sequential and in parallel. To know the exact activities executed in sequential or in parallel, one easy way to find out is to pay attention to the timestamp of each activity [12]. The relation is defined as parallel if the timestamp overlaps or in terms of linking one or more activities with one or more activities executed afterwards, meanwhile sequential relation has the timestamp which does not overlap and about linking one activity and another activity executed afterwards [13]. Because of that matters, using temporal pattern, event log is divided into several small groups to make mining process models and their relations easy.

In this research, seven types of temporal pattern between activities are presented as extended version of relations used in the double timestamp event log. This paper proposes Time-based α^+ Miner (TBA+M) algorithm to mine a process model along with their relations based on the event log. Modified Time-based Alpha Miner (MTBAM) [13] was introduced to upgrade the α algorithm based on time interval. However, this algorithm cannot mine length one loop (L1L) and length two loop (L2L) in the event log to be modelled in a business process. When compared to original α^+ algorithm [14, 15], the proposed algorithm can detect overlapping processes using temporal pattern and identifies them as parallel processes by defining new thresholds and also can discover length one loop (L1L) and length two loop (L2L). α^+ algorithm is chosen because it is the upgraded version of α algorithm [16]. However, the disadvantages of α^+ algorithm are it cannot distinguish the differences between AND and OR in parallel relations and still use sequential relations between activities to obtain a business process. In other words, α^+ algorithm can only determine whether the relations of business processes are sequential and parallel (AND, XOR) in sequential way. To cover the shortcomings of the α^+ algorithm, the modification of α^+ algorithm into Time-based α^+ algorithm (TBA+M) is proposed in order to be able to discover process models, including discovering L1L and L2L, as well as their sequential and parallel relations (AND, OR, XOR) by determining new thresholds so that the business process cannot only be discovered by using sequential way. To sum up, this research focuses on using temporal pattern, modifying α^+ algorithm into TBA+M algorithm and using fitness value as evaluation of discovered process models.

This research paper consists of four sections. Our process discovery approach related to this research will be explained in section 2. The experimental result and discussion will be presented in section 3. Last, the conclusion will be concluded this research paper in section 4.

2. RESEARCH METHOD

In section 2, we present the event log, temporal pattern and integrated discovery approach for discovering business processes in this research.

2.1. Event log

We evaluate an event log of Production from PT. YM; a yarn production company in Jakarta, Indonesia. We call it as EL1. Table 1 shows part of EL1 which contains Case ID, list of activities, start time, and complete time. EL1 has 100 cases and 11 activities. The activities are *Cotton Bales*, *Blowing and Picking*, *Carding*, *Pre Carding*, *Super Lap*, *Combing*, *Drawing*, *Drawing II*, *Roving*, *Spinning*, *Winding*. In the end, Petri Net form [15, 17] is used to present the final result of business process in our experiment.

2.2. Temporal pattern

The model of the temporal pattern is based on the category of the timestamp, especially double timestamp to mine the process model. Rizka et al. introduced the use of time-based in process mining algorithm [18]. In her research, the definition of time-based in process mining algorithm consists of *before and meets*, *overlaps*, *contains*, *is-finished-by*, *equals*, and *starts*. In this research, we present seven types of temporal pattern between activities which are extended version of standard relations in business processes. If previously there were only sequential and parallel relations, after modification, we divide the sequential relations into before and meet, whereas parallel relation becomes *the_same_start_time*, *the_same_complete_time*, *overlap*, *contain* and *equal*. Table 2 shows process models for all temporal patterns.

Definition 2.1. There are *event log* (EL) and *trace* (σ) wherein $\sigma \in EL$. The types of temporal pattern between activity X and activity Y ; $X(X_s, X_f)$ and $Y(Y_s, Y_f)$, according to which $X, Y \in EL$ can be classified into *before*, *meet*, *the_same_start_time*, *the_same_complete_time*, *overlap*, *contain* and *equal*.

Definition 2.2. Based on *event log* (EL), *trace* (σ) and activities $X, Y \in EL$, the relations of process models both sequential and parallel can be differentiated as follows:

$$\text{NotRelated}, X \# Y \text{ iff } XOY \wedge XOY$$

Sequence, $X \rightarrow Y$ iff $X > Y \wedge YOX$
 Parallel, $X \parallel Y$ iff $X > Y \wedge Y > X \vee \{XOY \vee X@Y \vee X_fY \vee X \approx Y \vee X\rho Y\}$
 AND, $X \bullet Y$ iff $X \parallel Y$ and there is no $X \otimes Y$ in σ in EL
 OR, $X \oplus Y$ iff $X \parallel Y$ and there is $X \otimes Y$ in σ in EL
 XOR, $X \otimes Y$ iff $X \parallel Y$ if there is only X or Y in σ in EL

Table 1. The fragment of EL1

Case ID	List of Activities	Start Time	Complete Time	Case ID	List of Activities	Start Time	Complete Time
YM01	Cotton Bales	20/06/2017 09.09	20/06/2017 12.28	YM02	Cotton Bales	23/06/2017 04.48	23/06/2017 15.15
YM01	Blowing and Picking	20/06/2017 13.42	20/06/2017 19.12	YM02	Blowing and Picking	23/06/2017 16.44	23/06/2017 22.11
YM01	Carding	20/06/2017 23.41	21/06/2017 08.32	YM02	Pre Carding	23/06/2017 23.26	24/06/2017 07.48
YM01	Pre Carding	21/06/2017 08.16	21/06/2017 10.10	YM02	Carding	24/06/2017 04.19	24/06/2017 07.48
YM01	Super Lap	21/06/2017 10.46	21/06/2017 15.34	YM02	Pre Carding	24/06/2017 07.55	24/06/2017 10.10
YM01	Combing	21/06/2017 16.57	21/06/2017 18.09	YM02	Super Lap	24/06/2017 11.33	25/06/2017 23.02
YM01	Drawing	21/06/2017 18.09	21/06/2017 20.15	YM02	Combing	25/06/2017 00.30	25/06/2017 03.02
YM01	Drawing II	21/06/2017 19.15	22/06/2017 10.51	YM02	Roving	25/06/2017 03.16	25/06/2017 05.11
YM01	Roving	22/06/2017 10.51	22/06/2017 16.28	YM02	Spinning	25/06/2017 06.00	25/06/2017 08.25
YM01	Spinning	22/06/2017 18.28	22/06/2017 23.42	YM02	Winding	25/06/2017 09.03	25/06/2017 12.45
YM01	Winding	22/06/2017 00.22	23/06/2017 04.48				

Table 2. Fragment of double timestamp EL2

<p><i>before</i> : wherein $X > Y$ iff $X_f \leq Y_s$ <i>meet</i> : wherein $X > Y$ iff $X_f \leq Y_s$</p> <p>Process model:</p>	<p><i>the_same_complete_time</i> : wherein $X_f Y$ iff $X_f = Y_f \wedge X_s < Y_s \wedge Y_x < X_f$</p> <p>Process model:</p>
<p><i>overlap</i>: wherein XOY iff $X_f > Y_s \wedge X_f < Y_f$ <i>contain</i>: wherein $X@Y$ iff $X_s < Y_s \wedge Y_f > X_s \wedge X_f > Y_f$ <i>equal</i>: wherein $X \approx Y$ iff $X_s = Y_s \wedge X_f = Y_f$</p> <p>Process model:</p>	<p><i>the_same_start_time</i>: wherein $X\rho Y$ iff $X_s = Y_s \wedge X_f = Y_f$</p> <p>Process model:</p>

2.3. Integrated discovery approach

The discovery approach for the overall business process is shown in Figure 1. This method uses event log (EL) as an input process. For all cases, we need to determine the temporal pattern and create process models based on temporal pattern. From the process models, we can obtain one business process. The next step is to classify all relations of business process (sequential and parallel). To evaluate the business process, we calculate the fitness value [19-21] and compare it with the original algorithm. In the end, business process and its sequential and parallel (AND, OR, XOR) relations are obtained.

This research proposed how to discover process models using Time-based $\alpha+$ Miner algorithm and temporal pattern. When the process model is obtained, we calculate the fitness value to evaluate the process model obtained by TBA+M algorithm as our proposed method and the process model generated by original $\alpha+$ algorithm.

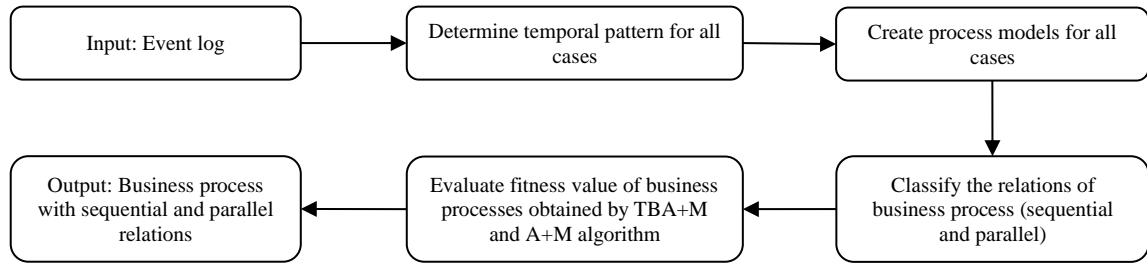


Figure 1. Process discovery approach

Step 1. Event log (EL) is the input for this approach. Generally, Case ID, List of Activities and Timestamp are needed when mining the process models.

Step 2. EL as input then is used to mine the process models using temporal pattern and Time-based $\alpha+$ Miner algorithm. The steps of process discovery using Time-based $\alpha+$ Miner algorithm:

Step 2.1. Define the temporal pattern for all cases in EL

Step 2.2. Generate process models based on the temporal pattern for all cases, including the relations of all activities. The steps to generate process models:

Step 2.2.1 Generate set of transition (T_L)

$$T_L = \{t \in T | \exists_{\sigma \in EL} t \in \sigma\} \quad (1)$$

Step 2.2.2 Generate set of input (T_I)

$$T_I = \{t \in T | \exists_{\sigma \in EL} t = first(\sigma)\} \quad (2)$$

Step 2.2.3 Generate set of output (T_o)

$$T_o = \{t \in T | \exists_{\sigma \in EL} t = last(\sigma)\} \quad (3)$$

Step 2.2.4 Generate the places (P_L)

$$P_L = \{p_{(A,B)} | (A, B) \in Y_L\} \cup \{I_L, O_L\} \quad (4)$$

Step 2.3. Determine all traces of EL which can be obtained from the process models. To get all the traces, the process models from Step 2.2. are stacked into one process model. The traces should include sequential and parallel relations.

Step 2.4. Obtain one business process as well as sequential and parallel relations

Step 2.5. Identify set of Loop (LL), namely length one loop (L1L) and length two loop (L2L) from EL

$$LL = \{t \in T_L | \exists_{\sigma} = t_1 t_2 \dots t_n \in W; i \in \{1, 2, \dots, n\} [t = t_{i-1} \wedge t = t_i]\} \quad (5)$$

Step 2.6. Specify the type of parallel relations: XOR, OR or AND XOR relation

$$\text{If Avg PPM} \leq \text{Min ASR in EL, then XOR} \quad (6)$$

OR relation

$$\text{If Min ASR} \leq \text{Avg PPM} \leq \text{Avg ASR in EL, then OR} \quad (7)$$

AND relation

$$\text{If Avg ASR} \leq \text{Avg PPM in EL, then AND} \quad (8)$$

where:

Min ASR : minimum value of all sequential relations in EL

Avg ASR : average of all sequential relations in EL

Avg PPM : average of parallel relations with the same parent activity in process model (the frequency both directly and indirectly of each activity followed by other activity)

Step 2.7. Display the final result of business process in Petri Net

$$\alpha(L) = (P_L, T_L, F_L) \quad (9)$$

where:

α : A business process in Petri Net form

P : place

In the business process, it is symbolized by circle shape and named as p1, p2, p3, etc.

T : transition

In the business process, it shows the activities with square shape

F : function

In the business process, function means an arc.

$F = \{P \times T\}$ means the arc connects the place (P) and transition (T)

Step 3. Evaluation of the process model. We calculate the fitness value after process model is obtained [19], and then we compare the results between process model obtained by Time-based $\alpha+$ Miner algorithm and process model generated from $\alpha+$ Miner algorithm. The fitness value is calculated using (10).

$$T_f = \frac{CaptCases}{LogCases} \quad (10)$$

where:

T_f : the fitness value

$CaptCases$: number of captured cases in EL

$LogCases$: number of cases in EL

3. RESULTS AND ANALYSIS

In this section, we do the experiment which will present the results that our proposed method can mine business processes using Time-based $\alpha+$ Miner algorithm and temporal pattern. As mentioned in section 2.1., a real-life event log is tested and analyzed in this experiment. Event log EL1 is in double timestamp as explained in Table 1.

3.1. Experimental results

From event log EL1 presented in Table 1, we have 100 cases and 11 activities to be implemented in this experiment. Based on all steps explained in section 2.3., we need to determine the input and the type of temporal pattern which are used for all cases. All cases in the event log EL1 have the same input which is activity *Cotton Bales*. The results of temporal pattern for two cases (YM01 and YM02) of EL1 are presented in Table 3. EL1 consists of *before*, *overlap*, *meet*, *the_same_complete_time*, *the_same_start_time* and *contain* for all 100 cases. However, the temporal pattern *equal* is not explained in this experiment. This is because *equal* does not occur in the event log EL1 after we analysis it based on the start time and complete time.

After all temporal patterns from the EL1 were obtained, we need to generate the process models including the relations of all activities based on the temporal pattern for all cases by following the results in Table 3 and the definition in section 2.2. In addition, we need to create the transition and place, and also determine the input and output for each case by following (1), (2), (3), (4).

Table 3. Temporal pattern for two cases

Case ID	List of Activities	Case ID	List of Activities
YM01	Cotton bales <i>before</i> Blowing and Picking	YM02	Cotton bales <i>before</i> Blowing and Picking
YM01	Blowing and Picking <i>before</i> Carding	YM02	Blowing and Picking <i>before</i> Pre Carding
YM01	Carding <i>overlap</i> Pre Carding	YM02	Pre Carding <i>the_same_complete_time</i> Carding
YM01	Pre Carding <i>before</i> Super Lap	YM02	Carding <i>before</i> Pre Carding
YM01	Super Lap <i>before</i> Combing	YM02	Pre Carding <i>before</i> Super Lap
YM01	Combing <i>before</i> Drawing	YM02	Super Lap <i>before</i> Combing
YM01	Drawing <i>overlap</i> Drawing II	YM02	Combing <i>before</i> Roving
YM01	Drawing II <i>meet</i> Roving	YM02	Roving <i>before</i> Spinning
YM01	Roving <i>before</i> Spinning	YM02	Spinning <i>before</i> Winding
YM01	Spinning <i>before</i> Winding		

Based on EL1, Case ID YM01 has activity *Cotton Bales* as input and activity *Spinning* as output as well as Case ID YM02. Transition presents the name of activity, meanwhile place is defined to connect two activities in a Petri Net form. In this experiment, we name the place as p1, p2, p3, etc. Figure 2 shows the process models based on Table 3 for Case ID YM01 and YM02. For the rest of 98 cases, we do the same

steps as Case ID YM01 and YM02. Figure 3 presents the discovered process models for Case ID YM01 and YM02 and also emphasizes that all cases of EL1 consist of sequential and parallel relations.

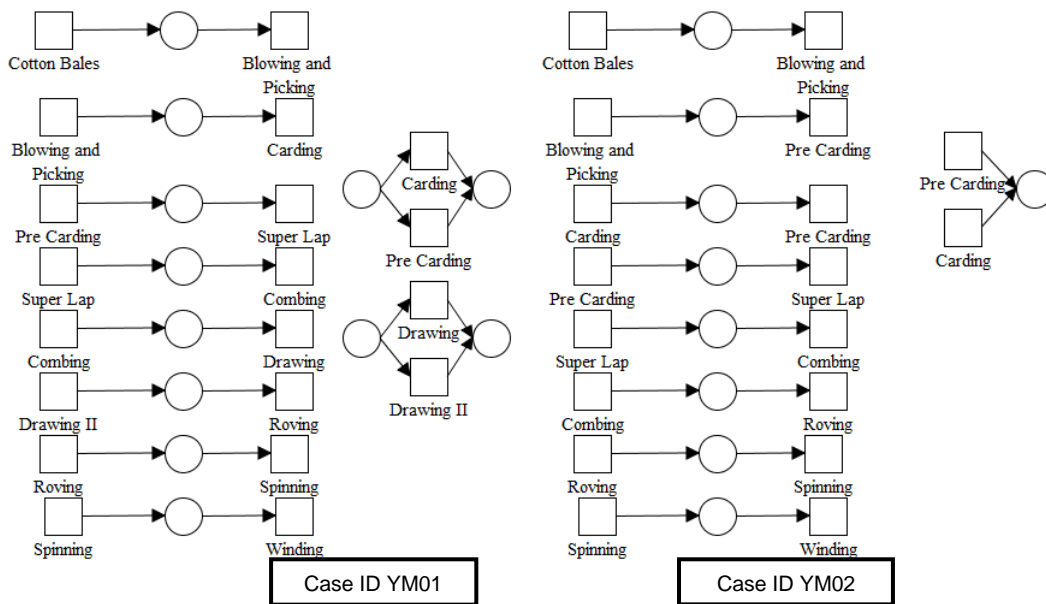


Figure 2. Process models generated from temporal pattern for case ID YM01 and YM02

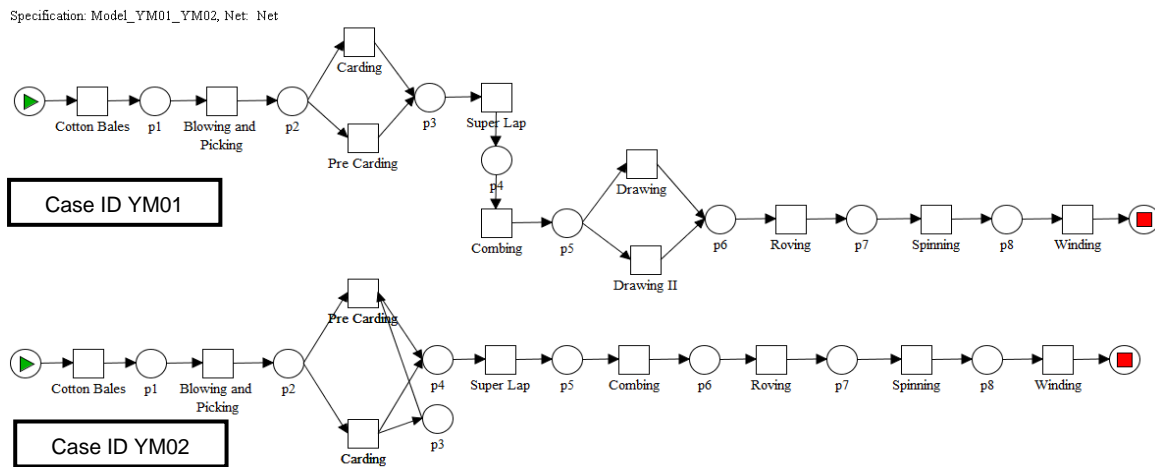


Figure 3. Process models for case ID YM01 and YM02

Next step, we also need to get all traces from all cases of EL1. Traces records sequence of activities which belong to the same case [22]. The easy way to get all traces, the two process models in Figure 3 and the other 98th cases are stacked into one business process. All traces of EL1 are presented in Table 4. The total number of traces in EL1 are 12 traces. Besides getting the number of traces, we also get the business process of EL1 which is the result of process discovery using TBA+M algorithm. The business process of EL1 is shown in Figure 4.

After the business process of EL1 including sequential and parallel relations was obtained, now we must identify whether this event log EL1 contains length one loop (L1L) and length two loop (L2L) by following (5). L1L and L2L mean one activity in one case may be executed multiple times. This condition is commonly known as loop [16, 23, 24]. We do analyzing for all 100 cases to find L1L and L2L, and the results show that there are some activities which executed multiple times in EL1. Table 5 presents the activities which are categorized as L1L and L2L in the process models.

Table 4. Traces of EL1

TRACES 1	TRACES 2	TRACES 3	TRACES 4	TRACES 5	TRACES 6
Cotton Bales	Cotton Bales	Cotton Bales	Cotton Bales	Cotton Bales	Cotton Bales
Blowing and Picking	Blowing and Picking	Blowing and Picking	Blowing and Picking	Blowing and Picking	Blowing and Picking
Carding	Pre Carding	Carding	Pre Carding	Carding	Pre Carding
Pre Carding	Carding	Pre Carding	Carding	Pre Carding	Carding
Super Lap	Super Lap	Super Lap	Super Lap	Super Lap	Super Lap
Combing	Combing	Combing	Combing	Combing	Combing
Drawing	Drawing	Drawing II	Drawing II	Roving	Roving
Drawing II	Drawing II	Drawing	Drawing	Spinning	Spinning
Roving	Roving	Roving	Roving	Winding	Winding
Spinning	Spinning	Spinning	Spinning		
Winding	Winding	Winding	Winding		
TRACES 7	TRACES 8	TRACES 9	TRACES 10	TRACES 11	TRACES 12
Cotton Bales	Cotton Bales	Cotton Bales	Cotton Bales	Cotton Bales	Cotton Bales
Blowing and Picking	Blowing and Picking	Blowing and Picking	Blowing and Picking	Blowing and Picking	Blowing and Picking
Carding	Pre Carding	Carding	Pre Carding	Pre Carding	Carding
Pre Carding	Carding	Pre Carding	Carding	Carding	Pre Carding
Super Lap	Super Lap	Super Lap	Super Lap	Pre Carding	Pre Carding
Combing	Combing	Combing	Combing	Super Lap	Super Lap
Drawing	Drawing	Drawing II	Drawing II	Combing	Combing
Roving	Roving	Roving	Roving	Roving	Drawing
Spinning	Spinning	Spinning	Spinning	Spinning	Roving
Winding	Winding	Winding	Winding	Winding	Spinning
					Winding

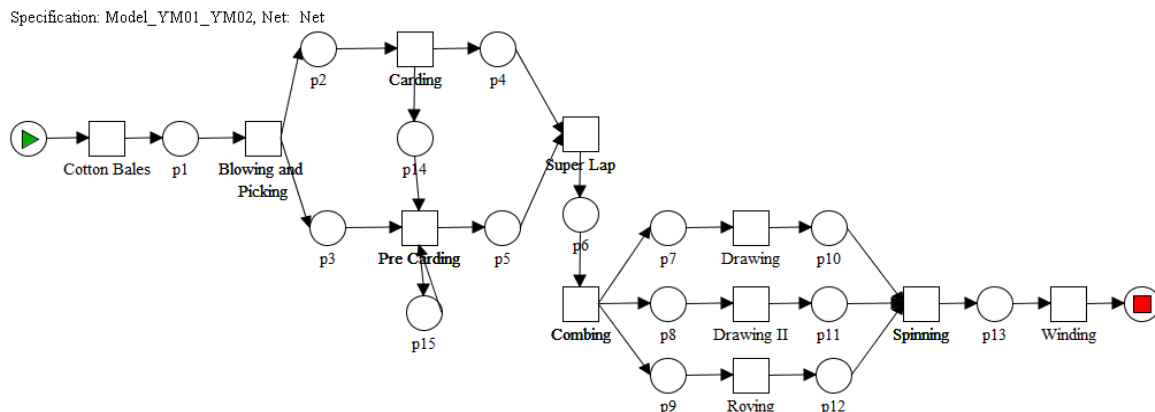


Figure 4. Business process of EL1 discovered by using the proposed method

Table 5. Activities executed multiple times in EL1

List of Activities	L1L	L2L
Pre Carding	Yes	-
Carding	-	Yes

3.2. Determining parallel relations of discovered business process

When discovered business process is obtained, the next step in our process discovery approach is to determine the type of parallel relations; XOR, OR, AND. We use (6), (7), and (8) to specify the type of parallel relations. The frequencies of sequential and parallel relations which are calculated from all cases of EL1 are activity *Cotton Bales* and activity *Blowing and Picking* have 100 relations, activity *Blowing and Picking* and activity *Carding* have 77 relations, activity *Blowing and Picking* and activity *Pre Carding* consist of 23 relations. Meanwhile, activities *Carding* and *Superlap*, activities *Pre Carding* and *Superlap* have 23 and 77 respectively. Activity *Superlap* and activity *Combing* have 100 relations in the EL1. In addition, Activity *Combing* which pair with activities *Drawing*, *Drawing II* and *Roving* consist of 73, 12, 15 respectively. Activities *Drawing* and *Drawing II* have 60 relations, whereas activities *Drawing II* and

Drawing have 9 relations. Activities *Drawing* and *Drawing II* lead to activity *Roving* consist of 13 and 3 relations. Last, activity *Roving* and activity *Spinning*, activity *Spinning* and activity *Winding* have the same 100 relations. All of them are in sequential relations. For activities *Carding* and *Pre Carding* and activities *Drawing*, *Drawing II* and *Roving* have 100 and 69 respectively. They are in parallel relations.

For the next step, we need to calculate minimum all sequential relations, average parallel relations and average all sequential relations. Table 6 presents the results of EL1 for the value of Min ASR, Avg PPM and Avg ASR. From Table 6 we get the information that activities *Drawing*, *Drawing II* and *Roving* have the value of Avg PPM higher than the value of Min ASR, but the value of Avg PPM is the same as the value of Avg ASR. So, the final relation of activities *Drawing*, *Drawing II* and *Roving* is OR as explained (7). While, for activities *Carding* and *Pre Carding*, the value of Avg PPM is higher than the value of Avg ASR. Based on (8), the relation is AND.

The very last step of our proposed approach, we present the AND and OR relations in Petri Net form into the discovered business process of EL1 by following (9). Figure 5 shows the final result of the business process of EL1 along with AND, OR relations and L1L, L2L.

Table 6. The value of min ASR, avg PPM and avg ASR to determine the parallel of EL1

Parallel Activities	The value of Min ASR	The value of Avg PPM	The value of Avg ASR	Parallel Gateway
Carding Pre Carding		100		AND
Drawing Drawing II Roving	3	52.3	52.3	OR

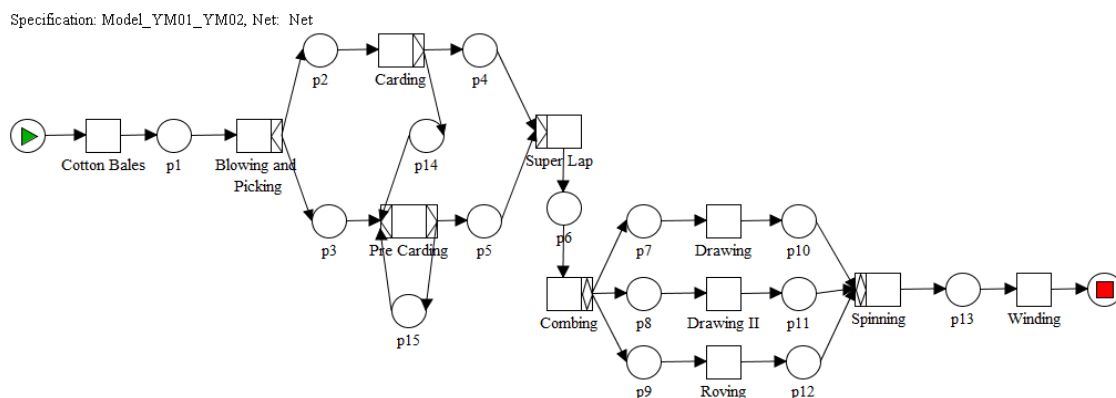


Figure 5. A business process of EL1 with AND, OR, L1L, L2L by using TBA+M algorithm

3.3. Evaluation of the business process

In this research, two comparison steps were done to evaluate the business process. Firstly, comparing the parallel relations between the proposed algorithm (TBA+M) and the original algorithm ($\alpha+$ Miner). Secondly, the fitness value of the business process is calculated using (10). Table 7 presents the parallel relations discovered by using $\alpha+$ Miner algorithm and Figure 6 shows the business process in the Petri Net form for the same cases of EL1 in Table 1 obtained by $\alpha+$ Miner algorithm. The main difference between Figures 5 and 6 is we can obtain OR relation in the business process in the Figure 5, meanwhile we can only distinguish AND relation in the Figure 6 among activities *Drawing*, *Drawing II* and *Roving*.

Generally, process discovery using the original $\alpha+$ Miner algorithm uses single timestamp event log [16, 25, 26]. It cannot distinguish the differences between AND and OR in parallel relations. The original $\alpha+$ Miner algorithm has difficulties in interpreting OR in a business process. This is because the original $\alpha+$ Miner algorithm only has 2 parallel relations, namely AND and XOR [16, 24]. If one activity is executed exactly once in business process, then it is categorized as XOR, meanwhile AND happens when one or more activities are allowed to be executed more than once in business process. Therefore, OR in the business process will be equated with XOR or AND, even though in reality those three relations (AND, OR, XOR) have differences in how to execute activities in the event log [4, 6, 13]. We can conclude that the proposed algorithm is better in determining parallel relations (AND, OR) than the original algorithm.

The calculation of fitness value becomes the last step for evaluating the business process. Fitness value is declared good if the value is closer to 1 [19, 20]. The calculation results of fitness value presented in

Table 8 give the information that the business process obtained by using the proposed algorithm fits the reality better than that of the original algorithm. All cases in EL1 can be captured and can be modelled into a business process.

From the evaluation stage, we can conclude that our proposed algorithm can mine the business process as well as the sequential and parallel (AND, OR, XOR) relations, which cannot be obtained by the original algorithm. The results of fitness value also clearly state that the new proposed algorithm gives the best value. Hence, Time-based $\alpha+$ Miner algorithm can be an extended version of $\alpha+$ Miner algorithm in process discovery.

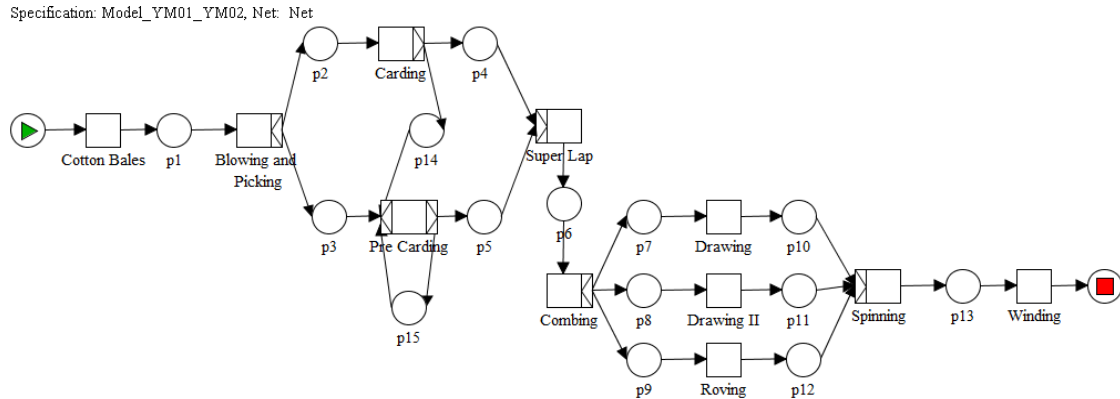


Figure 6. A business process of EL1 with AND, AND, L1L, L2L generated by $\alpha+$ Miner

Table 7. The parallel relations of EL1 discovered using $\alpha+$ Miner

Parallel Activities	Parallel Relations
Carding Pre Carding	AND
Drawing Drawing II Roving	AND

Table 8. The comparison of fitness value of EL1

Algorithms	Fitness Value
Time-based $\alpha+$ Miner	1.000
$\alpha+$ Miner	0.954

4. CONCLUSION

This research paper focused on introducing a new approach to mine the business processes using temporal pattern which implemented in Time-based $\alpha+$ Miner (TBA+M) algorithm. The proposed method defined seven temporal patterns which included both sequential and parallel relations. They are *before*, *meet*, *the_same_start_time*, *the_same_complete_time*, *overlap*, *contain* and *equal*.

TBA+M algorithm was the extended version of $\alpha+$ Miner algorithm. TBA+M combined with temporal pattern were used to mine the process model from the event log. Compared to original $\alpha+$ Miner algorithm, this modified algorithm could detect overlapping timestamp using temporal pattern and identifies it as a parallel process, discovered length one loop (L1L), length two loop (L2L) and specified the sequential and parallel (AND, OR, XOR) relations in a correct way. Meanwhile $\alpha+$ Miner algorithm was not able to distinguish the differences between AND and OR in parallel relations.

The experimental results presented that our new process discovery approach could mine the business processes with parallel AND and OR relations, which could not be obtained by the original $\alpha+$ Miner algorithm. In addition, the results of fitness value also clearly stated that the new method also gave better result rather than that of the original $\alpha+$ Miner algorithm

ACKNOWLEDGEMENTS

The completion of this research could not have been possible without the support from Computer Laboratory, Department of Information Systems, Universitas Airlangga and Information Management Laboratory, Department of Informatics, Institut Teknologi Sepuluh Nopember.

REFERENCES

- [1] X. Wang, P. Huang, S. Wu, and Z. Shi, "Measurement of Synergetic Degree of Enterprise E-Commerce Business Process," *Fourth International Conference on Business Intelligence and Financial Engineering*, 2011, doi: 10.1109/BIFE.2011.81.
- [2] J. Guo and Y. Zou, "A Business Process Explorer: Recovering Business Processes from Business Applications," in *15th Working Conference on Reverse Engineering*, 2018, doi: 10.1109/WCRE.2008.25.
- [3] A. Sienou, A. P. Karduck, E. Lamine, and H. Pingaud, "Business Process and Risk Models Enrichment: Considerations for Business Intelligence," *IEEE International Conference on e-Business Engineering*, 2018, doi: 10.1109/ICEBE.2008.123.
- [4] Y. A. Effendi and R. Sarno, "Implementation of the Semantic Web in Business Process Modeling Using Petri Nets," in *International Conference on Information and Communications Technology*, 2018, vol. 741-746, doi: 10.1109/ICOIACT.2018.8350724.
- [5] R. Saylam and O.K. Sahingoz, "Process mining in business process management: Concepts and challenges," *International Conference on Electronics, Computer and Computation (ICECCO)*, 2013, doi: 10.1109/ICECCO.2013.6718246.
- [6] W. M. P. Van der Aalst, "Process mining: discovery, conformance and enhancement of business processes," Springer Science and Business Media, 2011, doi: 10.1007/978-3-642-19345-3.
- [7] W. M. P. Van der Aalst and H. M. W. Verbeek, "Process discovery and conformance checking using passages," *Fundamenta Informaticae*, vol. 131, no. 1, pp. 103-138, 2011.
- [8] J. Joe, T. Emmatty, Y. Ballal, and S. Kulkarni, "Process mining for project management," *International Conference on Data Mining and Advanced Computing (SAPIENCE)*, 2016, doi: 10.1109/SAPIENCE.2016.7684142.
- [9] A. A. Kalenkova, W. M. P. Van der Aalst, I. A. Lomazova, and V. A. Rubin, "Process mining using BPMN: relating event logs and process models," *Software & Systems Modeling*, 2017, vol. 16, no. 4, pp. 1019-1048.
- [10] H. M. W. Verbeek, J. C. A. M. Buijs, B. F. Van Dongen, and W. M. P. Van der Aalst, "ProM 6: The Process Mining Toolkit," vol. 615, 2010.
- [11] M. R. Peña and S. Bayona-Oré, "Process Mining and Automatic Process Discovery," *7th International Conference on Software Process Improvement (CIMPS)*, 2018, doi: 10.1109/CIMPS.2018.8625621.
- [12] S. S. Pinter and M. Golani, "Discovering workflow models from activities' life spans," *Computers in Industry*, vol. 53, no. 3, pp. 283-296, 2014.
- [13] Y. A. Effendi and R. Sarno, "Modeling Parallel Business Process Using Modified Time-based Alpha Miner," *International Journal of Innovative Computing, Information and Control*, vol. 14, no. 5, 2018, doi: 10.24507/ijicic.14.05.1565.
- [14] W. M. P. Van der Aalst, "Business process management as the "Killer App" for Petri nets," *Software & Systems Modeling*, vol. 14, no. 2, pp. 685-691, 2015.
- [15] W. M. P. Van der Aalst and C.C. Stahl, "Modeling business processes: A petri net-oriented approach," MIT Press, 2011.
- [16] A. K. A. de Medeiros, B. F. van Dongen, W. M. P. van der Aalst, and A. J. M. M. Weijters. "Process Mining: Extending the α -algorithm to Mine Short Loops," 2004.
- [17] N. Tax, N. Sidorova, R. Haakma, and W. M. P. Van der Aalst, "Mining process model descriptions of daily life through event abstraction," *Proceedings of SAI Intelligent Systems Conference*, pp. 83-104, 2016.
- [18] R. A. Sutrisnowati, H. Bae, L. Dongha, and K. Minsoo, "Process Model Discovery based on Activity Lifespan," *International Conference on Technology Innovation and Industrial Management*, pp. 137-156, 2014, doi: 10.1016/j.eswa.2014.05.055.
- [19] Y. A. Effendi and R. Sarno, "Conformance Checking Evaluation of Process Discovery Using Modified Alpha++ Miner Algorithm," *International Seminar on Application for Technology of Information and Communication*, pp. 435 - 440, 2018, doi: 10.1109/ISEMANTIC.2018.8549770.
- [20] A. Burattin, F. M. Maggi, and A. Sperduti, "Conformance checking based on multi-perspective declarative process models," *Expert Systems with Applications*, vol. 65, pp. 194-211, 2016, doi: 10.1016/j.eswa.2016.08.040.
- [21] W. Chomyat, and W. Premchaiswadi, "Process mining on medical treatment history using conformance checking," in *14th International Conference on ICT and Knowledge Engineering (ICT&KE)*, 2016, doi: 10.1109/ICTKE.2016.7804102.
- [22] W. M. P. Van der Aalst, A. Adriansyah, and B. F. Van Dongen, "Causal Nets: A Modeling Language Tailored towards Process Discovery," *J.P.K.B. König, CONCUR-Concurrency Theory*, Springer Berlin Heidelberg, pp. 28-42, 2011, doi: 10.1007/978-3-642-23217-6_3.
- [23] D. C. Sofie, C. Jan, and P. Geert, "Improving the quality of the Heuristics Miner in Prom 6.2.," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7678-7690, 2014.
- [24] Li J, Liu D and Yang B. "Process Mining: Extending the Alpha algorithm to mine duplicate tasks in process logs," *Advances in Web and Network Technologies, and Information Management*, pp. 396-407, 2007.
- [25] A. Rogge-Solti, R. S. Mans, W. M. P. Van der Aalst, and M. Weske, "Repairing event logs using timed process models," *OTM Confederated International Conferences*, 2013.
- [26] W. Lijie and W. Jianmin, "Detecting implicit Dependencies Between Tasks from Event Logs," School of Software, Tsinghua University.