Determining Process Model Using Time-Based Process Mining and Control-Flow Pattern

Riyanarto Sarno*, Widyasari Ayu Wibowo, Kartini, Yutika Amelia, Kelly Rossa
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember,
Kampus ITS Sukolilo, Jalan Raya ITS, Surabaya, Jawa Timur 60111, (031) 5994251
*Corresponding author, e-mail: riyanarto@if.its.ac.id

Abstract

Determining right model of business process from event log is the purpose of process discovery. However some problems i.e the inability to discover OR, noise and event log incompleteness are emmerged while determining right model of business process. First, OR relation is often discovered as AND relation. Second, noise problem is occured when there are truncated and low frequency traces in event log. Thus control-flow pattern is used to solve issues of same noise relation frequency hence it discovers relation based on transaction function of activity. Consequently, it can refine non noise relation in business process model. Third, incompleteness leads to incorrect discovery of parallel process model; therefore we used Timed-based Process Mining which utilized non-linear dependence to solve the incompleteness. Finally this paper proposed combination of Timed-based Process Mining and control-flow pattern to discover OR and handle same frequency noise and incompleteness. From the experiment in section 3, this proposed method manages to get right process model from event log.

Keywords: *Conditional OR, Control-Flow Pattern, Incompleteness, Noise, Timed-based Process Mining*

1. Introduction

Determining model of business process from event log is the main purpose of process discovery. Process discovery is a challenging task in process mining. It is a set of techniques which automatically construct a model of an organization`s current activities and its major activities variations. These techniques use event log of activities within an organization. The business process model is analyzed to show the complexity of issues in activities and how to solve them. These issues exist in any field, e.g. business [1], environment [2, 3] smartphone [4], and fraud [5].

Process discovery comes up with many algorithms, e.g. alpha, alpha+, alpha++ [6]. The alpha, alpha+, and alpha++ cannot deal with noise, incompleteness issues and OR conditional. Heuristic miner algorithms [7, 9] come up to solve the noise problem. However, most of the algorithms are unable to find OR conditional model. The existing algorithm frequently discovers the OR conditional as AND parallel or XOR conditional. The thought of parallel model discovery will change the result of activities [8]. When “wait and see” behavior model synchronization is occured, it needs OR parallel to model the parallel split and join. The “wait and see” behavior model synchronization occured when the actor can choose only one activity, all activity, or more than one activity in parallel split and join. In this paper we proposed ideas to discover OR conditional within business process model.

One of important things from process mining is the idea of completeness which is related to noise. Incompleteness leads to false parallel relations discovery, e.g the discovered parallel relation is XOR but the right parallel relation in business process is OR. The new representation of OR-split uses combination the existing XOR-split and AND-split to make the model easier to be analyzed [13]. In other hand temporal activity-based algorithm [8] and control-flow pattern can handle discovery of business process model with incompleteness and same frequency noise issues. Non-linear dependence in temporal activity-based algorithm is used to solve incompleteness problem since it can discover more relation than linear dependence. Control-flow pattern is used to solve same amounts of noise frequency issues because it discovers relation based on transaction function of activity, therefore it can choose non noise relation in business process model.

Received November 12, 2015; Revised January 7, 2016; Accepted January 26, 2016

2. Research Method

In Figure 1 we describe the proposed method to discover the right model in business process. One of the methods used in this research is modified temporal activity-based algorithm. Temporal activity-based algorithm is modified to discover conditional OR and overcome noise and incompleteness. The modification is done by using non-linear dependence to deal with incompleteness and modifiying parallel relation to distinguish parallel AND and conditional OR. The non-linear dependence utilizes double timestamped event log to discover sequence and concurrent relation in a case in event log; whereas linear dependence utilizes single timestamped event log to discover only sequence relation in a case in event log [8]. Finally the discovered model can be matched with formal control-flow pattern existed in real business process model. In Figure 1 we explained analytical steps of proposed method.

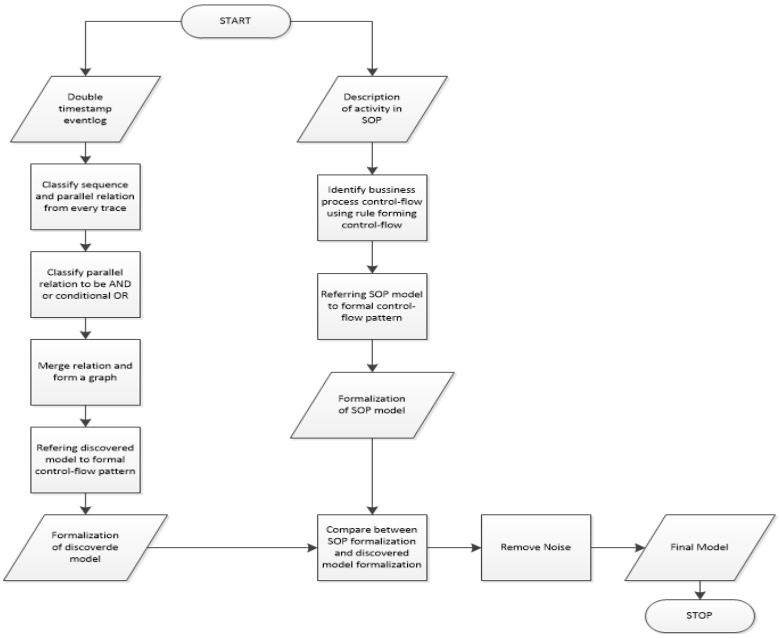
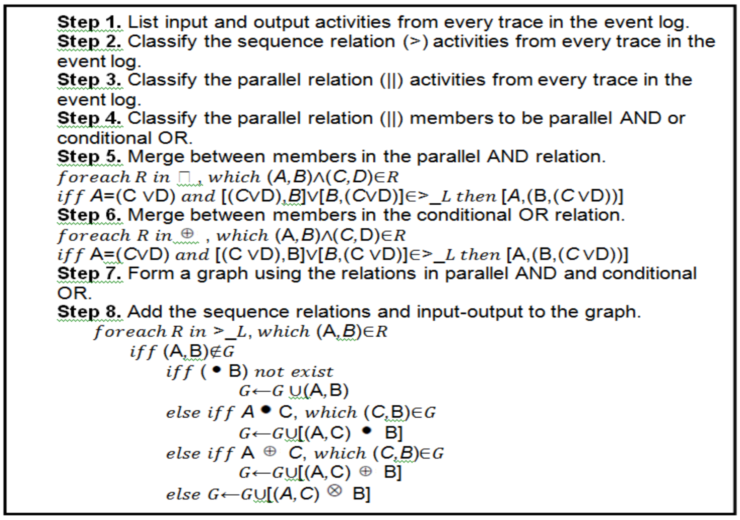

START
Double timestamp eventlog
Description of activity in SOP
Classify sequence and parallel relation from every trace
Identify bussiness process control-flow using rule forming control-flow
Classify parallel relation to be AND or conditional OR
Referring SOP model to formal control-flow pattern
Merge relation and form a graph
Formalization of SOP model
Refering discovered model to formal control-flow pattern
Formalization of discoverde model
Compare between SOP formalization and discovered model formalization
Remove Noise
Final Model
STOP

Figure 1. Flow chart of proposed method

2.1. Modification Timed-based Process Mining Algorithm



```
Step 1. List input and output activities from every trace in the event log.
Step 2. Classify the sequence relation (>) activities from every trace in the event log.
Step 3. Classify the parallel relation (||) activities from every trace in the event log.
Step 4. Classify the parallel relation (||) members to be parallel AND or conditional OR.
Step 5. Merge between members in the parallel AND relation.
$foreach\ R\ in$ □ $,\ which\ (A,B)\wedge(C,D)\in R$
$iff\ A=(C \vee D)\ and\ [(C\vee D),B]\vee[B,(C\vee D)]\in >\_L\ then\ [A,(B,(C\vee D))]$
Step 6. Merge between members in the conditional OR relation.
$foreach\ R\ in\ \oplus\ ,\ which\ (A,B)\wedge(C,D)\in R$
$iff\ A=(C\vee D)\ and\ [(C \vee D),B]\vee[B,(C \vee D)]\in >\_L\ then\ [A,(B,(C\vee D))]$
Step 7. Form a graph using the relations in parallel AND and conditional OR.
Step 8. Add the sequence relations and input-output to the graph.
    $foreach\ R\ in\ >\_L,\ which\ (A,B)\in R$
        $iff\ (A,B)\notin G$
            $iff\ (\bullet\ B)\ not\ exist$
                $G\leftarrow G \cup (A,B)$
            $else\ iff\ A \bullet C,\ which\ (C,B)\in G$
                $G\leftarrow G\cup[(A,C) \bullet B]$
            $else\ iff\ A \oplus C,\ which\ (C,B)\in G$
                $G\leftarrow G\cup[(A,C) \oplus B]$
            $else\ G\leftarrow G\cup[(A,C) \otimes B]$
```

Figure 2. Modification Timed-based Process Mining Algorithm

The Timed-based Process Mining algorithm is introduced by Rizka et. al. [8]. The algorithm is created based on several definitions to note the relation between activities in event log e.g before and meets, overlaps, contains, is-finished-by, equals, and starts [8]. However, the algorithm can not distinguish OR or AND relations. Thus the algorithm is modified by adding a step to distinguish OR and AND relations (shown in step 4-8). The modification algorithm is written in Figure 2.

2.2. Control-flow Patterns

In business process models there are often used process forms which are called Control-flow patterns. There are 20 patterns available for used in multi model diagram based on Aalst paper [10]. Those patterns are too abstract for practical use in business process modelling, thus we tried to formalize the patterns usage in two steps, Top Level Abstraction and Low Level Execution. Top Level Abstraction defines business process model in multi organizations platform while Low Level Abstraction is used to discover the real activities performed inside an organization.

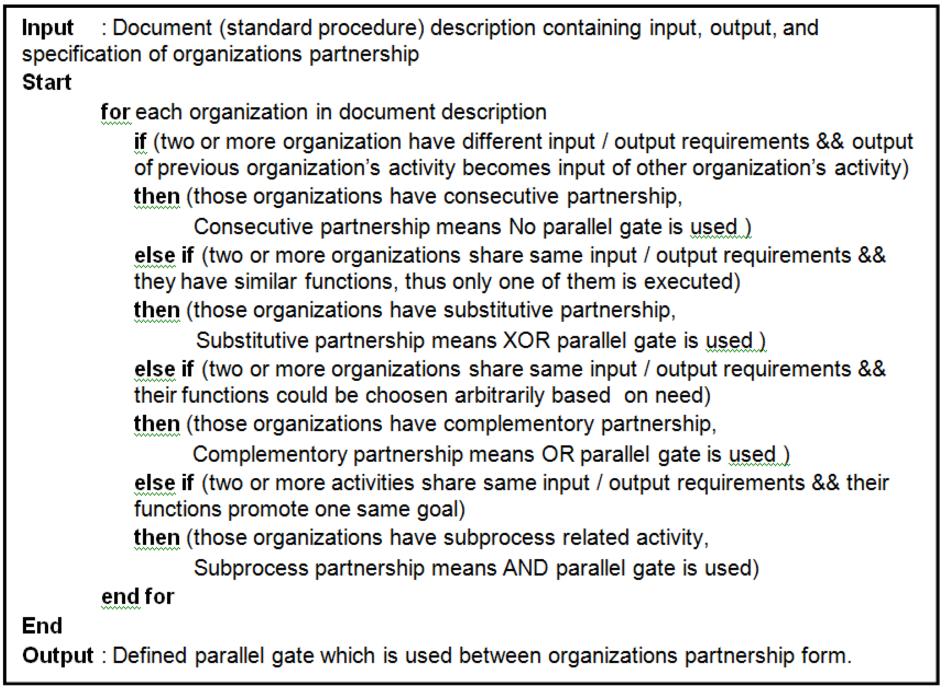
2.2.1. Top Level Abstraction

In real life application event-logs of a complete business process hardly can be found in one database. Each embroiled organization keeps their own data in separate databases to ensure data security. Thus to obtain a complete viewpoint in business process model we need to define the form of partnership between organizations participated in business process.The definition of partnership form between organization is called Top Level Abstraction. It is only explained the framework of a business process but not necessarily explained the activities which are executed by a certain organization. Pseudocode for Top Level Abstraction is listed in Figure 3.

There are four types of partnership which is regularly applied in multi-organization business process:

a. Consecutive Partnership
b. Substitutive Partnership
c. Complementory Partnership
d. Subprocess Partnership

Pseudocode Rule Defining Top Level Abstraction is explained in Figure 3.



```
Input   : Document (standard procedure) description containing input, output, and
specification of organizations partnership
Start
        for each organization in document description
            if (two or more organization have different input / output requirements && output
            of previous organization's activity becomes input of other organization's activity)
            then (those organizations have consecutive partnership,
                    Consecutive partnership means No parallel gate is used )
            else if (two or more organizations share same input / output requirements &&
            they have similar functions, thus only one of them is executed)
            then (those organizations have substitutive partnership,
                    Substitutive partnership means XOR parallel gate is used )
            else if (two or more organizations share same input / output requirements &&
            their functions could be choosen arbitrarily based on need)
            then (those organizations have complementory partnership,
                    Complementory partnership means OR parallel gate is used )
            else if (two or more activities share same input / output requirements && their
            functions promote one same goal)
            then (those organizations have subprocess related activity,
                    Subprocess partnership means AND parallel gate is used)
        end for
End
Output : Defined parallel gate which is used between organizations partnership form.
```

Figure 3. Rule of Top Level Abstraction

2.2.2. Low Level Execution

Defining form of patnership between embroiled organizations only explained outer framework of business process model. Thus we need to look up inside each organizations and model executed activities of real business process, this step called Low Level Execution. The model is incorrectly discovered while it contains the noise relation. Therefore, control-flow pattern of business process is employed to confirm the discovered relation in business process.

We acknowledge Aalst control-flow patterns [10] as theoretical guidance. We define 8 out of 20 available patterns [11] are able to help in deciding parallel model which is limited for organizational collaboration. Those the 8 patterns are expressed in formal and graphical way. Basic Patterns

This group of patterns contains elementary aspects of workflow process. Those patterns are listed below:

1) Sequence

The pattern as seen in Figure 4, defines simplest form of activity execution in control-flow. Sequence describes an activity in workflow is enabled after the completion of its input activity in the same process.

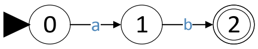


Figure 4. Sequence pattern

2) Parallel Split

The pattern as seen in Figure 5, defines splitting point in a workflow where a single thread of process divised into two or more branches of process control which can be executed silmultaneously in any order.

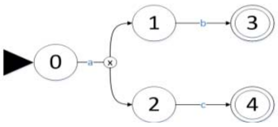


Figure 5. Parallel split pattern

3) Synchronization

The pattern as seen in Figure 6, defines a point in a workflow where multiple process converge into one single process, thus synchronization of multiple thread of process is happened.

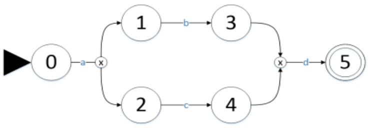


Figure 6. Synchronization pattern

4) Exclusive Choice

The pattern as seen in Figure 7, defines a point in a workflow where based on decision control only one out of some n multiple branches is executed.

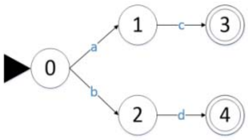


Figure 7. Exclusive choice pattern

5) Simple Merge

The pattern as seen in Figure 8, describes a workflow when two or more branches converge into a single process thread without synchronization. The assumption of this pattern is none of multiple branches is ever executed simultaneously.

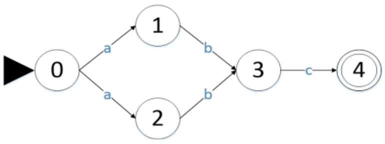


Figure 8. Simple merge pattern

a) Advance Conditional Branching Patterns

This group of patterns is slightly more complex than branching patterns in basic pattern, yet the patterns list in this group are usually often used in real life processes. Those patterns are listed below:

1) Multi Choice

The pattern as seen in Figure 9, is an improvement of Exclusive choice pattern. In Multi choice pattern branches are able to be executed in parallel or sequentially depending on the decision of execution time.

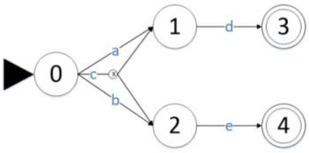


Figure 9. Multi choice pattern

2) Synchronizing Merge

The pattern as seen in Figure 10, combines between Synchronization and Simple Merge pattern, where the decision of sychronization or merge process depends on execution.

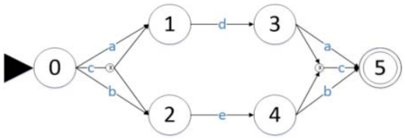


Figure 10. Synchronizing merge pattern

b) Structural Pattern

The pattern as seen in Figure 11, describes different restriction in workflow models such as cycles pattern. In this section, we present a pattern which represent typical workflow management systems structural restrictions.

1) Arbitrary cycles

Arbitrary cycles defines a point in a workflow where one or more activities can be executed repeatedly.

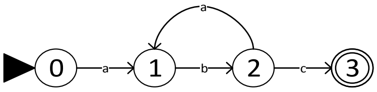
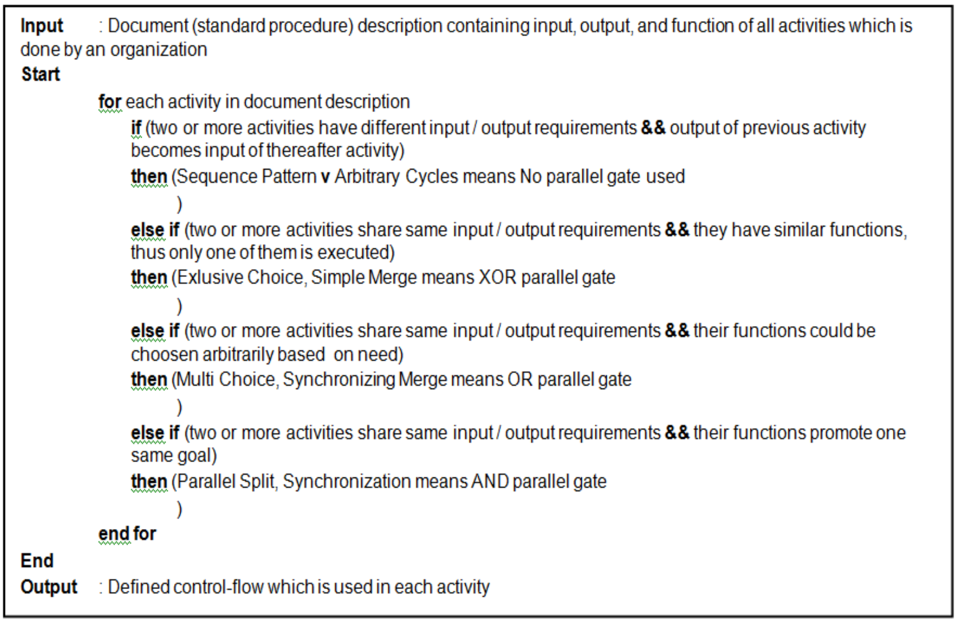


Figure 11. Arbitrary cycles pattern

The rule used to define the model is similar with Context Sensitive Grammar [12]. Pseudocode Rule Defining Low Level Execution is explained in Figure 12.



```
Input    : Document (standard procedure) description containing input, output, and function of all activities which is
done by an organization
Start
        for each activity in document description
            if (two or more activities have different input / output requirements && output of previous activity
            becomes input of thereafter activity)
            then (Sequence Pattern v Arbitrary Cycles means No parallel gate used
                  )
            else if (two or more activities share same input / output requirements && they have similar functions,
            thus only one of them is executed)
            then (Exlusive Choice, Simple Merge means XOR parallel gate
                  )
            else if (two or more activities share same input / output requirements && their functions could be
            choosen arbitrarily based  on need)
            then (Multi Choice, Synchronizing Merge means OR parallel gate
                  )
            else if (two or more activities share same input / output requirements && their functions promote one
            same goal)
            then (Parallel Split, Synchronization means AND parallel gate
                  )
        end for
End
Output   : Defined control-flow which is used in each activity
```

Figure 12. Rule of Low Level Execution

3. Results and Analysis

This research uses double time-stamped event log containing more than one organization executing activities in event log. The information attributes contained in event log are the number of case id, the activity in process, time-stamp of activity execution, and the organization executing the activity.

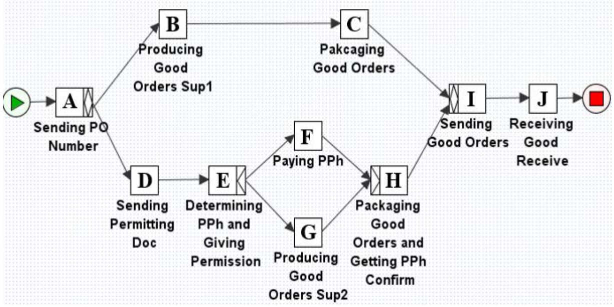
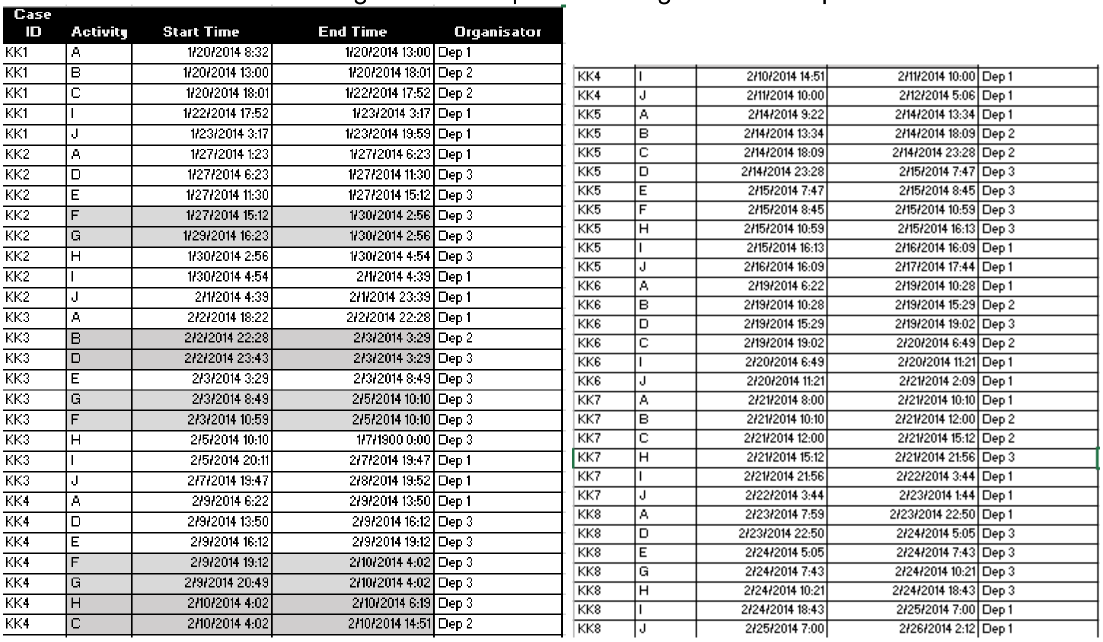


Figure 13. YAWL model of standard business process

Business process on Figure 13 executed as in event log is a process to buy goods for production from supplier by a company. There are two kind of supplier. The first supplier is in bonded zone of Customs and Excise and the second is out of it. The company sends purchase order number to supplier to order good for production (activity A). Then, the first supplier produces (activity B), packages (activity C), and sends the good order to company (activity I); whereas the second supplier sends permitting document to Customs and Excise for requesting transaction approvement between supplier and company (activity D). Then, Customs and Excise determines tax and approve the transaction (activity E). Furthermore, the second supplier pays the tax (activity F), produces (activity G), packages (activity H), and sends good order (activity I). Finally, company receives the good (activity J).

The event log which is us in this experiment is presented in Table 1.

Table 1. Single time stamped event log of business process



Case ID	Activity	Start Time	End Time	Organisator
KK1	A	1/20/2014 8:32	1/20/2014 13:00	Dep 1
KK1	B	1/20/2014 13:00	1/20/2014 18:01	Dep 2
KK1	C	1/20/2014 18:01	1/22/2014 17:52	Dep 2
KK1	I	1/22/2014 17:52	1/23/2014 3:17	Dep 1
KK1	J	1/23/2014 3:17	1/23/2014 19:59	Dep 1
KK2	A	1/27/2014 1:23	1/27/2014 6:23	Dep 1
KK2	D	1/27/2014 6:23	1/27/2014 11:30	Dep 3
KK2	E	1/27/2014 11:30	1/27/2014 15:12	Dep 3
KK2	F	1/27/2014 15:12	1/30/2014 2:56	Dep 3
KK2	G	1/29/2014 16:23	1/30/2014 2:56	Dep 3
KK2	H	1/30/2014 2:56	1/30/2014 4:54	Dep 3
KK2	I	1/30/2014 4:54	2/1/2014 4:39	Dep 1
KK2	J	2/1/2014 4:39	2/1/2014 23:39	Dep 1
KK3	A	2/2/2014 18:22	2/2/2014 22:28	Dep 1
KK3	B	2/2/2014 22:28	2/3/2014 3:29	Dep 2
KK3	D	2/2/2014 23:43	2/3/2014 3:29	Dep 3
KK3	E	2/3/2014 3:29	2/3/2014 8:49	Dep 3
KK3	G	2/3/2014 8:49	2/5/2014 10:10	Dep 3
KK3	F	2/3/2014 10:59	2/5/2014 10:10	Dep 3
KK3	H	2/5/2014 10:10	1/7/1900 0:00	Dep 3
KK3	I	2/5/2014 20:11	2/7/2014 19:47	Dep 1
KK3	J	2/7/2014 19:47	2/8/2014 19:52	Dep 1
KK4	A	2/9/2014 6:22	2/9/2014 13:50	Dep 1
KK4	D	2/9/2014 13:50	2/9/2014 16:12	Dep 3
KK4	E	2/9/2014 16:12	2/9/2014 19:12	Dep 3
KK4	F	2/9/2014 19:12	2/10/2014 4:02	Dep 3
KK4	G	2/9/2014 20:49	2/10/2014 4:02	Dep 3
KK4	H	2/10/2014 4:02	2/10/2014 6:19	Dep 3
KK4	C	2/10/2014 4:02	2/10/2014 14:51	Dep 2
KK4	I	2/10/2014 14:51	2/11/2014 10:00	Dep 1
KK4	J	2/11/2014 10:00	2/12/2014 5:06	Dep 1
KK5	A	2/14/2014 9:22	2/14/2014 13:34	Dep 1
KK5	B	2/14/2014 13:34	2/14/2014 18:09	Dep 2
KK5	C	2/14/2014 18:09	2/14/2014 23:28	Dep 2
KK5	D	2/14/2014 23:28	2/15/2014 7:47	Dep 3
KK5	E	2/15/2014 7:47	2/15/2014 8:45	Dep 3
KK5	F	2/15/2014 8:45	2/15/2014 10:59	Dep 3
KK5	H	2/15/2014 10:59	2/15/2014 16:13	Dep 3
KK5	I	2/15/2014 16:13	2/16/2014 16:09	Dep 1
KK5	J	2/16/2014 16:09	2/17/2014 17:44	Dep 1
KK6	A	2/19/2014 6:22	2/19/2014 10:28	Dep 1
KK6	B	2/19/2014 10:28	2/19/2014 15:29	Dep 2
KK6	D	2/19/2014 15:29	2/19/2014 19:02	Dep 3
KK6	C	2/19/2014 19:02	2/20/2014 6:49	Dep 2
KK6	I	2/20/2014 6:49	2/20/2014 11:21	Dep 1
KK6	J	2/20/2014 11:21	2/21/2014 2:09	Dep 1
KK7	A	2/21/2014 8:00	2/21/2014 10:10	Dep 1
KK7	B	2/21/2014 10:10	2/21/2014 12:00	Dep 2
KK7	C	2/21/2014 12:00	2/21/2014 15:12	Dep 2
KK7	H	2/21/2014 15:12	2/21/2014 21:56	Dep 3
KK7	I	2/21/2014 21:56	2/22/2014 3:44	Dep 1
KK7	J	2/22/2014 3:44	2/23/2014 1:44	Dep 1
KK8	A	2/23/2014 7:59	2/23/2014 22:50	Dep 1
KK8	D	2/23/2014 22:50	2/24/2014 5:05	Dep 3
KK8	E	2/24/2014 5:05	2/24/2014 7:43	Dep 3
KK8	G	2/24/2014 7:43	2/24/2014 10:21	Dep 3
KK8	H	2/24/2014 10:21	2/24/2014 18:43	Dep 3
KK8	I	2/24/2014 18:43	2/25/2014 7:00	Dep 1
KK8	J	2/25/2014 7:00	2/26/2014 2:12	Dep 1

We provide an example of Process Discovery using Timed-based Process Mining and Control-Flow Patterns as written below.

3.1. Step 1: Discovering Process Model

The first step is classify the sequence and parallel relation activities from every trace in eventlog. After classifying relation activities from every trace, merge it into a sequence relation activities and a parallel relation activities. The merging relations are described in Figure 14.

I = A, O = J
> = {{} -> A, A -> B , A -> D, B -> C, B->D, C -> D, C -> I, C->H, D -> E , E -> F, E -> G, F -> H, G -> H, H -> I, I -> J, J -> {}}
|| = {B || D, F || G, C || H}

Figure 14. Merging relations

After merging the relation activities, then classify parallel relations to be AND or conditional OR. The classifies are described in Figure 15.

$$\oplus = \{B \oplus D, C \oplus H\}$$
$$\bullet = \{ F \bullet G\}$$

Figure 15. AND or Conditional OR relations

The third step is form a graph based on the relation activities in Figure 14 and Figure 15. The graph can be modeled as Figure 16.

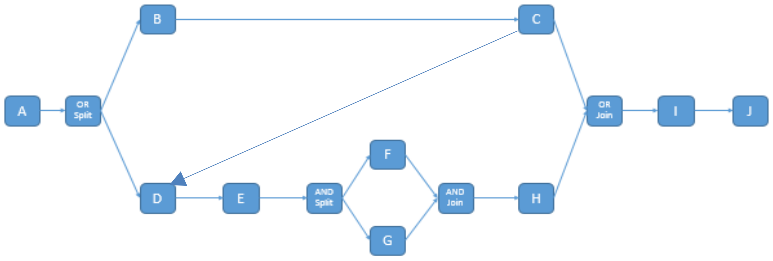


Figure 16. Final graph

3.2. Step 2: Discovering Standard Process Model using Control-flow Patterns

There are 3 organizations participated in whole business process : α, β , and γ Activity A, I, J is done by organization α, activity B and C is done by organization β, and activity D, E, F, G, H is done by organization γ. α is the main organization responsible for whole business process. α shares its responsibilities with β and γ to do the same goal : supply goods which is requested in activity A. The form of partnership between organization α and β, γ is defined as substitutive partnership. Thus we can obtain Top Level Abstraction model as Figure 17.

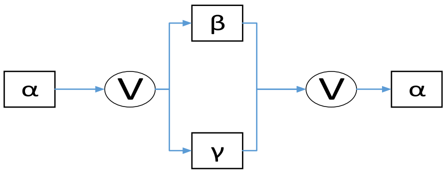


Figure 17. Top Level Abstraction model with OR gate

Activity A (purchasing order) is a starting activity in business process and followed by activity B (first supplier) and activity D (second supplier). Activity B and activity D share the same inputs and outputs, and company can optionally choose between executing activity B or D or both (since activity B and activity D are same, to choose supplier type). Thus activity B and D use Multi Choice pattern (see Figure 18). Activity B followed by activity C resulting activity B and C use Sequence pattern (see Figure 19). Activity D is followed by activity E, those two activities define as Sequence pattern. Next activities are activity F (paying taxes) and G(produce goods). Activity F and activity G must be done to complete the order, and both activities could be executed simultaneously, resulting activity F and G as Parallel pattern (see Figure 20). Activity I (send goods) and activity J (receive goods) are executed sequentially. We can define the Low Level Execution model for each organization as below.

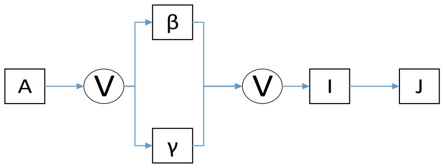


Figure 18. Low Level Execution model of α with OR gate

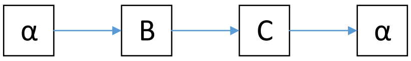


Figure 19. Low Level Execution model of β without any parallel gate

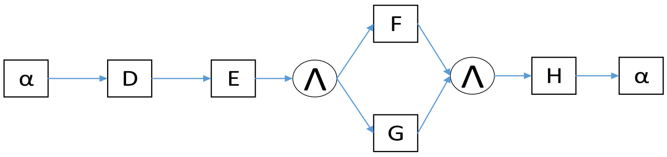


Figure 20. Low Level Execution model of γ with AND gate

The formalization of this business process model in Figure 13 will be presented as follow. Formalization for standard procedure of process is needed as right guidance in process execution. We represented its standard formalization as below:

W = {*S, A, N,* ∑*, α, E*} where:
S is a Start activity in workflow process
A is a finite set of Activities
N $\subseteq$ *A is a finite set of Nodes*
∑ *is finite set of task inputs/outputs*
α is the Node transition function
E is an End activity in workflow process
S = {A}
E = {J}
A = {A, B, C, D, E, F,G, H, I, J}
N = $\{n_A, n_B, n_C, n_D, n_E, n_F, n_G, n_H, n_I, n_J\}$
∑ = {a, b, c, d, e, f, g, h, i, j, k}
α = {

Sequence
$[n_B, c, n_C]$, $[n_D, d, n_E]$, $[n_I, i, n_J]$,
Parallel split & synchronization
$[n_E, e, n_F n_G]$, $[n_F n_G, f, n_H]$
Multi Choice
$[n_A, a, n_B]$, $[n_A, b, n_D]$, $[n_A, j, n_B n_D]$
Synchronizing Merge
$[n_C, g, n_I]$, $[n_H, h, n_I]$, $[n_C n_H, k, n_I]$

}

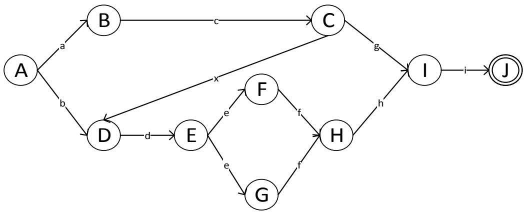


Figure 21. Formalized graph of dependency graph

In step 1 we obtained relation between activity as shown in Figure 21, thus we derived a standard formalization and compare this formalization and formalization of standard procedure to verify whereas there's noise in obtained model or not. Later in this step noise is deleted from obtained model.

W = {*S, A, N,* ∑*, α, E*} where:
S = {A}
E = {J}
A = {A, B, C, D, E, F,G, H, I, J}
N = $\{n_A, n_B, n_C, n_D, n_E, n_F, n_G, n_H, n_I, n_J\}$
∑ = {a, b, c, d, e, f, g, h, i, j, k, x}
α = {

Sequence
$[n_B, c, n_C]$, $[n_D, d, n_E]$, $[n_I, i, n_J]$, $[n_c, x, n_d]$,

Parallel split & synchronization

$[n_E, e, n_F n_G]$, $[n_F n_G, f, n_H]$

Multi Choice

$[n_A, a, n_B]$, $[n_A, b, n_D]$, $[n_A, j, n_B n_D]$

Synchronizing Merge

$[n_C, g, n_I]$, $[n_H, h, n_I]$, $[n_C n_H, k, n_I]$

}

Relation of $[n_C, x, n_D]$ is not available in relation set of standard procedure, thus this relation is labeled as noise and deleted from obtained model as seen in Figure 22.

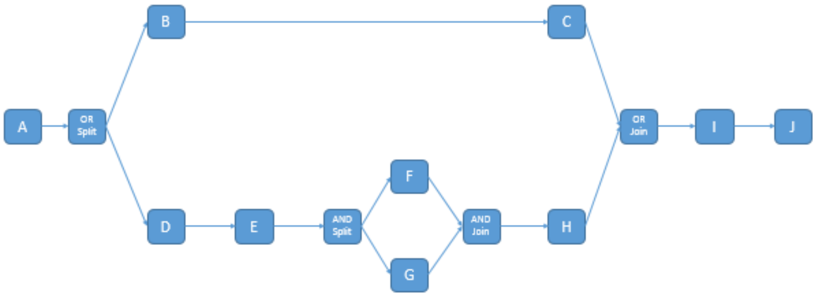


Figure 22. Discovered model with parallel gate

The formalization of business process obtained in Figure 22 is written as below,

W = {*S, A, N, ∑, α, E*} where:

S = {A}

E = {J}

A = {A, B, C, D, E, F,G, H, I, J}

N = $\{n_A, n_B, n_C, n_D, n_E, n_F, n_G, n_H, n_I, n_J\}$

∑ = {a, b, c, d, e, f, g, h, i, j, k, x}

α = {

Sequence

$[n_B, c, n_C]$, $[n_D, d, n_E]$, $[n_I, i, n_J]$

Parallel split & synchronization

$[n_E, e, n_F n_G]$, $[n_F n_G, f, n_H]$

Multi Choice

$[n_A, a, n_B]$, $[n_A, b, n_D]$, $[n_A, j, n_B n_D]$

Synchronizing Merge

$[n_C, g, n_I]$, $[n_H, h, n_I]$, $[n_C n_H, k, n_I]$

}

We compare the formalization in this step with formalization of standard procedure in step 2 and obtaining the same result. If there's a difference between formalization of standard procedure and obtained model, then the formalization of standard procedure model is the precise one.

4. Conclusion

In this paper, we proposed methods to determine process model by timed-based process mining and control-flow patterns. First, we discover sequence relation activities and parallel relation (AND or OR conditional) activities by timed-based process mining. After we discover relations, we model the relations to graph. Second, we re-model the graph to control-flow pattern. Third, we compare formalization of control-flow patterns in discovered graph and formalization of control-flow patterns in the business process model to confirm the right activity relations in graph. Confirmation is used to clear up the noise relation which has same frequency with non-noise relation. Finally, we obtain parallel relation of discovered process model is same as the parallel relation of business process model.

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] Adnen Sanaa, Samir Ben Abid, Abdennacer Boulila, Chokri Messaoud, Mohammed Boussaid, Najeh Ben Fadhel. Modelling hydrochory effects on the Tunisian island populations of Pancratium maritimum L. using colored Petri nets. *BioSystems.* 2015; 129: 19-24.

[3] Jie Yuan, David Oswald, Wei Li. Autonomous traking of chemical plumes developed in both diffusive and turbulent airflow environment using Petri nets. *Expert Systems with Application.* 2015; 42: 527-538.

[4] Victor RL, Shen Horng-Yih Lai, Ah-Fur Lai. The implementation of a smartphone-based fall detection system using a high-level fuzzy Petri net. *Applied Soft Computing.* 2015; 26: 390-400.

[5] R Sarno, Rahadian Dustrial Dewandono, Tohari Ahmad, Mohammad Farid Naufal, Fernandes Sinaga. Hybrid Association Rule Learning and Process Mining for Fraud Detection. *IAENG International Journal of Computer Science.* 2015; 42(2): 59-72.

[6] Wvd Aalst. Process Mining-Discovery, Conformance, and Enhancement of Business Processes. Netherlands, Springer. 2011: 95-125.

[7] Sofie De Cnudde, Jan Claes, Geert Poels. Improving the quality of the Heuristics Miner in Prom 6.2. *Expert Systems with Applications.* 2014; 41: 7678-7690.

[8] Riska A Sutrisnowati, Hyerim Bae, Dongha Lee, Minsoo Kim. *Process Model Discovery Based On Activity Lifespan*. in International Conference on Technology Innovation and Industrial Management. Seoul. 2014.

[9] Wvd Aalst, A Weijters. *Process mining with the heuristic miner algorithm*. In BETA Working Paper Series, WP 166. Eindhoven. 2006.

[10] N Mulyar, Wvd Aalst, HtH Arthur, N Russel. Towards a WPSL: A Critical Analysis of the 20 Classical Workflow Control-flow Pattern. *BPM Center Report.* 2006: 1-66.

[11] R Sarno, WA Wibowo, D Sunaryono, A Munif. *Developing Workflow Patterns Based on Functional Subnets and Control–Flow Patterns.* Proceeding of International Conference on Science in Information Technology (ICSITech). 2015.

[12] R Sarno, WA Wibowo, NF Ariyani, D Manfaat, MS Arif, I Baihaqi. *Context Sensitive Grammar for Composing Business Process Model Variants.* Proceeding of International Conference on Science in Information Technology (ICSITech). 2015.

[13] R Sarno, D Sunaryono, H Ginardi, PLI Sari. *Decision Mining for Multi Choice Workflow Patterns.* International Conference on Computer, Control, Informatics and Its Applications (IC3INA). 2013.