International Review on Computers and Software (I.RE.CO.S.), Vol. 11, N. 6
ISSN 1828-6003 June 2016

# Coupled Hidden Markov Model for Process Mining of Invisible Prime Tasks

Riyanarto Sarno[1], Kelly R. Sungkono[2]

**Abstract** – *Process mining provides process improvement in a variety of application domains. A primary focus of process mining is transferring information from event logs into process model. One of the issues of process mining is dealing with invisible prime tasks. An invisible prime task is an additional task in the process model to assist in showing real processes. However, a few of algorithm solves the issue. This research proposes an algorithm for dealing with invisible prime tasks. The proposed algorithm contains rules and equations utilizing probability of state transition of Coupled Hidden Markov and double time-stamped in event logs. The rules and equations are used for determining invisible prime tasks and parallel control-flows patterns. In addition to dealing with invisible prime tasks, the experiment results also show that the proposed algorithm obtains right parallel control-flow patterns from non-complete event logs. This proposed algorithm also decreases usage of the invisible prime task in A# algorithm without reducing the quality of discovered process models. It has proven with the fitness of process models obtained by the proposed algorithm are relatively high as those obtained by A# algorithm.* ***Copyright © 2016 Praise Worthy Prize S.r.l. - All rights reserved.***

***Keywords**: Coupled Hidden Markov Model, Double Time-stamped Event Log, Fitness, Invisible Prime Tasks, Process Mining*

## Nomenclature

| | |
|---|---|
| $\pi$ | Initial state probability vector |
| $a_{ij}$ | Probability of transitioning from state $i$ to state $j$ |
| *aot* | Number of activities in event log |
| *at* | Number of accurated traces (number of traces which are appropiate with process model) |
| *avgPP* | Average of PP as a threshold for differentiating parallel control-flow patterns |
| $A$ | State transition probability matrix |
| $b_j^h(k)$ | Probability of observation $k$ depends on state $j$ for Hidden Markov Model $h$ in Coupled Hidden Markov Model |
| $B$ | Observation probability matrix |
| *DP* | Pairs of activities which have relation between each other (Dependency Pairs) |
| *ea* | Number of excessing activities at the endpoint of the trace |
| $FT(X_t)$ | Finish time of activity at time = $t$ |
| $h$ | Type of Hidden Markov Model in Coupled Hidden Markov Model |
| *ma* | Number of missing activities from the trace |
| *minPP* | Minimal value of PP as a threshold for differentiating parallel control-flow patterns |
| $n_{nextstate(i)}$ | Total of state $i$ to other states wherein there are probability of transitioning from state $i$ to state j |
| $n_S$ | Total of state in Coupled Hidden Markov Model |
| *NDP* | Pairs of activities which have not relation between each other (Non-Dependency Pairs) |
| RM $\left(S_i,\left[S_j,S_k,...,S_n\right]\right) = \frac{A_{jj}+A_{jk}...+A_{nn}}{n[S]}$ | Relation measure of state $S_i$ as a parameter to differentiate the parallel control-flow patterns |
| $SDm(x)$ | The percentage of activities in traces of event log $x$ depicted in process models (Specific Depiction Measure of event log $x$) |
| $S_j^h$ | State at time=j for Hidden Markov Model $h$ in Coupled Hidden Markov Model |
| $S_N$ | The end state in Coupled of Hidden Markov Model |
| $ST(X_{t+1})$ | Start time of activity at time $t$+1 |
| $t$ | Number of traces in equation $Dm(x)$ or time in other equations |
| $q_t$ | State at time $t$ |
| $S_i \not\rightarrow S_j$ | No relation from state $i$ to state $j$ |


DOI: 10.15866/irecos.v11i6.9555

| | |
|---|---|
| $S_i \rightarrow S_j$ | Relation of state $i$ to state $j$ |
| Z | State $j$ which has a probability of transtioning from state $i$ |

# I. Introduction

Nowadays, information systems record events of their processes. Process mining is a discipline for discovering a process model by observing the events from information systems automatically [1]. The discovered process models are guidances for analyzing performance of processes in the applications [2], [3]. The discovered process models are also guidances for finding solution of complex issues in activities. These issues occur in any fields, especially business [4], and fraud [5].

There are a variety of issues in discovering a process model. The several issues are determining parallel control-flow patterns and dealing with invisible prime tasks. Invisible prime tasks are additional tasks in process models to assist in showing real processes. Invisible prime tasks did not appear in event logs.

There are many situations to leading process models using invisible prime tasks [6]. The first situation is many processes allow skip, redo or jump current tasks and the processes need tasks to exhibit that situation. The second situation is process models need present more than one control-flows containing same activities.

The example of second situation is task B and task C must be executed parallelly after task A, but sometimes task C is executed after task B. By evaluating the situation, there are two parallel control-flow patterns containing task A. The control-flow patterns are XOR relation between task A and task B,C,D and AND relation between task A and task B,D. Invisible prime tasks are used for linking that control-flows.

A few of researches discuss about dealing with invisible prime tasks. A# algorithm is the most detailed algorithm for solving that problem. A# algorithm separates invisible prime tasks into five types. The types are initialize, redo, skip, switch and finalize.

Initialize and finalize invisible prime tasks are invisible prime tasks which solve the second situation. The initialize invisible prime task is used if beginning task is involved. Finalize invisible prime task is used if ending task is involved [6].

Skip, redo and switch invisible prime tasks are invisible prime tasks which solve the first situation.

The skip invisible prime task is used for skipping the execution of one or more tasks.

The redo invisible prime task is used for repeating tasks. The switch invisible prime task is used for switching the execution of tasks [6]. Because few of researches suggest a solution of dealing with the invisible prime task, this research is challenged to discuss about that issue. Not only dealing with the invisible prime task, but this research also discusses about obtaining parallel control-flow patterns in non-complete event logs.

This research proposes an algorithm to obtain a process model which deals with invisible prime tasks and non-complete event logs. An event log indicates as non-complete event log if it can not shows parallel relation between activities.

For example, task B and task C are executed parallelly, but the event log only records the execution of task C after the execution of task B, not vice versa. By simply relying on event log, the discovered relation of task B and C is sequence, whereas the real relation is parallel. [7], [8] discuss about double time-stamped utilization for anticipating non-complete event logs. By referring to [7], [8], the proposed algorithm utilizes double time-stamped of event log to deal with non-complete event logs. Coupled Hidden Markov Model is widely used in many issues, i.e. audio-visual speech recognition [9], [10], bearing fault recognition [11], biosignal interaction [12] and social network [13].

However, no research utilizes Coupled Hidden Markov Model in process mining. The previous research, [14], suggests an algorithm to discover parallel control-flow patterns using probability of state transition of Hidden Markov Model.

This research suggests rules and equations in a proposed algorithm to discover parallel control-flow patterns and invisible prime tasks using probability of state transition and observations in Coupled Hidden Markov Model. In addition to discovering process model which deal with the invisible prime task and non-complete event log, evaluating a discovered process model is also important.

The evaluation is based on the fitness. A process model has good value of fitness if all traces in event logs are represented in the process model.

This research is constructed as follows: Section II presents Coupled Hidden Markov Model for Activity Relation Determination. This section also reviews high-level Petri nets as literature for determining Coupled Hidden Markov Model and equations to evaluate the fitness of a discovered process model. Section III describes the proposed algorithm. Section IV reports the steps and final outputs of the experimental process. The last section, Section V, presents the conclusions of the research.

# II. Research Method

Coupled Hidden Markov Model for Activity Relation Determination is explained in this section. A collection of event logs which has double time-stamped is data for determining the Coupled Hidden Markov Model. The double time-stamped contains of start time and end time of activities in each case of event log. This double-time stamped are used for discovering sequential and parallel conditions. As a literature for determining Coupled Hidden Markov Model, high-level Petri net is described in this section.

This section also explains the equations to measure quality of the discovered process model. Calculating fitness is chosen as the quality measurement of process models.

### II.1. High-level Petri Nets

Petri net is one of the representations of the process model. Petri net connects some places with some transitions forming a two-way graph. Each place has one or more transitions which occur if their place traversed [1]. There are two types of Petri net. They are classical Petri nets and high-level Petri nets.

The detail of those types is explained in [15]. This research only discusses about high-level Petri nets as literature for determining states and observations in Coupled Hidden Markov Model. High-level Petri net is extended by color [15]. The color is a representation of attribute in tokens. A transition of a Petri net is enabled if a token traverse its input place. The color is utilized in deciding traversed place in choice condition.

Petri net is divided into four types of routing for modeling relations among the activities. The types are sequential routing, parallel routing, conditional routing and iteration routing [15]. Sequential routing occurs if an activity has relation with one activity. The example of sequential routing is displayed in Fig. 1. Activity A and activity B are included in sequential routing because activity B only has relation to activity A and vice versa. Place p1 is post-condition of activity A and pre-condition of activity B. It is also the model of relationship between activity A and B if a token signified by a dot traverse.

Parallel routing occurs if the activities must be executed, but the order of execution can be changeable. Activity B and C are the example of parallel routing in Fig. 2. Conditional routing occurs if the order of execution depends on attributes from previous activities.

By observing Fig. 3, activity B and C is conditional routing. *p2* as an input place of activity B and *p3* as an input place of activity C are traversed by a token if the color of token qualifies the condition of those places.

The conditions are signified by x in Fig. 3. Iteration routing occurs if some activities are executed more than one time in one process. The example of iteration routing is displayed in Fig. 4.

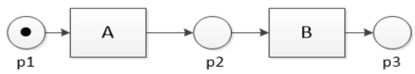


Fig. 1. Sequential Routing

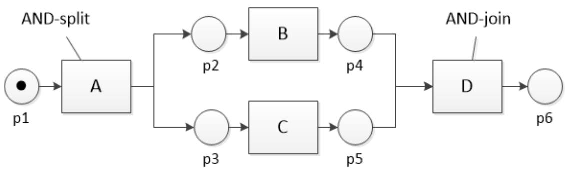


Fig. 2. Parallel Routing

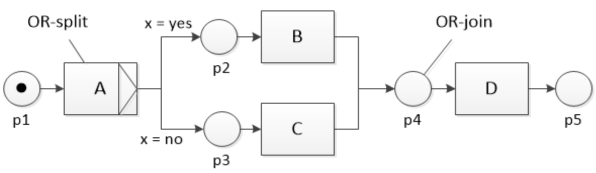


Fig. 3. Conditional Routing

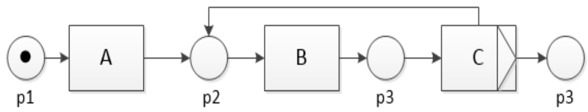


Fig. 4. Iteration Routing

### II.2. Coupled Hidden Markov Model of Activity Relation Determination

Coupled Hidden Markov Model is an extension model of Hidden Markov Model. Coupled Hidden Markov Model consists of a multiple Hidden Markov Model wherein all of states from each Hidden Markov Model are dependent on the states of all Hidden Markov Model in previous time slice [11].

Initialization of Hidden Markov Model in Coupled Hidden Markov Model is *h*= [1, 2]. Each Hidden Markov Model has a set of states $S^h = \left(S_1^h, S_2^h, ..., S_n^h\right)$ and a sequence of observations $O_t = \left(O_t^h\right) = \left(O_t^1, O_t^2\right)$ [11].

The elements of Coupled Hidden Markov Model are defined in Eq. (1):

$$CHMM = \left(\pi, A, B\right) \tag{1}$$

Coupled Hidden Markov Model has one or many initial states. π is an initial state probability vector. The π contains probabilities of initial states in $S_i = S_i^h$ .

That probability is symbolized by $\pi_i$. The $\pi_i$ is defined in Eq. (2):

$$\pi_i = P\left(q_1 = S_i\right) = \prod_{h=1}^{2} P\left(q_1^h = S_i^h\right) \tag{2}$$

A is a state transition probability matrix. A contains probabilities of transitioning from states $S_i = S_i^h$ to states $S_j = S_j^h$ [6].

The probability of transitioning is symbolized by $a_{ij}$ and is defined in Eq. (3):

$$a_{ij} = P\left(q_{t+1} = S_j \mid q_t = S_i\right) = \prod_{h=1}^{2} P\left(S_j^h \mid S_i^h\right) \tag{3}$$

B is an observation probability matrix. B contains the probabilities of observations which depend on certain states. The $b_j\left(O_t\right)$ which is a symbolized probability of observation is defined in Eq. (4):

$$b_j\left(O_t\right) = \prod_{h=1}^{2} P\left(O_t^h \mid q_t^h = S_j^h\right) \tag{4}$$

This research builds Coupled Hidden Markov Model which is adapted from Petri net explained in Section II.1.

The observations of Coupled Hidden Markov Model are attributes of activities and the name of activities. Each Hidden Markov Model in Coupled Hidden Markov Model has same states. The states are places of Petri net.

By considering Petri net in Section II.1, the number of places is same as the number of activities. It is because every activity has own input places. The attributes as the observation are depend on the next places of Petri net as the states. The additional attribute named 'anonym' is depend on the place which has no attributes.

The illustration about Coupled Hidden Markov Model of activity relation determination is presented in Fig. 5.

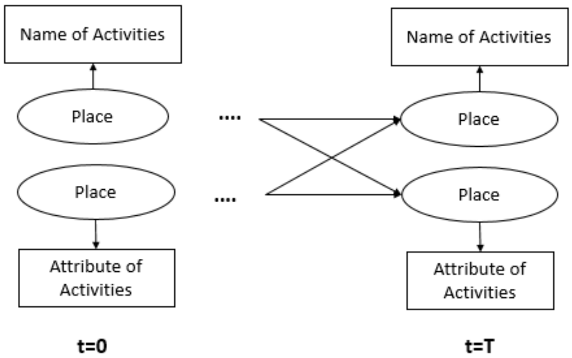


Fig. 5. Coupled Hidden Markov Model of Activity Relation Determination

The initial guess of states and observations probabilities are mined from traces. This research uses 20% of traces in the event log.

### II.3. The Fitness of Discovered Process Models

The quality of process discovery can be assessed by evaluating their discovered outputs, i.e. process models. Evaluating process discovery refers to comparing discovered process models from different algorithms. The final decision determines the best algorithm among the algorithms [16]. Process mining reports have not yet defined the generic construction to evaluate the process models. However, the validity and the fitness are already defined as the domain to evaluate process models.

The fitness is a quality measurement of the discovered process model, by calculating the percentage of recognized traces included in the discovered process model. The high value of fitness shows that a lot of the traces are already depicted, whereas the low value of fitness shows that a few of the traces are depicted [16].

Reference [16], there are equations for calculating the fitness of the process model. This research calls the equations as Depiction Measure (*Dm*) and Specific Depiction Measure (*SDm*). *Dm* is the percentage of traces depicted in process models and *SDm* is the percentage of activities in traces depicted in process models. *Dm* is earned by dividing the accurate traces with the total of traces and *SDm* is earned by dividing the accurate activities with the total of activities in traces. The *Dm* is defined in Eq. (5) and the *SDm* is defined in Eq. (6):

$$Dm(x) = \frac{at}{t} \tag{5}$$

$$SDm(x) = \frac{1}{2}\frac{(aot - ma)}{aot} + \frac{1}{2}\frac{(aot - ea)}{aot} \tag{6}$$

## III. Proposed Method

This section explains the details of the proposed algorithm. It is divided into 2 steps: predicting the probabilities of Coupled Hidden Markov Model and determining process model.

### III.1. Predicting the Probabilities of Coupled Hidden Markov Model

Coupled Hidden Markov Model is trained with the Baum-Welch algorithm. This research modifies Baum-Welch algorithm by adding equation at the end of Baum-Welch algorithm.

The additional equation is used to handle non-complete event log. Non-complete event log occurs if two activities have parallel relations, but only one side of dependency relations of those activities are restored in the event log. For example, activity B and C actually have parallel relations, but the event log only restore the execution from activity B to activity C.

If only considering the event log, activity B and C have sequential relations, not parallel relations. This additional equation shows the non-visible relation of activities by utilizing double-time stamped.

The additional equation is described in Eq. (7):

$$\begin{gathered}\text{If } ST(X_{t+1}) \geq FT(X_t) \text{ and } a_{ji} = 0, \text{then:} \\ a_{ji} = a_{ij}, a_{kj} = a_{ki}, a_{il} = a_{jl}\end{gathered} \tag{7}$$

This research gives a simplified example. There are traces, i.e. ABDE and ACE, have been observed.

The ABDE appears once and ACE appears twice. The start time of event D in the event log is before the finish time of event B. Event A has attribute 'no' in trace ACE and has attribute 'yes' in trace ABDE.

Considering to the traces, the research discovered five events, such as A, B, C, D, E. These events are the observations as name of activities. Because the events except event A did not have attributes, so the attribute of those activities is 'anonym'.

Under Section II.2, a number of states are five. They are Place 1 (P1), Place 2 (P2), Place 3 (P3), and Place 4 (P4) and Place 5 (P5). P1 has event A as its observation. P2 has event B as its observation. P3 has event C as its observation. P4 has D as its observation. P5 has E as its observation. P1, P4 and P5 have 'anonym' as their observations. P2 has 'yes' and P3 has 'no' as their observations. A new Coupled Hidden Markov Model of simplified example obtained by Modified Baum-Welch algorithm is displayed in Table I until Table IV.

TABLE I
THE INITIAL STATE PROBABILITY OF NEW COUPLED HIDDEN MARKOV MODEL

| πn | |
|---|---|
| State | Probability |
| P1 | 1 |

TABLE II
THE STATE TRANSITION PROBABILITY MATRIX OF NEW COUPLED HIDDEN MARKOV MODEL

| An | | | | | |
|---|---|---|---|---|---|
| Beginning state | Next State | Probability | Beginning state | Next State | Probability |
| P1 | P2 | 0.25 | P2 | P5 | 0.49 |
| P1 | P3 | 0.5 | P3 | P5 | 1 |
| P1 | P4 | 0.25 | P4 | P2 | 0.51 |
| P2 | P4 | 0.51 | P4 | P5 | 0.49 |

TABLE III
THE OBSERVATION PROBABILITY MATRIX OF NEW COUPLED HIDDEN MARKOV MODEL FROM NAME OF ACTIVITIES

| Bn | | | | | |
|---|---|---|---|---|---|
| State | State | State | State | State | State |
| P1 | P1 | P1 | P1 | P1 | P1 |
| P2 | P2 | P2 | P2 | P2 | P2 |
| P3 | P3 | P3 | P3 | P3 | P3 |

TABLE IV
THE OBSERVATION PROBABILITY MATRIX OF NEW COUPLED HIDDEN MARKOV MODEL FROM ATTRIBUTE OF ACTIVITIES

| Bn | | | | | |
|---|---|---|---|---|---|
| State | Observation | Probability | State | Observation | Probability |
| P1 | 'anonym' | 1 | P4 | 'anonym' | 1 |
| P2 | 'yes' | 1 | P5 | 'anonym' | 1 |
| P3 | 'no' | 1 | | | |

### *III.2. Determining Process Model*

There are four conditions in the process model. The four conditions are sequence condition, parallel condition, loop condition and invisible prime task condition. Sequence condition appears when the state has dependency relations with one state. Sequence condition shows sequence control-flow pattern which is called sequence relation in the process model.

Parallel condition appears when the state has dependency relations with more than one state. Parallel condition shows parallel control-flow patterns in the process model. Parallel control-flow patterns are XOR, OR and AND relation. Loop condition appears when the state has dependency relations with itself or with its previous state. In many researches like [16], [17], the loop conditions are divided in 2 parts: length one loop and length two loop. Length one loop appears when the state has dependency relation with itself and length two loop appears when the state has dependency relation with its previous state.

The loop conditions are classified as sequence conditions in process model. Invisible prime task condition appears when process model needs invisible prime tasks to modeling the real process. A# algorithm determines invisible prime tasks into five types. The detail of types is described in [6]. The proposed rules are used for determining the relations in all conditions.

TABLE V
RULES OF DETERMINING ACTIVITY RELATIONS

| Number | Rules |
|---|---|
| 1 | Counting the beginning state and the next state for each state from the state transition probability matrix of Coupled Hidden Markov Model. If the beginning state is more than one, then doing step 2. If the next state is more than one, then doing step 6. If the beginning and next state are one, then doing step 10. |
| 2 | Determining Dependency Pairs (DP) described in Equation (9) and Non-Dependency Pairs (NDP) described in Equation (10) of beginning state.<br>If it has DP and the total of DP is less than NDP, then doing step 3.<br>If it has NDP and the total of NDP is less than DP, then doing step 4.<br>Else, doing step 5. |
| 3 | $S_i$ $\rightarrow$ [Invisible prime task, $S_j$ in NDP $\wedge$ not in DP] = XOR.<br>Invisible prime task $\rightarrow$ [ $S_j$ in DP $\wedge$ not in NDP] = Equation Relation. |
| 4 | $S_i$ $\rightarrow$ [Invisible prime task, $S_j$ in NDP $\wedge$ not in DP] = AND.<br>Invisible prime task $\rightarrow$ [ $S_j$ in DP $\wedge$ not in NDP] = XOR. |
| 5 | Doing Equation (10) until (12). |
| 6 | Determining Dependency Pairs (DP) described in Equation (8) and Non-Dependency Pairs (NDP) described in Equation (9).<br>If it has DP and the total of DP is less than NDP, then doing step 7.<br>If it has NDP and the total of NDP is less than DP, then doing step 8.<br>Else, doing step 9. |
| 7 | [Invisible prime task, $S_j$ in NDP $\wedge$ not in DP] $\rightarrow$ $S_i$ = XOR.<br>[ $S_j$ in DP $\wedge$ not in NDP] $\rightarrow$ Invisible prime task = Equation Relation.<br>[ $S_j$ in DP $\wedge$ not in NDP] $\rightarrow$ $S_i$ = [ $S_j$ in DP $\wedge$ not in NDP] $\rightarrow$ Invisible prime task |
| 8 | [Invisible prime task, $S_j$ in DP $\wedge$ not in NDP] $\rightarrow$ $S_i$ = AND.<br>[ $S_j$ in NDP $\wedge$ not in DP] $\rightarrow$ Invisible prime task = XOR.<br>[ $S_j$ in NDP $\wedge$ not in DP] $\rightarrow$ $S_i$ = [ $S_j$ in NDP $\wedge$ not in DP] $\rightarrow$ Invisible prime task |
| 9 | State $\rightarrow$ Next State = Equation Relation |
| 10 | $S_i$ $\rightarrow$ $S_j$ = sequence. |

The rules are described in Table V:

$$DP = \left[ S_j, S_k \right] \Leftrightarrow \left( a_{jk} > 0 \ \vee \ a_{kj} > 0 \right) \tag{8}$$

$$NDP = \left[S_j, S_k\right] \Leftrightarrow a_{jk} = 0 \tag{9}$$

$$\begin{aligned} &\mathit{if}\ \left(a_{ij} > 0 \wedge S_i \not\to S_j\right) \vee a_{ii} \neq 0 \\ &\mathit{then}\ \left(S_i \to S_j\right) \vee \left(S_i \to S_i\right) = \mathit{sequence} \end{aligned} \tag{10}$$

$$\begin{aligned} &\mathit{if}\ n_{nextstate(i)} = n_S - 1, \mathit{then} \\ &\begin{pmatrix} S_i \to \left[\mathit{Invisible\ Task}\right], \\ S_j \neq S_N \end{pmatrix} = \text{Equation Relation} \\ &\text{and } \mathit{Invisible\ Task} \to S_N = \mathit{sequence} \end{aligned} \tag{11}$$

$$\mathit{else}\ S_i \to S_j = \text{Equation Relation} \tag{12}$$

Equation Relation is used for determining parallel condition.

By referring to [14], the equation relation is described in Eq. (13) until (18):

$$\text{RM}\left(S_i, \left[S_j, S_k, \ldots, S_n\right]\right) = \frac{a_{jj} + a_{jk} \ldots + a_{nn}}{n_S - 1} \tag{13}$$

$$\text{avgPP} = \frac{\sum_{j=1}^{Z} a_{ij}}{Z} \tag{14}$$

$$\text{minPP} = \min_{j=1}^{Z}\left(a_{ij}\right) \tag{15}$$

$$\mathit{if}\ \text{Relation Measure}\left(S_i\right) \leq \text{minPP then XOR} \tag{16}$$

$$\text{if minPP} < \text{Relation Measure}\left(S_i\right) < \text{avgPP then OR} \tag{17}$$

$$\text{if avgPP } \leq \text{Relation Measure}\left(S_i\right) \text{ then AND} \tag{18}$$

According to the example in Section III.1, this research determines relations based on rules in Table V.

The research gives the step by step of determining the relation of P1. Because P1 has more than one next place based on Table II, Eq. (8) in rule (1) is fulfilled so the step 2 is observed. DP of P1 are [P2, P4], [P4, P2] and NDP of P1 are [P2, P3],[P3, P2], [P3, P4],[P4, P3].

Total of DP of P1 is 2 and total of NDP of P1 is 4. Based on this information, rule (3) is chosen. P1 qualifies rule (2) and (3), so the result are P1 $\to$ [Invisible prime task,P3] and [Invisible prime task, [P2, P4]. The invisible prime task is signified by 0. The other places are also observed based on Table V. The final relations of activities are described in Table VI. Building the process model by changing the states in table relation of activities with their depending observations as name of activities.

The process model of simplified example is displayed in Fig. 6.

TABLE VI
RELATION OF ACTIVITIES

| An | | | | | |
|---|---|---|---|---|---|
| Beginning State | Next State | Relation | Beginning State | Next State | Relation |
| P1 | 0,P3 | XOR | P4 | 1 | Sequence |
| 0 | P2,P4 | AND | P3,1 | P5 | XOR |
| P2 | 1 | Sequence | P2,P4 | 1 | AND |
| P3 | P5 | Sequence | | | |

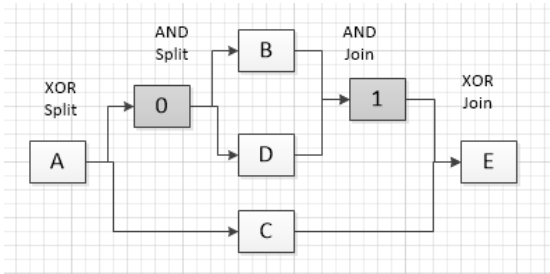


Fig. 6. The process model

# IV. Results and Analysis

## IV.1. Experiment Data

This research uses many event logs for determining invisible prime tasks explained in [11]. This research explains step by step from one of the event log.

The piece of the event log is displayed in Fig. 7.

| Case ID | Activity | Start Stamp | End Stamp | Attribute |
|---|---|---|---|---|
| PP1 | A | 6/20/2014 10:32 | 6/20/2014 13:42 | Accepted |
| PP1 | C | 6/20/2014 13:42 | 6/20/2014 16:52 | |
| PP1 | E | 6/20/2014 16:52 | 6/20/2014 20:02 | |
| PP1 | F | 6/20/2014 18:27 | 6/20/2014 21:37 | |
| PP1 | G | 6/20/2014 21:37 | 6/21/2014 0:47 | |
| PP2 | A | 6/21/2014 0:47 | 6/21/2014 3:57 | Rejected |
| PP2 | B | 6/21/2014 3:57 | 6/21/2014 7:07 | |
| PP2 | D | 6/21/2014 5:32 | 6/21/2014 8:42 | |
| PP2 | G | 6/21/2014 8:42 | 6/21/2014 11:52 | |
| PP3 | A | 6/21/2014 11:52 | 6/21/2014 15:02 | Rejected |
| PP3 | B | 6/21/2014 15:02 | 6/21/2014 18:12 | |
| PP3 | D | 6/21/2014 16:37 | 6/21/2014 19:47 | |
| PP3 | G | 6/21/2014 19:47 | 6/21/2014 22:57 | |
| PP28 | A | 7/4/2014 21:17 | 7/5/2014 0:27 | Accepted |
| PP28 | C | 7/5/2014 0:27 | 7/5/2014 3:37 | |
| PP28 | F | 7/5/2014 3:37 | 7/5/2014 6:47 | |
| PP28 | E | 7/5/2014 5:12 | 7/5/2014 8:22 | |
| PP28 | G | 7/5/2014 8:22 | 7/5/2014 11:32 | |
| PP29 | A | 7/5/2014 11:32 | 7/5/2014 14:42 | Accepted |
| PP29 | C | 7/5/2014 14:42 | 7/5/2014 17:52 | |
| PP29 | F | 7/5/2014 17:52 | 7/5/2014 21:02 | |
| PP29 | E | 7/5/2014 19:27 | 7/5/2014 22:37 | |
| PP29 | G | 7/5/2014 22:37 | 7/6/2014 1:47 | |

Fig. 7. The piece of the event log

## IV.2. Process Discovery using Proposed Algorithm

This section describes the steps of process discovery using the proposed algorithm with the event log presented in Section IV.1.

### IV.2.1. Predicting the Probabilities of Hidden Markov Model

The detail of determining observations and states is explained in Section III.

This research uses trace PP1 until PP6 as training data. By considering training data, the observations of Coupled Hidden Markov Model are described in Table VII and Table VIII.

TABLE VII
THE OBSERVATION PROBABILITY MATRIX OF NEW COUPLED HIDDEN MARKOV MODEL FROM NAME OF ACTIVITIES

| State | Observation | State | Observation |
|---|---|---|---|
| P1 | A | P5 | E |
| P2 | B | P6 | F |
| P3 | C | P7 | G |
| P4 | D | | |

TABLE VIII
THE OBSERVATION PROBABILITY MATRIX OF NEW COUPLED HIDDEN MARKOV MODEL FROM ATTRIBUTE OF ACTIVITIES

| State | Observation | State | Observation |
|---|---|---|---|
| P1 | 'anonym' | P5 | 'anonym' |
| P2 | 'Rejected' | P6 | 'anonym' |
| P3 | 'Accepted' | P7 | 'anonym' |
| P4 | 'anonym' | | |

By utilizing all trace in event logs described in Section IV.1, Coupled Hidden Markov Model is obtained by Modified Baum-Welch algorithm.

The Coupled Hidden Markov Model shows in Table IX until Table XII.

TABLE IX
THE INITIAL STATE PROBABILITY OF NEW COUPLED HIDDEN MARKOV MODEL

| πn | |
|---|---|
| State | Probability |
| P1 | 1 |

TABLE X
THE STATE TRANSITION PROBABILITY MATRIX OF NEW COUPLED HIDDEN MARKOV MODEL

| An | | | | | |
|---|---|---|---|---|---|
| Beginning state | Next State | Probability | Beginning state | Next State | Probability |
| P1 | P2 | 0.25 | P4 | P2 | 0.5 |
| P1 | P3 | 0.5 | P4 | P7 | 0.5 |
| P1 | P4 | 0.25 | P5 | P6 | 0.5 |
| P2 | P4 | 0.5 | P5 | P7 | 0.5 |
| P2 | P7 | 0.5 | P6 | P5 | 0.5 |
| P3 | P5 | 0.5 | P6 | P7 | 0.5 |
| P3 | P6 | 0.5 | | | |

TABLE XI
THE OBSERVATION PROBABILITY MATRIX OF NEW COUPLED HIDDEN MARKOV MODEL FROM NAME OF ACTIVITIES

| Bn | | | | | |
|---|---|---|---|---|---|
| State | Observation | Probability | State | Observation | Probability |
| P1 | A | 1 | P5 | E | 1 |
| P2 | B | 1 | P6 | F | 1 |
| P3 | C | 1 | P7 | G | 1 |
| P4 | D | 1 | | | |

TABLE XII
THE OBSERVATION PROBABILITY MATRIX OF NEW COUPLED HIDDEN MARKOV MODEL FROM ATTRIBUTE OF ACTIVITIES

| Bn | | | | | |
|---|---|---|---|---|---|
| State | Observation | Probability | State | Observation | Probability |
| P1 | 'anonym' | 1 | P5 | 'anonym' | 1 |
| P2 | 'Rejected' | 1 | P6 | 'anonym' | 1 |
| P3 | 'Accepted' | 1 | P7 | 'anonym' | 1 |
| P4 | 'anonym' | 1 | | | |

*IV.2.2. Building Process Model*

The research follows rules described in Table V for determining relation of activities in process model.

The final relations of activities are described in Table XIII.

TABLE XIII
RELATION OF ACTIVITIES

| An | | | | | |
|---|---|---|---|---|---|
| Beginning State | Next State | Relation | Beginning State | Next State | Relation |
| P1 | 0,P3 | XOR | P5 | 2 | Sequence |
| 0 | P2,P4 | AND | P6 | 2 | Sequence |
| P2 | 1 | Sequence | 1,2 | P7 | XOR |
| P3 | P5,P6 | AND | P2,P4 | 1 | AND |
| P4 | 1 | Sequence | P5,P6 | 2 | AND |

Building the process model by changing the state in table relation of activities with their depending observations as name of activities.

The discovered process model using proposed algorithm is shown in Fig. 8.

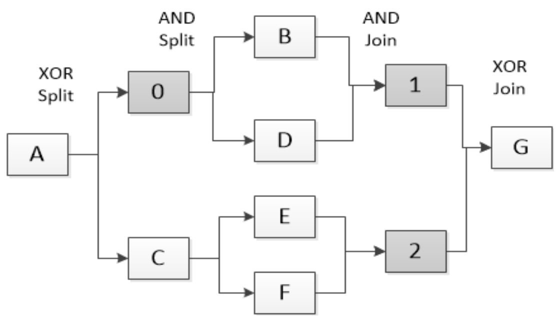


Fig. 8. Discovered Process Model using proposed algorithm

The other event logs as the experiment data are determined in Table XIV.

TABLE XIV
EVENT LOGS

| Event Log | Trace |
|---|---|
| W1 | [ABE],[ADE],[ACE],[ACE],[ABDE] |
| W2 | [ABCD],[AD],[ABBCD], [ABBCD] |
| W3 | [AD],[ABD],[ACD],[ABCD] |
| W4 | [ABDF],[ABEF],[ACDF] |

The discovered process models using proposed algorithm based on event logs in Table XIV are shown in Fig. 9 until Fig. 12.

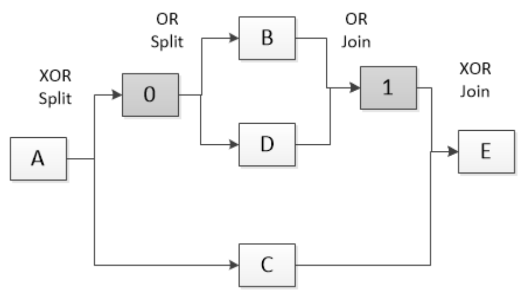


Fig. 9. Discovered Process Model using proposed algorithm based on Event Log W1

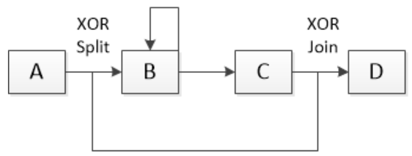


Fig. 10. Discovered Process Model using proposed algorithm based on Event Log W2

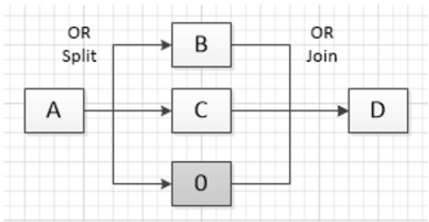


Fig. 11. Discovered Process Model using proposed algorithm based on Event Log W3

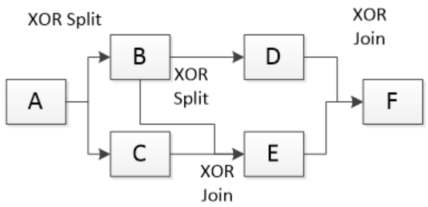


Fig. 12. Discovered Process Model using proposed algorithm based on Event Log W4

### IV.2.3. Process Discovery Using another Algorithm

As a model comparison, this research discovers the process model using A# algorithm [11] with event log from Section IV.1 and Table XIV.

The discovered process models using A# are shown in Fig. 13 until Fig. 17.

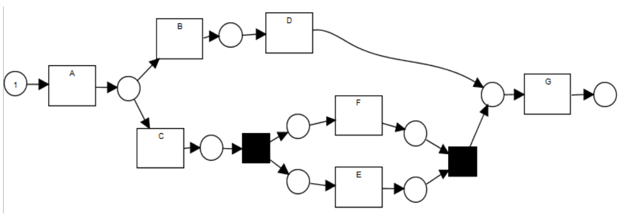

Fig. 13. Discovered Process Model using A# algorithm based on Event Log in Section IV

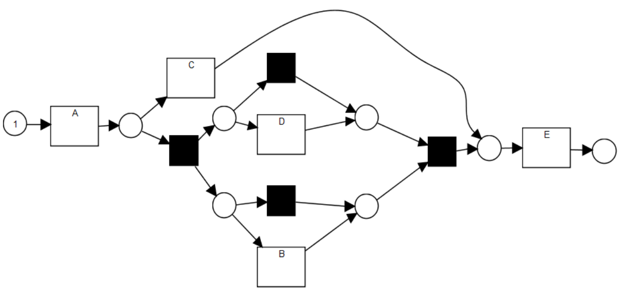

Fig. 14. Discovered Process Model using proposed algorithm based on E0vent Log W1

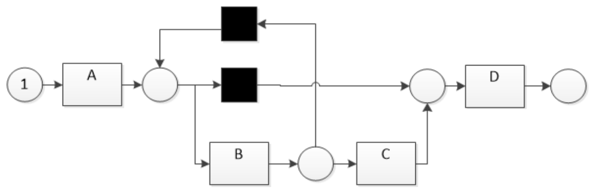

Fig. 15. Discovered Process Model using proposed algorithm based on Event Log W2

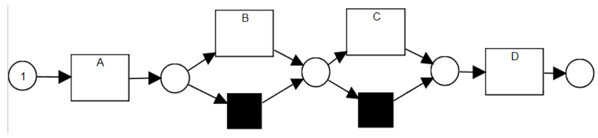

Fig. 16. Discovered Process Model using proposed algorithm based on Event Log W3

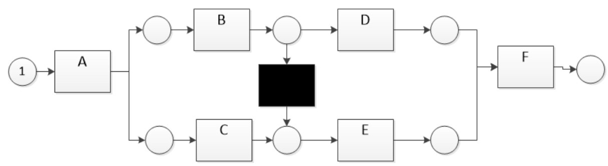

Fig. 17. Discovered Process Model using proposed algorithm based on Event Log W4

### IV.2.4. The Fitness of Discovered Process Models

Reflecting on the fitness equations in Section II.2, the discovered process models from three algorithms can be compared. The two algorithms are the proposed algorithm and A# algorithm. Based on Depiction Measure in Eq. (5) and Specific Depiction Measure in Eq. (6), the fitness values for each algorithm are displayed in Table XV and Table XVI.

TABLE XV
FITNESS VALUES USING THE PARSING MEASURE

| Depiction Measure (Dm) | | | | |
|---|---|---|---|---|
| Event Log | Method/algorithm | Dm | at | t |
| Section IV.1 | Proposed Algorithm | 1 | 30 | 30 |
| | A# algorithm | 1 | 30 | 30 |
| W1 | Proposed Algorithm | 1 | 5 | 5 |
| | A# algorithm | 1 | 5 | 5 |
| W2 | Proposed Algorithm | 1 | 3 | 3 |
| | A# algorithm | 1 | 3 | 3 |
| W3 | Proposed Algorithm | 1 | 4 | 4 |
| | A# algorithm | 1 | 4 | 4 |
| W4 | Proposed Algorithm | 1 | 3 | 3 |
| | A# algorithm | 1 | 3 | 3 |

TABLE XVI
FITNESS VALUES USING THE CONTINUOUS PARSING MEASURE

| Depiction Measure (Dm) | | | | | |
|---|---|---|---|---|---|
| Event Log | Method/algorithm | SDm | Aot | ea | ma |
| Section IV.1 | Proposed Algorithm | 1 | 138 | - | - |
| | A# algorithm | 1 | 138 | - | - |
| W1 | Proposed Algorithm | 1 | 20 | - | - |
| | A# algorithm | 1 | 20 | - | - |
| W2 | Proposed Algorithm | 1 | 12 | - | - |
| | A# algorithm | 1 | 12 | - | - |
| W3 | Proposed Algorithm | 1 | 12 | - | - |
| | A# algorithm | 1 | 12 | - | - |
| W4 | Proposed Algorithm | 1 | 12 | - | - |
| | A# algorithm | 1 | 12 | - | - |

## V. Conclusion

This paper has proposed an algorithm based on Coupled Hidden Markov Model to discover parallel business process from event logs in non-complete event logs. The proposed algorithm consists of three steps. First, the probabilities are estimated by Modified Baum-Welch method. Second, the relations are determined using the proposed rules and equations.

Finally, the process model is established based on the mined relations. The evaluation results showed that the proposed algorithm could illustrate invisible prime tasks in the process model. The invisible prime tasks can make the process model more suitable with the real processes.

This research also could determining parallel relations, i.e. XOR, OR and AND relations, from non-complete event logs. The evaluation results also showed that the usage of invisible prime tasks by proposed algorithm is less than by A# algorithm.

Even though the proposed algorithm used less invisible prime tasks, the fitness of discovered process models obtained by the proposed algorithm is relatively high as those obtained by A# algorithm. This proposed algorithm is promising because there are only few research of process mining utilized Coupled Hidden Markov Model.

## Acknowledgements

We would like to thank the Institut Teknologi Sepuluh Nopember and the Ministry of Research, Technology and Higher Education of Indonesia for supporting this research.

## References

[1] 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/10.4258/hir.2014.20.2.157

[2] N. Y. Setiawan, R. Sarno, Multi-Criteria Decision Making for Selecting Semantic Web Service Considering Variability and Complexity Trade-Off, (2016) *Journal of Theoretical and Applied Information Technology,* 86 (2), pp. 316-326.

[3] Sarno, R., Sari, P. L. I., Ginardi, H., Sunaryono, D., Decision mining for multi choice workflow patterns, *Computer, Control, Informatics and Its Applications (IC3INA)* (Page: 337-342 Year of Publication: 2013).
http://dx.doi.org/10.1109/IC3INA.2013.6819197

[4] Sarno, R., Pamungkas, E. W., Ginardi, H., Sunaryono, D., Clustering of ERP business process fragments, *Computer, Control, Informatics and Its Applications (IC3INA)* (Page: 319 - 324, Year of Publication: 2013).
http://dx.doi.org/10.1109/IC3INA.2013.6819194

[5] R. Sarno, R. D. Dewandono, T. Ahmad, M. F. Naufal, F. Sinaga, Hybrid Association Rule Learning and Process mining for Fraud Detection, (2015) *IAENG International Journal of Computer Science*, 42 (2), pp. 59-72.

[6] L. Wen, J. Wang, W. M. P. van der Aalst, B. Huang, J. Sun, Mining process models with prime invisible prime tasks, (2010) *Data & Knowledge Engineering,* 69 (10), pp. 999-1021. http://dx.doi.org/10.1016/j.datak.2010.06.001

[7] R. Sarno, W. A. Wibowo, Kartini, F. Haryadita, Determining Model Using Non-Linear Heuristics Miner and Control-Flow Pattern, (2016) *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 14 (1), pp. 349-360.
http://dx.doi.org/10.12928/telkomnika.v14i1.3257

[8] Sarno, R., Effendi, Y., Haryadita, F., Modified Time-Based Heuristics Miner for Parallel Business Processes, (2016) *International Review on Computers and Software (IRECOS)*, 11 (3), pp. 249-260.
http://dx.doi.org/10.15866/irecos.v11i3.8717

[9] Nefian, Liang, L., Pi, X., Xiaoxiang, L., Mao, Murphy, A coupled HMM for audio-visual speech recognition, *IEEE International Conference on Acoustics Speech and Signal Processing* (Year of Publication: 2002).
http://dx.doi.org/10.1109/icassp.2002.1006167

[10] A H. Abdelaziz, S. Zeiler, D. Kolossa, Learning Dynamic Stream Weights for Coupled-HMM-based Audio-visual Speech Recognition, (2015) *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* 1 (1).
http://dx.doi.org/10.1109/taslp.2015.2409785

[11] H. Zhou, J. Chen, G. Dong, H. Wang, H. Yuan, Bearing fault recognition method based on neighbourhood component analysis and coupled hidden Markov model, (2016) *Mechanical Systems and Signal Processing,* 66, pp. 568-581.
http://dx.doi.org/10.1016/j.ymssp.2015.04.037

[12] Rezek, I., Roberts, S.J., Estimation of coupled hidden Markov models with application to biosignal signal interaction modelling, *Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop* (Year of Publication: 2000).
http://dx.doi.org/10.1109/nnsp.2000.890160

[13] Raghavan, V., ver Steeg, G., Galstyan, A., Tartakovsky, A.G., Coupled hidden markov models for user activity in social networks, *IEEE International Conference on Multimedia and Expo Workshops* (Year of Publication: 2013).
http://dx.doi.org/10.1109/icmew.2013.6618397

[14] Sarno, R., Sungkono, K., Hidden Markov Model for Process Mining of Parallel Business Processes, (2016) *International Review on Computers and Software (IRECOS)*, 11 (4), pp. 290-300.
http://dx.doi.org/10.15866/irecos.v11i4.8700

[15] W. M. P. van der Aalst, The application of Petri nets to Workflow Management, (1998) *Journal of Circuits, Systems and Computers*, 8 (1), pp. 21-66.
http://dx.doi.org/10.1142/s0218126698000043

[16] S. De Cnudde, J. Claes, G. Poels, Improving the Quality of the Heuristics Miner in Prom 6.2, (2014) *Expert System with Application*, 41, pp. 7678-7690.
http://dx.doi.org/10.1016/j.eswa.2014.05.055

[17] Weijters, A.J.M.M., Ribeiro, J.T.S., Flexible Heuristics Miner (FHM), *Proceedings of IEEE Symposium on Computational Intelligence and Data Mining* (Year of Publication: 2011).
http://dx.doi.org/10.1109/cidm.2011.5949453

## Authors' information

[1,2]Informatics Department, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia.

**Riyanarto Sarno** received M.Sc and Ph.D in Computer Science from the University of Brunswick Canada in 1988 and 1992. In 2003 he was promoted to a Full Professor. His teaching and research interest includes Internet of Things, Process Aware Information Systems, Intelligent Systems and Smart Grids.
E-mail: riyanarto@if.its.ac.id

**Kelly R. Sungkono** was born in Surabaya, Indonesia. She is currently a B.Sc student of Informatics Departement, Institut Teknologi Sepuluh Nopember Surabaya, Indonesia. Her major skills and researches are in process mining. She is also interested in business process models and data models.
E-mail: kelsungkono@gmail.com