ICIC Express Letters (ISSN 2185-2766), Vol. 2, No. 2, April 2011, pp. 473-479

Bayesian network for finding best combination of performers in BPM Environment

Bernardo Nugroho Yahya[1], Arif Wibisono[2], Hyerim Bae[1], Kwangyeol Ryu[1]
[1]Industrial Engineering Department
[2]Logistics Information Technology Department
[1,2] Pusan National University, Pusan 609-735, South Korea
Email : {bernardo, wibisono, hrbae, kyryu} @pusan.ac.kr

Abstract

Business process management system (BPMS) is an essential element in the business environments as a technology for delivering services to customers. In order to satisfy the customers, Business Process (BP), as a series or network of value-added activities, should be performed by relevant performers. One of the mechanisms to manage performers in BPMS is Separation-of-Duty (SoD) authorization policy under role-based access control (RBAC) mechanism. In the case of dynamic SoD, user assignment based on the constraints occurs in the run-time environment that results various execution time which can surely affect the optimality of completion time. This paper provides a method to find the best combination of performers based on execution time of process instances by employing Bayesian Network (BN). There are two main contributions of this paper. First, it proposes a simple translation in mapping BPM Notation (BPMN) to BN. Second, team sequence algorithm to find best combination of performers in SoD environment is composed. We also determine the most critical variable that can impact on the overall system. We believe that this approach can assist business entities that want to improve their BP performance.

Keywords: Bayesian network, separation of duties, role based access control, workflow

1. Introduction. A Business Process Management System (BPMS) is an integrated software that helps to coordinate and manage work processes through set of activities [4]. Allocating right user into right activity is essential to improve activity performance which also pertain user assignment strategy which can increase customer satisfaction in term of speed and QoS. Thus, users' skills, competency and experience may influence these factors.

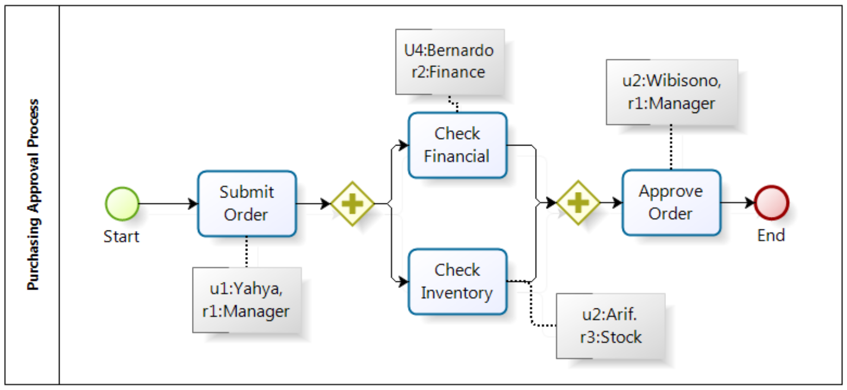


Figure 1. Example of BP with SoD policy at "Submit Order" and "Approve Order" activity

One issue that may hinder the user assignment is Separation-of-Duties (SoD) in the Role-Based Access Control (RBAC) approach. RBAC allows a user with a specific role to execute any activity under his/her permission assignment. However, in order to avoid a fraud, concept of SoD has been proposed. Fraud is a crime of obtaining money or some other benefit by deliberate deception. A cashier obtains money by deceiving selling ticket or a supervisor approves an amount of money without right are the examples of a fraud. For that reason, SoD does not allow a user to execute more than one conflicting tasks with the same role [2,3].

Figure 1 shows a purchasing approval process. Considering the SoD environment, we focus only on a role manager. Suppose that there are two users with the role of "Manager", Mr. Yahya and Mr. Wibisono. The role "Manager" can execute two activities: "Submit Order" and "Approve Order".

In order to avoid fraud, a user has to execute exactly one activity among the two in a process instance. Each user's skill, competency and experience are different, thus they impact on the activity performance. We can say that the performance of Mr. Yahya and Mr. Wibisono is different at "Submit Order" activity (also in the "Approve Order"). Hence, the overall process performance corresponds to the sequence order of performers in two activities. Therefore, predicting the best combination of performers is one method to improve the process performance.

The aim of this paper is to find the best combination of performers in SoD that can minimize the system lateness. Using the process logs, we can measure the lateness probability with specific condition. Bayesian Network (BN) is a tool to measure the probability dependency among activities based on performers. As a result, organizations can design a user assignment strategy based on BN to increase the quality of service (QoS) of BP. The rest of this paper is organized as follows. Section 2 discusses with regard to the research background. Section 3 shows the mechanism using BN. Some examples are shown in the section 4. Finally, we conclude the approach in section 5.

2. Research Background. The formal definition of SoD is that it is a security policy that is used to formulate multi-person control policies, requiring two or more distinct people to be responsible for the completion of a task or set of task [2,3]. In doing so, fraud is discouraged by distinguishing of the responsibility between more than one users.

Although there are many variations of dynamic SoD, this paper uses object and operational SoD. It implies that restricted roles may contain common members as long as the union of the activities the roles perform does not contain all the activities in a complete BP. It prevents any one person from performing an entire BP. Thipse et al. discussed an application with regard to SoD in BPMS [10].

BN can represent the probabilistic relationships between a set of random variables and their conditional dependences via a directed acyclic graph (DAG) [6]. In workflow environment, the network can be used to compute the probabilities of the process performance under such a QoS circumstances. Thus, this paper attempts to consider a workflow modeled by BPMN to be a BN.

3. Proposed approach using Bayesian Inference in BN. The definition of process structure and path we use in this paper proposed by Bae et al. [1].

Definition 3.1. Process structure is represented by a directed graph $P = (A,L)$ which consists of sets of node A and sets of arcs L

- $A = \{a_i|i=1,2,...N\}$ is the sets of activities, where a_i is the i-th activity, and N is total number of activities in P.
- $L \subseteq \{(a_i,a_j)|\ a_i,a_j \in A$ and $i \neq j\}$is link among activities where an element (a_i,a_j) represents that a_i immediately precedes $a_{j.}$
- For a split activity a_i, such that $|SA_i| > 1$, where $SA_i=\{a_{j+}|(a_i,a_{i+})\in L\}$, $f(a_i)$ = 'AND' if all a_{i+}'s should be executed; otherwise, $f(a_{i+})$= 'XOR'
- For a merge activity a_i, such that $|MA_i| > 1$, where $MA_i=\{a_{j-}|(\ a_{i-},a_i)\in L\}$, $f(a_{i-})$ = 'AND' if all a_{i-}'s should be executed; otherwise, $f(a_{i-})$= 'XOR'

In workflow environment, RBAC is frequently used to apply SoD policy. Ferraiolo et al. [3] defined the RBAC as follows.

Definition 3.2. The Role-based Access Control (RBAC) model consists of U, R, PM, PA, UA that represent a set of users, roles, permissions, permission assignment and user assignment, respectively.

- $U = \{u_j|j=1,2,...,J\}$ is the sets of users, where u_j is the j-th user, and J is total number of users.
- $R = \{r_k|k=1,2,...,K\}$ is the sets of roles, where r_k is the k-th user, and K is total number of roles.
- $PM = \{pm_l|l=1,2,...,L\}$, where PM ($\subseteq Op\times A$) is a set of permissions that assign operation Op to activity A, pm_l is the l-th permission, and L is total number of permission.
- PA ($\subseteq PM\times R$) is a set of permission assignment that assigns the permission need to complete their jobs to roles
- UA ($\subseteq U\times R$) is a set of user assignment that assigns user to roles.

One of the QoS for our measurement in the BPMS is process lateness. We define the process lateness as it describes in definition 3.

Definition 3.3. Lateness is the total of completion time of a job that summarizes total earliness and total tardiness incurred in the system for that job. The higher positive value of lateness namely "Late", the lower QoS will be which is in turn lowering system performance.

In order to predict the lateness of activity with prior knowledge about precedence activity, we use Bayesian inference. We denote $\bar{a}_i$ to represent that i-th activity is being late (lateness). The bayes theorem of conditional distribution of $\hat{a}_i$ given that i-th activity is executed by u_j is defined at equation (1). This is also valid for conditional distribution of $\bar{a}_i$.

$$P(\bar{a}_i \mid u_j) = \frac{P(\bar{a}_i \cap u_j)}{P(u_j)} = \frac{P(u_j \mid \bar{a}_i)P(\bar{a}_i)}{P(\bar{a}_i)} = \frac{P(u_j \mid \bar{a}_i).P(\bar{a}_i)}{[P(u_j \mid \bar{a}_i).P(\bar{a}_i)]+[P(u_j \mid \bar{a}_i)P(\bar{a}_i)]} \quad (1).$$

4. Basic Mapping BPMN Notation to Bayesian Network. Figure 3 shows BPs with three different structures; they are *sequence*, *XOR*-block and *AND*-block. In the *sequence* structures, an activity is followed by other activities consecutively. *Sequence* structure is finite and is able to appear multiple times both on block and outside block. *XOR*-block and *AND*-block contain parallel paths in which the control flow is different. *XOR*-block rule allow the activity flow going through exactly one path from the split gateway into merge gateway. On the other hand, *AND*-block requires the execution of all parallel paths from the split gateway and synchronizes all the paths in the merge gateway. To detect the *XOR*- and *AND*-block, we developed serial and parallel block detection as a modification from previous research [1]. Figure 2 shows the algorithms for block detection and for building the Bayesian network without conditional probability.

Notation

G: process definition network
N: the sets of nodes in G
A: set of arcs in G
v, v': an activity in G
s: the start activity of G
pred (v): the sets of activities immediately preceeding *v*.
succ(v): the sets of activities immediately succeeding *v*.
w (v): the water level of *v*.
w-list: the sets of water-levels.

merge: variable name
perf(*v*) : the set of performers in the activity *v*

fire(*succ*(*v*)) :
"AND" if ALL of *succ*(*v*) have to be executed
"XOR" if exactly one activity in *succ*(*v*) have to be executed

Serial Block

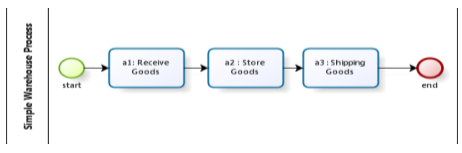
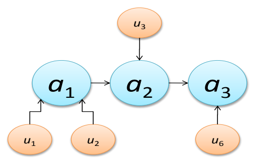
```
PROCEDURE Serial-block-detection (in G, out (SB, w-list))
b:=min(w-list); LOOP := T; QUEUE:={s};
while (LOOP := T) do
  if (QUEUE ≠ ø ) then return null;
    let v be the first element of the QUEUE;
    remove v from QUEUE;

  if (w(v) = b && | succ(v)|=1 &&
pred(succe(v))|=1) then
    LOOP := F;

    SB:=SerialFrom (v);
    w(SB) := w(v);

  else
     append succ(v) to QUEUE;
        addEdgeFromTo(v,succ(v));
      if (perf(v) ≠ ø ) then
        addEdgeFromTo(perf(v),v);
      end
 end
end Serial-Block-Detection
```

Parallel Block

```
PROCEDURE Parallel-block-detection (in G, out (PB, w-list))
b:=min(w-list); LOOP :=T ; QUEUE:={s};
c = 1;
merge : = null;
while (LOOP = T ) do
  let v be the first element of QUEUE;
  remove v from QUEUE;
  if (w(v) = b) then
      LOOP := F;
      PB:= succ(pred(v));
      w(PB) :=w(b) * |succ(pred(v));
      update w-list;
else append succ(v) to QUEUE;
       addEdgeFromTo(v, succ(v));
       addEdgeFromTo(v, PB)
       if fire(succ(v) ="OR") then
          addEdgeFromTo(v, PB);
       end;
       if fire(succ(v) ="AND") then
          merge = "1";
          addEdgeFromTo(v, merge);
       end;
       if (perf(v) ≠ ø ) then
          addEdgeFromTo(perf(v), v);
       end
  End
 if  (merge ≠ null ) then
        addEdgeFromTo(merge, PB);
  end
end Parallel-block-detection
```

Figure 2. Algorithms for Block Detection and Build Bayesian Network

Based on figure 3, the BN generation of *sequence* is typically the same with *DAG* generation with additional user variable. Moreover, *AND*-block has an additional variable compared with *XOR*-block. There is a Noisy-*OR* variable (named as "merge") to handle *AND*-join synchronization problem that may lead to deadlock. The join probability of "merge" will influence the block B_1 as block detection in the process. The rest calculation will follow the general Bayesian Inference as defined at equation (1).

Structure	BPMN	BN
Sequence		

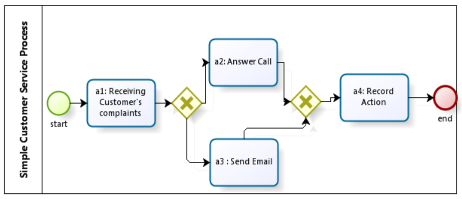
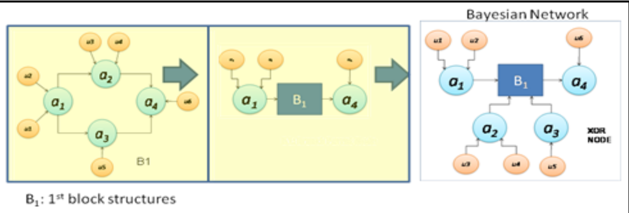
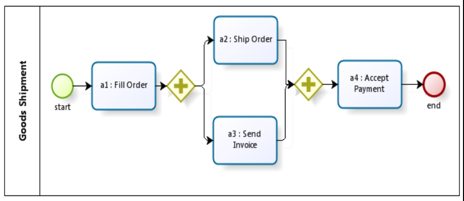
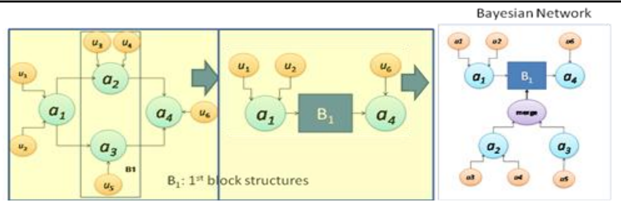
XOR-block	 Simple Customer Service Process start a1: Receiving Customer's complaints a2: Answer Call a3 : Send Email a4: Record Action end	 Bayesian Network B1 B1: 1st block structures XOR NODE
AND-block	 Goods Shipment start a1 : Fill Order a2 : Ship Order a3 : Send Invoice a4 : Accept Payment end	

Figure 3. Workflow Structures mapping from BPMN to BN

5. Application of Bayesian Network in lateness prediction. The workflow using BPMN in figure 1 is mapped into BN as shown in figure 4. Using the basic information of BPMN (A, R, U) and other given information, we obtain the conditional probability and join probability for each variable. This paper measured the process lateness prediction using Bayesian Inference with given information from process logs. The result is shown in the first column of figure 4. By giving other *evidence* for lateness, it shows that the probability value of sequence performers u_1 and u_2 (0.3333) is higher than u_2 and u_1 (0.2517). Thus, we choose the combination performers of u_2 and u_1 that give less probability for lateness.

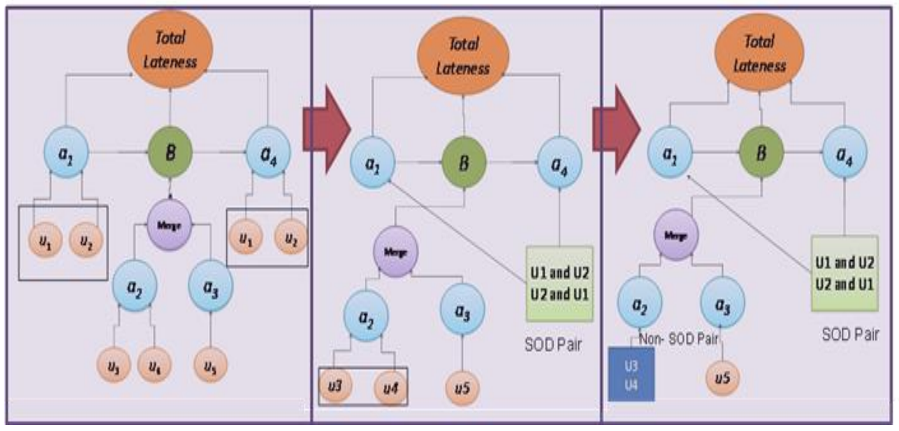
$A = \{a_1, a_2, a_3, a_4\}$ $R = \{r_1, r_2, r_3\}$ $U = \{u_1, u_2, u_3, u_4, u_5\}$ $r_1 = \{u_1, u_2\}$, $r_2 = \{u_3, u_4\}$, $r_3 = \{u_5\}$ P(TotalLateness=*late*|evidence, SoD Pair =”$u_1$and u_2”) = 0.3333 P(TotalLateness=*late* |evidence, SoD Pair =”$u_2$and u_1”) = 0.2517	 Total Lateness Merge U1 and U2 U2 and U1 SOD Pair Non- SOD Pair U3 U4

Figure 4. Step of translation the workflow of figure 1 into BN

6. Sensitivity analysis. A sensitivity analysis was undertaken to determine the variable that has the most impact on process lateness with performers' u_2 and u_1. By using the in-built NETICA “Sensitivity to findings” function, the sensitivities are expressed in term of Entropy reduction which is a concept borrowed from Information Theory [7].

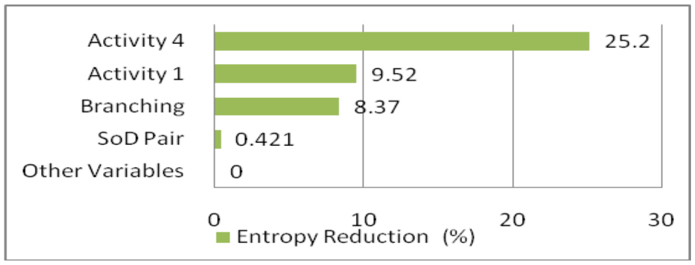


Figure 5. Entropy reduction result based on NETICA

Based on figure 4, by sorting the value in descending order, we can say that the first variable (activity 4) is the most influence factor to the total lateness (25.2%). It is followed by activity 1 (9.52%) and branching (8.37%) respectively, while other variables have little or no impact to the Total Lateness. Those result may partially happened because of the structure of the workflow process which lead the establishment of the BN. Using this information, process manager can see "activity 4" as the most critical process and subsequently can indicate following processes in priority risks in terms of reducing process lateness.

7. Conclusions. In this paper, we utilized Bayesian Network as a tool to result a better analysis of process execution in BPMS. There are two main contributions of this approach. First, we proposed a method to map BPMN into Bayesian Network, which is limited to the control flow of sequential, and parallel (AND & XOR). Second, we measured the execution probability based on process logs. Bayesian inference is used to infer the execution probability of respective users in SoD policy to find the best performers to minimize the process lateness.

Our measurement shows that best performers in SoD environment are those who have small process lateness probability. Additionally, it also shows that this is an effective approach to find the most critical activity with regard to process lateness. To the best of our knowledge, this is a novel approach to determine the best performers in BPMS with SoD environment using BN. There are some issues left for further research that are extending the experiments with more complicated process which can include higher number of activities and process structures (OR and LOOP) and considering cost and time aspects for QoS.

Acknowledgement. This work was supported by a Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (The Regional Research Universities Program/ Research Center for Logistics Information Technology).

REFERENCES

[1]. Bae, J., Bae, H., Kang, S., Kim, Y., (2004), "Automatic Control of Workflow Processes using ECA Rules", IEEE Transactions on Knowledge and Data Engineering, Vol. 16, no. 8, pp. 1010-1023.
[2]. Bertino, E., Ferrari, E., (1999), "The specification and enforcement of authorization constraints in workflow management systems", ACM Transactions on Information and System Security 2 (1), February 1999, pp. 65 – 104
[3]. Ferraiolo, D.F., Kuhn, D.R., Chandramouli, R., (2007), "Role based Access Control", Artech House
[4]. WFMC, (1994), "Workflow Reference Model.", Technical report, Workflow Management Coalition, Brussels
[5]. Thipse, A., Hewett, R., (2008), "Verification of Dynamic Separation of Duty Policy for Role-based Business Process", Region 5 Conference, Kansas city, pp. 1-6.

[6]. Li, N., Tripunitara, M.V., Bizri, Z., (2007) "On Mutually Exclusive Roles and Separation-of-Duty", ACM Trans. Inform. Syst. Security, vol. 10, no. 2

[7]. Han Jiawei., Kamber. M. (2006). "Data.Mining.Concepts.and.Techniques",San Fransisco: Morgan Kaufmann