A DYNAMIC AND HUMAN-CENTRIC RESOURCE ALLOCATION FOR MANAGING BUSINESS PROCESS EXECUTION

Arif Wibisono[1], Amna Shifia Nisafani[1*], Hyerim Bae[2], You-Jin Park[3]

[1]Department of Information Systems
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
*Corresponding author's e-mail: amna@is.its.ac.id

[2]Department of Industrial Engineering
Pusan National University
Busan, Korea

[3]School of Business Administration
Chung-Ang University
Seoul, Korea

Generally, resource allocation is essential to the efficient operational execution. More specifically, resource allocation for semi-automatic business processes might be more sophisticated due to human involvement. To this point, human performances are oscillating over time. Hence, upfront and static resource allocation might be suboptimal to deal with human dynamics. For this reason, this research suggests a dynamic and human-centric resource allocation to organize human-type resources in semi-automatic business process. Here, we use Bayesian approaches to predict resource's performances according to historical data set. As a result, we can construct a dynamic priority rule to assign a job to a specific resource with the highest probability to work faster. Finally, we demonstrate that our approach outperforms other priority rules: Random, Lowest Idle, Highest Idle, Order, and previously developed Bayesian Selection Rule from the total completion time and waiting time point of view.

Keywords: dynamic resource allocation, machine learning, dynamic dispatching rule, dynamic priority rule, naïve bayes

1. INTRODUCTION

The advancement of information technology has forced many companies to adopt information technology products to support their daily operations such as software for managing customer (e.g. customer relationship management), managing supplier (e.g. supplier management system), and managing inventory and production (e.g. Enterprise Resource Planning). One of the information technology products that many companies bring into their business ecosystem is business process management system (BPMS). BPMS is software to plan, execute, control, monitor, and evaluate business process (BP) within companies (Wibisono *et al*., 2015). In order to improve the efficiency of business process (BP), researchers have investigated some scheduling concepts for BP by organizing resources during BP execution (Bae, Lee and Moon, 2014) (Eder *et al*., 2003) (Huang, Lu and Duan, 2012) (Huang, van der Aalst and Lua, 2011) (Rhee, Bae and Kim, 2004) (Wu *et al*., 2009) (Zhao and Stohr, 1999) (Yahya *et al*., 2011) (Nisafani *et al*., 2014).

Generally, resources in a business process can be divided into human resources and facilities, for example, machines, vehicles, storages, etc. While facility-type resource is applicable to manufacturing-related processes within most of its execution, human-type resource is predominantly found in organization-related processes such as quote-to-order, procure-to-pay, order-to-cash, application to approval, and issue-to-resolution (Dumas et al., 2013). For example, Figure 1 shows an order-to-cash related process, which involves human-type resources and is commonly available in many wholesaling companies.

Compared to facility-type resource, managing human-type resource is more challenging for the following two reasons. First, human-type resources have a lower performance consistency than facility-type resources. For instance, an operator might perform persistently in his/her first three hours and then his/her speed gradually decreases. After having lunch, the operator might increase his/her speed but is a bit more slowly than the speed in the morning. As time progresses, the speed becomes constant from 2 PM until the end of the day (Wibisono et al., 2015). Thus, it is truly unfeasible to expect constant performance of human-type resource over a consecutive period. Second, generally, even though researchers acknowledge that individual

ISSN 1943-670X

performances may have a strong relationship with individual's knowledge, physical condition and emotional/mental state, it is hard to explain how these factors are related together to introduce such performance level. For these reasons, developing an accurate method to predict the performances of human-type resource is demanding, especially, to construct a reliable performance-aware resource allocation.

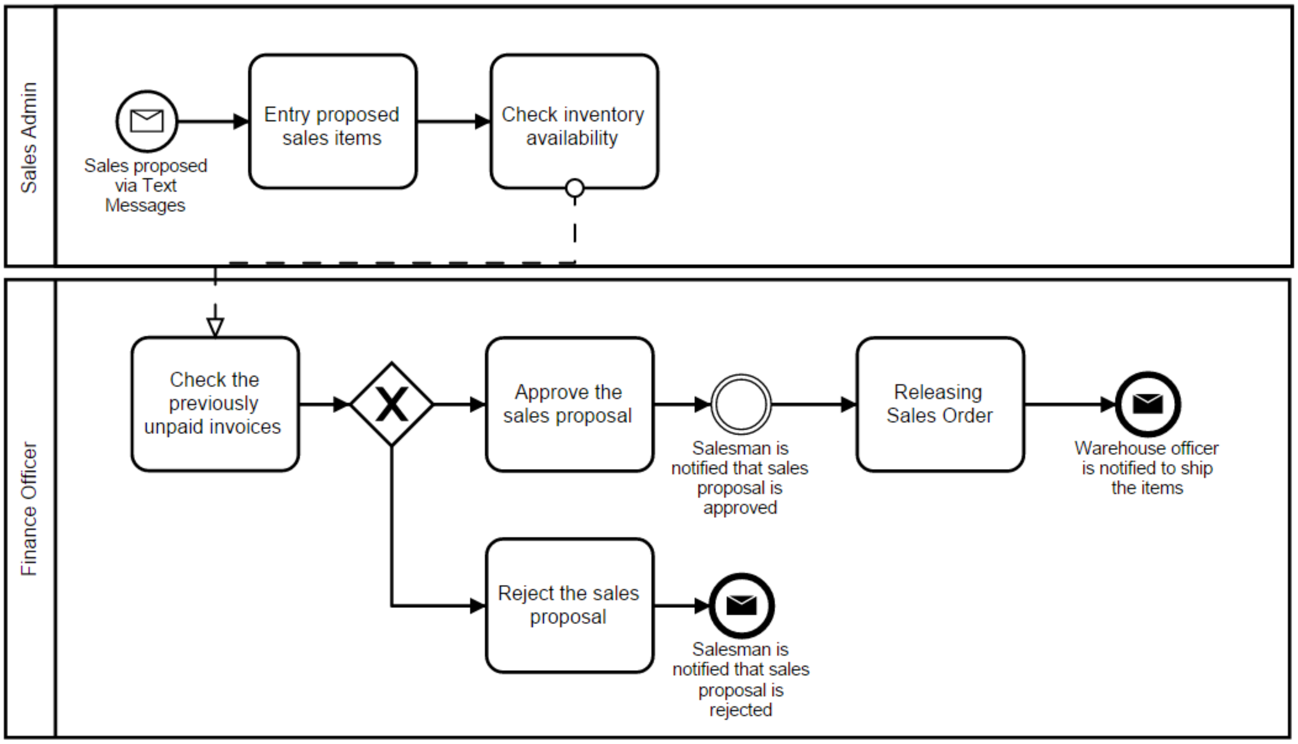


Figure 1. Issuing Sales Order as an example of organizational process which intensively involves human-type resources

The notion of BP scheduling has drawn a significant amount of attentions among many researchers. Zhao and Stohr suggested a method to reduce handle reworks in claim handling system (Zhao and Stohr, 1999). Bae *et al.* (Bae, Lee and Moon, 2014) developed BP execution plan using Mixed Integer Programming (MIP) by accommodating BP semantics and another possibility paths in the BP structure. Eder *et al.* (Eder et al., 2003) constructed an individual-oriented timetable to foresee future incoming jobs in order to cut back both the rate of time constraint violation and turnaround time. By the way, most of the aforementioned papers focus on upfront planning without considering resource performance dynamics. Here, upfront planning usually encompasses any effort to plan resource allocation prior the BP execution comes about. Despite its popularity, upfront planning might introduce a longer completion time due to sub-optimal resource allocation since human-type resource demonstrates performance fluctuations over time.

The objective of the study is to create a dynamic, human-centric, and performance-aware resource allocation by means of attaching Naïve Bayes Model (NBM) in the resource allocation algorithm. As the BP execution occurs, the resource allocation algorithm selects a resource with the highest performance prediction to carry out the incoming job. For each job accomplishment, the resource allocation algorithm then saves the job's completion time and categorizes the completion time as three categories, that is, slow, or moderate, or fast. Afterwards, resource allocation algorithm inserts this performance statistics into a historical data set, and, additionally, the resource allocation algorithm regularly updates NBM with the new historical data set. By doing so, we could enhance the NBM precision.

The remainder of the paper is organized as follows: Section. Section 2 discusses related studies of performance-aware resource allocation in business process management environment. Section 3 describes the suggested method. Section 4 explains the experiment results in comparison with other resource allocation algorithms as well as in depth analysis of the algorithm performances. Section 5 conclude the study and delineate potential future researches.

2. RELATED STUDIES

2.1 Administering Resource Allocation in Business Process Management Setting

Many noble procedures have been proposed to tackle scheduling issues in wide range of various areas, for example, manufacturing, transportation, project management, etc. Wu *et al* (Wu et al., 2009) forecasted the future workload by accommodating factors affecting workload such as service time, replication overhead, transition probability for additional service etc. Huang *et al.* (Huang, van der Aalst and Lua, 2011) advised the combination of Markov decision process and reinforcement learning to construct a resource allocation algorithm. Ha *et al.* (Ha et al., 2006) offered a real-time resource allocation algorithm by considering resource's limited capacity. Huang *et al.* (Huang, Lu and Duan, 2012) proposed rule to measure resource performance according to four important factors to enhance BP execution, which are availability, corporation, preference, and competence.

A great deal of previous researches into business process scheduling has emphasized on upfront planning in which might be inappropriate to cope up with resource performance dynamics. Most of the aforementioned researches supposed that resource performances are constant regardless the time; unfortunately this assumption is untrue especially if the resource-type is human. As a result, as time progresses any upfront resource planning will experience accuracy declines, unless the period of two upfront resource planning is very short so that approaching real time. Again, unfortunately this solution may be infeasible because the completion time to perform upfront resource planning would be longer than the resource execution itself. So, it is very important to take into account a real-time, human-centric, and performance-aware resource allocation for executing business process in order to upgrade system performance regularly.

To date, there have been limited studies on the development of a real time, human-centric and performance-aware resource allocation for executing business process because of the complexity of accurately modeling human behavior as well as embedding the model to existing real-time scheduling concepts. Liu Yingbo *et.al* (Liu et al., 2008) proposed a semi-automatic method for allocate staffs in the workflow. They generated models from workflow event log using supervised machine learning algorithms (C.45, SVM, and Naïve Bayes) and used it to make prediction of the best human resource. The works was significant; however, they did not assume the fluctuations of the resource performances due to the differences of the day time (such as morning, afternoon, and evening) in which apparent in many organizational processes.

Yang *et.al* (Yang et al., 2012) advised BNRR (Bayesian Network-based Resource Recommendation) to suggest the most skillful sets of resources for a business process based on event logs. The approach suggested in this research takes into account both the information about the resource dependency and the information about the resource capability simultaneously by using Bayesian Network. Even though the approach is meaningful, a drawback becomes visible since the approach requires one-to-one transformation from process model into Bayesian Network. In the one-to-one transformation, an activity within the process model will be exactly mapped into a node in the Bayesian Network. In addition, in the one-to-one transformation, if an immediate predecessor activity has a link to its successor(s), then the causal relationship link(s) is/are built from the predecessor to its successor(s). Now, each immediate predecessor activity will be the parent (in terms of Bayesian Network) of its immediate successor(s). Consequently, the issue with the one-to-one transformation is that its inability to address one of the Bayesian Network (BN) properties that are directed acyclic graph (DAG) which involves a property that forbids any closed loop structures in the BN. Therefore, any process model that follows one-to-one transformation rule (such as Yahya *et.al* (Yahya *et al.*, 2011)) will be challenging when modeling iterative structure in which prevalent in many business processes. Furthermore, assuming that every immediate predecessor activity is the most activity to its immediate successor(s) requires statistical validation in order to avoid modeling inaccuracy. Recent researches showed that, using K2 algorithm, some of the predecessor activities have demonstrated strong influences to their successor(s) and have not been necessarily to their immediate successor(s) (Wibisono et al., 2014).

Furthermore, Nisafani *et al.* (Nisafani *et al*., 2014) suggested a dynamic, human-centric, and performance-aware resource allocation on a simulated semi-automatic business process by establishing a BN in the simulation model. The BN covered various aspects that influence the BP execution quality namely inter-arrival time, workload, working hours, and daytime. The study indicated that the proposed resource allocation surpassed four static resource-based priority rule: longest-idled, index-ordered, random allocation and shortest-idled in terms of average cycle time,average waiting time, and average completion time.

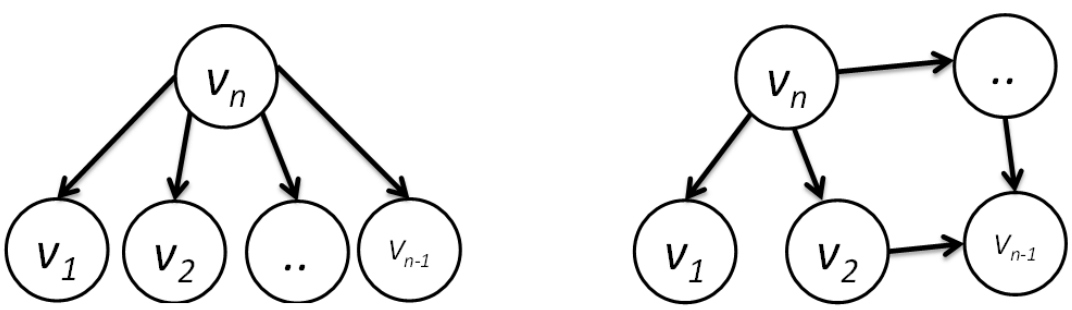
One of the greatest challenges of Nisafani *et al.* (Nisafani *et al.*, 2014) was that the proposed BN model had no statistical validation (Wibisono *et al*., 2015), since the construction of BN model was according to the author's heuristics. Heuristics is often a solution in making BN model since BN model aspects and their relations are difficult to capture due to their abstract natures. So, heuristics can be used whenever the modeler assumes that he/she has deep understanding about aspects in the BN model and their relations. This assumption might be untrue due to different system perspectives and the system's evolutions; thus, it might be insidious in the long run. To illustrate, at some points of time, a worker with long queue does not necessarily show performance deterioration. In reverse, a worker with an immense workload does not necessarily show performance increase. Moreover, these two situations can shift one to another in the long run to result different characteristics. Hence, whenever we run a system with an invalidated BN model embedded in the running resource allocation algorithm, we might have a sub optima from total completion time point of view.

Actually, statistical validation is just an assessment method to measure the BN Model quality, but it is not a course to generate a good-quality BN model. Instead, it is crucial to frequently invoke a time consuming algorithm (e.g. K2 Algorithm) in order to construct a good-quality BN model. However, even though a good-quality BN model will result on a more accurate prediction (Sutrisnowati, Bae and Park, 2014), as the system grows, K2 algorithm might not be a good alternative because of its time consuming nature. Therefore, it is very important to find a method that is not time consuming to recurrently reconstruct BN model as well as is able to avoid modeler's erroneous assumptions. Thus, in this research, we use Naïve Bayesian Approach (NBA) conjecture to make more appropriate BN model. There is a rule in NBA indicating that each aspect (or node) is independent to other aspects except ~~to~~ one aspect so called target. One of the excellent characteristics of the NBA is that its ability to "still" provide a good forecast even though we find that two or more aspects are dependent each other (Whitten, Frank and Hall, 2011). By incorporating NBA, we could maintain a fair execution time and avoid far deviations in human judgment.

Instead of making new algorithm, this study puts more in depth analysis of what Wibisono et al (Wibisono *et al*., 2015) have developed. In this case, we use Analysis of Variance (ANOVA) test to assess algorithms' performances. So that, we can perceive this study as an extension of what Wibisono *et.al* (Wibisono *et al*., 2015) have done.

2.2 Naïve Bayes as the subset of Bayesian Network

A BN is a Directed Acyclic Graph (DAG) and is beneficial to model uncertainty. It comprises set of directed links and nodes to articulate probabilistic causal dependencies among nodes (Pritsker and O'Reilly, 1999). These associations are then assessed by constructing conditional probability table (CPT) within each node to represents random variables. Consider that $G=(V,E)$ be a DAG with a node set V and a link set E, and let $X = (X_v)_{v \in V}$ be a set of random variable indexed by v (Wibisono *et al*., 2015).



(a) (b)

Figure 2. Example of Naive Bayes (a) and Example of Bayesian Network (b)

Naïve Bayes is a special form of BN. While BN allows immense variance of DAG structure to occur as shown in Figure 2 (b), Naïve Bayes has very special structure and restricts other structure to occur. The Naïve Bayes Structure should consist of exactly one parent node with several child nodes. Every child node is connected with the parent node with a single arrow as shown in Figure 2 (a) and it is impossible for each node to have any link to other child nodes except its parent. The benefit of having such structure is that we could avert sophisticated computation compare to other BN structure due to Naïve Bayes simplicity. Furthermore, Naïve Bayes basically assumes that each child node is independent to other child nodes, but this assumption of independency might be an issue (Cheng and Grainer, 1999). However, whenever the historical data is abundant, Naïve Bayes demonstrates its superiority to other algorithms (Langley, Iba and Thompson, 1992).

3. SUGGESTED APPROACH

3.1 Process Structure

Definition 1. (Process Structure)

We define process structure as Rhee *et al.* (Rhee, Bae and Kim, 2004) presented. A process structure is a directed graph $P=(A, L, F)$ consisting of sets of node A, sets of arcs L and the labeling function F:

- $A=\{a_i \mid i = 1, \ldots, N\}$ is the set of activities, where a_i is the i^{th} activity and N is the total number of activities in P.
- $F \subseteq \{(f_s, f_m)\}$ is the set of labeling function, where fs is the split function and f_m is the merge function.
- $L \subseteq \{(a_{i-}, a_{i+}) \mid a_{i-}, a_{i+} \in A$ and $i+ \neq i-\}$ is the set of links where an element (a_{i-}, a_{i+}) represents a_{i-} immediately precedes a_{i+}.
- For a split activity a_i, such that $|SA_i| > 1$, where $SA_i=\{a_{i+} \mid (a_i, a_{i+}) \in L\}$, $f(a_i)$ = 'AND' if all a_{i+} 's should be executed; otherwise $f(a_{i+})$ = 'OR'.
- For *a* merger activity a_i such that $|MA_i| > 1$, where $MA_i = \{a_{i-} \mid (a_{i-}, a_i) \in L\}$, $f(a_{i-})$='AND' if all a_i should be executed; otherwise $f(a_{i-})$ = 'OR'.

3.2 Incorporating Naïve Bayes in the suggested approach

The Naïve Bayes of our suggested approach, that is, Naïve Bayes Model (NBM) comprises five nodes namely Day Time, Queue, Human Performance, Inter-arrival, and Activity (see Figure 3). Each node signifies aspects joined in BP execution. As indicated earlier, we have two types of nodes: parent node and child nodes. Parent node is the focus of NBM and contains performance prediction in which aspects that influenced performance are captured throughout all child nodes. The detail information of NBM is available at Table 1.

Table 1. Nodes in the NBM

No	Nodes/Factors	Possible States	Notes
1	Human Performance	Low, Medium, High	Human resource performance prediction **(Parent Node)**
2	Queue	Low, Medium, High	Queue in front of the activity
3	Inter-arrival rate	Short, Medium, Long	Average of the systems' inter-arrival time/hours.
4	Performer	{human resource name}	
5	Activity	{activity name}	
6	Day time	Morning, Afternoon, Evening	The working shift

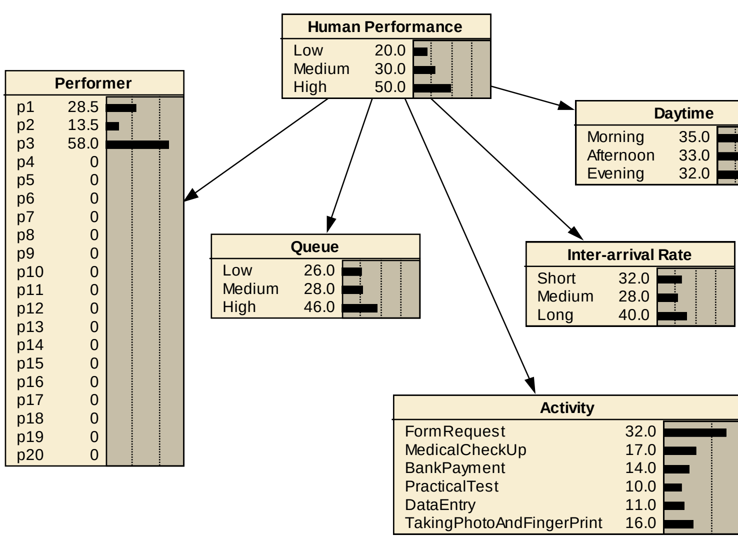


Figure 3. Naïve Bayes Model (NBM)

In order to measure the human performance, we observe the evidence of the other five nodes. An illustration about how the NBM works is available in Figure 4 below. For example, consider that a job comes to an activity, that is, "*medical checkup*", at time t and we need to allocate a right resource to undertake this job. Moreover, suppose we have found that at the job arrival: the daytime = "*morning*," the inter-arrival = "*short*," human performance = "*high*", and the queue = "*high*". From the NBM we could determine that $p3$ has the highest priority to do this job (probability 80%). Unless $p3$ is busy; thus unavailable, $p1$ is the second choice to replace $p3$ to perform the job (probability: 15%).

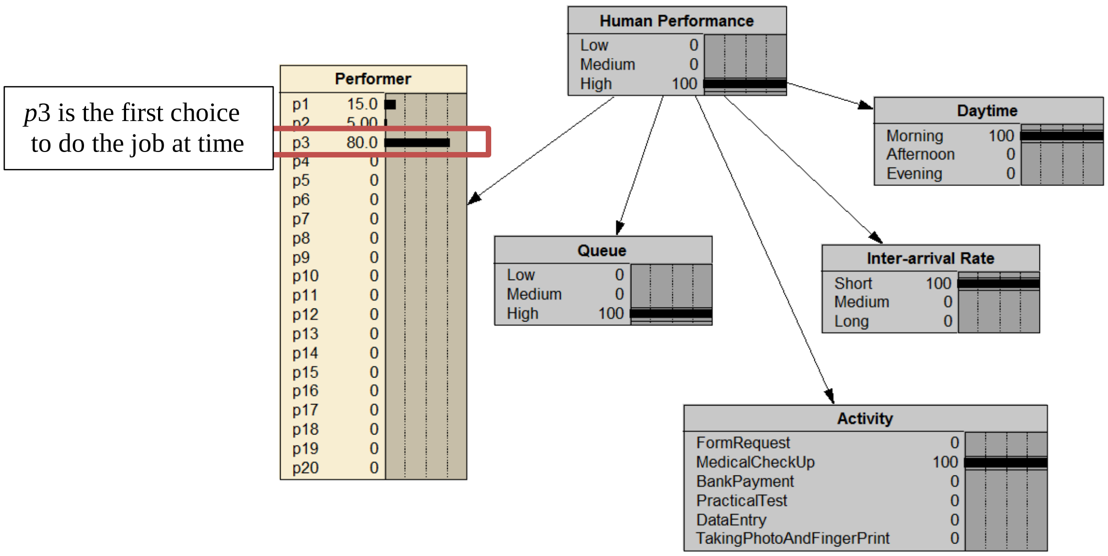


Figure 4 Illustration of the NBM execution with some evidences at time = t, Selected resource will be decided by the high performance resource prediction in which produced by the NBM

3.3 Naïve Bayesian Selection Rule as The Real-Time Human Centric Resource Allocation

We denote the real time and human centric resource allocation proposed in this research as Naïve Bayes Selection Rule (NBSR). Within NBSR we attach NBM as the model to determine most suitable resource. NBSR has the same steps as previously developed BSR algorithm by Nisafani *et al.* (Nisafani et al., 2014) except the utilized BN model. That is, while NBSR uses NBM, the BSR use a BN, which includes expert judgment. The reason we use NBM is that the performance prediction accuracy of Nisafani *et al.* (Nisafani et al., 2014)'*s* BN would be continuously deteriorated as the simulation model run in the long period of time since the expert judgment was set only once prior the simulation execution took place.

The NBSR assigns the best human resource to do a job in the a_i at time t, and *NBSR* (t, a_i) utilizes some parameters as follows:

- $R_a = \{r_n | n=1, 2, ..., N\}$ is the set of human resources where r_n is the n^{th} human resource and N is the total number of resources employed in a_i
- $Q_a(t)$ is the queue before a_i at time t
- *BN* represents the used NBM
- $D_a(t) \in$ {morning, evening, afternoon} is daytime at time t
- $I(t) \in$ {low, medium, high} is the inter-arrival rate at time t

The code in Figure 5 shows how the NSBR algorithm works in more detail. The algorithm will foresee the human resource performance as a certain job arrives, and it will select one resource to perform the job according to NBM recommendation. To this point, the algorithm gathers the NBM recommendation by calling *do_inference* function. Within *do_inference* a probability function is invoked:

$$P(Human\_Performance = \text{“High”} | Activity = a_i, Queue = Q_a(t), Humanresource = r_n, Daytime = D_a(t), Inter\text{-}arrival = I(t)).$$

By having this function, we can get the suitable resource given that activity a_i, Queue $Q_a(t)$, inter-arrival $I(t)$, human resource r_n,, and Day time $D_a\ (t)$.

```
FUNCTION SELECT RESOURCE (a_i, Q_a(t), R_a, BN, D_a (t), I(t))
BEGIN
BOOLEAN loop := TRUE ;
RESOURCE res ;
DOUBLE temp := -9999;
        //very big negative number, indicating no human resource is selected
WHILE (loop = TRUE)
{
        FOR (INT index :=0; index<size(R_a) ; index++)
        {
            value : = do_inference(a_i, Q_a(t), R_a, BN, D_a (t), I(t))
            IF (temp < value && r_index IS IDLE) THEN
                temp := value;
                res = r_index ;
                // r_index is the resource in the R_a with index = index
            END IF
        }
        IF (res != NULL) THEN
         loop := FALSE;
      END IF
}
  RETURN res;
END
```
Figure 5 NBSR Algorithm (Source: Nisafani *et al.* (Nisafani *et al*., 2014))

Table 2. Variable Classification

No	Nodes/Aspects	Possible States	Classification Method
1	Human Performance	Low	Processing time (a_i, t) < Quartile$_2$ (a_i)
		Medium	Processing time $(a_i, t) \geq$ Quartile$_2$ (a_i) and Processing Time $(a_i, t) \leq$ Quartile$_3$ (a_i)
		High	Processing time (a_i, t) > Quartile$_3$ (a_i)
2	Queue	Low	Queue Length (a_i, t) < Quartile$_2$ (a_i)
		Medium	Queue Length $(a_i, t) \geq$ Quartile$_2$ (a_i) and Queue Length $(a_i, t) \leq$ Quartile$_3$ (a_i)
		High	Queue Length (a_i, t) > Quartile$_3$
3	Inter-arrival rate	Short	Inter-arrival (t) < Quartile$_2$ (*System Inter-arrival*)
		Medium	Inter-arrival $(t) \geq$ Quartile$_2$ (*System Inter-arrival*) and Inter-arrival $(t) \leq$ Quartile$_3$ (*System Inter-arrival*)
		Long	Inter-arrival (t) > Quartile$_3$ (*System Inter-arrival*)
4	Performer	{human resource name}	-
5	Activity	{activity name}	-
6	Day time	Morning	8 AM - 12 AM
		Afternoon	12 AM – 16 PM
		Evening	16 PM – 21 PM

3.4 Updating NBM after a job accomplishment

Frequent updating of the NBM is conducted once a job leaves the activity. During that time, we could get the continuous value of each node. We then classify the continuous value into discrete value (see Table 2). Afterwards the updating processes can be made. In addition, for the classification, we utilize lower quartile and upper quartile to group Human Performance, Queue, and Inter arrival. The reason of using quartile is that we need to capture normal and abnormal (high or low) range of each nodes/aspect. Since the normal range is quite far which is from lower quartile to upper quartile, we may expect a fair utilization since resource allocation is distributed among resources. The utilization might decrease if a resource demonstrates a constant exceptional performance, thus making him/her allocated every time a job comes. However, this situation is certainly impossible because human cannot work in a regular pace.

4. EXPERIMENTS AND RESULTS

4.1 Comparison with other priority rules for resource allocation

Table 3 lists all priority rules in our simulation model. This study incorporates four static priority rules, which are prevalent in many manufacturing processes to compare the efficiency of our proposed method (Pritsker and O'Reilly, 1999). We denote these priority rules as static because they do not accommodate any resource performance dynamics. In addition, we also show previously developed BSR as a benchmark of our proposed dynamic algorithm.

Table 3. Priority rules in the simulation

No	Priority Rule	Description	Natures
1	ORDER	Allocate available resource in the preferred order	Static
2	LIDDLE	Allocate resource according to the largest idle	Static
3	SIDDLE	Allocate resource according to the smallest idle	Static
4	RANDOM	Allocate resource randomly	Static
5	BSR	Nisafani [8]'s algorithm, Allocate available resource in the preferred order suggested from BN	Dynamic
6	NBSR	Proposed algorithm	Dynamic

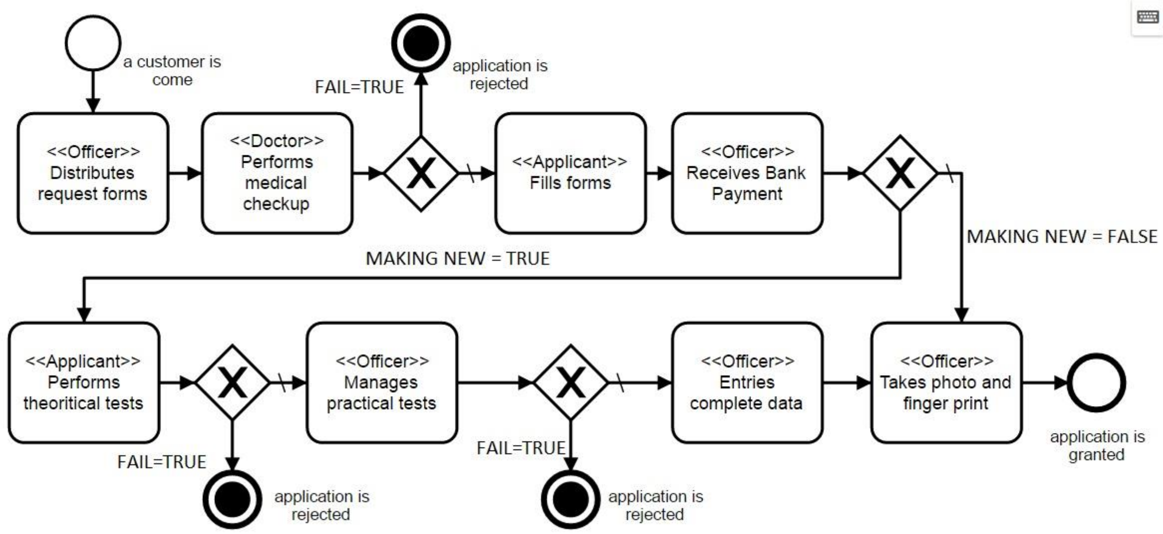


Figure 6 Driver License Application Process

4.2 Experiments Result

The simulation model from Nisafani *et al.* (Nisafani *et al*., 2014) captured a semi-automatic real world driver license application process in Indonesia (see Figure 6). In general, there were eight activities in which two activities were belonged to applicants and the rest six activities were belonged to the police department officers. The six activities are "distribute request form", "perform medical checkup", "receive bank payment", "managing practical test", "entry complete data", and "take photo and fingerprint". While the rest two activities are "fill forms" and "perform theoretical test". More specifically, three officers are dedicatedly responsible to each activity and the transfers of the officer among activities are prohibited. Since, the objective is to allocate the available officers to the appropriate position as best as they can; all applicants' self-service activities were excluded from the BSR or NBSR. We run the simulation model for 13 hours a day and every officer has his/her own distinct distributions for three working shift: morning, afternoon and evening. The number of replication is 50. We use 50 replications to ensure that the data used for Analysis of Variance (ANOVA) test is normally distributed. The result of each replication will be compared using one-way ANOVA test. By using ANOVA test, we will investigate whether there is significant performance improvement of NBSR compare to other algorithms in terms of total completion time and total waiting time. There are two dependent variables: total waiting time and total completion time, while the independent variable is the priority rules. Therefore, in the ANOVA test, there will be two categories of hypothesis: completion time-related and waiting time related hypothesis. For the completion time hypothesis, we have two hypotheses: H_0: $\mu_{Random} = \mu_{Order} = \mu_{BSR} = \mu_{NBSR} = \mu_{SILDLE} = \mu_{LIDLE}$ and H_1: $\mu_{Random} \neq \mu_{Order} \neq \mu_{BSR} \neq \mu_{NBSR} \neq \mu_{SILDLE} \neq \mu_{LIDLE}$ where μ is mean of total completion time. For the waiting time hypothesis, we have two hypotheses: H_0: $\mu_{Random} = \mu_{Order} = \mu_{BSR} = \mu_{NBSR} = \mu_{SILDLE} = \mu_{LIDLE}$ and H_1: $\mu_{Random} \neq \mu_{Order} \neq \mu_{BSR} \neq \mu_{NBSR} \neq \mu_{SILDLE} \neq \mu_{LIDLE}$ where μ is mean of total waiting time. We employ 5% level of significance.

Based on Table 4, NBSR outperforms other priority rules with the lowest mean of 31.20 minutes, followed by BSR, RANDOM, and ORDER priority rules with mean of 32.84, 34.66, and 35.88 minutes, respectively. We can see that SIDLE and LIDLE have the worst performance among all the algorithms compared and NBSR has the lowest value for maximum processing time, 37 minutes. Even though ORDER method does not perform quite well compare to BSR and NBSR, the minimum processing time of Order method is 20 minutes, which is equal to the minimum processing time of NBSR.

Table 4. Processing Time Among Priority Rules

Rule	**N**	**Mean**	**Std. Deviation**	**Std. Error**	**95% Confidence Interval for Mean**		**Minimum**	**Maximum**
					Lower Bound	Upper Bound		
RANDOM	50	34.66	4.049	.573	33.51	35.81	28	40
ORDER	50	35.88	5.709	.807	34.26	37.50	**20**	45
BSR	50	32.84	4.635	.656	31.52	34.16	24	44
NBSR	**50**	**31.20**	**4.247**	**.601**	**29.99**	**32.41**	**20**	**37**
SIDLE	50	36.36	5.178	.732	34.89	37.83	25	50
LIDLE	50	36.14	3.860	.546	35.04	37.24	28	40
Total	300	34.51	4.997	.289	33.95	35.08	20	50

Based on

Table 5Table 5, we can see that NBSR has the least mean of waiting time with 150.96 minutes, followed by BSR with 158.68 minutes. Moreover, NSBR outperforms other priority rules by producing the lowest maximum waiting time, 189 minutes of 200 minutes. In contrast with NBSR, BSR produces the highest maximum waiting time, 200 minutes, along with RANDOM and LIDLE method. Even though NBSR has the lowest value of maximum waiting time among all priority rules; BSR outperforms NBSR with the lowest minimum waiting time, 120 minutes. It can be concluded that sometimes BSR has more superior performance than the performance of NBSR in terms of customer waiting time.

Table 5. Waiting Time among Priority Rules

Rule	**N**	**Mean**	**Std. Deviation**	**Std. Error**	**95% Confidence Interval for Mean**		**Minimum**	**Maximum**
					Lower Bound	**Upper Bound**		
RANDOM	50	165.12	22.812	3.226	158.64	171.60	129	200
ORDER	50	161.62	21.975	3.108	155.37	167.87	129	199
BSR	50	158.68	22.162	3.134	152.38	164.98	**120**	200
NBSR	**50**	**150.96**	**20.023**	**2.832**	**145.27**	**156.65**	123	**189**
SIDLE	50	166.20	20.192	2.856	160.46	171.94	124	199
LIDLE	50	165.24	23.271	3.291	158.63	171.85	130	200
Total	300	161.30	22.232	1.284	158.78	163.83	**120**	200

Table 6. Test of Homogeneity of Variances for Waiting Time

Levene Statistic	df1	df2	Sig.
.625	3	196	.599

Table 7. Test of Homogeneity of Variances for Processing Time

Levene Statistic	df1	df2	Sig.
1.522	5	294	.183

Table 8. ANOVA for Processing Time

	Sum of Squares	Df	Mean Square	F	Sig.
Between Groups	1086.187	5	217.237	10.009	.000
Within Groups	6380.760	294	21.703		
Total	7466.947	299			

Table 9. ANOVA for Waiting Time

	Sum of Squares	Df	Mean Square	F	Sig.
Between Groups	8400.417	5	1680.083	3.544	.004
Within Groups	139386.980	294	474.105		
Total	147787.397	299			

Table 10. Multiple Comparisons for Processing Time

(I) Rule	(J) Rule	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
RANDOM	ORDER	-1.220	.932	1.000	-3.98	1.54
	BSR	1.820	.932	.776	-.94	4.58
	NBSR	3.460*	.932	.004	.70	6.22
	SIDLE	-1.700	.932	1.000	-4.46	1.06
	LIDLE	-1.480	.932	1.000	-4.24	1.28
ORDER	RANDOM	1.220	.932	1.000	-1.54	3.98
	BSR	3.040*	.932	.019	.28	5.80
	NBSR	4.680*	.932	.000	1.92	7.44
	SIDLE	-.480	.932	1.000	-3.24	2.28
	LIDLE	-.260	.932	1.000	-3.02	2.50
BSR	RANDOM	-1.820	.932	.776	-4.58	.94
	ORDER	-3.040*	.932	.019	-5.80	-.28
	NBSR	1.640	.932	1.000	-1.12	4.40
	SIDLE	-3.520*	.932	.003	-6.28	-.76
	LIDLE	-3.300*	.932	.007	-6.06	-.54
NBSR	RANDOM	-3.460*	.932	.004	-6.22	-.70
	ORDER	-4.680*	.932	.000	-7.44	-1.92
	BSR	-1.640	.932	1.000	-4.40	1.12
	SIDLE	-5.160*	.932	.000	-7.92	-2.40
	LIDLE	-4.940*	.932	.000	-7.70	-2.18
SIDLE	RANDOM	1.700	.932	1.000	-1.06	4.46
	ORDER	.480	.932	1.000	-2.28	3.24
	BSR	3.520*	.932	.003	.76	6.28
	NBSR	5.160*	.932	.000	2.40	7.92
	LIDLE	.220	.932	1.000	-2.54	2.98
LIDLE	RANDOM	1.480	.932	1.000	-1.28	4.24

	ORDER	.260	.932	1.000	-2.50	3.02
	BSR	3.300*	.932	.007	.54	6.06
	NBSR	4.940*	.932	.000	2.18	7.70
	SIDLE	-.220	.932	1.000	-2.98	2.54
*. The mean difference is significant at the 0.05 level.						

One assumption that should be met in order to conduct ANOVA analysis is the homogeneity test. Based on Table 6 and Table 7, the variances of all selection rules algorithm seems homogeneous in both waiting time and processing time with significances greater than 0.05.

Since the variances are homogeneous, we can utilize the ANOVA test. Both Table 8 and Table 9 show the result of ANOVA test for processing time and waiting time respectively. The significance for processing time and waiting time is less than 0.05 (0.000 for processing time and 0.004 for waiting time), so we conclude that there is significant difference in terms of processing time and waiting time for different priority rules. The details of multiple comparisons are available at Table 10 and Table 11 for processing time and waiting time, respectively.

Based on Table 10, NBSR and BSR have significant difference of mean processing time with RANDOM, ORDER, SIDLE and LIDLE priority rules at the 0.05 level. In terms of waiting time (see Table 11), at the 0.05 level, the significant difference is occurred for NBSR with RANDOM, SIDLE and LIDLE. It means that the amount of total waiting time reduction produced by NBSR is quite large.

Table 11. Multiple Comparisons for Waiting Time

(I) Rule	(J) Rule	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
RANDOM	ORDER	3.500	4.355	1.000	-9.39	16.39
	BSR	6.440	4.355	1.000	-6.45	19.33
	NBSR	14.160*	4.355	.019	1.27	27.05
	SIDLE	-1.080	4.355	1.000	-13.97	11.81
	LIDLE	-.120	4.355	1.000	-13.01	12.77
ORDER	RANDOM	-3.500	4.355	1.000	-16.39	9.39
	BSR	2.940	4.355	1.000	-9.95	15.83
	NBSR	10.660	4.355	.224	-2.23	23.55
	SIDLE	-4.580	4.355	1.000	-17.47	8.31
	LIDLE	-3.620	4.355	1.000	-16.51	9.27
BSR	RANDOM	-6.440	4.355	1.000	-19.33	6.45
	ORDER	-2.940	4.355	1.000	-15.83	9.95
	NBSR	7.720	4.355	1.000	-5.17	20.61
	SIDLE	-7.520	4.355	1.000	-20.41	5.37
	LIDLE	-6.560	4.355	1.000	-19.45	6.33
NBSR	RANDOM	-14.160*	4.355	.019	-27.05	-1.27
	ORDER	-10.660	4.355	.224	-23.55	2.23
	BSR	-7.720	4.355	1.000	-20.61	5.17
	SIDLE	-15.240*	4.355	.008	-28.13	-2.35
	LIDLE	-14.280*	4.355	.017	-27.17	-1.39
SIDLE	RANDOM	1.080	4.355	1.000	-11.81	13.97
	ORDER	4.580	4.355	1.000	-8.31	17.47
	BSR	7.520	4.355	1.000	-5.37	20.41
	NBSR	15.240*	4.355	.008	2.35	28.13
	LIDLE	.960	4.355	1.000	-11.93	13.85
LIDLE	RANDOM	.120	4.355	1.000	-12.77	13.01
	ORDER	3.620	4.355	1.000	-9.27	16.51
	BSR	6.560	4.355	1.000	-6.33	19.45

	NBSR	14.280*	4.355	.017	1.39	27.17
	SIDLE	-.960	4.355	1.000	-13.85	11.93
*. The mean difference is significant at the 0.05 level.						

5. CONCLUSION

This study proposes a real time and human-centric resource allocation in the business process management setting. The essence of the proposed algorithm is the usage of Naïve Bayes Model to predict the resource's performance. We benchmark our proposed algorithm with four static priority rules and similar Bayes approach for real time scheduling. According to ANOVA test at 0.05 significance level, there is a significant improvement of using NBSR compare to four static priority rules: ORDER, LIDDLE, SIDDLE, and RANDOM in terms of both completion time and waiting time. However, even though NBSR has outperformed BSR but the difference is less significant. Overall, the result demonstrates that our proposed algorithm exceeds all priority rules in terms of completion time and waiting time. Further research might focus to resource transfer and resource down time.

REFERENCES

Bae, H., Lee, S., & Moon, I. (2014, June). Planning of business process execution in Business Process Management environments. *Information Sciences, 268*, 357–369.

Cheng, J., & Grainer, R. (1999). Comparing Bayesian Network Classifiers. Morgan Kauffman Publishers.

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. (2013). *Fundamentals of Business Process Management.* Berlin, Germany: Springer.

Eder, J., Pichler, H., Gruber, W., & Ninaus, M. (2003). Personal schedules for workflow system. *Lecture Notes Computer Science*, 216–231.

Ha , B.-H., Bae, J., Park, Y., & Kang, S.-H. (2006). Development of process execution rules for workload balancing on agents. *Data Knowledge Engineering, 56*, 64–84.

Huang, Z., Lu, X., & Duan, H. (2012). Resource behaviour measure and application in business process management. *Expert systems with applications, 39*, 6458–6468.

Huang, Z., van der Aalst, W., & Lua, X. (2011, January). Reinforcement learning based resource allocation in business process management. *Data & Knowledge Engineering, 70*(1), 127–145.

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. *Tenth National Conference of Artifical Intelligence.*

Liu, Y., Wang, J., Yang, Y., & Sun, J. (2008). A semi-automatic approach for workflow staff assignment. *Computer in Industry , 59*(5), pp. 463-476.

Nisafani, A., Wibisono, A., Kim, S., & Bae, H. (2014). Bayesian Selection Rule for Human-Resource Selection in Business Process Management Systems. *Journal of Society for e-Business Studies, 17*(1).

Pritsker, A., & O'Reilly, J. (1999). *Simulation with Visual SLAM and AweSim.* WestLafayette, Indiana: System Publishing Corporation.

Rhee, S. H., Bae, H., & Kim, Y. (2004). A Dispatching Rule for Efficient Workflow. *Concurent Engineering: Research and Applications, 12*(4), 305-318.

Sutrisnowati, R. A., Bae, H., & Park, J. (2014). Bayesian Network Learning for Port Logistics Process Knowledge Discovery. *21*(3).

Whitten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques: Practical Learning Tools and Techniques, Third Edition.* United States: Morgan Kauffman.

Wibisono, A., Nisafani, A. S., Bae, H., & Park, Y.-J. (2014). Early detection for supply chain disruption using bayesian network. *Asia Pacific Business Process Management.* Brisbane, Australia.

Wibisono, A., Nisafani, A. S., Bae, H., & Park, Y.-J. (2015). On-the-fly Performance-Aware Human Resource Allocation in the Business Process Management Systems Environment. In *Lecture Notes in Business Information Processing* (Vol. 215, pp. 70-80). Springer.

Wu, B., Chi, C.-H., Chen, Z., Gu, M., & Sun, J. (2009, March). Workflow-based resource allocation to optimize overall performance of composite services, Future Generation Computer Systems. *Future Generation Computer Systems, 25*, 199-212.

Yahya, B. N., Wibisono, A., Bae, H., & Ryu, K. (2011). Bayesian network for finding best in BPM Environment. *ICIC Express Letters, 2*(2), pp. 473-479.

Yang, H., Wen, L., Liu, Y., & Wang, J. (2012). An Approach to Recommend Resources for Business Processes. In P. Herrero, H. Panetto, R. E. Meersman, & T. E. Dillon (Ed.), *OTM Workshops*, *Lecturer Notes in Computer Science*, *7567*, pp. 662-665.

Zhao, J., & Stohr, E. (1999). Temporal workflow management in a claim handling system. *The international joint conference on work activities coordination and collaboration*, (pp. 187-195). New York.