International Journal of Innovative Computing, Information and Control
Volume **8**, Number **7(B)**, July **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **5381–5399**

TOOL SUPPORT FOR PROCESS MODELING USING PROXIMITY SCORE MEASUREMENT

Bernardo Nugroho Yahya[1], Hyerim Bae[1], Joonsoo Bae[2,*] and Ling Liu[3]

[1]Department of Industrial Engineering
Pusan National University
30-san, Jangjeon-dong, Geumjong-gu, Busan 609-735, South Korea
{ bernardo; hrbae }@pusan.ac.kr

[2]Department of Industrial and Information Systems Engineering
Chonbuk National University
664-14 1-Ga Deokjin-dong, Jeonju, Jeonbuk 561-756, South Korea
*Corresponding author: jsbae@chonbuk.ac.kr

[3]College of Computing
Georgia Institute of Technology
266 Ferst Dr, Atlanta, GA 30332-0765, USA
lingliu@cc.gatech.edu

Received March 2011; revised August 2011

Abstract. *In Business Process Management System (BPMS), process modeling is a troublesome task for a designer with little or insufficient experience. It is widely recognized in practice that only a proficient process designer is able to utilize process modeling tools effectively. Furthermore, although a process modeling tool can be effective in BPMS, a considerable amount of effort is required from enterprises in order to reconfigure business processes for convenient process modeling environments. This paper proposes a proximity score measurement approach to facilitate process modeling. Our approach has three salient features. First, it utilizes a proximity score to provide an analysis about the degree to which an activity is related with another activity in business processes. We argue that this analysis is critical in assisting process designers to initiate their process design with the best possible process reference model. Second, we developed a suite of methods for convenient process modeling, particularly suitable for novice designers, including the method of determining the proximity score measurement (PSM), and the methods of finding the respective process reference model and calculating homogeneity. We demonstrate that a process reference model is a convenient and effective way for a designer lacking experience to be guided to design his own process model. The homogeneity score can help the process designer to determine the suitable class to which a new model may belong. This further facilitates the versioning of the process model. Third but not the least, we develop a prototype of our system and conduct the experiments to evaluate the effectiveness of our approach. Our experimental results show that the proximity score measurement approach is efficient and effective for process designers to perform process modeling in BPMS environments.*

Keywords: Business process modeling, Workflow management, Business process management, Selection process, Proximity

1. **Introduction.** Business Process Management System (BPMS) is popular in business environments as an emerging technology for delivering services to customers. BPMS provides business users with a simple set of tools with which they can model human-centric processes and enable processes to be easily "orchestrated" in order to improve the process efficiency and quality of business operations [1, 2]. Furthermore, BPMS facilitates a service provider in creating a customized process design to meet customers' needs [3-5]. A process

model, the result of process designing, is not just a graphical representation but also serves as a means of communication between stakeholders and system designers. Moreover, a process model should be representative, easy to understand, easy to use, optimized to the appropriate level of detail, and support abstraction [7]. A process modeling tool, therefore, is indispensable to a process designer [3, 6, 8].

A fair amount of research to date has been devoted to the methods for creating the most effective design processes. This line of research primarily focused on the correctness of the process models. Surprisingly very few have emphasized on the importance of process modeling convenience and developed a process modeling tool that offers process modeling convenience as one of the basic features. It is recognized that a designer with little prior design experience may build a process model that is inadequate or ineffective in terms of meeting the requirements of its customers. Without clear understanding of process objectives and sufficient knowledge of the previous process variants, the process modeling often leads to different levels of customer dissatisfaction [6, 9]. Furthermore, in order to meet the individual preferences of different customers, process modeling also requires personalized customization. Such customization naturally generates several variants from a single process model [10, 11]. Thus, a process modeling tool should be able to capture and incorporate knowledge extracted from a process repository. Such capability is critical in meeting different customization requirements, improving the usability and enhancing the convenience and user experience in process modeling. We believe that the knowledge built into a process modeling tool should guide both novice and experienced process designers in building an effective and satisfactory process model.

Process similarity or closeness is an important measure of the relationships between different variants resulting from the same original process and between different processes with similar domain-specific requirements. Although several process modeling tools [7] have been proposed to date, few of them have provided a quantitative approach to capturing and representing the closeness of existing process variants. Many have reported that by utilizing a similar and previously designed process model, a process modeling tool can help a novice user to design a new process model more easily and efficiently [12-15].

With these objectives in mind, in this paper we develop a closeness measurement, called Proximity Score Measurement (PSM) based on the concept of path, distance, reachability in graph theory [10]. Using PSM, we can statistically measure the homogeneity of PSM among process variants and evaluate the homogeneity of a group of processes. This helps a process designer to find the best cluster of processes in terms of design and customization requirements. Another interesting challenge is the complexity of large search space when computing PSM for processes in a large process repository. We develop a heuristic approach based on A-star algorithm to efficiently find the most representative process among process variants based on PSM.

Process mining is another line of research that is closely relevant to the research of convenient process modeling problem presented in this paper. Some researchers have considered process mining with log events [2, 18]. Other researchers have approached the process mining by studying the similarity between two individual processes [19, 20]. This paper, to the best of our knowledge, is the first one that provides a systematic approach to computing and employing the PSM measures among a group of processes in establishing and promoting convenient process modeling. This paper makes three distinct contributions. First, we formally define a set of basic concepts critical in modeling business processes, including path, sequence, parallelism, reachability, distance. Based on these formal concepts, we introduce the activity proximity score (APS) to measure the distance between two activities in a process. Second, we formally introduce Proximity Score Measurement (PSM) as a novel method to assist process designers in the business

process modeling phase. Finally, we present an experimental study and analysis with several process variants for better consideration of the approach. We show that our PSM approach is efficient and effective for process designers to perform process modeling in BPMS environments.

The rest of this paper is structured as follows. Section 2 introduces background of this paper that elucidates the goal of our research. Section 3 describes the analytical approach to PSM-based process modeling in the business process environment. PSM-based related application and experiments are explained in Section 4. We outline the related work in Section 5 and conclude the paper in Section 6.

2. Background.

Processes can be viewed as collections of decision models, each of which are identified by a type of decision and contain a sequence of processing tasks [3]. The term business process modeling is used to incorporate all activities relating to the transformation of knowledge about business systems into models that describe the processes performed by organizations. Process modeling has always been at the core of BPMS. Models enable the system to initiate relevant activities so that business objectives can be achieved [3, 7, 8]. In this study, the business process model notation defined in Definition 2.1 is used.

Definition 2.1. *Process Model. We define a process model p as a tuple of $< A, L >$ and labeling function f, each element of which is defined below.*

- *$A = \{a_i | i = 1, \ldots, I\}$ is a set of activities where a_i is the i-th activity of p and I is the total number of activities in p.*
- *$L \subseteq \{l_{ij} = (a_i, a_j) | a_i, a_j \in A\}$ is a set of links where l_{ij} is the link between two activities a_i and a_j in a process. The element (a_i, a_j) represents the fact that a_i immediately precedes a_j.*
- *For a split activity a_i such that $|SA_i| > 1$, where $SA_i = \{a_j | (a_i, a_j) \in L\}$, $f(a_i) = AND$ if all of the succeeding activities should be executed; otherwise, $f(a_i) = OR$.*
- *For a merge activity a_i such that $|MA_i| > 1$, where $MA_i = \{a_j | (a_j, a_i) \in L\}$, $f(a_i) = AND$ if all of the preceding activities should be executed; otherwise, $f(a_i) = OR$.*
- *Start activity (a_S) is an activity with an empty set of preceding activity, $|MA_S| = 0$. End activity (a_E) is an activity with an empty set of succeeding activity, $|SA_E| = 0$.*

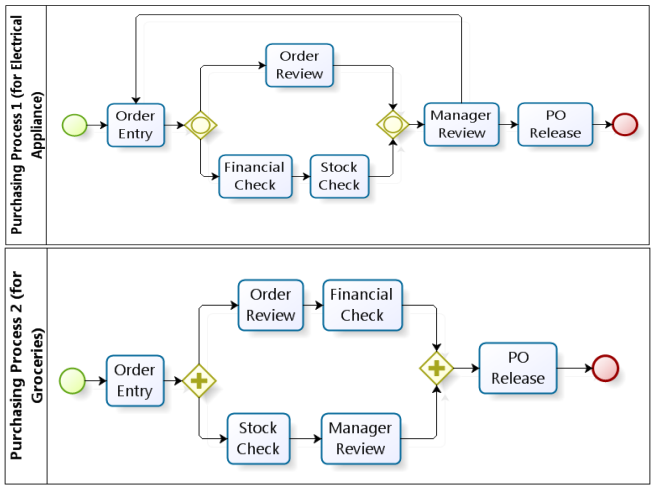

FIGURE 1. Example process variants in a repository

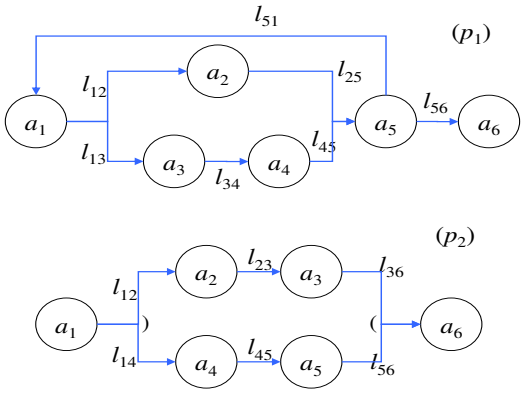

FIGURE 2. Abstraction of Figure 1 process into graph

Figure 1 represents two process variants in the repository. Based on the standard purchasing process, Order_Entry activity should be the start activity, and PO_Release activity should be the end activity. In order to customize the process according to customer requirements or specific product types, a designer should have good knowledge of how to model a new process variant. A designer should refer many times to existing processes to stay close to the variants built previously. Without using proximity measure, a user may have to incur time-consuming repetitive comparisons to obtain a new process model. Our proximity-based approach to process modeling can minimize process redundancy and other time-consuming and error-prone procedures.

We use directed graph (digraph) to calculate the proximity of activities in business processes. Digraph is one of the most representative graphical notations representing business processes. Using the graph matching approach, we can measure the similarity value between two processes [12, 13, 15, 16, 21]. As one of the intentions of this study was to graph-match based on business process properties such as AND-, OR-, and *iterative*-block activity, which is not covered by [12, 16], measurement of the path sequence is necessary. The present study considered all processes and all activity sequences in the processes.

Figure 2 is an abstraction into a graph of the business process shown in Figure 1. Activity a_1 in p_2 (Figure 2) is identified as the AND-split activity linked to a_2 and a_4 by using the symbol ")", otherwise the OR-split (as shown by activity a_1 in p_1). In the case of merge, the symbol "(" represents AND-merge activity (a_6 in p_2), otherwise OR-merge (a_5 in p_1).

With regard to process variants, customized processes also can incur combinatorial problems in modeling activity from its start to the end. When the number of activities in process variants gets increased, the problem in finding the reachability graph will increase exponentially and often becomes intractable. Thus, the PSM approach will also deploy a heuristic search algorithm to overcome the computation problem.

3. Proximity Score Measurement.

By considering graph theory approach, we describe the proximity score measurement in this section.

3.1. Proximity score measurement (PSM) approach.

This study aimed to derive a proximity score among activities in a process variant. Before the score is obtained, the distance between activities is calculated as denoted in Definition 3.1.

Definition 3.1. *Path, Distance, Reachability. We define a set of paths PA_{ij} from an activity a_i to another activity a_j as an edge-connectivity. Since there might be multiple*

paths between the two activities, an element, pa_t, of the set is defined as the t-th path. $PA_{ij} = \{pa_t | t = 1, 2, 3, \ldots, T\}$

- $pa_t = a_i, a_k, a_{k+1}, \ldots, a_{k+m}, \ldots, a_{k+M}, a_j$, $(a_i, a_k) \in L$, $(a_{k+M}, a_j) \in L$, $(a_{k+m}, a_{k+m+1}) \in L$, $\forall m$
- *T is the number of paths from* a_i *to* a_j

If there is a path from a_i *to* a_j*, we say that* a_j *is reachable from* a_i*, and to represent that reachability, we use a '*$\rightarrow$*' notation. Alternatively, we use a "*$||$*" notation to represent that activity* a_j *is not reachable from* a_i*.*

- $a_i \rightarrow a_j : a_j$ *is reachable from* a_i
- $a_i || a_j : a_j$ *is not reachable from* a_i

When there is a path $pa_t (\in PA_{ij})$ *from* a_i *to* a_j*, the distance between the two activities is*

$$d_t = |pa_t| - 1 = M + 1 \tag{1}$$

When we consider the path between two activities, there can in fact be multiple paths, owing to split structures, between the two. Multiple paths exist where there are split and merge activities. Those activities are considered as a single block structure. Bae et al. [22] explains more about the block structure as it pertains to parallel activities such as the AND-, OR-, and iterative-blocks. For this reason, we introduce the concept of the average distance between two activities that are reachable.

Definition 3.2. *Average Path Distance. Average Path Distance is the average distance among several paths, from a split activity* a_s *to a merge activity* a_m*.*

– Average Path Distance of AND-block

The average path distance $\bar{d}_{sm}$ *between* a_s $(|SA_s| > 1)$ *and* a_m $(|MA_m| > 1)$ *is denoted as*

$$\bar{d}_{sm} = \frac{\sum_{pa_t \in PA_{sm}} d_t}{|SA_s|} \tag{2}$$

where $|SA_s|$ *is the number of forks in the split activity.*

– Average Path Distance of OR-block

The average path distance $\bar{d}_{sm}$ *between* a_s $(|SA_s| > 1)$ *and* a_m $(|MA_m| > 1)$ *is denoted as*

$$\bar{d}_{sm} = \sum_{pa_t \in PA_{sm}} pr_t * d_t \tag{3}$$

where pr_t *is the probability of executing the t-th path between the s-th activity and the m-th activity and* $\sum_t pr_t = 1$*. The initialization value of* pr_t *is usually determined by experts or any previous experiences. The* pr_t *is equal to 1 if the relationship of activity* a_i *to* a_j *is direct sequential order. Example obtained from Figure 2* (p_1) *is*

$$\bar{d}_{15} = \frac{1+2}{2} = 1.5$$

– Average Path Distance of iterative-block

Let the iteration go from a_j *back to* a_i*, and let the probability of re-execution be* pr_l *(see Figure 3). To measure the activity distance in the iterative-block (the iteration from* a_j *to* a_i*), first we need to calculate the looping distance, defined as* $\bar{d}_{ij}$*. The existence of the iterative-block is specified by a process design user at process build-time, and the re-execution probability can be determined from previous data or by expert knowledge. We can calculate the distance between two activities, both of which are located within the*

iterative-block, by multiplying the distance by the re-execution probability, as shown in Equation (4). It is denoted as

$$\bar{d}_{ij} = d_{ij} + pr_l.d_{ij} + pr_l^2.d_{ij} + \ldots = \frac{d_{ij}}{1 - pr_l} \tag{4}$$

where d_{ij} is the distance between a_i and a_j in the iterative-block.

FIGURE 3. Iterative-block abstraction model

Definition 3.3. *Activity Proximity Score (APS). We define Q_{ij} as the existence probability of path pa_s (PA_{ij}) from a_i to a_j in all existing processes, which probability is called the Total Activity Proximity Score (TAPS). To compute Q_{ij}, we have to obtain the activity proximity value in each process. The value, which is denoted by q_{ij}^k, is defined as*

$$q_{ij}^k = \frac{h^k(i,j)}{d_{ij}^k} \tag{5}$$

is the average path distance between activity a_i and a_j of the k-th process. Each pair of activities (a_i and a_j) has a single value of q_{ij}^k, $k = 1, 2, 3, \ldots, K$, where is the APS of the k-th process index, and K is the total number of processes. If there is no relationship between activity a_i and a_j in the k-th process, or if it is denoted as $a_i || a_j$, then $q_{ij}^k = 0$. If the activity relationship is reachable ($a_i \rightarrow a_j$) with distance $\bar{d}_{ij} = 1$ in the k-th process, we obtain $q_{ij}^k = 1$. The APS value, q_{ij}^k, can have a value between 0 and 1, and a high value of APS indicates that the distance between the two activities is reachable. To gain the average proximity score (Q_{ij}) of activity a_i and a_j considering all K process variants, we should sum all q_{ij}^k and divide it by K. The average proximity score, equal to the existence probability of activity a_i and a_j among K process variants, is measured by the equation

$$Q_{ij} = \frac{\sum_{k=1}^{K} q_{ij}^k}{K} \tag{6}$$

If activities a_i and a_j are adjacent in all process variants, i.e., q_{ij}^k is equal to 1 for all K, then Q_{ij} is definitely equal to 1.

Definition 3.4. *Total Proximity Score (ρ_k) Total Proximity Score (TPS) of a process p_k, which is denoted by ρ_k can be represented by the following expression.*

$$\rho_k = \frac{\sum_{i=1}^{m} \sum_{j(\neq i)=1}^{m} \frac{Q_{ij}}{\bar{d}_{ij}^k}}{\sum_{i,j} h^k(i,j)} \tag{7}$$

TPS of a process variant is a total score to represent the proximity of it among other process variants. To enable process comparison, we need to assess the proximity score of each process. We denote ρ_k to measure the TPS of the K-th process. The TPS is determined by the summation of all existing Q_{ij} over the distance divided by the total combination of pair activities that can occur in the new process. The denominator $\bar{d}_{ij}^k$ in Equation (7) has a different function from that of the denominator in Equation (5). It is used to normalize the TPS based on each activity relationship property. Additionally,

parallel activity is inconsequential to the scoring. Thus, the score is divided by the number of reachable activities that exist in the K-th process variant.

Definition 3.5. *Possible Predecessors and Possible Successors. Based on the APS function, we mathematically formalize the definition of possible predecessors and possible successors as follows.*
$\bullet pred(a_n) \subseteq \{a_r | \exists k \text{ such that } q^k_{rn} > 0\}$
$\bullet succ(a_n) \subseteq \{a_u | \exists k \text{ such that } q^k_{nu} > 0\}$

TABLE 1. Activity descriptions in process variants

Activity	Activity Name	Description
a_1	Order Entry	Receiving an order (create a PO [purchase order])
a_2	Order Review	Review order based on the PO
a_3	Financial Check	Check the financial history of the customer
a_4	Stock Check	Check the stock based on the product ordered
a_5	Manager Review	Review the activities and decisions of predecessors'
a_6	PO Release	Activity to release the PO

3.2. **Procedure to measure proximity.** To demonstrate the PSM, we conducted a simple case study using process variants. Suppose that six process variants which consist of the activities defined in Table 1 are stored in the process repository, and a user has designed a new process with these activities as in Figure 5. We considered a new process design (p_{new}) with 6 activities (defined in Table 1), and there were 6 existing process variants in our process repository (Figure 4). Based on the variants in the repository, we calculated the PSM for the new process. The procedure is as follows.

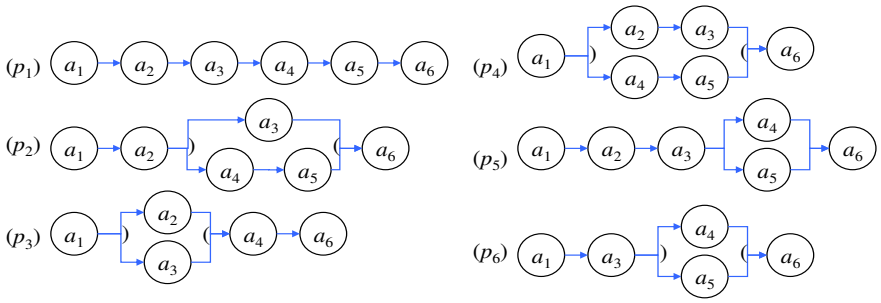

FIGURE 4. Process variants and APS measurement

(Step 1) Calculation of APS: For every pair of activities in a process, the proximity score (q^k_{ij}) can be obtained from Equation (5) with $h^k(i,j)$ and d^k_{ij} values. For example, we obtained the values $h^1(3,5) = 1$ and $d^1_{35} = 2$. Thus, we calculated $q^1_{35} = 0.5$.

(Step 2) Calculation of TAPS: The TAPS, Q_{ij}, can be obtained as follows.

$$Q_{12} = \frac{\frac{1}{1}+\frac{1}{1}+\frac{1}{1}+\frac{1}{1}+\frac{1}{1}+\frac{1}{1}+0}{6} = \frac{5}{6} \quad Q_{35} = \frac{\frac{1}{2}+0+0+0+\frac{1}{1}+\frac{1}{1}}{6} = \frac{2.5}{6}$$

Table 2 summarizes all of the Q_{ij}.

(Step 3) Calculation of TPS: The TPS of the new process illustrated in Figure 5 is obtained as follows.

$$\rho_{new} = \frac{0.83 + 0.67 + 0.53 + \ldots + \frac{0.29}{3}}{15-3} = \frac{5.12}{12} = 0.426$$

TABLE 2. TAPS measurement (Q_{ij})

	a_1	a_2	a_3	a_4	a_5	a_6
a_1	–	0.83	0.67	0.53	0.32	0.29
a_2	0	–	0.67	0.5	0.22	0.33
a_3	0	0	–	0.67	0.42	0.64
a_4	0	0	0	–	0.5	0.75
a_5	0	0	0	0	–	0.83
a_6	0	0	0	0	0	–

The TPS of the new process variant was calculated for comparison with other existing process variants.

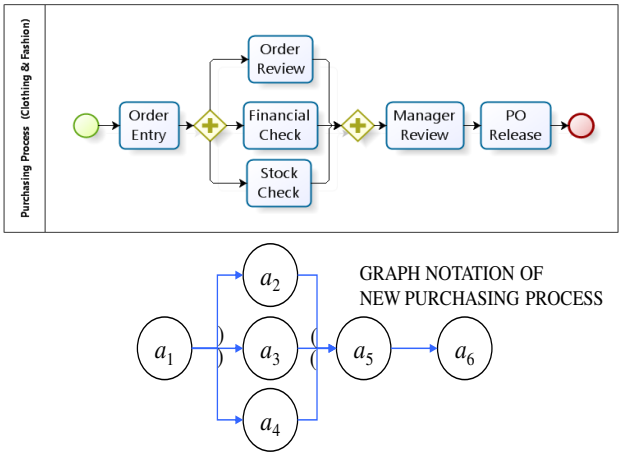

FIGURE 5. A new process variant designed by a user

Let the total number of activities in a process be N. In order to find the activity pair, the computation complexity becomes $O(N^2)$ times. To develop efficient solutions to reachability problems, the concept of transitive closure is employed. Transitive closure in graph theory involves the construction of a path, by means of a data structure mechanism, which path renders a possible solution to the reachability problem [10]. The Warshall algorithm, as proposed for solution of the transitive closure problem, includes the t-th iteration of loop sets in order to find any pairs of activities with indices greater than 1, since there are T paths from a_i to a_j (Equation (1)). Thus, there will be necessary $|pa_t|-1$ times calculations (equal to $N+1$ activities) to measure the path distance from a_i to a_j. The complexity of computing the TPS, therefore, is $O(N^3)$.

4. Applications and Experiments.

In this section, we discuss the applications of PSM in the business process field. We use a Business Process design analysis scheme incorporating Proximity Score Measurement (BP-PSM) to represent our approach.

4.1. Finding process reference using heuristic.

Since the TAPS (Q_{ij}) represents the proximity of all pairs of activity relationships, we can create a new process model that optimizes the sum of Q_{ij}. We believe that this model can represent the characteristics of all variants in the repository, and thus can be a reference model. Theoretically, a possible reference model can be found by finding the optimal path from a start activity to an end activity. However, deriving a reachability graph from a group of activities is

a computational problem that can increase exponentially in complexity, often becoming intractably large. This can be formalized as a combinatorial optimization problem, which issue is discussed in Yahya *et al.* [23]. In our approach, a heuristic search algorithm by finding the minimum cost path is deployed. The algorithm is guaranteed to produce the optimum result within a reasonably acceptable amount of time when the admissibility is proved.

In this section, we present a method of finding process reference models using a well-known heuristic approach, A-star algorithm. This algorithm is generally used to estimate the shortest distance from start activity to end activity, which equals the travel distance plus the predicted distance ahead. Hence, A-star tries to minimize the following equation to find the next step to a goal.

$$f(n) = g(n) + h(n) \tag{8}$$

In Equation (8), $f(n)$ is the evaluation function, $g(n)$ is the actual cost of the path from a start node to a node n with minimum cost, and $h(n)$ is the actual cost of an optimal path from n to a preferred goal node [24]. In our study, we modified the functions ($g(n)$ and $h(n)$) in order to be able to obtain the highest proximity score by minimizing the $f(n)$ evaluation function value. In applying the A-star algorithm to business process models, the proximity score is used for calculating both $g(n)$ and $h(n)$. In our method, $g(n)$ indicates the path calculation from start activity a_1 to an activity a_n, for several alternative paths. Meanwhile, $h(n)$ represents the estimation of the proximity score for the best path, from all activities in the CLOSED list to the end activity.

A-star Algorithm for finding process reference (A*-PR)

1. Create a process sub-graph p^s, consisting solely of the start activity. Put a_1 as a start activity on an activity list called OPEN. Create a list called CLOSED that is initially empty.
2. Select the first activity on OPEN, remove it from OPEN, and put it on CLOSED. Call this activity a_n.
3. If $a_n = a_E$, exit successfully with the process p^s as a reference process.
4. Expand node a_n, generating the set, M, of which each element is an activity that has any proximity to a_n as a succeeding activity of it, that is, $M = \{a_i | \exists K \text{ such that } q^k_{ni} = 1\}$.
5. For each member of M, a_m, update p^s by deciding whether to link it to a_n.

5-1 If $a_m \notin$ OPEN and $a_m \notin$ CLOSED

Establish a link from a_n to a_m (l_{nm}).

5-2 Add a_m to OPEN.

5-3 For each member a_m in OPEN or CLOSED

Find an activity a_n that has the minimum of $\widetilde{\delta}_{nm}$.

5-4 For each member a_m in CLOSED

Find an activity a_j that is already in $pred(a_m)$

if $\widetilde{\delta}_{nm} < \widetilde{\delta}_{jm}$, establish a link l_{nm}; otherwise, establish a link l_{jm} for any a_j which satisfy the following condition in the repository.

($|SA_j| > 1$ and ($\exists K, j^+$ such that $q^k_{jj^+} = 1$)) or ($|MA_j| > 1$ and ($\exists K$, j^- such that $q^k_{j^-j} = 1$)).

6. Reorder the list OPEN in order of increasing $\hat{f}$ values. (Ties among minimal values are resolved in favor of the deepest node in the search tree).

The evaluation function $\hat{f}(n)$ attempts to measure the proximity score based on path generation by $\hat{g}(n)$ and $\hat{h}(n)$. In order to find the minimum value of evaluation function $f(n)$, we define an estimated cost function ($\widetilde{\delta}_{ij}$) for paths a_i and a_j (Equation (12)). $\widetilde{d}_{ij}$ denotes the estimated distance of the i-th and j-th activities where $a_i \rightarrow a_j$ at $pred(a_n)$.

The notation a_u represents the members of $succ(a_n)$ until a preferred end activity (a_E). $\hat{g}(n)$ is the proximity evaluation function from a_S to a_n with minimum cost thus far found by A*-PR, and $\hat{h}(n)$ is an estimation of the proximity evaluation function of an optimal path from a_n to a preferred end activity a_E. To normalize the value of $\hat{f}(n)$, we divide the summation of $\hat{g}(n)$ and $\hat{h}(n)$ by the total combination of existing paths in the process generation. The notations of $\hat{f}(n)$ are defined in Equation (12).

$$\widetilde{\delta}_{ij} = \frac{\widetilde{d}_{ij}}{Q_{ij}} \tag{9}$$

$$\hat{g}(n) = \sum_{a_i, a_j \in pred(a_n) \cup \{a_n\} \text{ and } a_i \to a_j} \widetilde{\delta}_{ij} \tag{10}$$

$$\hat{h}(n) = \sum_{a_i \in pred(a_n) \cup \{a_n\}, a_u \in succ(a_n), a_i \to a_j} \widetilde{\delta}_{iu} \tag{11}$$

$$\hat{f}(n) = \frac{\hat{g}(n) + \hat{h}(n)}{\sum_{i,j \in A} h^s(i,j)} \tag{12}$$

The A*-PR in our study is admissible, that is, $\hat{h}(n) \leq h(n)$ for all n. Figure 6(a) illustrates how the A*PR works, and also represents the admissibility of A*-PR. In this search approach, the best reference model will satisfy $\hat{f}(E) = f(S)$. It is certain that when $\hat{f}(E) = f(S)$, $\hat{h}(n) \leq h(n)$. Figure 6(b) presents an example with $S = 1$ and $E = 6$. Suppose that we want to measure the evaluation function when $n = 4$; we can obtain the analysis for $\hat{h}(n)$ and $h(n)$ using an estimated cost function ($\widetilde{\delta}_{ij}$) and an actual cost function (δ_{ij}) as follows:

$$\hat{h}(4) = \widetilde{\delta}_{16} + \widetilde{\delta}_{26} + \widetilde{\delta}_{36} + \widetilde{\delta}_{46} \text{ (when it connects } a_n \text{ to end activity } a_6)$$
$$h(4) = \delta_{15} + \delta_{25} + \delta_{35} + \delta_{45} + \delta_{16} + \delta_{26} + \delta_{36} + \delta_{46} + \delta_{56}.$$

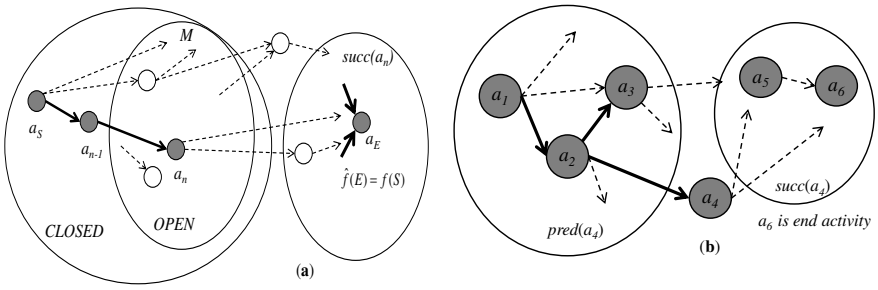

FIGURE 6. Mechanism of finding process reference model

The detailed explanation of the proof of the admissibility of A*-PR is provided in Yahya *et al.* (2010) [25]. It shows that this algorithm is guaranteed to find an optimal path to the goal, owing to its admissibility. By means of the A*-PR algorithm, we obtained the reference model shown in Figure 7.

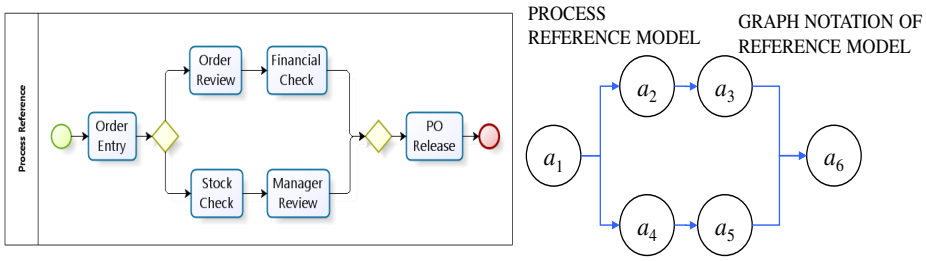

FIGURE 7. Process reference model with minimum value of $f(n)$

4.2. **Process model versioning based on process homogeneity.** Another application of the BP-PSM is process versioning by analysis of the homogeneity of process variants. Using the TPS value (ρ_K), we can analyze the degree to which process versions of a process model are homogeneous. We can conduct a homogeneity test for a newly designed process. If it is statistically significant that it belongs to a group of process variants in terms of the TPS value, we can make it a new member of the group. Otherwise, we have to create a separate, new process model.

Previously, Figure 4 represented variant processes originating from a typical purchasing process. Many custom customers would like to have a process customized to meet their needs. Some activities, accordingly, should be aligned with those customers' behaviors. Therefore, a process designer must design a customized process within the scope of the process variants. In such a setting, the activity flow might differ; however, the business objective should remain the same. Figure 5 shows a new purchasing process variant for a buyer who deals in clothing and fashion products. The BP-PSM can help a designer model a new process while remaining close to the same process variants. Otherwise, the system triggers a mechanism to generate a new process group.

We need to ensure that the new process (ρ_{new}) in Figure 5 is homogeneous with the group of process variants in Figure 4. The PSM of all of the existing process variants (Table 3) was assumed to be a normal distribution. A t-test statistical analysis hypothesis was carried out to determine whether a new process is homogeneous to a group of process variants. The statistical analysis represented in Figure 8 showed the T value result (–2.45) and the associated p-value (0.058). This p-value indicates that there is a 5.8% probability that we would have obtained our sample if the μ was actually 0.426388. It is certain that the new created process was statistically consistent among the existing process variants, since the p-value was greater than the α-level ($\alpha = 0.05$).

TABLE 3. TPS of each process variant

Process	#Activity	$\sum h^k(i,j)$	TPS
1.	6	15	0.356922
2.	6	14	0.413579
3.	5	9	0.429074
4.	6	11	0.384242
5.	6	14	0.403631
6.	5	9	0.415185

4.3. **Process design guidance.** To create a new process variant, a designer should have knowledge of the existing process design. He/she should know the structure of

One-Sample T: New Process
Test of μ = 0.4268388 vs not μ = 0.426388

Variable	N	Mean	StDev	SE Mean	95% CI	T	P
C1	6	0.400439	0.025986	0.010609	(0.373169, 0.427709)	-2.45	0.058

FIGURE 8. Statistical analysis in homogeneity

most process variants in the repository so as to avoid process redundancy. The BP-PSM helps the designer to determine the activity flow with regard to the proximity of existing activities. The system will display the nearest succeeding activity based on a descending order of Q_{ij}.

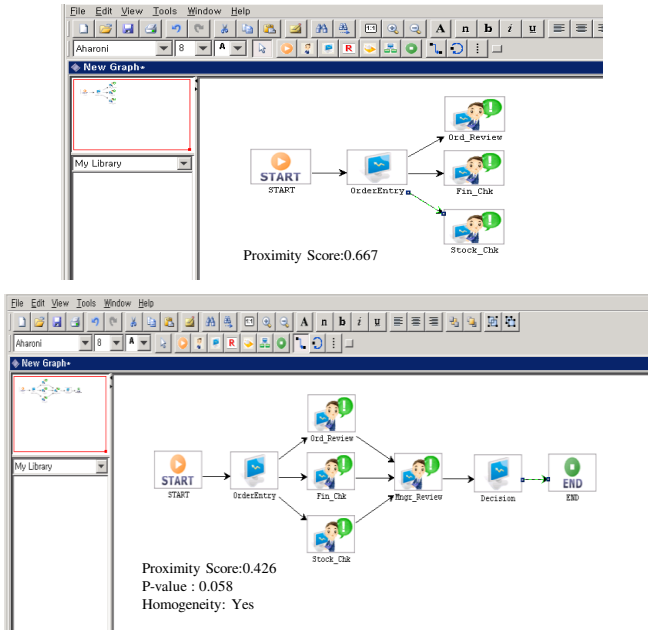

FIGURE 9. A process design tool with proximity score calculation

The prototype of the process design is shown in Figure 9. A user who intends to design a new process can select the appropriate activity from the list. The system reckons the proximity score between each of the connected activities, either directly or indirectly connected. Eventually, the system shows a proximity score for the chosen activities in real time. The proximity score will not show any compliance result if the process has not reached the end activity a_E. Once the end activity a_E is selected by the user, the system directly generates a homogeneity analysis among other existing process variants, according to the p-value and consistency results.

4.4. **Experiments.** We conducted experiments to validate the effectiveness of the BP-PSM on ten process groups with various numbers of activities. Table 4 lists the properties of process data employed. There are 10 groups of processes, each of which consists of 10 process variants with different sizes of processes.

Using our data in Table 4, we can analyze the degree of sequentiality and the degree of parallelity in the business process repository, as denoted in Equations (13) and (14), respectively. The degree of sequentiality is a measure of the ratio of the sequence activity

in a process, including the average distance in blocks. Equation (13) shows that the sequentiality is equal to the maximum distance of a process divided by the number of activities in it. A shorter distance, then, indicates the possibility of high parallelity. The degree of sequentiality and parallelity metric are modification of sequentiality and separability ratio in Mendling, 2008 [26]. A process model with a high sequentiality ratio should be less likely to contain errors than one with a low sequentiality ratio. In contrast to the parallelity ratio, sequentiality relates to the fact that sequences of consecutive tasks are the simplest components of a process model. High parallelity ratio shows that most activities existed in the parallel block imply an increase in complexity of overall model.

$$sequentiality = \frac{max\ dist}{\#\ of\ activity} \tag{13}$$

$$parallelity = \frac{(avg.\#\ of\ block) * (avg.dist\ in\ blocks)}{max\ dist} \tag{14}$$

The degree of parallelity, by contrast, is a measure of the ratio of the parallel blocks based on the total distance of a process. Equation (14) denotes parallelity as the multiplication of the average number of blocks and the average distance in the blocks, divided by the maximum distance (max dist). The results for the degrees of sequentiallity and parallelity are shown in the two right-most columns of Table 4. In process group IX and X, the parallelity exceeds one since the process may consist block and distance in block more than the average value (high standard deviation).

Table 4. Degree of *sequentiality* and *parallelity*

Group	# of Activity	Max Distance	Avg.# of block	Avg. distance in block	Sequentiality	Parallelity
I	4-7	3.48	1.1	2.08	0.570	0.657
II	9-11	5.68	1.8	2.705	0.546	0.857
III	14-15	8.7	3	2.307	0.588	0.795
IV	19-20	11.7	3.4	2.258	0.616	0.656
V	21-25	12.467	4.1	2.773	0.537	0.911
VI	28-30	16.205	4.8	2.626	0.567	0.778
VII	32-35	16.168	5.1	2.951	0.488	0.931
VIII	35-40	17.811	5.5	2.866	0.465	0.885
IX	40-45	18.628	7.4	3.061	0.433	1.216
X	43-50	19.502	6.7	3.067	0.413	1.053

Table 5 shows the TPS values of the different process groups. The number of activities in the process variants affects the values of the TPS, in that a greater number of activities bring about a longer distance. Thus, certainly, it decreases the TPS value, and the means of which are plotted in Figure 10. Figure 11 displays a detailed measurement result, in which each process index for each process group is categorized into three subgroups: high, medium, and low. By calculating the average activity number based on the low-mid-high types, we could certify that a greater number of activities produce a lower value of TPS. We also found that a greater value of activity number and of standard deviation results in a decrement of TPS deviation. In other words, a greater number of activities show a tendency to be more homogeneous (Figure 12).

TABLE 5. Process variant examples

Group		# of Activity	Mean of activity#	St Dev. of Activity#	TPS mean	TPS St Dev.
low	I	4-7	6.1	0.994	0.706	0.062
	II	9-11	10.4	0.699	0.529	0.039
	III	14-15	14.8	0.422	0.442	0.046
mid	IV	19-20	19	1.247	0.376	0.031
	V	21-25	23.2	1.549	0.352	0.047
	VI	28-30	28.6	1.505	0.297	0.029
	VII	32-35	33.1	1.449	0.288	0.036
high	VIII	35-40	38.3	1.767	0.267	0.035
	IX	40-45	43	1.563	0.258	0.035
	X	43-50	47.2	2.394	0.249	0.037

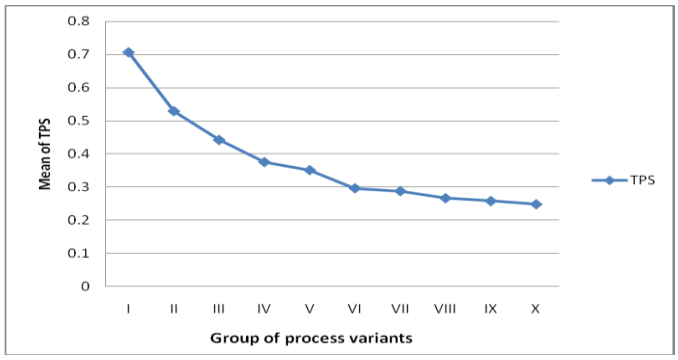

FIGURE 10. Mean TPS with different process sizes

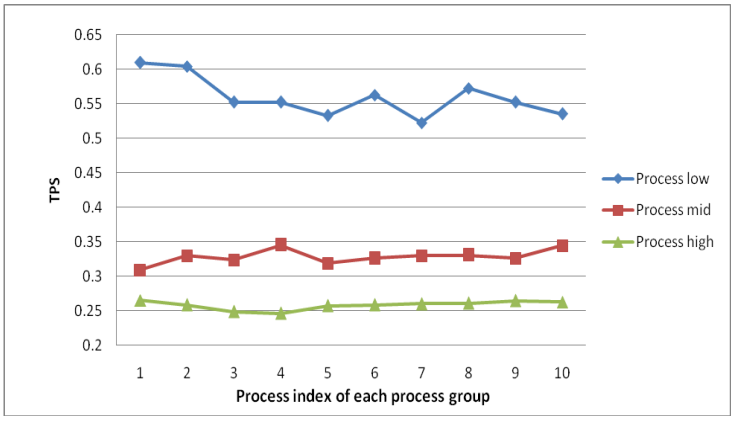

FIGURE 11. TPS value based on low-mid-high of process sizes

We analyzed the TPS value execution time for the existing 10 process groups listed in Table 4. The execution time is plotted in Figure 13. As is apparent, the execution time increases as the process size becomes larger.

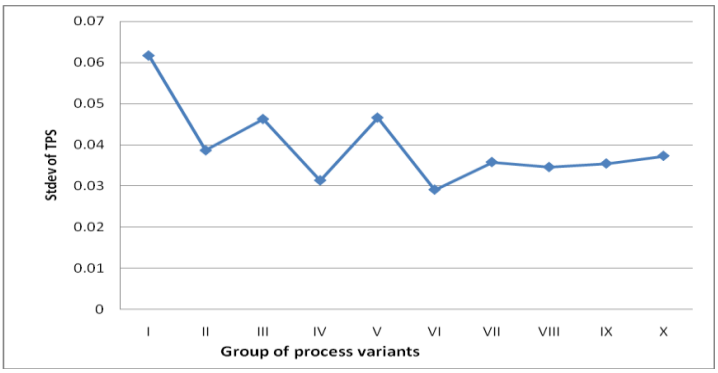

FIGURE 12. Deviation of TPS with different process sizes

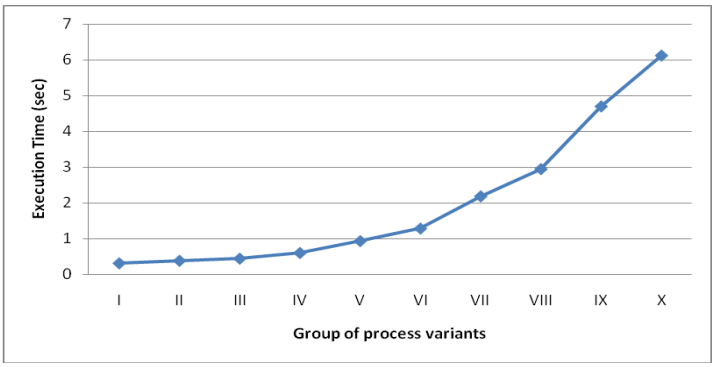

FIGURE 13. Execution time with different process sizes

5. **Related Work and Discussion.** There are two areas of research which are mostly related to this work: One is process customization and the other is process mining. In this section we give a brief overview of these two areas of research.

5.1. **Process customization.** Customer-focused strategies and customized services have become increasingly popular in the 2000s [27]. However, process customization in BPMS has many things to deal with. For example, to understand customer needs is one of the difficult tasks of a process designer. Nevertheless, customization can offer the competitive advantage of increased customer value and better service [11, 27].

Thomke [28] stated that the impact of a change in a business process can extend beyond the specific aspect that has been changed, to affect, for example, preconditions, the inputs or outputs requirement for other activities. Therefore, customers' new requirements will necessitate further efforts to meet the requisite quality of service. The approach to process design selection pursued in the present study did not address attribute changes, leaving that issue to further research. Nonetheless, Soffer's notion significantly influenced the ideas that were pursued.

We adopted the concept of "customization" from [29]. This study employed a mass customization strategy in order to design travel packages minimizing the operation and processing costs to the service provider and maximizing customer satisfaction. Hidden relations discovered using data mining tools were used to identify the rules of association with this mechanism.

For customer convenience, a process modeler should substitute a homogeneity tool for similarity measurement. The process-homogeneity concept addresses the issue of customer process design being the same as prior processes, whereas similarity measurement aims to enumerate the distance between the two processes and provide information on which process is the most similar according to the similarity value. Most of the previous research has treated similarity measurement rather than homogeneity [12, 13, 15].

5.2. **Process mining approach.** A workflow-clustering method based on process similarity, BPClustering proposed by [15] is a two-phase approach to classifying domains and analyzing patterns in a process model. Domain classification executes an activity similarity measure, whereas pattern analysis runs a transition similarity measure. An implementation using cosine measures for the similarity of either activity or transition is claimed to support process repository analysis and new process design. This approach affords distance information between two processes rather than homogeneity among all stored processes in a database.

Another approach, called Business Process Similarity Analysis Tools (BPSAT), proposed by [12], measures similarity using a process dependency graph that is converted into a normalization matrix. The distance measure used in this approach is considered as a quantitative and qualitative tool in process mining. Moreover, the important aim of this approach is to reduce or minimize the costs incurred in the design phase. The process dependency graph represents the relationship among activities within a process. However, there is no exact value by which to express the split and merge activities.

Process Variants Mining (PVM) proposed by [14] was designed to satisfy the need for deriving a process model that is easily configurable. The authors claimed that their approach could create a generic process model allowing for easy and optimized configuration of process variants. Weijters and Aalst proposed a technique for process mining using WF-Nets [18]. This technique validates workflow processes by uncovering and measuring the discrepancies between a build-time model and a run-time execution process. The pertinent paper provided insight into the construction of the dependency and frequency of activities in a process instance. Aalst *et al.* presented an algorithm to extract a process model from such a log and to represent it in terms of a Petri net [17]. This research attempted to demonstrate that it is not possible to discover any arbitrary workflow process.

Some applications that can implement this approach are the process revision and versioning systems. Homogeneity analysis helps the process designer to determine whether a new process is homogenous within a particular cluster or requires a new process model. The idea of homogeneity, as discussed in Section 4.2, can be applied to addressing the issues of process revision and versioning as well. When the result statistically proves that the null hypothesis is rejected, the system automatically generates a new version instead of a revision. Reader can refer to [22] for detail issue about process version. This surely facilitates a process designer's delimitation of the scope of a process modeling system in terms of creating new process version.

In addition to measurement of the proximity of process structure, the dependency among activities can be a consideration. Activity dependency [30] and policy-driven process mapping [31] are some of the research on mapping activities in a process. Volkner *et al.* [4] proposed a decision support technique for support of business process planning. Binding the task and UI proposed in [32] is considered as one approach to measure the dependency. However, we left this issue over for further research.

5.3. **Discussion issues.** Table 6 compares the present qualitative analysis with previous similar studies. BP-PSM offers a greater benefit by measuring all processes simultaneously

rather than performing 1-to-1 process similarity measurements by BPSAT and BPClustering. BP-PSM can assess the distance of all related activities, which is an advantage over BPSAT. The split-merge detection function is an additional function and an additional improvement on BPSAT. BP-PSM can measure the proximity of process instances to determine the closeness of existing instances and find, thereby, the reference models. WM-EL and GMA can be applied for finding reference models of process instances; however, BP-PSM offers greater ease of process modeling. For all of these reasons, BP-PSM is considered to be a better, more convenient approach to process design by novice designers.

TABLE 6. BP-PSM qualitative analysis compared with previous studies

	BPSAT [12]	BPClustering [15]	WM-EL [17]	GMA [13]	BP-PSM
Process Measurement	1-to-1	1-to-1	All process instances	All processes (models)	All processes (models & instances)
Activity Measurement	Adjacent activities	All activities	All activities	Similar activities	All activities
Split-Merge	undetected	Detected	Detected	Detected	Detected
Measurement Method	δ-comparability	Similarity Measure	α-algorithm	Graph edit similarity	Proximity score measurement (PSM)

There is a need to improve the mechanism with regard to establishing the process reference model. First, the search process p^s is unable to specify the parallel block, neither the AND- nor the OR-block. In addition, it is also necessary to improve the search process for certain level of sub-processes. Second, we limited the process variants so as to disregard the loop function in finding the reference model. In the case of the loop, some extended variables and rules should be established in order to create a more representative reference model. Third, the search process p^s, as was the case with regard to the loop function, did not take the activity semantic dependency into account. Fourth, degrees of sequentiality and parallelity that can be an additional analysis tool in the modeling system are not considered. All of these issues will be the subjects of further research.

6. **Conclusions.** We have presented a proximity score measurement based approach to processing modeling, called BP-PSM. Our approach has three salient features. First, it utilizes a proximity score to provide an analysis of the degree to which an activity is related with another activity in business processes. We argue that this analysis is critical in assisting process designers to initiate their process design with the best possible process reference model. Second, we developed a suite of methods for BP-PSM based convenient process modeling, particularly suitable for novice designers, including the method of determining the proximity score measurement (PSM), and the methods of finding the respective process reference model and calculating homogeneity.

We developed a heuristic algorithm, called A*-PR, to address the complexity of combinatorial problems involved in PSM computation. Our mechanism to determine the homogeneity of new process designs among existing processes is novel. We demonstrated that a BP-PSM based process reference model is a convenient and effective way for a

designer lacking experience to be guided to design his own process model. The homogeneity score can help the process designer to determine the suitable class to which a new model may belong. Third but not the least, we developed a prototype of our system and conducted the experiments to evaluate the effectiveness of our approach. Our experimental results show that the PSM approach is efficient and effective for process designers to perform process modeling in BPMS environments. Our research continues along several dimensions. First, we continue our investigation on extending the BP-PSM to incorporate cost and time dimension in improving business process modeling convenience. Each relationship score can connect to a cost or time value, indicating the effectiveness of this approach. Second, we are extending the homogeneity of the attributes of activities to an activity dependency approach in order to support process versioning and the revision of configuration in terms of both the process structures and the activity attribute properties.

Acknowledgment. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0025650). The last author is partially sponsored by grants from NSF NetSE, NSF CyberTrust, an IBM SUR grant, and a grant from Intel Research Council.

REFERENCES

[1] H. Smith and P. Fingar, *Business Process Management – The Third Wave*, Meghan Kiffer Pr., 2006.
[2] W. M. P. van der Aalst and K. M. van Hee, *Workflow Management: Models, Methods, and System*, MIT Press, Cambridge, 2002.
[3] G. M. Giaglis, A taxonomy of business process modeling and information systems modeling techniques, *The International Journal of Flexible Manufacturing Systems*, vol.13, no.2, pp.209-228, 2001.
[4] P. Volkner and B. Werners, A decision support system for business process planning, *European Journal of Operation Research*, vol.125, no.3, pp.633-647, 2000.
[5] M. Laguna and J. Marklund, *Business Process Modeling, Simulation, and Design*, Perason Prentice Hall, USA, 2004.
[6] A. A. Rad, M. Benyoucef and C. E. Kuziemsky, An evaluation framework for business process modeling languages in healthcare, *Journal of Theoretical and Applied Electronic Commerce Research*, vol.4, no.2, pp.1-19, 2009.
[7] R. Lu and S. Sadiq, A survey of comparative business process modeling approaches, *BIS 2007, LNCS*, vol.4439, pp.82-94, 2007.
[8] A. M. Magdaleno, C. Cappelli, F. Baiao, F. Santoro and R. M. Araujo, A practical experience in designing business processes to improve collaboration, *BPM 2007 Workshops, LNCS*, vol.4928, pp.156-168, 2008.
[9] A. Hallerbach, T. Bauer and M. Reicher, Context-based configuration of process variants, *The 3rh International Workshop on Technologies for Context-Aware Business Process Management (TCoB 2008)*, Barcelona, Spain, pp.31-40, 2008.
[10] R. Sedgewick, *Algorithms in Java, Part 5 Graph Algorithm*, Addison-Wesley, Pearson Education, 2004.
[11] H. Simon and R. J. Dolan, Price customization, *Marketing Management*, vol.7, no.3, pp.11-17, 1998.
[12] J. Bae, L. Liu, J. Caverlee, L. J. Zhang and H. Bae, Development of distance measures for process mining, discovery and integration, *International Journal of Web Services Research*, vol.4, no.4, pp.1-17, 2007.
[13] R. Dijkman, M. Dumas and L. Garcia-Banuelos, Graph matching algorithms for business process model similarity search, *BPM 2009, LNCS*, vol.5701, pp.48-63, 2009.
[14] C. Li, M. U. Reichert and A. Wombacher, Issues in process variants mining, *Technical Report TR-CTIT-08-10*, Center for Telematics and Information Technology, University of Twente, Enschede, 2008.
[15] J. Y. Jung, J. Bae and L. Liu, Hierarchical clustering of business process models, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12, pp.4501-4511, 2009.
[16] B. N. Yahya, H. Bae and J. Bae, Process design selection using proximity score measurement, *BPM 2009 Workshop, LNBIP*, vol.43, pp.330-341, 2009.

[17] W. M. P. van der Aalst, T. Weijters and L. Maruster, Workflow mining: Discovering process models from event logs, *IEEE Transactions on Knowledge and Data Engineering*, vol.16, no.9, pp.1128-1142, 2004.
[18] A. J. M. M. Weijters and W. M. P. van der Aalst, Process mining discovering workflow models from event-based data, *Proc. of the 13th Belgium-Netherlands Conference Artificial Intelligence*, 2001.
[19] M. O. Dayhoff, R. M. Schwartz and B. Orcutt, A model of evolutionary change in proteins, *Atlas of Protein Sequence and Structure*, vol.5, pp.345-352, 1978.
[20] S. Henikoff and J. G. Henikoff, Amino acid substitution matrices from protein blocks, *Proc. of Natl. Acad. Sci.*, USA, vol.89, no.22, pp.10915-10919, 1992.
[21] F. Buckley and F. Harary, *Distance in Graphs*, Addison-Wesley Publishing Company, 1990.
[22] H. Bae and Y. Seo, BPM-based integration of supply chain process modeling, executing and monitoring, *International Journal of Production Research*, vol.45, no.11, pp.2545-2566, 2007.
[23] B. N. Yahya, H. Bae, J. Bae and D. Kim, Generating valid reference business process model using genetic algorithm, *International Journal of Innovative Computing, Information and Control*, vol.8, no.2, pp.1463-1477, 2012.
[24] P. Hart, N. J. Nilsson and B. Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions of Systems Science and Cybernetics*, vol.4, no.2, pp.100-107, 1968.
[25] B. N. Yahya and H. Bae, Finding process reference model using A*-PR heuristic algorithm, *Technical Report*, Pusan National University, Busan, South Korea, 2010.
[26] J. Mendling, Metric for process models, *Lecture Notes in Business Information Processing*, vol.6, pp.103-133, 2008.
[27] P. Suomala, M. Sievanen and J. Paranko, Customization from the after sales point of view – Implications of product and item customization for spare-part business, *Technovation*, vol.24, pp.831-840, 2004.
[28] S. Thomke and E. v. Hippel, Customers as innovators: A new way to create value, *Harvard Business Review*, 2002.
[29] B. Al-Salim, Mass customization of travel packages: Data mining approach, *International Journal Flexible Manufacturing System*, vol.19, no.4, pp.612-624, 2007.
[30] J. Kang and J. F. Naughton, Schema matching using interattribute dependencies, *IEEE Transactions on Knowledge and Data Engineering*, vol.20, no.10, pp.1393-1407, 2008.
[31] H. J. Wang, J. L. Zhao and L. J. Zhang, Policy-driven process mapping (PDPM): Discovering process models from business policies, *Decision Support Systems*, vol.48, no.1, pp.267-281, 2009.
[32] Y. Zou, Q. Zhang and X. Zhao, Improving the usability of e-commerce applications using business process, *IEEE Transactions on Software Engineering*, vol.33, no.12, pp.837-854, 2007.
[33] H. Bae, E. Cho and J. Bae, A version management of business process models in BPMS, *AP-Web/WAIM 2007 Ws, LNCS*, vol.4537, pp.534-539, 2007.