International Journal of Innovative Computing, Information and Control
Volume **5**, Number **12**, December **2009**

ICIC International ©2009 ISSN 1349-4198
pp. 1–ISII08-123

HIERARCHICAL CLUSTERING OF BUSINESS PROCESS MODELS

Jae-Yoon Jung*, Joonsoo Bae**,[1] and Ling Liu***

*Department of Industrial Engineering
Kyung Hee University
Yongin-si, Gyeonggi-do, 446-701, Republic of Korea
jyjung@khu.ac.kr

**Department of Industrial and Information Systems Engineering
Chonbuk National University
Jeonju-si, Jeonbuk, 561-756, Republic of Korea
jsbae@chonbuk.ac.kr

***College of Computing
Georgia Institute of Technology
Atlanta, GA, United States
lingliu@gatech.edu

Received July 2008; revised December 2008

Abstract. *Business process is collection of standardized and structured tasks inducing value creation of a company. Nowadays, it is recognized as one of significant intangible business assets to achieve competitive advantages. We introduce a novel approach to business process analysis, which has more and more significance as process-aware information systems that are spreading widely over a lot of companies. In this paper, a methodology of business process clustering based on process similarity is proposed. The purpose of business process clustering is to analyze accumulated process models in order to assist new process design or process reengineering. The proposed methodology exploits structural similarity metrics of business processes. We illustrated the methodology with example processes inducing the hierarchical merged models from the process clusters.*

Keywords: Hierarchical clustering, Business process management (BPM), Workflow

1. **Introduction.** The business process management (BPM) is spreading to implement process-aware information systems for the purpose of various business innovation techniques, such as real-time enterprises, balanced score-card, and knowledge management. Many researches on BPM have provided methodologies of modeling and analyzing business process for the last decade. Especially, as more and more companies introduce process-aware information systems, classification and typology of business processes are getting more significant. Some researches on the issues have been mainly made to present process reference models based on generic business activities (for instance, Process Handbook project [1]) and to provide standard process models in electronic business (for instance, RosettaNet PIPs [2]). However, such researches do not support a method of company's analyzing the own process assets which are being accumulated in a variety of process-aware information systems.

In this paper, a methodology of clustering business processes is presented as one of the means to analyze business processes accumulated in information systems. We define business process clustering as a procedure of measuring the similarities among process models

[1] Corresponding Author

A preliminary and shorter version of this paper is presented in Proc. of the IEEE International Conference on Service Computing, IEEE Computer Society, pp.613-616, 2008.

and discovering the groups of similar processes. The results of business process clustering can be used to recommend appropriate process models or reengineering business processes by analyzing the patterns in each group. In our research, a structural process similarity measure is suggested to cluster business process models. The measure is extended from graph similarity measures in order to reflect on the characteristics of business process modeling which contains the dependency among activities, such as AND, XOR, and OR splits and loop.

The paper is organized as follows. We first discuss related work on business process analysis in Section 2. A similarity measure of business process is presented in Section 3, and the methodology of business process clustering is proposed along with an example of insurance processes in Sections 4 and 5. Finally, Section 6 concludes the paper.

2. **Related Work.** Although business process management systems are widely used in many enterprises, the research on design and analysis about complex business process is still not mature. One research result about business process types and classification for general purpose process is Process Handbook of MIT [1]. In this book, there are more than 5000 types of process warehouse in each domain and function. The functional categories are purchasing, supply chain management, marketing, sales, information systems, finance, engineering, and so forth, and the processes are defined according to the characteristics an task of each functional category. This is a typical research about process warehouse construction on the basis of domain and purpose. And this research provides a reference model for the process design by using process classification. Although the reference model for the business process design is the same concept in this paper, it provides a reference model not based on their own processes but based on the general and standard for all enterprises. Our research in this paper can provide a reference model by using the existing own processes in a specific organization.

One representative process analysis research on its own organization can be process mining [4-6]. Process mining can infer process model itself by analyzing execution results and event logs in the process warehouse or can analyze the process execution characteristics such as inter-relationship among activities and workload allocation to the participants. The process mining research has similar background with our research in that it also uses the process-based information system in order to analyze business process. The difference between process mining and our research is that process mining utilizes only process execution log data while our research compares and analyzes the process model itself. Another approach using event log data is proposed based on Petri-Net execution behavior [7]. This research proposed and implemented notions of fitness, precision, and recall in the context of Petri-Net in order to quantify process equivalence. However, this approach needs to have event log with typical execution sequences as a starting point.

The research on structural analysis of process models for the process design and improvement can be classified to process inheritance, process comparison metrics, and process evaluation metrics. Process inheritance tried to find the dynamic extension possibility of process by analyzing dependency relationship between process models [3,8]. This research was proposed to support dynamic change of process in case of process model expansion or change management in an enterprise. Process comparison metrics were introduced to design a new process model collaboratively by many experts [9] or recommend process models for design support [10]. The researches have analyzed many attributes of process such as structure, participant and condition. Recently research on process metrics is proposed to evaluate the stability and validity of process model. Cardoso [11] proposed process model complexity metrics, which computes the split and merge complexity in process model in order to reduce execution errors. Reijers and Vanderfeesten [12] tried to generate a balanced process model in the sense of cohesion and integration metrics, which is the level of information coupling between process models. The researches on process inheritance, comparison metric, and evaluation metric utilize process structure analysis and support process design through the concept of dynamic change possibility, similarity, complexity, cohesion and integration. The our clustering methodology is similar to the above researches in that it also adopts a process structural approach to the process model comparison, while our approach is differentiated in that it can recommend candidate process models to support process design.

3. **Similarity Measure for Business Processes.** In generally, business process is designed and automated in workflow engines of enterprise information systems, such as ERP (Enterprise Resource Planning), PLM (Product Lifecycle Management), and SCM (Supply Chain Management) [13]. In such systems, business process is often designed as a specific labeled directed graph, such as UML Activity Diagram [14], EPC (Event-driven Process Chain) [15], Petri-Net [16]. Moreover, XPDL (XML Process Definition Language), the standardized exchange format proposed by WfMC (Workflow Management Coalition) [17], and BPMN (Business Process Modeling Notation) recommended by OMG [18] are also based on graph-based process modeling.

3.1. **Process vectors.** We assume that business processes of a company are stored in process asset library (PAL) which can be expressed in $R = \langle A^{\diamond}, W^{\diamond} \rangle$. $A^{\diamond}$ is a set of the activities that can be used for designing business process models, and $W^{\diamond}$ is a set of the process models. The process model is designed as tuple $W = \langle A, T, \mathit{Split}, \mathit{Join} \rangle$, where A is a subset of $A^{\diamond}$ and include only the activities used in W, and T is the dependency relation among the activities in W. That is, T can be defined as $T \subset (A - A_s) \times (A - A_F)$, where A_s and A_F are the sets of starting and ending activities in W, respectively. Besides, *Split* and *Join* are the functions of mapping transitions T to specialized control-flow. $\mathit{Split} : T \rightarrow \{AND, XOR, OR\}$ and $\mathit{Join} : T \rightarrow \{AND, XOR, OR\}$.

In our research, two types of vector models are defined to express the structure of a business process model W: activity and transition vectors. The vectors are used to calculate the co-occurrence of activities and their dependencies in two processes.

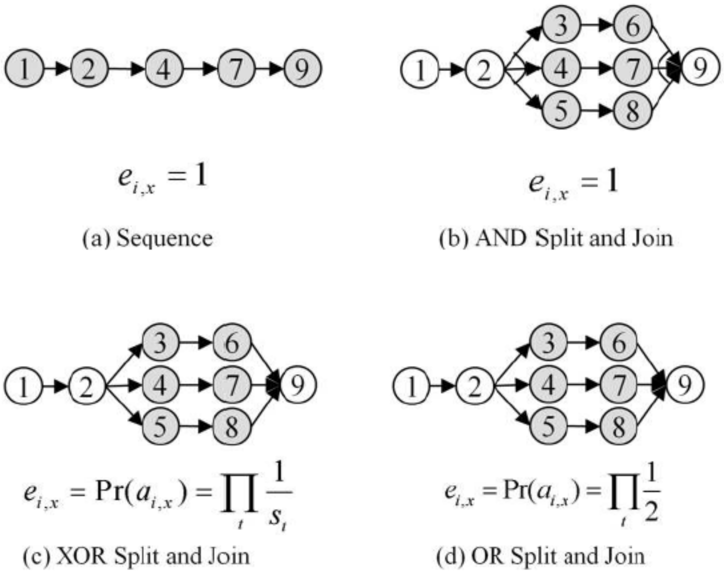
Suppose the total number of processes in R is $N = |W^{\diamond}|$, and that of activities in R is $n = |A^{\diamond}|$. First, the activity vector of process $P_x (1 \leq x \leq N)$ is an n-dimensional vector a_x, the element of which is the execution probability of the i-th activity in P_x, denoted $e_{i,x} (1 \leq x \leq n)$. Second, the transition vector of P_x is an n^2-dimensional vector t_x, the element of which is multiplication of execution probabilities of two activities and distance weight $(1/d_{ij,x})$ between the two. The following formula is expressing the two process vectors.

$$a_x = (a_{i,x}),\ a_{i,x} = (e_{i,x}),\ \mathit{where}\ i = 1, ..., n \tag{1}$$

$$t_x = (t_{ij,x}),\ t_{ij,x} = \frac{1}{d_{ij,x}} e_{i,x} e_{j,x},\ \mathit{where}\ i = 1, ..., n \tag{2}$$

The execution probability and distance weight are described in the followings.

(1) Execution probability. Activities in a process model often have different execution probabilities in dependence on control-flow patterns in the model. Let us see the examples in Figure 1. The activities in (a)*Sequence* and (b)*AND* split and join of the figure must be executed only once unless the process is cancelled in the enactment. Note that just as in *Sequence* pattern, activities in *AND* split and join are necessarily


1
2
3
4
5
6
7
8
9
$e_{i,x} = 1$
$e_{i,x} = 1$
$e_{i,x} = \Pr(a_{i,x}) = \prod_t \frac{1}{s_t}$
$e_{i,x} = \Pr(a_{i,x}) = \prod_t \frac{1}{2}$

(a) Sequence

(b) AND Split and Join

(c) XOR Split and Join

(d) OR Split and Join

Figure 1. Execution probability of activities in each control-flow patterns

executed once although the execution timing is synchronously. Therefore, the execution probabilities of activities in the two patterns are always 1.

On the other hand, the certain execution cannot be guaranteed to activities between XOR split and join or those between OR split and join as shown in Figure 1 (c) and (d). If we do not have the knowledge of execution rates in historical process log, we can assume that the branches in XOR split have the same probabilities, i.e. $Pr(a) = 1/s$ (where, s is the number of branches). In addition, in the same assumption, the branches in OR split have only two cases, doing or not. Therefore, the branches in OR split can be expected to have the probability of 0.5. However, if we can obtain the experimental probabilities of the activities, the data can be surely adopted in the probabilities of the activities.

(2) Distance weight. Distance weight was devised to express the delicate precedence among activities. Since business process is a collection of activities and their dependencies, the order or precedence among activities must be critical structure. Unfortunately, normal transitions are not enough to compare such precedence between two processes since they are only expressing the adjacent dependencies.

The motivation of distance weight can be easily comprehended with the comparison among four similar processes without any common transition shown in Figure 2. Each process is modified into the next by simple change such as activity insertion and replacement, and parallelism. But, the pairs do not share any transition with each other although activities 1 and 3 have strong precedence. Especially, P_4 has two similar transitions to P_3, but they are constrained transitions with XOR split and join.

The simple example shows the limitation of transition in expressing the precedences of activities and the structures. For the reason, we adopt implicit transitions of considering complete precedences among all activities. Implicit transitions are additional arcs which are created between two activities if the two without a normal transition have explicit precedence in a process model. For instance, activities 1 and 3 in P_3 and P_4 are connected with implicit transition.

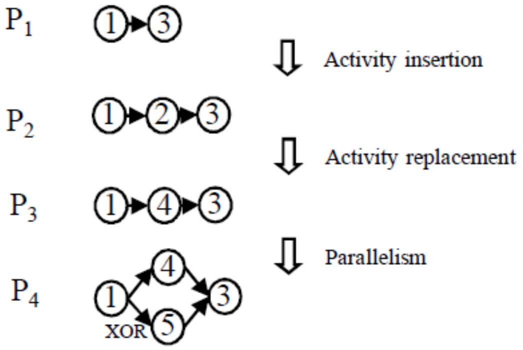


FIGURE 2. Four similar processes without a common arc

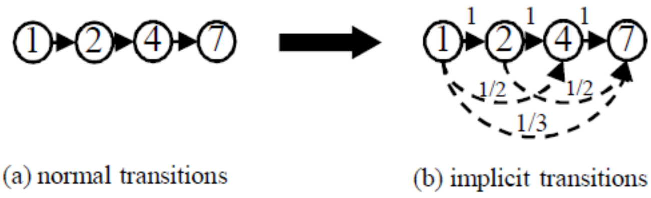
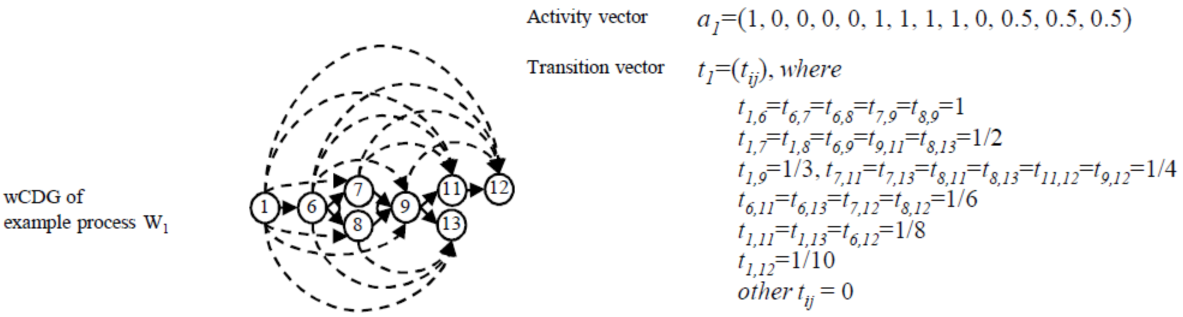


FIGURE 3. Transformation to wCDG



Activity vector a_1=(1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0.5, 0.5, 0.5)

Transition vector $t_1=(t_{ij})$, *where*

$t_{1,6}=t_{6,7}=t_{6,8}=t_{7,9}=t_{8,9}=1$
$t_{1,7}=t_{1,8}=t_{6,9}=t_{9,11}=t_{8,13}=1/2$
$t_{1,9}=1/3$, $t_{7,11}=t_{7,13}=t_{8,11}=t_{8,13}=t_{11,12}=t_{9,12}=1/4$
$t_{6,11}=t_{6,13}=t_{7,12}=t_{8,12}=1/6$
$t_{1,11}=t_{1,13}=t_{6,12}=1/8$
$t_{1,12}=1/10$
other $t_{ij}=0$

FIGURE 4. Example of process vectors

Furthermore, the process model is transformed into weighted Complete Dependency Graph (wCDG), which does not only contain normal transitions, but also implicit transitions with distance weights. The distance weight has the inverse value of distance between two activities. While the distance weight of every normal transition is 1, while that of implicit transition has the value between 1 and 0. Figure 3 shows an example of transformation to wCDG. The dotted arcs are the added implicit transitions. The distance weight can be used as a means of expressing complete precedence among activities in comparing two process models.

In summary, let us see an example of activity and transition vectors of the process model in Figure 4. The activity vector of W_1 is a_1=(1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0.5, 0.5, 0.5). Note that activities a_{11}, a_{12}, and a_{13} have the execution probabilities of 0.5 because they are following the XOR split of fan-out=2. And, the transition vector of W_1 is $t_1=(t_{i,j})$ where $t_{1,6} = t_{6,7} = t_{6,8} = t_{7,9} = t_{8,9} = 1$, $t_{1,7} = t_{1,8} = t_{6,9} = t_{9,11} = t_{8,13} =$

$1/2$, $t_{1,9} = 1/3$, $t_{7,11} = t_{7,13} = t_{8,11} = t_{8,13} = t_{11,12} = t_{9,12} = 1/4$, $t_{6,11} = t_{6,13} = t_{7,12} = t_{8,12} = 1/6$, $t_{1,11} = t_{1,13} = t_{6,12} = 1/8$, $t_{1,12} = 1/10$, and the other $t_{i,j}$'s are zero. For example, $t_{1,12} = 1/10$ is the multiplication of two execution probabilities ($e_1 = 1$ and $e_{10} = 1/2$) and the distance weight $1/d_{1,10} = 1/5$.

3.2. **Similarity measure.** In our research, Cosine coefficient is adopted to measure the similarity between two process models. We have two types of process models: activity and transition vectors. Cosine coefficient quantifies higher value as two vectors have more common elements with higher values. In detail, if the activity vectors of two processes share more elements near 1.0, Cosine coefficient of the activity vectors has higher value. In the same way, if the transition vectors of two contain closer precedences between two shared activities, Cosine coefficient of the transition vectors has higher value. The similarity measures of activity and transition vectors are defined as following formulas.

$$sim_{act}(P_x, P_y) = \frac{a_x \cdot a_y}{|a_x||a_y|} = \frac{\Sigma a_{i,x} a_{i,y}}{\sqrt{\Sigma a_{i,x}^2 \Sigma a_{i,y}^2}} \tag{3}$$

$$sim_{trans}(P_x, P_y) = \frac{t_x \cdot t_y}{|t_x||t_y|} = \frac{\Sigma t_{i,x} t_{i,y}}{\sqrt{\Sigma t_{i,x}^2 \Sigma t_{i,y}^2}} \tag{4}$$

In summary, activity vectors can be used to compare how common activities two models contain, while transition vectors can be done to compare how similar flows they follow. Hereby, total similarity of two process models can be obtained by weighted sum of Cosine coefficients of activity and transition vectors as the following equation. • is a blending factor of two similarities

$$sim_(P_x, P_y) = \alpha sim_{act}(P_x, P_y) + (1 - \alpha) sim_{trans}(P_x, P_y) \tag{5}$$

4. **Business Process Clustering.** Business process clustering is composed of the following steps. First, business process models in a process repository are transformed to vector models based on their activity and transitions. Second, the similarity values among the models are calculated by Cosine measure. Finally, the process models are grouped by using agglomerative hierarchical clustering algorithm [19]. Figure 5 shows the pseudo code of the algorithm for business process clustering.

The algorithm is started with a business process repository $R = (A^{\diamondsuit}, W)$, where $A^{\diamondsuit}$ is an activity space, and W is a set of business processes represented in a tuple $W = \langle A, T, Split, Join \rangle$. In the tuple, A is a set of the activities which were contained in the business process w (i.e. $A \subset A^{\diamondsuit}$ and $w \in W$), and T is a set of the transitions among a's in w. Besides, $Split$ and $Join$ are the constraint functions of T, representing the control flows of split and join, respectively. The two functions are called before accomplishing the core clustering of business processes.

BPClusteirng algorithm includes two functions: *create_process_vectors* function generates process vector set V^R, and *vector_similarity* function measure the similarity among the vectors, $S(V_f^R)$. The two functions are called before accomplishing the core clustering of business processes.

At the beginning, the algorithm regards all processes as initial clusters. In other words, it creates the $|W|$ clusters where each process is mapped to a cluster (Lines 4 to 5). Next, the algorithm repeats the procedure of clustering the clusters until the number of clusters decreases to the given number k (Lines 6 to 16). After finding the most similar pairs of clusters (Line 7), it merges the two clusters c_u and c_v to c_m (Line 8), and updates the

```
**Input**: A process repository $R = (A^\diamond, W)$, the number of clusters $k$, a blending factor $\alpha$.
**Output**: A clustering result $C^R = \{(C, M(W), S(C))|$ result clusters $C$, the membership of processes $M(W)$, similarity matrix between clusters $S(C)\}$.
**Algorithm** *BPClustering* (**in** $(R, k, \alpha)$, **out** $C$)
1:  $V^R := create\_process\_vectors(R)$;
2:  $S(V^R) := vector\_similarity(V^R, \alpha)$;
3:  create $|W|$ clusters;     //make initial clusters
4:  **for** each $w_i \in W$ **do** $M(w_i) = c_i$;
5:  $S(C) := S(V^R)$;
6:  **while** $|C| > k$ **do**      //agglomerative clustering
7:      $(c_u, c_v) = find\_nearest\_pair(S(C))$;
8:      $delete(c_u, c_v, C); add(c_m, C)$;     //update clusters
9:      **for** each $w \in W$ **do**                 //update the membership
10:         **if** $M(w) == c_u$ or $M(w) == c_v$ **then** $M(w) = c_m$;
11:     **end for**
12:     **for** each $c_i \in C(i \neq u, v)$ **do**    //update similarity matrix of the new cluster
13:         $sim(c_m, c_i) = \frac{|c_u|sim(c_u,c_i)+|c_v|sim(c_v,c_i)}{c_u+c_v}$;
14:         $S(C) \leftarrow sim(c_m, c_i)$;
15:     **end for**
16: **end while**
```

```
**Function** *create_process_vectors* (**in** $R$, **out** $V^R$)
**for** each $w \in W$ **do**
    $a_w = 0; t_w = 0$;
    **for** each $a^i \in A^\diamond$ **do**
        **if** $a^i \in A_w$ **then** $a_{i,w} = execution\_rate(w, a^i)$;
    **end for**
    **for** each $a^i, a^j \in A^\diamond$ **do**
        **if** $a^i \in A_w$ and $a^j \in A_w$ **then** $t_{ij,w} = a_{i,w}a_{i,w}distance(w, (i, j))$;
    **end for**
    $A^R \leftarrow a_w; T^R \leftarrow t_w$;
**end for**
**return** $V^R = (A^R, T^R)$;
```

```
**Function** *vector_similarity* (**in** $(V^R, \alpha)$, **out** $S(V^R)$)
**for** each $w_x, w_y \in W$ **do**
    $sim_{act}(w_x, w_y) = \frac{a_x \cdot a_y}{|a_x||a_y|}$;
    $sim_{tran}(w_x, w_y) = \frac{t_x \cdot t_y}{|t_x||t_y|}$;
    $sim(w_x, w_y) = \alpha sim_{act}(w_x, w_y) + (1 - \alpha) sim_{tran}(w_x, w_y)$;
    $S(V^R) \leftarrow sim(w_x, w_y)$;
**end for**
**return** $S(V^R)$;
```

Figure 5. Hierarchical algorithm for business process clustering.

membership of all the processes in c_u and c_v into c_m (Lines 9 to 11). Next, the similarity values of the new cluster c_m with the others c_i's are calculated by the following equation (Lines 12 to 15).

$$sim(c_m, c_i) = \frac{|c_u|sim(c_u, c_i) + |c_v|sim(c_v, c_i)}{c_u + c_v}; \tag{6}$$

The k final clusters are achieved by $(N - k)$ time merges.

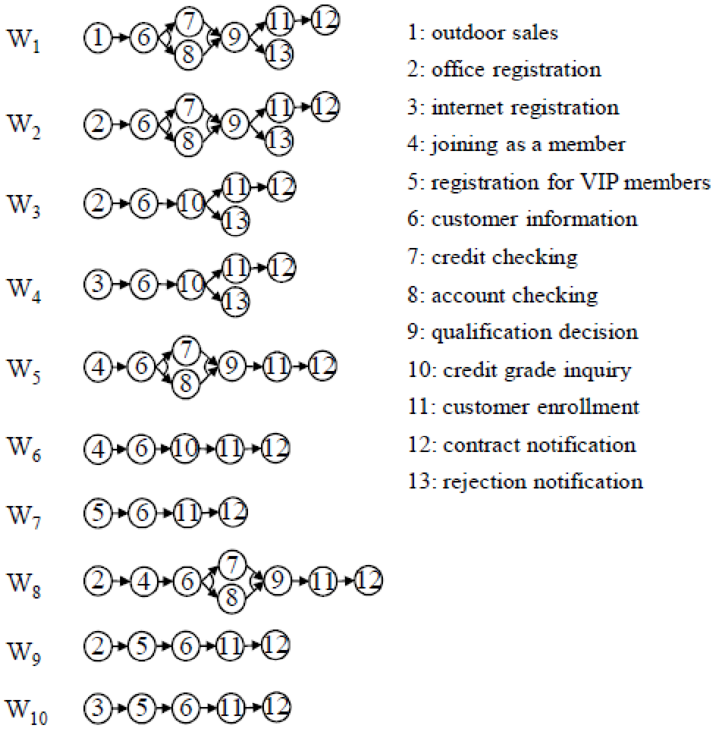


FIGURE 6. Example of insurance processes

The total time complexity of *BPlusteirng* algorithm is $O(n^2N^2)$, where $n = |A^\Diamond|$ and $N = |W|$. The pure clustering algorithm has $O(N^3)$ complexity because it performs the $(N - k)$ times of the merges and each merge needs $N \times N$ comparisons. However, *create_process_vectors* function has $O(nN)$ and $O(n^2N)$ in calculating activity and transition vectors, respectively, and *vector_similarity* function has $O(nN^2)$ and $O(n^2N^2)$ for activity and transition vectors, respectively.

5. **Example.** An example of business process clustering is presented to help readers comprehend the proposed algorithm. Figure 6 illustrates the example of 10 insurance processes W_1 to W_{10}.

W_1 to W_4 start with the registration in outdoor, offices, or web sites, and W_5 to W_7 are for the exiting general or VIP members. Especially, W_8 to W_{10} are for the members who visit offices or web sites. Some processes need only the activity of credit grade inquiry, and others need additional activity for credit assessment request according to the members or the registration places. Besides, the processes for qualified customers do not include the activity of reject notification.

For the process repository, the size of activity space $A^\Diamond$ is $n = 13$, and that of process set W is $N = 10$. First, ten process models are transformed into the pairs of activity and transition vectors. Figure 4 in Section 3.1 showed the example of transforming W1 into two process vectors. The vectors of the other processes have been obtained in the same ways.

The similarity values among the 10 process were calculated on the basis of activity and transition vectors by using adjustment factor $\alpha = 0.5$ (See Table 2). Because the most similar pair of processes are (W_5, W_8), the two are first clustered into C_1 and the similarities between C_1 and W_i's $(i \neq 5, 8)$ are updated by the equation $sim(c_m, c_i) =$

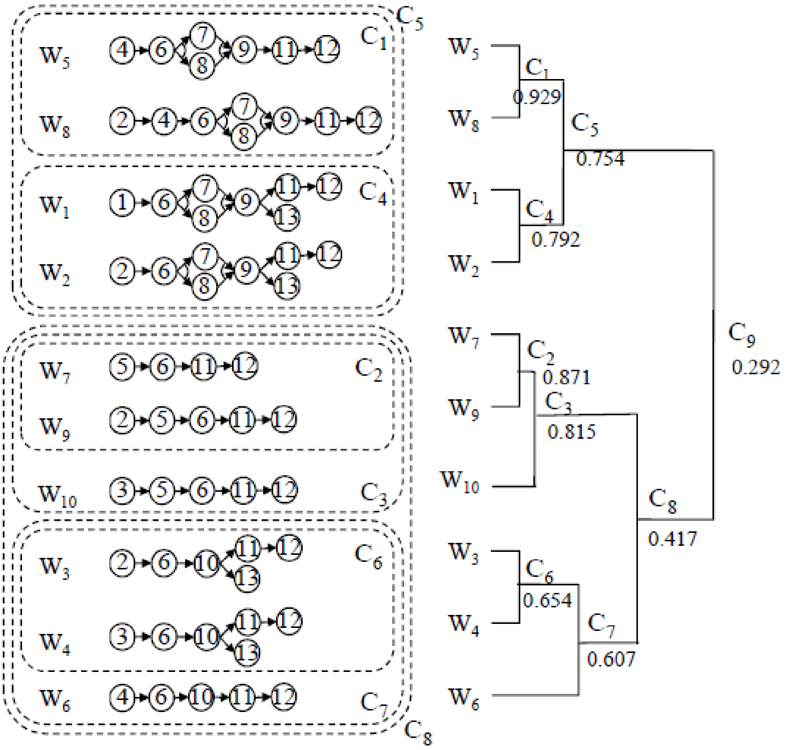


FIGURE 7. Clusters of insurance processes

TABLE 1. Similarities between insurance processes.

	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}
W_1	1.000	0.792	0.207	0.207	0.748	0.219	0.257	0.695	0.227	0.227
W_2		1.000	0.429	0.207	0.748	0.219	0.257	0.825	0.369	0.227
W_3			1.000	0.654	0.230	0.607	0.345	0.354	0.495	0.305
W_4				1.000	0.230	0.607	0.345	0.215	0.305	0.495
W_5					1.000	0.514	0.411	0.929	0.361	0.361
W_6						1.000	0.531	0.478	0.466	0.466
W_7							1.000	0.382	0.871	0.871
W_8								1.000	0.44	0.336
W_9									1.000	0.759
W_{10}										1.000

$\frac{|c_u|sim(c_u,c_i)+|c_v|sim(c_v,c_i)}{c_u+c_v}$. In the same way, the clustering is iterated until $k = 1$. The result of hierarchical clustering was made illustrated in Figure 7. The floating numbers in the dendrogram are the similarity values between two clusters.

Let us look into the result. W_5, W_8, W_1, and W_2 in Cluster C_5 were the processes with parallel block of activities A_6, A_7, A_8, and A_9, and the block represents the credit assessment with credit and account checking. And, W_7, W_9, and W_{10} in Cluster C_3 were the insurance processes for VIP members. Finally, W_3, W_4, and W_6 in Cluster C_7 were the processes which contains serial block of inquiring customer grade with customer information. We can induce the merged models of the process clusters as shown in Figure 8. The merged models are not yet enough integrated to use them as process models.

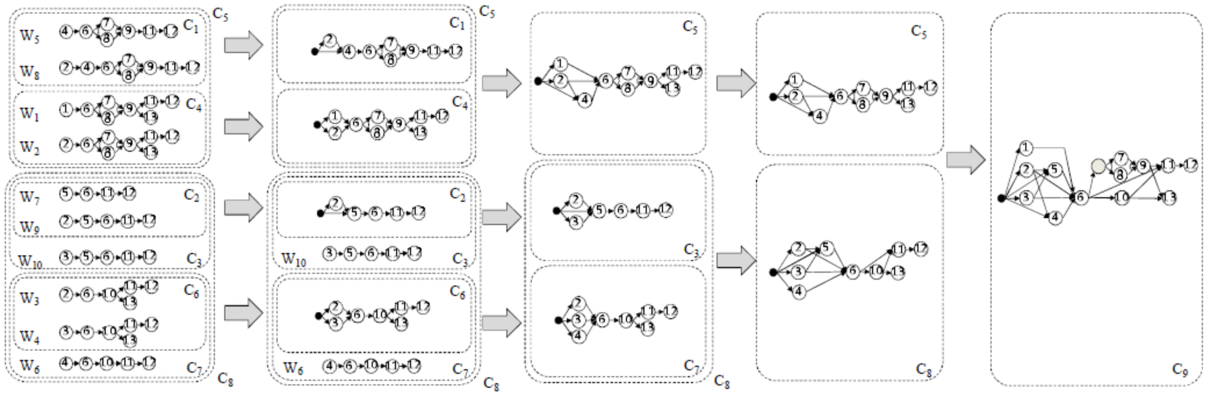


Figure 8. Merged models of insurance processes

They can be augmented by adding the condition to their control-flow after referring to the semantics of original process models. The detailed procedure is omitted since it is beyond this paper.

6. **Conclusions.**

Business processes are considered one of core business assets of raising competitiveness of enterprises, and recently they are standardized and automated in process-aware information systems. Since those systems are continuously accumulating business processes, systematic methods of analyzing and improving the process asset is getting necessary. To address the issue, we proposed a methodology of comparing business process models and discovering the similar processes.

In our research, business process models are first transformed to vector models based on their structures such as activities and transitions, and the vectors are compared by Cosine similarity measure. Finally, the models are clustered by the agglomerative hierarchical clustering algorithm. As illustrated the example processes in Section 5, structurally similar processes are clustered in the same cluster step by step. The results of clustering can be utilized to reengineer the process models or support new process design by extracting their common patterns and structures.

As process-aware information systems spread over companies, the analysis of business process models which were being accumulated in the systems is getting more significant. A variety researches on analysis and reuse of business processes, such as business process clustering, are expected to be helpful enough for companies to designing new processes and reengineering existing processes.

Acknowledgment. This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2008-331-D00709). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] T. W. Malone, K. Crowston, and G. A. Herman, *Organizing Business Knowledge: The MIT Process Handbook*, The MIT Press, Cambridge, MA. 2003.

[2] *RosettaNet Implementation Framework: Core Specification 2.0*, RosettaNet. http://www.rosettanet.org/.

[3] W. M. P. van der Aalst and T. Basten, Inheritance of business processs: An approach to tackling problems related to change, *Theoretical Computer Science*, vol.270, no.1, pp.125-203, 2002.

[4] W. M. P. van der Aalst and A. J. M. M. Weijters, Process mining: A research agenda, *Computers in Industry*, vol.53, no.3, pp.231-244, 2004.

[5] G. Schimm, Mining exact models of concurrent workflows, *Computers in Industry*, vol.53, no.3, pp.265-281, 2004.

[6] M. H. Jansen-Vullers, W. M. P. van der Aalst, and M. Rosemann, Mining configurable enterprise information systems, *Data and Knowledge Engineering*, vol.56, no.3, pp.195-244, 2006.

[7] A. K. A. de Medeiros, W. M. P. van der Aalst, and A. J. M. M. Weijters, Quantifying process equivalence based on observed behavior, *Data and Knowledge Engineering*, vol.64, no.1, pp.55-74, 2008.

[8] H. Kim, J.-Y. Jung, and S.-H. Kang, Extensible collaborative process composition using workflow inheritance, *IE Interfaces*, vol.16, pp.49-54, 2003.

[9] J. Bae, B. Kwon, J.-Y. Jung, and S.-H. Kang, Workflow collaboration design using similarity measures among process definitions, *Review of Korean Society for Internet Info.*, vol.6, no.1, pp.52-61, 2005.

[10] J. Jung and J. Bae, Workflow clustering method based on business process similarity, in *Computer Science and Applications*, M. Gavrilova et al. (eds.), Berlin Heidelberg, Springer-Verlag, LNCS 3981, pp.379-389, 2006.

[11] J. Cardoso, How to Measure the control-flow complexity of web processes and workflows, in *Workflow Handbook 2005*, L. Fischer (eds.), WfMC, Lighthouse Point, FL, pp.199-212, 2005.

[12] H. A. Reijers and I. T. P. Vanderfeesten, Cohesion and coupling metrics for workflow process design, in *Business Process Management*, J. Desel, B. Pernici, and M. Weske (eds.), Berlin Heidelberg, Springer-Verlag, LNCS 3080, pp.290-305, 2004.

[13] L. Zhao, L. Qu, and M. Liu, Disruption coordination of closed-loop supply chain network (I) - Analysis and simulations, *International Journal of Innovative Computing, Information and Control*, vol.4, no.11, pp.2955-2964, 2008.

[14] OMG, *Business Process Modeling Notation (BPMN) Specification*, Final Adopted Specification, dtc/06-02-01, Object Management Group, 2006.

[15] J. Mendling and W. M. P. van der Aalst, Formalization and verification of EPCs with OR-joins based on state and context, *Proc. of the 19th Int. Conf. on Advanced Info. Systems Eng.*, pp.439-453, 2007.

[16] W. M. P. van der Aalst and A. ter Hofstede, YAWL: Yet another workflow language, *Information Systems*, vol.30, pp.245-275, 2005.

[17] Workflow Management Coalition, *Workflow Process Definition Interface - XML Process Definition Language Version 2.0*, WFMC-TC-1025, WfMC, 2005.

[18] R. Agarwal and A. P. Sinha, Object-oriented modeling with UML: A study of developers' perceptions, *Communications of ACM*, vol.46, no.9, pp.248-256, 2003.

[19] Y. Kusunoki, M. Inuiguchi, and J. Stefanowski, Rule induction via clustering decision classes, *International Journal of Innovative Computing, Information and Control*, vol.4, no.10, pp.2663-2677, 2008.