World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:6, 2009

A Simplified and Effective Algorithm Used to Mine Similar Processes: An Illustrated Example

Min-Hsun Kuo and Yun-Shiow Chen

***Abstract*—**The running logs of a process hold valuable information about its executed activity behavior and generated activity logic structure. Theses informative logs can be extracted, analyzed and utilized to improve the efficiencies of the process's execution and conduction. One of the techniques used to accomplish the process improvement is called as process mining. To mine similar processes is such an improvement mission in process mining. Rather than directly mining similar processes using a single comparing coefficient or a complicate fitness function, this paper presents a simplified heuristic process mining algorithm with two similarity comparisons that are able to relatively conform the activity logic sequences (traces) of mining processes with those of a normalized (regularized) one. The relative process conformance is to find which of the mining processes match the required activity sequences and relationships, further for necessary and sufficient applications of the mined processes to process improvements. One similarity presented is defined by the relationships in terms of the number of similar activity sequences existing in different processes; another similarity expresses the degree of the similar (identical) activity sequences among the conforming processes. Since these two similarities are with respect to certain typical behavior (activity sequences) occurred in an entire process, the common problems, such as the inappropriateness of an absolute comparison and the incapability of an intrinsic information elicitation, which are often appeared in other process conforming techniques, can be solved by the relative process comparison presented in this paper. To demonstrate the potentiality of the proposed algorithm, a numerical example is illustrated.

***Keywords*—**process mining, process similarity, artificial intelligence, process conformance.

I. Problem Background

NORMALLY, there requires a number of years taken to train a novice to become a veteran; thus, how to shorten working training time becomes a paramount issue for a business running. Trying to preserve experienced knowledge about the business conduction behavior into a knowledge base may be one of the ways to achieve the goal of shortening training time. The experienced works, service and design processes are such knowledge deserving of being saved into the informative knowledge base of an enterprise. Besides, the success of reengineering a process and eliciting required knowledge hidden in the working process also contributes to enrich the knowledge base and to advance the quality of the working processes through extracting the knowledge base enriched. In order to achieve these destinations, process mining techniques may be a best instrument [1].

Process mining is an emerged technique used to check the conformance of processes or to discover a new process from their historical logs in the form of distinct occurred activities (events). This conformance or discovery enables an enterprise to learn from the mined processes for making the best business conduction decisions. However, if activities are distributed or unstructured at different positions in a process, there is much more critical to correctly track their executed traces (activity sequences) for mining the best process out. An appropriate process mining whose mechanism must function with the accurate logged traces tracking [2].

This paper presents a simplified and effective process conforming algorithm, in which two similarity coefficients of the relative processes are designed to perform the task of tracking traces in logs of the processes, and shows its potentiality through the illustration of an example of process mining. The proposed algorithm uses the activity sequences (traces) of processes as an assistance of gathering structural activity logs and further to define the two similarities related to the processes being mined. The first similarity defines the relationship between processes; the second expresses the degree of the identical (or similar) relationships among the mining processes. The activity sequences can be used to draw a parallel between various processes for finding the deviations from a certain required (experienced) process. How similar are the two or more processes in terms of activity sequences? 30%, 80% or 100%, is the major goal to be investigated in this paper.

Traditionally, the bi-similarity or activity trace equivalence was used to parallel a normalized process with other processes [3]. Its yes/no binary solution (i.e., two processes are completely same or not) to a process mining problem, however, seems not to be plausible in real applications because (1) different process configurations may not allude to different activity sequences, and (2) the same set of activity sequences generated by different processes should not be concluded only by counting the frequency of activity sequences. Thereby, conforming or comparing processes should be performed in accordance of the process's gradualness in terms of devious activity (sub)sequences.

In this paper, the aforementioned two process similarity coefficients are formulated to get the answers to the above two problems by respectively considering the following two aspects: (1) how similar is the gradualness of the mining processes to the

Min-Hsun Kuo is a graduate student with the Industrial Engineering and Management Department, Yuan Ze University, Chung Li, Taiwan.

Yun-Shiow Chen, Ph.D., is a professor with the Industrial Engineering and Management Department, Yuan Ze University, Chung Li, Taiwan. (e-mail: ieyschen@saturn.yzu.edu.tw).

experienced one, and (2) how much of the gradualness in the event log can be generated by the mined process.

The rest of this paper is organized as follows. Section 2 introduces the characteristics of the process mining. Section 3 reviews the process mining literatures where the main idea of this study was inspired from. Section 4 proposes the simplified process mining algorithm in detail. Section 5 explains the numerical results by illustrating an example published elsewhere. Finally, Section 6 concludes the findings of the proposed algorithm.

II. Introduction to Process Mining and Petri Net

Process mining functions as the supervision, discovery, evaluation, correction and improvement of real working processes. It can find out whether, which, what, where, and how a process's activities are distinct or deviatory from those events existed or executed in the normalized process by means of eliciting knowledge from the activity logs of the processes. Thus, it is particularly interested in a situation where the activities of a process should be self-organized and differentiated prominent behavior from other processes. The more situations where people, matters, time, place, equipment, and other activities information can be deviated or emerged, the more curious demeanors hidden in the processes should be looked into and found out.

By eliciting, comparing, or adapting a number of real experienced process execution logs stored in the knowledge management (KM) system of an enterprise, a process mining technique can be utilized to find or to work out new business process problems. There are several such business KM systems advocated and sold in the commercial market as BPM (Business Process Management), WfM (Workflow Management), EC (Electronic Commerce) systems, ERP (Enterprise Requirements Planning), new product development (NPD), software design improvement, object configuration analysis, CRM (Customers Retention Management) systems, MfM (Manufacturing Flowcharts Management), and suchlike. These KM systems are able to save lots of real working experiences, empirical data, rules and activity treatment steps, and then are applied to (re)structure or (re)configure business conduction processes on the basis of meeting the requirements of the enough executive performance [4]. Conceiving and modeling such a KM system, however, is a sophisticate task with time consuming and excessive workload, for all of different process mining approaches existed.

In process mining, an activity can be defined as its related people, matters, time, place, equipment, exception, transaction, temporal and spatial data, working type, numerical data, descriptive words, rules and semantics, and suchlike. An activity sequence can be seen as a trace occurred in a process. There are a number of traces occurred in a process. A family of traces comprises the log of a process. One of the major challenges to collect data records in the prerequisite for mining process is to relate activities to become traces so as further to correspondingly correlate the traces among different processes. Generally, all process mining techniques exploit at least one log of activity traces as their input data and as an initiative step to engage in the tasks of the mining processes. The requirements for the log creation should be in accordance with the process mining algorithm exploited. The algorithm also should be deliberated and selected due to the fact that different algorithms frequently lead to achieve various goals of mining processes. Three kinds of the goal are described as follows [4].

The first kind of the goal is to mine the conformance between processes. In process conformance, a known normalized process model is needed and used as a reference model for checking whether the activity logs of other mining processes correctly meet its normative requirements and specifications. In other words, the conformance comparison aims at the discovery of a process model based on required event log. During the conforming comparison, the inspections of the variations or deviations between the comparing process models are performed, the locations of the divergences are indicated, the reasons of the differentiations are explained, and meanwhile the degrees of those deviations are evaluated.

The second sort of the goal is to discover a new process through a self-organized mining process. The self-organization does not need a known normalized process model but needs the experienced records and empirical data of an unknown process to be fumbled and embodied through the function of rearranging and reorganizing activity records and data. The self-organization mining algorithm can create and structure a new process model according to the information gathered from the historical event records of an unstructured process.

The third kind of the goal is to mine the extension of a process. Like conformation mining, a normalized process model is required to be known in advance; afterwards, certain new information or perspectives can be added to it for the fertilization of its logging contents; in other words, the goal of extending process model is not to do some comparison and conformation with other process models, but to let more normative or normalized activity information get into an existing process for the enhancement of the further process executions, business conductions, and possible real applications. A process model can be more mushroomed by dynamically irrigating and adapting its logging data.

To get to the above mining goals, certain approaches have been proposed and will be summarized in the next section, Literature Review. While, it seems that almost all of the approaches are originated from the ideas of Wil M.P. van der Aalst (http://is.tm.tue.nl/staff/wvdaalst/), a pioneer of process mining, and his leading teams. Petri net, developed by him, is one of the techniques used to model and mine processes, and also is the most popular net that has been used to derive more improved process mining methods.

A Petri net is an oriented bipartite graph constructed with nodes and the directed connections or arcs between them. There are two different typed nodes, place (drawn by a circle) and transition (drawn by a rectangle with a marked input), and the same typed nodes cannot be linked in the process of drawing a Petri net. The marked input is used to distinguish transitions in different subnets. A place may have a zero or a token (a unique identity, ID) drawn with black dot. The number of tokens may change during the execution of the Petri net. A circle place functions as an input to a rectangle transition if there draws a directed arc from the place to the transition; on the other hand, a place is deemed as an output from a transition if a directed arc pulls the transition towards the place [5] [6].

A process can be corresponding modeled by a Petri net. Places and transitions are suitable to abstract the activity records and to extract the detailed empirical data stored in the log of the process. Each activity is correspondent to a transition. The corresponding transitions are interconnected via the logical sequential places that are taken to model the flow of the activities. The tokens, each with a unique ID, can stand for different people actions, tooling names, document codes, working signals, and such elements of a process. Accordingly, the priority and the posterity of activities are orientated and how a process works can be shown in the Petri net crafted [6].

The ontology skills may be applied to map the correspondences between a process and a Petri net, so that the process's situations can be constructed and mined on the basis of the operations of the Petri net. Van Dongen, et al. [7] described a nested Petri net could be crafted; namely, a complicate process can be divided into several simpler sub-processes so that the complication can be hierarchically observed and be clearly solvable subnet by subnet.

In [2], four kinds of major process mining method were surveyed:

A. Abstraction-Based Algorithm

This kind of algorithms constructs a Petri-like net (structured workflow Petri net) based on an abstraction of the log and uses an α-algorithm to mine the log. This α mining algorithm keeps the characteristic that if the log is a complete one generated by a model belonging to the given class, then that model can be rediscovered. Three sequential running phases are composed of such kind of algorithms: abstraction, induction and construction [8]. The activity records in a process log are firstly abstracted to become the represent able schema based on real various workflows. The logic of the workflows is the clues of the induction used to deploy the Petri-like net structure. The construction of the net is completed if the process inductions meet the real workflow requirements, and the constructed net model can be a useful representative model to support the conductions of the process.

B. Heuristic-Based Algorithm.

The α mining algorithm cannot deal with the situation of the presence of noises, which they are the exceptional traces existed in an incorrect or incomplete log. So, a pure abstraction-based algorithm might not construct an applicable process model if some unclear data are involved in the log. Mruster, et al. [9] developed an empirical method that constructs the process model consisted of the activity relations in the process log containing noise and imbalance data. To predict the activity relationships, two rule sets were induced from the process logs: one is used to detect all causal relations. After the causal relations are found, another rule set detects whether the exclusive/parallel activity relations share the same cause or the same direct successor. By remedying the causal and exclusive/parallel relations, the noise and imbalance data can be removed, a process model can be built as a corresponding Petri net, and an α algorithm can be used to infer the Petri net that explains the data.

C. Meta-Heuristic-Based Algorithm.

Although the use of heuristics has led to the development of algorithms that are more robust to noise, the algorithms described above, however, are still much dependent on local information in activity log. In other words, the heuristic way of inducing the oriented relations is pretty much based on the precedence and succession of activity orientations themselves. Besides, there few abstraction or heuristic methods are able to simultaneously handle all common process constructs and noise involved. To tackle all these issues, other kinds of algorithms functioning with the global search for the process log should be developed. Genetic algorithms (GAs) are a sort of such algorithms [10].

GAs are adaptive search methods that mimic the process of cell evolution. They start with an initial population of individuals. Every individual is assigned a fitness function used to measure its quality of matching the requirements of a process. In the case of process mining, an individual is a possible process model and the fitness is a function that evaluates how well an individual is able to reproduce the behavior in the log. Since GAs are developed to benefit discovery of process models that capture the most frequent behavior in a log, they work better when the traces in a log are grouped based on some similarity criteria between elements of the log [10].

D. Semantics-Based Algorithm.

Although the aforementioned process mining techniques provide feedback about different perspectives of process models to reuse the log data for the conformance or discovery of processes, the degree of their automation and reuse is somewhat limited because it is based on activities in the form of data strings which lack the abstraction level required for the process analysts, thus it is hard to express business operations by means of the encoded data strings. Consequently, the process mining algorithm based on the semantic relationships between activities has been proposed to enhance the expression of the data string coding [11].

Generally, a semantic process analysis environment has three steps: (1) the creation of ontology that capture the meanings of different data and records in process models, (2) the semantic annotation of processes with the defined ontology, and (3) the definition of semantic versions of executable process mining techniques. Dang, et al., [7] presented such a semantic ontology method as these three steps for a practical medical workflows application.

For an overview of process mining techniques, readers are suggested to refer [2] [4] [12]. In what follows, certain published papers specified with particular process mining methods or related this paper will be reviewed.

III. Literature Review

Wen, et al., [12] proposed a new process mining technique called β-algorithm involving the information of both start and complete events. The β-algorithm was utilized to compare the process parallelism and able to conquer the inability of using α-algorithm to mine short loops existing in a process. DWF-net (direct workflow net) modified from Petri-net was newly proposed and its theoretical foundation was also developed for supporting the results of numerical examples illustrated in their paper.

Another net called NDT-WF-Net extended with the addition of records of model running time to the transition status of activities in Petri net was newly proposed by [13]. For each timed activity in a process model, there two kinds of time function were used to respectively define the minimum and maximum duration of each activity. If a temporal activity contains the so-called activity tokens in their NDT-WF-Net, its corresponding activity in other process models is being mined. Based on the mined NDT-WF-Net, the earliest time and the latest time to start each activity can be evaluated and recorded. Also, the latest time can be determined by the earliest time of each activity so as to get the resources ready for the activity as soon as possible. In their paper, mathematical certifications were derived to support the NDT-WF-Net mining operations that can discover the spatial (structural) with temporal information from the timed running logs of the process.

Ho, et al. [14] applied process mining technique to aid a supply chain management (SCM) so that the SCM system can function with shorter delivery time, more flexible inventory and higher customer satisfaction with the learning ability of a back-propagation neural network. The system they proposed is named CDPMS (cooperative distributed process mining system), which is composed of the so-called process mining engine (PME) and dynamic rule refinement engine (DR^{2}M). Each PME has the following three modules: process monitoring using OLAP (online analytical processing), quality prediction using ANN (artificial neural network), and decision support system implemented with fuzzy rules to select corrective actions for continual quality improvements. DR^{2}M involves two main modules, online process mining module and rule refinement module. The former is responsible for analyzing the process data imported from the PMEs through the XML (extensible markup language) translator, and the latter for tracking and modifying the fired rules in each PME which lead to poor quality. Successful operations of CDPMS depend on the effective XML communication for collaborating, coordinating and sharing information over intranets, extranets and the internet.

Since a business piles up lots of policy documents used to perform its conduction processes, and the policies must be frequently reviewed to ensure their correctness and consistency. Li, et al. [15] developed an automatic policy document management system named PBPM (policy-based process mining) to find the process variation and to progress the policy execution. A parse tree is applied to mine the contexts of the policy documents by identifying the process components (i.e., activities in terms of texts) so that PBPM can automatically extract process models from narrative business policies, thus to reduce the human's recognition load of processes. Another property of PBPM is its process mining mechanism based on the ability of leveraging unstructured policies for process discovery. This characteristic is not like that of tracing structured event logs in typical process mining techniques.

A serious of process mining techniques based on genetic algorithms (GAs) has been proposed by [3] [10] [16] [17]. Their genetic mining technique can answer to the question of what the process picturing the current status looks like, but in each of their papers addressed, only genetic fitness formulation was considered. This individual consideration reduces the suitability of the process conforming procedure involving a large of activity traces. The reason why the reduction may occur could be that the genetic fitness between the event log and the process requirements may not be properly defined to deal with the loops with long repeated trace checking.

This paper builds on the preliminary work reported in [3] [16] [17]. The genetic fitness function has been modified to become two simpler similarity functions to overcome the lengthy comparisons which may cause to incorrectly parallel activity sequences between different mining process models, and the GA algorithm has also been simplified to an effective heuristic one.

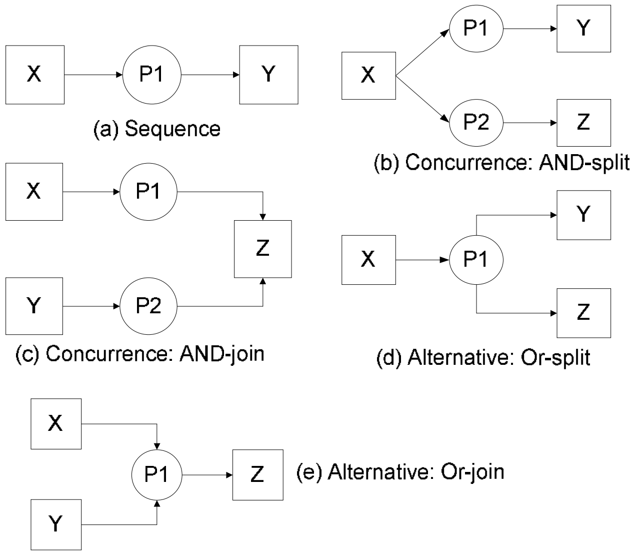
IV. The Proposed Processing Mining Algorithm

As aforementioned in Section 1, most process mining techniques can only perform the comparison of activity sequences between two processes that are the same or not, a sort of bi-similarity comparison, and meanwhile the activity relationships such as sequence, concurrence, alternative and loop are little discussed. Besides, if the sequences of activities that occur relatively close to each other in a given partial order also do not be taken into account of the similarity conformance, the different number of the activity relationships between two processes may cause the difficulty in defining the process similarity. To solve these problems, this section presents an algorithm to conform the deviations between mining models by computing the level of similarity among different processes, counting the frequency of sequential activity occurrences, and analyzing the similar percentage of the partial activity sequences in different process models.

During the proposed iterative process conformance comparisons, with the consideration of counting frequently occurring activities in a sequence, once such activity sequences are known, they can be applied to describe or program the behavior of the process. This section addresses an iterative heuristic approach with firstly using two rather simple activity relationship metrics to initiate the proposed process conformance, which they are so-called Transitions-Places table (T-P table) and Places-Transitions table (P-T table) and are tabulated from the logic activity directions depicted in the Petri nets. The T-P table records the connection from transitions to places, while the P-T table shows the reverse directions. From these two tables, the relationship of activities can be seen and then to induce the activity flows of the process in the form of Petri nets.

Figure 1 illustrates the elemental five types of sequential activity relationships in a Petri net; they are sequence, concurrences (AND-split and AND-join) and alternatives (OR-split and OR-join). Their corresponding T-P table and the

P-T table are shown in Table 1 and Table 2, respectively.



(a) Sequence

(b) Concurrence: AND-split

(c) Concurrence: AND-join

(d) Alternative: Or-split

(e) Alternative: Or-join

Fi.g.1 Five elemental activity relationships in the form of Petri net

TABLE I
T-P DIRECTION FOR THE FIVE ACTIVITY RELATIONSHIPS

From\To	(a)		(b)		(c)		(d)		(e)	
	P_1	P_2	P_1	P_2	P_1	P_2	P_1	P_2	P_1	P_2
X	1	0	1	1	1	0	1	0	1	0
Y	0	0	0	0	0	0	0	1	1	0
Z	0	0	0	0	0	0	0	0	0	0

TABLE II
T-P DIRECTION FOR THE FIVE ACTIVITY RELATIONSHIPS

	From\To	X	Y	Z
(a)	P_1	1	0	0
	P_2	0	0	0
(b)	P_1	0	1	0
	P_2	0	0	1
(c)	P_1	0	1	1
	P_2	0	0	0
(d)	P_1	0	0	1
	P_2	0	0	1
(e)	P_1	0	0	1
	P_2	0	0	0

The proposed iterative heuristic process conformance algorithm is distinguished with the following four steps:

Step 1. Tabulating the T-P tables and P-T tables on the basis of Petri net drawn from the process log. The contents of the T-P table and the P-T table are defined by the numbers of both sequential activity occurrences and places in a process model and the activity relationships extracted out of the T-P and the P-T tables. Thereby, the two similarity conformance coefficients can be derived.

Step 2. Modifying the two kinds of tables established in Step 1 for refining the relationships between process activities. The relationship refinement is treated until the activity direction logic meets the requirements of operations of Petri net. Operationally, the sequence-typed net is like a concurrence (two sequences in parallel) one, thus the mathematical operations of these three types of the elemental nets are similar. The sequence net, on the other hand, can be a partial sequence of the alternative net; therefore, the alternative nets can be adjusted to become the sequence ones. This adjustment can make the counting of the activity relationships be more consistent and time saving.

Step 3. Building a new table in which all relationships between the comparing process models are integrated. The integrated table functions as two operations, Modeling and Comparison. If a directional activity relationship is existed in a process model, the entry located at the intersection of the integrated table is filled with 1; otherwise, zero. The Modeling function whose values are written in the last row of the integrated table (representing by "Count," in Table 5 of the next section) records the frequency of the sequential activity occurrences in each process. The Comparison function, on the other hand, shows the directional relationships of process models and the difference between the two processes. This difference is indicated with 0 or 1. A slash in the table is used to discriminate the meaning of the relationships and the process identity (ID). If the corresponding relationship can be found in none of the comparing process models, the right side of the slash is written with 0; otherwise, 1 if the same relationships can be determined in both process models. The left side of the slash represents the ID of the process, and also is filled with 0 or 1. As any difference between the two models is existed, the code of model ID will be 1; otherwise, 0.

Step 4. Computing the two similarity coefficients for the process conformance comparisons. Equations (1) and (2) as following represent the two similarity equations. Equation (1) is to count the numbers of sequential activities and relationships. Selecting a process model as a normalized one used to conform whether the other comparing processes meet the requirements normalized by the selected process, and equation (2) is used to evaluate the percentages of the similar (identical) activity (sub)sequences between two processes. The second similarity is treated as an assistance of the first similarity so as to understand how likeness between two processes is. The usage of the frequencies of sequential activity occurrences is to conform whether they are consistent to the resulting Petri net.

$$D_{ab} = \frac{A_{a\cap b}}{A_{a\cup b}} \times \frac{N_{a\cap b}}{N_{a\cup b}} \tag{1}$$

$$C_{ab} = \frac{N_{a \cap b}}{N_a} \tag{2}$$

where

D_{ab}: similarity of process models *a* and *b*.

C_{ab}: percentages of the similar (identical) relationships of processes *a* and *b* with respect to process a.

$A_{a \cap b}$: number of the total similar (identical) activities of processes *a* and *b*.

$A_{a \cup b}$: number of the activities of processes *a* and *b*.

$N_{a \cap b}$: number of the total similar (identical) relationships of processes *a* and *b*.

$N_{a \cup b}$: number of the relationships of processes *a* and *b*.

N_a: number of the relationships of process *a*.

V. Findings of the Algorithm Implementation

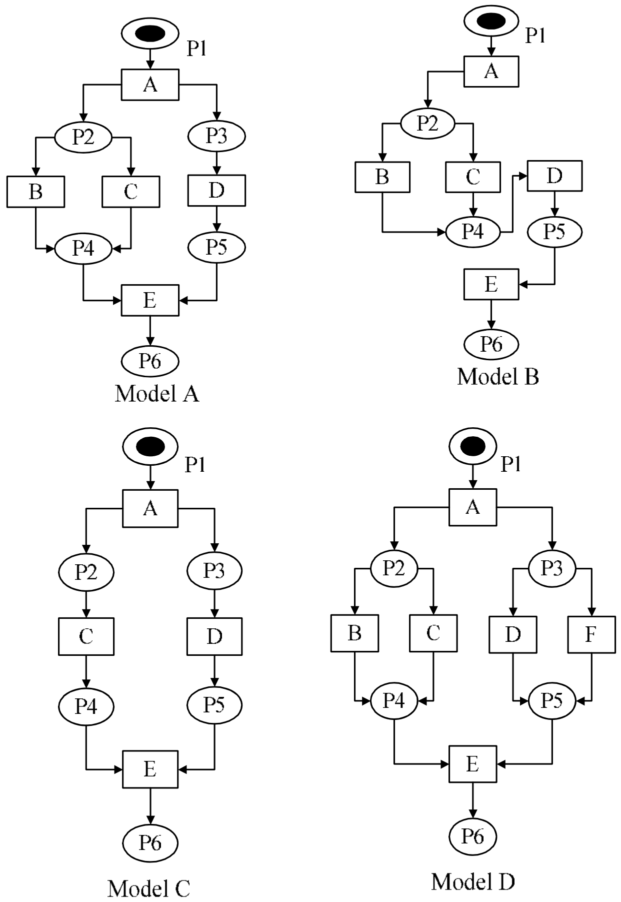


Fig 2. An example of illustrating the proposed algorithm [3]

In this section, a published numerical example is illustrated for the demonstration of the proposed algorithm used to perform the conformation of the requirements and specifications existing in processes. Figure 2 shows the example taken from [3]. In the figure, Model A is considered as a normal process used to draw a parallel between the four process models. Their corresponding T-P tables and the P-T tables are transformed and tabulated in Table 3. The relationships of activity sequences of the four process models in the table are the information input to the proposed algorithm. Owing to the limited length of this paper, the detailed mathematical operations of the remaindering three steps are omitted, and will be presented with more computer experimental results in other academic journals. The five tables, Table 4 to Table 8, respectively show the results of the four algorithmic operation steps depicted in the previous section.

TABLE III
(STEP 1) RELATIONSHIPS OF SEQUENTIAL CONSEQUENCES OF THE FOUR PROCESS MODELS

Process Model	Model (A)	Model (B)	Model (C)	Model (D)
Relationship	$Start \rightarrow A$	$Start \rightarrow A$	$Start \rightarrow A$	$Start \rightarrow A$
	$A \rightarrow (\{BC\}\{D\})$	$A \rightarrow \{BC\}$	$A \rightarrow (\{C\}\{D\})$	$A \rightarrow (\{BC\}\{DF\})$
	$(\{BC\}\{D\}) \rightarrow E$	$\{BC\} \rightarrow D$	$(\{C\}\{D\}) \rightarrow E$	$(\{BC\}\{DF\}) \rightarrow E$
	$E \rightarrow End$	$D \rightarrow E$	$E \rightarrow End$	$E \rightarrow End$
		$E \rightarrow End$		

TABLE IV
(STEP 2) RESULT OF REFINING RELATIONSHIPS OF THE ACTIVITY SEQUENCES IN TABLE III

Process Model	Model (A)	Model (B)	Model (C)	Model (D)
Relationship	$Start \rightarrow A$	$Start \rightarrow A$	$Start \rightarrow A$	$Start \rightarrow A$
	$A \rightarrow (\{B\}\{D\})$	$A \rightarrow \{B\}$	$A \rightarrow (\{C\}\{D\})$	$A \rightarrow (\{B\}\{D\})$
	$A \rightarrow (\{C\}\{D\})$	$A \rightarrow \{C\}$	$(\{C\}\{D\}) \rightarrow E$	$A \rightarrow (\{B\}\{F\})$
	$(\{B\}\{D\}) \rightarrow E$	$\{B\} \rightarrow D$	$E \rightarrow End$	$A \rightarrow (\{C\}\{D\})$
	$(\{C\}\{D\}) \rightarrow E$	$\{C\} \rightarrow D$		$A \rightarrow (\{C\}\{F\})$
	$E \rightarrow End$	$D \rightarrow E$		$(\{B\}\{D\}) \rightarrow E$
		$E \rightarrow End$		$(\{B\}\{F\}) \rightarrow E$
				$(\{C\}\{D\}) \rightarrow E$
				$(\{C\}\{F\}) \rightarrow E$
				$E \rightarrow End$

TABLE V
(STEP 3) RESULT OF THE COMPARING PROCESS MODELS BASED ON THE NORMALIZED MODEL A

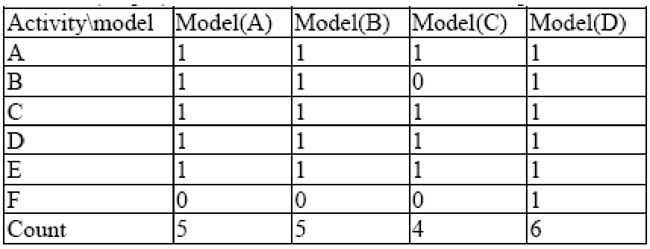
Relationship\ Model	Modeling Function				Comparison Function		
	Model(A)	Model(B)	Model(C)	Model(D)	(A)(B)	(A)(C)	(A)(D)
$Start \rightarrow A$	1	1	1	1	0/1	0/1	0/1
$A \rightarrow (\{B\}\{D\})$	1	0	0	1	A /1	A /1	0/1
$A \rightarrow (\{B\}\{F\})$	0	0	0	1	0/0	0/0	D/1
$A \rightarrow (\{C\}\{D\})$	1	0	1	1	A /1	0/1	0/1
$A \rightarrow (\{C\}\{F\})$	0	0	0	1	0/0	0/0	D/1
$(\{B\}\{D\}) \rightarrow E$	1	0	0	1	A /1	A /1	0/1
$(\{B\}\{F\}) \rightarrow E$	0	0	0	1	0/0	0/0	D/1
$(\{C\}\{D\}) \rightarrow E$	1	0	1	1	A /1	0/1	0/1
$(\{C\}\{F\}) \rightarrow E$	0	0	0	1	0/0	0/0	D/1
$A \rightarrow \{B\}$	0	1	0	0	B/1	0/0	0/0
$A \rightarrow \{C\}$	0	1	0	0	B/1	0/0	0/0
$\{B\} \rightarrow D$	0	1	0	0	B/1	0/0	0/0
$\{C\} \rightarrow D$	0	1	0	0	B/1	0/0	0/0
$D \rightarrow E$	0	1	0	0	B/1	0/0	0/0
$E \rightarrow End$	1	1	1	1	0/1	0/1	0/1
Count	6	7	4	10	2/11	4/6	6/10

From Table 5, the relationships which cause the difference between any two models can be observed. For example, the relationships between Models A and B, $A \rightarrow (\{B\}\{D\})$,

$A \to (\{C\}\{D\})$, $(\{B\}\{D\}) \to E$, $(\{C\}\{D\}) \to E$, $A \to \{B\}$, $A \to \{C\}$, $\{B\} \to D$, $\{C\} \to D$ and $D \to E$, show the two models in difference. The first four relationships are observed in Model A and the remaindering activity sequences are just in Model B. As to the Models A and C, $A \to (\{B\}\{D\})$ and $(\{B\}\{D\}) \to E$ are the two relationships appeared only in Model A. Therefore, it is reasonable to regard Model C as a sub-model of Model A; analogously, Model A can be seen as a sub-model of Model D.

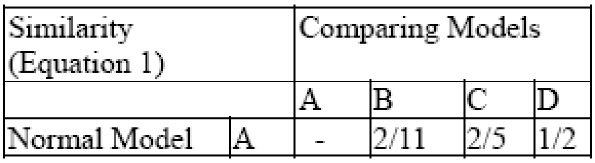
From Table 7, it can be observed that Model A is closer to Model D than to Model C. From reviewing the columns of Table 8, the frequency of the sequential activity occurrences of Model D completely is identical to that of Model A, and three fifths of the sequential occurrences of activities of Model A are the same as that of Model D. Moreover, Model A is identical to Model C as well, but its 50% of the activity sequences cannot be included in Model C. Relatively, Model A and Model D are much similar with each other

TABLE VI
(STEP 3) RESULT OF THE DIRECTIONAL RELATIONSHIPS OBSERVED FROM THE FOUR MODELS



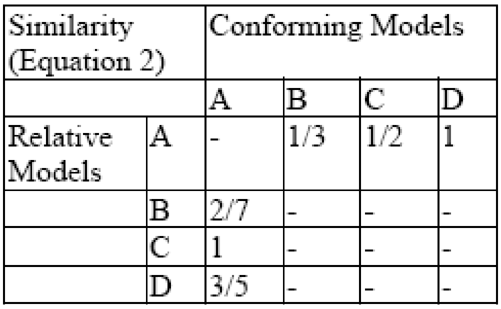
Activity\model	Model(A)	Model(B)	Model(C)	Model(D)
A	1	1	1	1
B	1	1	0	1
C	1	1	1	1
D	1	1	1	1
E	1	1	1	1
F	0	0	0	1
Count	5	5	4	6

TABLE VII
(STEP 4) SIMILARITY BETWEEN THE FOUR PROCESS MODELS ON THE BASIS OF MODEL A



Similarity (Equation 1)		Comparing Models			
		A	B	C	D
Normal Model	A	-	2/11	2/5	1/2

TABLE VIII
(STEP 4) PERCENTAGE OF SIMILARITY ACTIVITY OF THE FOUR PROCESS MODELS



Similarity (Equation 2)		Conforming Models			
		A	B	C	D
Relative Models	A	-	1/3	1/2	1
	B	2/7	-	-	-
	C	1	-	-	-
	D	3/5	-	-	-

VI. Conclusions

To the best of our knowledge, the proposed algorithm is a much simpler and an effective method than those known algorithms (like GA) that might use certain impractical fitness functions to perform the conformation task of process mining. It is the first approach that applies two easily understandable similarity coefficients, defined by the directional sequences of related activities and the relationships between the sequences, to heuristically accomplish the process conformance. Notwithstanding, the proposed algorithm neither considers the time influence on the conformance procedure, nor the detection of the noise appearing in event logs. At this moment, the proposed algorithm is being developed towards discovering process activities from unstructured processes with other factors that may be influential characteristics of process logs. The initial investigated results were presented with only a known example in this paper, but more numerical experiments will be performed by the developing computer program, which is being implemented with Borland C++ language. Also, more practical case studies can be expected to be done, and published elsewhere.

References

[1] van der Aalst, W.M.P., H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, H.M.W. Verbeek, "Business process mining: An industrial application," *Information Systems*, vol.32, no.5, pp.713-732, 2007.

[2] van Dongen, B. F., A. K. Alves de Medeiros and L. Wen, "Overview and Outlook of Petri Net Discovery Algorithms," in *Transactions on Petri Nets and Other Models of Concurrency II*, *Lecture Notes in Computer Science 5460,* pp.225-242, 2009.

[3] van der Aalst, W.M.P., A.K. Alves de Medeiros and A.J.M.M. Weijters, "Process equivalence: comparing two process models based on observed behavior," in *2006 International Conference on Business Process Management, Lecture Notes on Computer Science*, vol.4102, pp.129-144, 2006.

[4] Tiwari, A., C.J. Turner and B. Majeed, "A review of business process mining: state-of-the-art and future trends," *Business Process Management Journal*, vol.14, no.1, pp.5-22, 2008.

[5] van der Aalst, W. M. P. and K. M. van Hee, "Business process redesign: A Petri-net-based approach," *Computers in Industry*, vol.29, no. 1-2 , pp.15-26, 1996.

[6] Salimifard, K. and M. Wright, "Petri net-based modeling of workflow systems: An overview," *European Journal of Operational Research*, vol.134, no.3, pp.664-676, 2001.

[7] Dang, J., A. Hedayati, K. Hampel and C. Toklu, "An ontological knowledge framework for adaptive medical workflow," *Journal of Biomedical Informatics*, vol.41, no.5, pp.829-836, 2008.

[8] Li, Jiafei, Dayou Liu and Bo Yang, "Process mining: extending *α* -Algorithm to mine duplicate tasks in process logs," in *Advances in Web and Network Technologies, and Information Management, Lecture Notes in Computer Science 4537,* pp. 396-407, 2007.

[9] Mruster, L., Weijters, A.J.M.M., Wil M.P. van dre AALST and Antal van den Bosch, "A Rule-Based Approach for Process Discovery: Dealing with Noise and Imbalance in Process Logs," Data Mining and Knowledge Discovery, vol.13, no.1, pp.67-87, 2006.

[10] de Medeiros, A. K. A., A. J. M. M. Weijters and W. M. P. van der Aalst, "Genetic process mining: an experimental evaluation," *Data Mining and Knowledge Discovery*, vol.14, no.2, pp.245-304, 2007.

[11] Dijkman, R.M., M. Dumas, C. Ouyang, "Semantics and analysis of business process models in BPMN," *Information and Software Technology*, vol.50, no.12, pp.1281-1294, 2008.

[12] Wen, Lijie, Jianmin Wang, M. P. W. van der Aalst, Biqing Huang and Jiaguang Sun "A novel approach for process mining based on event types,"

Journal of Intelligent Information Systems, vol.32, no.2, pp.163-190, 2009.

[13] Duan, H., Qingtian Zeng, Huaiqing Wang, Sherry X. Sun and Dongming Xu, "Classification and evaluation of timed running schemas for workflow based on process mining," *Journal of Systems and Software*, vol.82, no.3, 2009, pp.400-410.

[14] Ho, G. T. S., H. C. W. Lau, S. K. Kwok, C. K. M. Lee and W. Ho, "Development of a co-operative distributed process mining system for quality assurance," *International Journal of Production Research*, vol.47, no.4, pp.883-918, 2009.

[15] Li, Jiexun, Harry Jiannan Wang, Zhu Zhang and J. Leon Zhao, "A policy-based process mining framework: mining business policy texts for discovering process models," *Information Systems and E-Business Management*, pp.to be published, 2009.

[16] Rozinat, A. and W.M.P. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Information Systems*, vol.33, no.1, pp.64-95, 2008.

[17] de Medeiros, A.K. Alves, 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.

Min-Hsun Kao a Ph.D. student studying industry engineering at the Yuan Ze university, Taiwan. During her research studies, she participated in several research projects in the areas of process modeling and data mining based on statistical methods. Currently she is working at the department for an instructor to teach engineering statistics and an assistant of statistical quality and experiments laboratory of the department.

Yun-Shiow Chen a full professor and the dean of industrial engineering department of Yuan Ze university Taiwan. She got her Ph.D. degree from the Iowa State University in USA. She is interested in the statistical learning methods, experiment design and quality control. She directs the laboratory of statistical quality and experiments laboratory, is leading a team of research project called Using Statistical Methods to Process Mining.