Formal Concept Analysis for Process Enhancement Based on a Pair of Perspectives

Madori IKEDA, Keisuke OTAKI, and Akihiro YAMAMOTO

Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
{m.ikeda, ootaki}@iip.ist.i.kyoto-u.ac.jp, akihiro@i.kyoto-u.ac.jp

Abstract. In this paper, we propose to use formal concept analysis for process enhancement, which is applied to enterprise processes, e.g., operations for patients in a hospital, repair of imperfect products in a company. Process enhancement, which is one of main goals of process mining, is to analyze a process recorded in an event log, and to improve its efficiency based on the analysis. Data formats of the logs, which contain events observed from actual processes, depend on perspectives on the observation. For example, events in logs based on a so-called process perspective are represented by their types and time-stamps, and observation based on a so-called organization perspective records events with organizations relating the occurrence of them. The logs recently became large and complex, and events are represented by many features. However, previous techniques of process mining take a single perspective into account. For process enhancement, by formal concept analysis based on a pair of features from different perspectives, we define subsequences of events whose stops are fatal to execution of a process as weak points to be removed. In our method, the extent of every concept is a set of event types, and the intent is a set of resources for events in the extent, and then, for each extent, its weakness is calculated by taking into account event frequency. We also propose some basic ideas to remove the weakest points.

Keywords: formal concept analysis, process mining, business process improvement, event log

1 Introduction

In this paper, we show a new application of formal concept analysis, *process enhancement* (or *business process improvement*), which is one of main goals of *process mining*. We show that formal concepts are useful to discover *weak points* of processes, and that a formal concept lattice works as a good guide to remove the weak points in the process enhancement.

Formal concept analysis (FCA for short) is a data analysis method which focuses on relationship between a set of objects and a set of attributes in data. A concept lattice, which is an important product of FCA, gives us valuable insights from a dual viewpoint based on the objects and the attributes. Moreover, because of its simple and strong definition, various types of data can be translated for FCA, and so FCA attracts attention across various research domains.

Process mining [9,13] is a relatively young research domain, and is researched for treating enterprise processes recorded in event logs, e.g., operations for patients in a hospital, repair of imperfect products in a company. It provides a bridge between business process management (BPM for short) [12] and data mining. BPM has been investigated pragmatically, and data formats, softwares, and management systems are proposed for manipulating processes. Like recent data represented as "big data", the event logs also became huge and complicated. Thus, BPM researchers need theoretically efficient approaches for handling such big data. This is also the recent trend of data mining. Though many results produced in the last decade of process mining, there are still many challenges [11], and we work with FCA on two of them: "combining process mining with other types of analysis" and "dealing with complex event logs having diverse characteristics". We treat business process improvement which is an essential goal of process mining as a application of FCA. In order to achieve it, so many matters should be considered. At first, we have to decide features of a process which are modified for improvement, and there are various types of features to represent the process. In order to categorize the features, six central perspectives have been proposed [4,8]. For improvement in the target features, many modifications can be constructed. According to [8], there are 43 patterns of the modifications. We also have to evaluate the improvement, so an improvement measure is needed for the evaluation. Based on principal aspects of processes, *time*, *quality*, *cost*, and *flexibility*, four types of measures are considered [4,8]. In this paper, for making a process robust and reliable, we focus on two of the perspectives to detecting weak points of the process which are subsequences of events. For the detection, our method calculated a *weakness degree* regarded as one of cost measures for each subsequence which is represented by the extent of a formal concept.

This paper is organized as follows. In the next section, we introduce process mining and give a running example, and then, we show the problem tackled in this paper. In Section 3, we explain our process enhancement method. Conclusions are placed in Section 4.

2 Process Mining

In this section, we outline process mining with an example and show the problem which we try to solve.

2.1 Event Logs Observed from Actual Processes

Process mining has three types: *process discovery*, *process conformance checking*, and *process model enhancement*. Every type strongly focuses on and starts from facts observed from actual processes. It is the main difference from BPM (Business Process Management) [12] and also from WFM (Workflow Management) [6]. They are past fields of process mining and rely on prior knowledge.

The observed facts are recorded in *event logs*, and so the logs are the most important materials in process mining.

Actual event logs are usually represented in a semi-structured format like MXML [15] and XES [17]. Theoretically, every *event log* can be simply formalized as a pair (F, E) of a finite set F of *features* and a finite set E of *events*. Every feature $f \in F$ is a function from E to its *domain* D_f, and every event $e \in E$ is recorded in the form of $(f_1(e), f_2(e), ..., f_{|F|}(e))$ in $\prod_{i=1}^{|F|} D_{f_i}$. Each event corresponds with an occurrence or a task which are found by observation of an actual process. The observation is performed based on *perspectives*, and the set of features is decided by depending on them. Mathematically, a set P of the perspectives satisfies that every perspective $p \in P$ is a non-empty subset of F. Though six central perspectives which are called *process*, *object*, *organization*, *informatics*, *IT application*, or *environment* are proposed [4, 8], there are no standards for deciding P should be adopted in the observation. The set of perspectives P varies from an observation to another based on aims of process mining, kinds of processes executed by organizations, sensor systems installed to organizations, and many other factors. There are however some fundamental perspectives which are currently adopted in construction of event logs. Our approach focuses on two of these. One of them is the process perspective (it is sometimes called a *control-flow perspective*), which is focusing on how process occurs. If a process is observed based on the perspective, the set of features in its event log must include an *event type feature*, a *time stamp feature*, and *a case feature*. The case feature makes clear which case each event occurs in (note that some researches regard the case feature as a feature based on another perspective, *a case perspective*). Based on such a perspective, event logs clarify ordering of events for each case, and the set E of events can be treated as a partially ordered set $(E, \leq)$, so we sometimes use E as the poset $(E, \leq)$ in this paper. A sequence of events occurring in a case which are ordered based on time is called a *trace*. At the same time, the process can be observed based on the organization perspective, which is another fundamental perspective. The perspective focuses on where the occurrence happens or who performs the task, and event logs based on it must have a *place feature*, a *resource feature*, or an *employee feature*. In this paper, we assume that a given event log records statistically enough events.

Example 1 As a running example, we show a process which is handling a request for compensation within an airline. Customers may request the airline to compensate for various reasons, e.g., delay of flight or its cancelation. In such situations, the airline has to examine the validity of the request and needs to pay compensation if it is unquestionable. Table 1 shows an event log recording the compensation process which is partially quoted from [13]. In this example, an event means a task executed by an employee: the first event in the table shows that a task called "register request" is executed as the beginning of Case 1 by Pete at 11:02 on 30 Dec., 2010. In this log, the features Case ID, Event type, and Time are based on the process perspective. Resource feature is based on the organization perspective and represents human resources needed for each of the event. Cost feature comes from another perspective. The log also shows that

Table 1. An event log $L = (F, E)$ recording a compensation process of an airline: each row shows an event which is represented by five features.

Case ID	Event type	Resource	Cost	Time(dd-mm-yyy.hh:mm)
1	register request	Pete	50	30-12-2010.11:02
1	examine thoroughly	Sue	400	31-12-2010.10:06
1	check ticket	Mike	100	05-01-2011.15:12
1	decide	Sara	200	06-01-2011.11:18
1	reject request	Pete	200	07-01-2011.14:24
2	register request	Mike	50	30-12-2010.11:32
2	check ticket	Mike	100	30-12-2010.12:12
2	examine casually	Sean	400	30-12-2010.14:16
2	decide	Sara	200	05-01-2011.11:22
2	pay compensation	Ellen	200	08-01-2011.12:05
3	register request	Pete	50	30-12-2010.14:32
3	examine casually	Mike	400	30-12-2010.15:06
3	check ticket	Ellen	100	30-12-2010.16:34
3	decide	Sara	200	06-01-2011.09:18
3	reinitiate request	Sara	200	06-01-2011.12:18
3	examine thoroughly	Sean	400	06-01-2011.13:06
3	check ticket	Pete	100	08-01-2011.11:43
3	decide	Sara	200	09-01-2011.09:55
3	pay compensation	Ellen	200	15-01-2011.10:45

three cases are observed and recorded as three traces, and that their length are 5, 5, and 9, respectively.

2.2 Models of Processes

Models of processes are also important in process mining because they are deeply related with the three types of process mining: models are extracted from event logs by the process discovery, they are used with event logs for the process conformance checking and for the process model enhancement. Note that different types of models can be considered, and have been researched because of various aims of mining. Some models have been proposed for extract procedure of processes, e.g., Petri net [16], Business process modeling notation (BPMN) [3], Event-driven process chain (EPC) [7], and UML activity diagram [2]. These procedure models express workflow of a process clearly as directed graphs. For another aim, expressing how resources are involved in a process or how resources are related with each other, social network models are proposed [10,14]. A working-together social network expresses relations among resources which are used in the same case. A similar-task social network ignores cases but focuses on relations among resources used together for the same event. A handover-of-work social network expresses handovers from resources to resources in cases.

All of these models are developed for expression, and do not provide any analytical function. In other words, they only push event logs into their format,

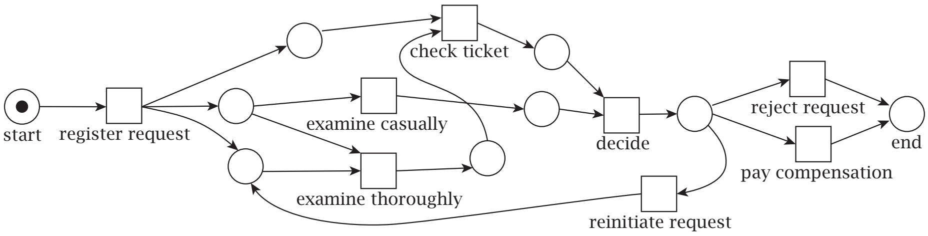

Fig. 1. A Petri net of the compensation process: every square called a transition indicates an event, and every circle called a place represents a state of the process.

and analysis is not their duty. However, for process enhancement, we need some analytical function for evaluating the enhancement. In addition, models focusing on one perspective are apt to neglect other perspectives. For example, the procedure models focusing on the process perspective do not contain information about resources which are observed based on the organization perspective. On the contrary, the social networks focusing on the organization perspective make correlations among resources explicit but make workflows which are observed based on the process perspective unclear. For our goal, detecting weak points of a process, we claim that its weakness should be measured based on at least two perspectives. This work thus relates to process model enhancement which is to extend a process model.

Example 2 Figure 1 shows a procedure model which is expressed in terms of a Petri net [16] extracted from the event log shown in Table 1. This model explicitly expresses the workflow of the compensation process and makes it clear which event happens before/after another event. On the other hand, the model ignores other perspectives: information derived from Resource and Cost features are not expressed at all in the model. Figure 2 shows a similar-task social network [10, 14] generated from the same event log. This model clarifies relations among employees sharing the same tasks, but it does not care about the ordering of events.

2.3 Weak Points Detection for Process Enhancement

Our final goal is process enhancement. For the goal, we propose to detect subsequences of events from a given event log as *weak points* which should be removed. Actually, our method does not decide whether or not subsequences of events are weak points. Instead, the method estimates the *weakness* for each of some subsequences of events and expresses it in a number called a *weakness degree*. Then, some weaker subsequence of events should be removed for the enhancement.

For the definition of the weakness degree, there are various candidates. If the process perspective is focused, sequences of events taking a lot of time in a process must be its weak points. Another type of weak points are looping sequences which

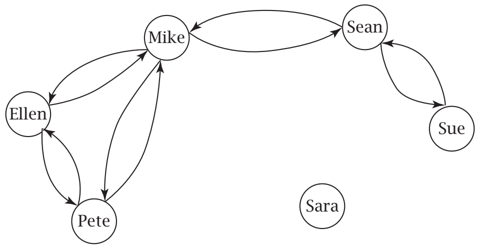

Fig. 2. A similar-task social network of the compensation process: every circle indicates an employee, and an edge is drawn between employees if their tasks are statistically similar.

many cases have to take. In the running example, it is reasonable to take costs of events into account for weakness. In this work, we focus on *importance* of a subsequence of events and *loads* of it. The importance is decided based on the process perspective and on the organization perspective. More precisely, a subsequence of events in an event log is considerable if the events are executed by a small number of resources in the log. Loads of the important sequence increase if the sequence appears many times in the log. In our method, important sequences of events having heavy loads are weak points of a process.

Example 3 In the running example, the subsequence "decide" executed by Sara should be regarded as weaker than the others. Because the subsequence is important due to the fact that it can be executed only by Sara, and because the event, "decide" by Sara, is very frequent. Only from the Petri net shown in Figure 1, it can be induced that the event "decide" is important in the process. It is also induced only from the social network shown in Figure 2 that Sara takes some important role. However, these models do not show explicitly that "decide" by Sara is important and has an impact on the process.

3 Process Enhancement via FCA

We adopt FCA for mining weak points of processes, so we firstly introduce the definitions of formal concepts and formal concept lattices with referring to [1,5]. Then, we explain our method.

3.1 From an Event Log to a Concept Lattice

A *formal context* is a triplet $K = (G, M, I)$ where G and M are mutually disjoint finite sets, and $I \subseteq G \times M$. Each element of G is called an *object*, and each element of M is called an *attribute*. For a subset of objects $A \subseteq G$ and a subset of attributes $B \subseteq M$ of a formal context K, we define $A^I = \{ m \in M \mid \forall g \in A. (g, m) \in I \}$, $B^I = \{ g \in G \mid \forall m \in B. (g, m) \in I \}$, and a pair (A, B) is a *formal concept* if $A^I = B$ and $A = B^I$. For a formal concept

$c = (A, B)$, A and B are called the *extent* and the *intent*, respectively, and let $\mathrm{Ex}(c) = A$ and $\mathrm{In}(c) = B$. For arbitrary formal concepts c and c', we define an order $c \leq c'$ iff $\mathrm{Ex}(c) \subseteq \mathrm{Ex}(c')$ (or equally $\mathrm{In}(c) \supseteq \mathrm{In}(c')$). The set of all formal concepts of a context $K = (G, M, I)$ with the order $\leq$ is denoted by $\underline{\mathfrak{B}}(G, M, I)$ (for short, $\underline{\mathfrak{B}}(K)$) and is called the *formal concept lattice* (concept lattice for short) of K. For every object $g \in G$ of (G, M, I), the formal concept $(\{\, g \,\}^{II}, \{\, g \,\}^{I})$ is called the *object concept* and denoted by γg. Similarly, for every attribute $m \in M$, the formal concept $(\{\, m \,\}^{I}, \{\, m \,\}^{II})$ is called the *attribute concept* and denoted by μm.

In our method, a formal context is obtained by translation from an event log, and then weak point mining is performed with a concept lattice constructed from the context. Suppose that the event log consists of two types of features, one of them is based on the process perspective, and that the other is based on the organization perspective. In this paper, the first one is called an *event-type feature* and is denoted by f_e, and the second is called a *resource feature* and is denoted by f_r. Note that the event-type feature represents types of events, not cases, and not time. This assumption is not strong because such features are very fundamental and are adopted in XES [17] in fact. From such an event log $L = (F, E)$ that $F \supseteq \{\, f_e, f_r \,\}$, a formal context $K_L = (G, M, I)$ is translated where $G = D_{f_e}$, $M = D_{f_r}$, $I = \{\, (g, m) \in G \times M \mid \exists e \in E. f_e(e) = g \wedge f_r(e) = m \,\}$. In the context $K_L = (G, M, I)$, $(g, m) \in I$ means that events sorted into g need a resource m. For every element $(g, m) \in I$ of the formal context K_L, we additionally define

$$\mathrm{freq}((g, m)) = |\, \{\, e \in E \mid f_e(e) = g \wedge f_r(e) = m \,\} \,|.$$

This function outputs frequency of events which are sorted into an event-type g and need resource m in the event log L.

Example 4 In the running example, "Event type" corresponds to the event-type feature, and "Resource" corresponds to the resource feature. Therefore, a formal context $K_L = (G, M, I)$ shown in Table 2 is obtained from the event log shown in Table 1. For example, $\mathrm{freq}((\mathrm{register\,request}, \mathrm{Pete})) = 2$ shows that an event "register request" by Pete is observed twice in construction of the event log in Table 1.

From a formal context K_L translated from an event log L, a concept lattice $\underline{\mathfrak{B}}(K_L)$ is constructed for process enhancement. Each formal concept $c = (A, B)$ of the concept lattice $\underline{\mathfrak{B}}(K_L)$ represents a pair of a set A of event-types and a set B of resources needed for events in A. For every formal concept $c \in \underline{\mathfrak{B}}(K_L)$, we define

$$\mathrm{Ex}_\gamma(c) = \{\, g \in \mathrm{Ex}(c) \mid \gamma g = c \,\}, \text{and}$$
$$\mathrm{In}_\mu(c) = \{\, m \in \mathrm{In}(c) \mid \mu m = c \,\}.$$

By extending freq for I, we also define

$$\mathrm{freq}(c) = \sum_{g \in \mathrm{Ex}(c)} \sum_{m \in \mathrm{In}(c)} \mathrm{freq}((g, m)).$$

Table 2. A formal context $K_L = (G, M, I)$ constructed from the event log L of the compensation process: elements of G are listed in the left most column, elements of M are listed in the first row, and every cell indicates $\mathrm{freq}(i)$ for $i \in I$ unless $\mathrm{freq}(i) = 0$.

	Pete	Sue	Mike	Sara	Sean	Ellen
register request	2		1			
examine throughly		1			1	
check ticket	1		2			1
decide				4		
reject request	1					
examine casually			1		1	
pay compensation						2
reinitiate request				1		

The value $\mathrm{freq}(c)$ is the sum of frequencies of events which are sorted into an event-type $g \in \mathrm{Ex}(c)$ and need a resource $m \in \mathrm{In}(c)$.

Example 5 Figure 3 shows a concept lattice $\underline{\mathfrak{B}}(K_L)$ of the context $K_L = (G, M, I)$ shown in Table 2. For example, the left most circle in the figure indicates a formal concept $c_2 = (\{\,\text{check ticket}, \text{pay compensation}\,\}, \{\,\text{Ellen}\,\})$. The sum of frequencies $\mathrm{freq}(c_2) = 3$ means that a task "check ticket" or "pay compensation" executed by Ellen appears three times in the event log L shown in Table 1.

3.2 Calculating Weakness Degrees

As we mentioned in Section 2.3, for every subsequence of events which is the extent of a formal concept, we define the weakness degree, and the weakness is estimated from its importance and its loads.

The importance is estimated based on both of the process perspective and the organization perspective. Every formal concept $(A, B) \in \underline{\mathfrak{B}}(K_L)$ is based on both of the perspectives because A is a set of event-types observed from the process perspective and B is a set of resources observed from the organization perspective. Such a formal concept is considered to represent that accomplishing all the events in A needs at least one of the resources in B and that every resource in B can execute all the events in A. From this consideration, we define the importance $\mathrm{imp}(c)$ of the subsequence $\mathrm{Ex}(c)$ of a formal concept $c \in \underline{\mathfrak{B}}(K_L)$ as

$$\mathrm{imp}(c) = \frac{1 + |\mathrm{Ex}_\gamma(c)|}{1 + |\mathrm{In}(c)|} \times \frac{1 + |\mathrm{Ex}(c)|}{1 + |\mathrm{In}_\mu(c)|}.$$

We call this an *importance factor*. Roughly speaking, this factor becomes large when a small number of resources are needed for a large number of events. The first term means the ratio of the number of events to the number of resources which can accomplish the events. In other words, if some or many events rely on

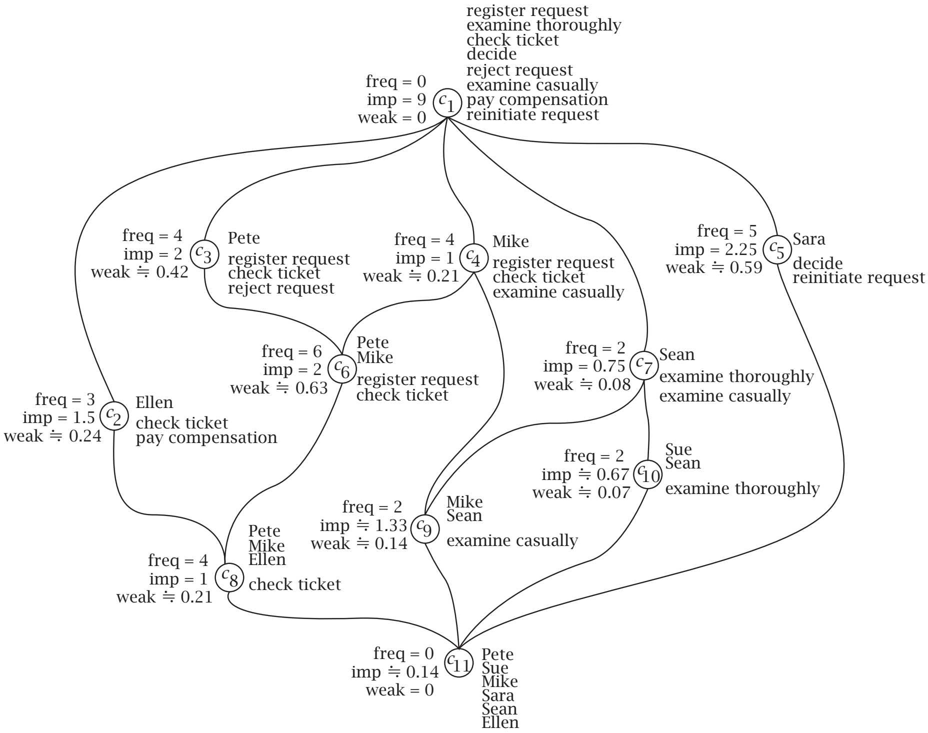

Fig. 3. A formal concept lattice $\underline{\mathfrak{B}}(K_L)$ constructed from the formal context K_L: Each circle represents a formal concept $c \in \underline{\mathfrak{B}}(K_L)$. Each edge represents an order $\leq$ between two concepts, and the greater concept is drawn above, and transitional orders are omitted. Every formal concept c accompanies with Ex(c) and In(c) on its right side and with freq(c), imp(c), and weak(c) on its left side.

little resources then the term is large. The second means the ratio of the number of resources to the number of events which are executed by the resources. It becomes large, if some or little resources are exhausted by many events. Also, we define load(c) of the subsequence Ex(c) as

$$\mathrm{load}(c) = \frac{\mathrm{freq}(c)}{|E|}$$

and call it a *load factor*. This is a ratio of frequency of events in the sequence Ex(c) to frequency of the whole events E. Then, for the subsequence Ex(c), the weakness degree weak(c) is defined as

$$\mathrm{weak}(c) = \mathrm{imp}(c) \times \mathrm{load}(c).$$

When an important sequence Ex(c) takes a heavy load, weak(c) becomes large. In other words, the weakness degree numerically shows liableness of trouble with Ex(c) to cause the whole process down. By extending this definition, the weakness of the whole process can be expressed as $\sum_{c \in \underline{\mathfrak{B}}(K_L)} \mathrm{weak}(c)$.

Example 6 In Figure 3, importance factors and weakness degrees of every subsequence of events Ex(c), $c \in \underline{\mathfrak{B}}(K_L)$ are also drawn. The importance factors show that the sequence of tasks $\mathrm{Ex}(c_5) = \{\, \mathrm{decide}, \mathrm{reinitiate\,request} \,\}$ executed by Sara is the most important. Indeed, there is no employee who can execute the tasks "decide" and "reinitiate request", but Sara. On the other hand, the weakness degrees show that the sequence $\mathrm{Ex}(c_6) = \{\, \mathrm{register\,request}, \mathrm{check\,ticket} \,\}$ of tasks is the weakest, and that the most important sequence $\mathrm{Ex}(c_5)$ is the secondary weakest. This reversal of roles is caused by their load factors. The total weakness of the whole process $\sum_{c \in \underline{\mathfrak{B}}(K_L)} \mathrm{weak}(c)$ is around 2.59.

3.3 Removing Weak Points

A process recorded in an event log L can be enhanced by removing the weakest point or by reducing the total weakness $\sum_{c \in \underline{\mathfrak{B}}(K_L)} \mathrm{weak}(c)$. Though there are many ways for achieving the enhancement, in this paper, we achieve it by operations to an original formal context $K_L = (G, M, I)$ which remove some weakest formal concepts from its concept lattice $\underline{\mathfrak{B}}(K_L)$, or which totally reduce $\sum_{c \in \underline{\mathfrak{B}}(K_L)} \mathrm{weak}(c)$. We here show some basic ideas for such operations.

Observing the definitions about the weakness shows that there are three plans for the reduction: reducing importance factors, reducing load factors, and decreasing the number of formal concepts. Though there are many operations achieving the plans, realizable operations are restricted by considering that we try to manage an actual enterprise process. Reduction of importance factors can be achieved by increasing the number of resources to the number of events requiring the resources. Also, reducing events can decrease importance factors, but we do not adopt this way because it has a risk that the process never works. In other words, we try to enhance processes by investment in equipment not by polishing processes. Besides, reducing load factors is not reasonable for our method, because we do not have control of frequency of events. Thus, our enhancement operations are to increase resources for events requiring them or to decrease formal concepts.

For enhancement of a process recorded in an event log L, we show two kinds of such operations. The first kind is adding $(g, m) \notin I$ such that $g \in \mathrm{Ex}(c)$ and $m \in M$ to I for removing a formal concept c from $\underline{\mathfrak{B}}(K_L) \ni c$. This means to expand flexibility of resources, e.g., updating machines, and expanding applicability of materials by an innovation. We have to note that the total weakness is not always reduced in this case. The second is adding m such that $m \notin M$ and $(g, m) \notin I$ such that $g \in \mathrm{Ex}(c)$ to M and I, respectively. This can reduce the total weakness $\sum_{c \in \underline{\mathfrak{B}}(K_L)} \mathrm{weak}(c)$. This means introducing new resources for sequences of events Ex(c). For example, purchase of the same machines as existing ones, and using a substitute to make up a shortage of materials. In order to decide properly which kind of operations is executed, we need other factors, e.g., execution time of the process, or costs and easiness of applying the operations.

Example 7 In the running example, there are some choices for removing the weakest sequence $\mathrm{Ex}(c_6) = \{\,\text{register ticket}, \text{check ticket}\,\}$. For example, addition of $(\text{register request}, \text{Ellen})$ to I which means that Ellen gets an ability to "register request" can remove the weak point. It removes the concept c_6, changes c_2 into $(\{\,\text{register request}, \text{check ticket}, \text{pay compensation}\,\}, \{\,\text{Ellen}\,\})$, and c_8 into $(\{\,\text{register request}, \text{check ticket}\,\}, \{\,\text{Pete}, \text{Mike}, \text{Ellen}\,\})$, respectively. If we assume that "register request" is shared equally by Pete, Mike, and Ellen, the numbers are changed: $\mathrm{freq}(c_2) = 4$, $\mathrm{imp}(c_2) = 2$, $\mathrm{weak}(c_2) \risingdotseq 0.42$, $\mathrm{freq}(c_3) = 3$, $\mathrm{imp}(c_3) = 2$, $\mathrm{weak}(c_3) \risingdotseq 0.32$, $\mathrm{freq}(c_8) = 7$, $\mathrm{imp}(c_8) = 2.25$, $\mathrm{weak}(c_8) \risingdotseq 0.83$. In this case, the total weakness increases to around 2.66. Employing a new person, Bob, having ability to execute "register request" is an operations of the second type. This is to add $\text{Bob} \notin M$ to M and to add $(\text{register request}, \text{Bob}) \notin I$ to I. In this case, a new concept $c_{12} = (\{\,\text{register request}\,\}, \{\,\text{Bob}\,\})$ is generated, and then, the total weakness decrease to 2.17 by assuming that "register request" is shared equally by Pete, Mike, and Bob. Because $\mathrm{weak}(c_3)$ and $\mathrm{weak}(c_6)$ decrease to around 0.32 and around 0.26, respectively, and $\mathrm{weak}(c_{12}) \risingdotseq 0.05$.

4 Conclusions

In this paper, we propose to apply FCA (formal concept analysis) to process enhancement. FCA is to analyze data from a dual viewpoint which is based on objects and attributes. Processes are recorded in event logs which are constructed by observation based on some perspectives. We assign a pair of the process perspective and the organization perspective to the objects and the attributes of FCA in order to investigate weak points of a process. Weakness of a sequence of events executed by resources is calculated by importance and loads of it.

There are many problems to be solved. Our weakness of process is not defined from enough analysis because only two features from two perspectives are considered. For improving a process more efficiently, we need to take into account other features across other perspectives in weak point detection. For example, using a time-stamp feature enables us to detect bottleneck of a process, using a cost feature enables us to find costly sequences. It may be achieved by combining other process models with our concept lattice. We also have to refine the operations for removing weak points. In our method, the number of the choices for enhancement sometimes becomes so large. A plan of the refinement is to estimate in advance the total weakness of a reinforced process for each of the choices. Combining other models is also useful. For example, combining procedure models with our method can suggest some effective operations from the many choices. Because such models sufficiently treat order of events in traces which is ignored by our lattice based approach. On the other hand, there are many constraints on resources in practical processes, e.g., some materials can be substituted few materials but the others can not, and employees are divided into groups in a company. In order to reduce the choices based on such constrains, social network models might be useful.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 26280085.

References

1. B. A. Davey, and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2002.
2. M. Dumas, and A. ter Hofstede. UML Activity Diagrams as a Workflow Specification Language. In M. Gogolla and C. Kobryn, editors, *UML 2001 The Unified Modeling Language. Modeling Languages, Concepts, and Tools*, *Lecture Notes in Computer Science*, vol. 2185, pp. 76–90, 2001.
3. R. Flowers, and C. Edeki. Business Process Modeling Notation. *International Journal of Computer Science and Mobile Computing*, vol. 2, issue 3, pp. 35–40, 2013.
4. F. Forster. The Idea behind Business Process Improvement : Toward a Business Process Improvement Pattern Framework. *BP Trends*, pp. 1–14, 2006.
5. B. Ganter, and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag New York Inc., 1997.
6. D. Georgakopoulos, M. Hornick, and A. Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, vol. 3, issue 2, pp. 119–153, 1995.
7. J. Mendling, and M. Nüttgens. EPC markup language (EPML): an XML-based interchange format for event-driven process chains (EPC). *Information Systems and e-Business Management*, vol. 4, issue 3, pp. 245–263, 2006.
8. H. A. Reijers, and S.L. Mansar. Best Practices in Business Process Redesign: An Overview and Qualitative Evaluation of Successful Redesign Heuristics. *Omega*, vol. 33, issue 4, pp. 283–306, 2005.
9. A. Shtub, and R. Karni. *ERP - The Dynamics of Supply Chain and Process Management*. Springer, 2010.
10. M. Song, and W. van der Aalst. Towards Comprehensive Support for Organizational Mining. *Decis. Support Syst.*, vol. 46, issue 1, pp. 300–317, 2008.
11. W. van der Aalst, et al.. Process Mining Manifesto. In F. Daniel, K. Barkaoui and S. Dustdar, editors, *Business Process Management Workshops*, *Lecture Notes in Business Information Processing*, vol. 99, pp. 169–194, 2012.
12. W. van der Aalst, A. ter Hofstede, and M. Weske. Business Process Management: A Survey. In W. van der Aalst and M. Weske, editors, *Business Process Management*, *Lecture Notes in Computer Science*, vol. 2678, pp. 1–12, 2003.
13. W. van der Aalst. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.
14. W. van der Aalst, H. Reijers, and M. Song. Discovering Social Networks from Event Logs. *Comput. Supported Coop. Work*, vol. 14, issue 6, pp. 549–593, 2005.
15. W. van der Aalst, B. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. Weijters. Workflow Mining: A Survey of Issues and Approaches. *Data Knowl. Eng.*, vol. 47, issue 2, pp. 237–267, 2003.
16. B. van Dongen, A. Alves de Medeiros, and L. Wen. Process Mining: Overview and Outlook of Petri Net Discovery Algorithms. In K. Jensen and W. van der Aalst, editors, *Transactions on Petri Nets and Other Models of Concurrency II*, *Lecture Notes in Computer Science*, vol. 5460, pp. 225–242, 2009.
17. H. Verbeek, J. Buijs, B. van Dongen, and W. van der Aalst. XES, XESame, and ProM 6. In P. Soffer and E. Proper, editors, *Information Systems Evolution*, *Lecture Notes in Business Information Processing*, vol. 72, pp. 60–75, 2011.