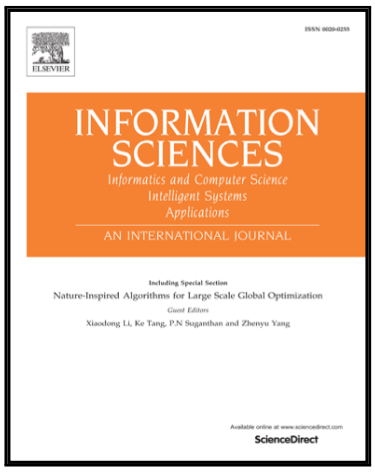

Using contextualized activity-level duration to discover irregular process instances in business operations

Ping-Yu Hsu , Yu-Cheng Chuang , Yao-Chung Lo , Shuang-Chuan He

PII: S0020-0255(16)31240-3
DOI: 10.1016/j.ins.2016.10.027
Reference: INS 12586

To appear in: *Information Sciences*

Received date: 10 April 2016
Revised date: 7 October 2016
Accepted date: 9 October 2016

Please cite this article as: Ping-Yu Hsu , Yu-Cheng Chuang , Yao-Chung Lo , Shuang-Chuan He , Using contextualized activity-level duration to discover irregular process instances in business operations, *Information Sciences* (2016), doi: 10.1016/j.ins.2016.10.027

Using contextualized activity-level duration to discover irregular process instances in business operations

Ping-Yu Hsu[a,*], Yu-Cheng Chuang[a], Yao-Chung Lo[a], Shuang-Chuan He[a]

[a]*Department of Business Administration, National Central University, No.300, Jhongda Rd., Jhongli Dist., Taoyuan City, Taiwan (R.O.C.)*
[*]*Corresponding author: Ping-Yu Hsu*
E-mail: pyhsu@mgt.ncu.edu.tw
Address: No.300, Jhongli Dist., Taoyuan City, Taiwan (R.O.C.)
Tel.: 886-3-4227151 ext: 66168

ABSTRACT

Effective time management is one of the most crucial characteristics of a successful business. For most businesses, time management is an area that has much scope for further improvement. Irregularities in the execution duration of business processes impede corporate agility and can incur severe consequences, such as project failures and financial losses. Efficient managers must constantly identify potential irregularities in process durations to anticipate and avoid process glitches. This paper proposed a k-nearest neighbor method for systematically detecting irregular process instances in a business using a comprehensive set of activity-level durations, namely execution, transmission, queue, and procrastination durations. Moreover, because agents, customers, and other variables influence the progress of processes, contextual information was presented using fuzzy values. The values and corresponding membership functions were used to adjust the durations of each activity. This proposed method was applied to the system logs of a medium-sized logistics company to identify irregularities. Experts confirmed that 81% of the identified irregular instances were abnormal.

Keywords: Activity-level duration; Fuzzy set; Process instances; Process irregularities; Workflow

1. Introduction

Only agile organizations can thrive in a constantly changing and unpredictable business environment. Agile businesses seek operational excellence by continually improving business processes [5], which are influenced by time management. Irregularities in the execution duration of business processes impede corporate agility and can yield severe consequences, such as project failures and financial losses [6]. Therefore, competent managers must constantly identify potential irregularities in durations to anticipate and avoid process glitches.

Ko [26] defined a business process as "a series or network of value-added activities, performed by their relevant roles or collaborators, to purposefully achieve the common business goal." Essentially, processes can be sequentially divided into activities or tasks that are conducted by people playing various roles.

Because of the abundance of enterprise resource planning (ERP) and business process management softwares, numerous studies have focused on mining process information from system logs [6, 10, 15, 21, 25, 26, 29, 45]; most of these studies have used the data to reconstruct workflows and compare the consistency of the mined and documented workflows.

Outlier detection, although a relevant subject, has been scarcely discussed in the literature [6, 10, 12, 15, 21, 22, 24, 25, 33, 45]. Needleman and Wunsch [33], Chuang et al. [15], Wang et al. [45], Bouarfa and Dankelman [10], and Huanget al. [21] have identified abnormal sequences of process activities. Bezerra and Wainer [6] proposed a system for identifying rarely used sequences to simplify workflow design during system implementation. Jakkula et al. [22] identified process outliers through irregular temporal relationships and Kang et al. [24] used the local outlier factor (*LOF*) to examine abnormal termination of work processes.

Except for Kang et al. [24], which focused on process durations, none of the aforementioned studies have used durations to identify abnormal process instances. However, studies on time management in business operations have suggested that researchers should address activity levels for effectively controlling and monitoring the progress of business processes [26, 29]. Collectively, these studies have investigated the monitoring of the execution, transmission, queue, and procrastination durations of activities.

Because business processes are performed by people (agents), such as sales and production managers, who assume roles and are affected by other people, such as customers and coworkers, their influence on activity duration should be considered when irregular process instances are identified. For example, an

inexperienced salesperson may need more time to complete a sales order placed by a new customer compared with the time required by an experienced salesperson taking orders from a recurring customer. The features of agents involved in activities are considered the contextual information of the activities. Contextual information describes the circumstances in which an activity is executed. In addition to agent features, information regarding other circumstances can also be included. For example, this study considered the features of customers whose orders triggered a process instance.

The goal of the study, therefore, was to identify irregular processes using activity-level durations and contextual information. To simplify the description of contextual information, fuzzy values were applied for representing such information; furthermore, the fuzzy values and associated membership functions were used for adjusting the activity-level durations. The distances between instances were calculated using the differences between adjusted durations. The k-nearest neighbor (*k*-NN) algorithm was then used for identifying irregular process instances that were distant from other instances.

The effectiveness of the proposed approaches was demonstrated using process execution logs, employee information, and customer transactions extracted from the ERP system of a medium-sized logistics company. The data set contained 2,169 process instances, of which 21 were identified as irregular using the proposed method. The results were presented to a group of domain knowledge experts for verification [4]. These experts, who on average had 14 years of work experience in the industry, confirmed that 81% of the identified process instances were abnormal. By contrast, only 9% of the identified outlier process instances using process durations based on the *LOF* were confirmed as outliers in the same environment setting.

The remainder of the paper is organized as follows: Section 2 presents a review of the literature. Sections 3 and 4 detail the methodologies and related definitions of the study, including duration definitions, contextual variables, fuzzy contextual values, and contextual membership-value-adjusted durations. The effectiveness of the assessment is presented in Section 5. Finally, Section 6 provides a conclusion and suggestions for future studies.

2. Literature review

2.1 Enterprise modeling

Efficient business process management is critical to a firm's survival and prosperity in an extremely competitive business environment [2]. A business process is conducted by more than one person or department,

possibly involving multiple people, machines, and systems from different departments or organizations that collaborate to achieve a common business goal [26]. Therefore, each activity in a business process includes agents, which might be either machines or people, required to implement and accomplish the required tasks and jobs. Because of the importance of business processes, new models to describe them have been proposed consistently [13].

Business processes can be divided into four classes, namely, production workflows, administrative workflows, ad hoc workflows, and collaborative workflows [2]. Production and administration workflows are characteristically repetitive and predictable, whereas ad hoc and collaborative workflows require more intensive participant involvement and are more difficult to predict in terms of duration. The study was designed to identify outliers in production and administration workflows.

Proposed by Keller in 1992, the event process chain (EPC) modeling method has been a critical tool for presenting business process models. The method presents the operation of business processes in an easily comprehensible graphical manner [43] and has been extensively employed by numerous large software vendors, such as Systems, Applications and Products (SAP). The EPC modeling method comprises three main elements: (1) functions, which are activities to be performed; (2) events, which describe the state of the functions before and after the execution; and (3) logical connectors, which are AND, XOR, and OR. This study adopted the EPC modeling method to present business processes.

Fig. 1 illustrates a process of international trade depicted using the EPC modeling method. The process begins when a customer places an order and ends when the cargo is delivered and the corresponding receivable is recorded.

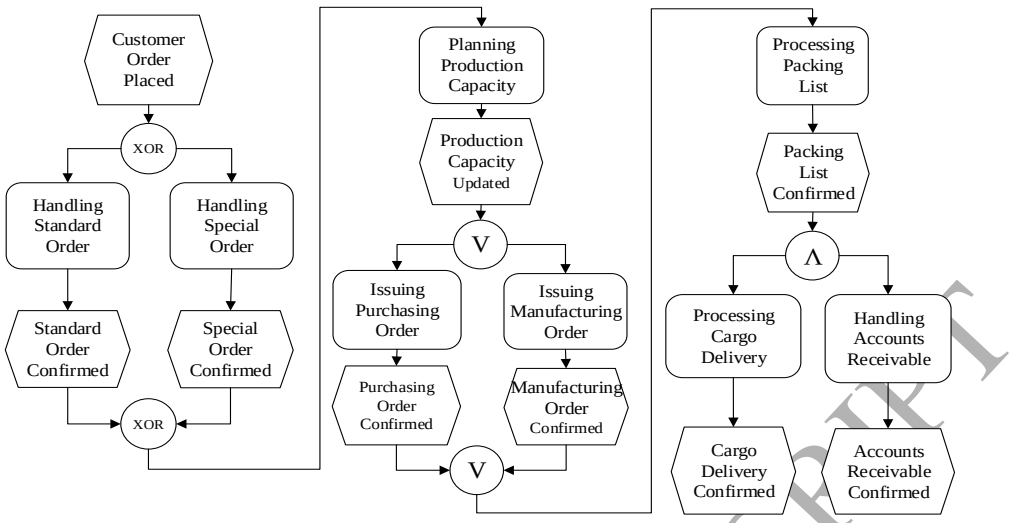

Fig. 1. Sample process of international trade.

2.2 Irregularity detection in business processes

Numerous studies have introduced process mining techniques that are based on system logs [6, 9, 15, 25, 27, 28, 29, 34, 45, 46]. However, most of these studies have focused on reconstructing business processes using data logs ; relatively fewer researchers have attempted to identify irregularities using system logs [5, 10, 12, 15, 21, 22, 24, 25, 27, 36, 38, 45]. For example, Xu et al. [47] and Roy et al. [36] have indicated that the design of business process outlier detection or error analysis and models in business processes have not been adequately considered, making it difficult to achieve the objective of improving the exception management in the supply chain. Lee et al. [28] echoed the argument and highlighted that eliminating unexpected irregularities in process instances could minimize business overhead costs and maximize profits.

Limited studies have investigated process anomaly detection and even fewer of them have aimed at improving process efficiency. Bezerra et al. [6] proposed a system for identifying rarely used activities to simplify workflow design during system implementation. Kim et al. [25] presented an outlier detection technique for monitoring database activity. Chuang et al. [15] and Wang et al. [45] have proposed algorithms for detecting abnormal activity sequences to alert managers about process irregularities. Kang et al. [24] adapted the *LOF* to detect abnormal terminations of business processes in real-time. Bouarfa and Dankelman [10] proposed building consensus sequences for disease treatments using medical data from various sources. A stepwise sequence alignment algorithm was used to detect outliers during medical care [33]. Huanget al. [21] proposed detecting abnormal treatment of audit data stored in electronic medical records by constructing a

graph of clinical pathways from the database. In the graph, each record was recorded as a path, each vertex represented an attribute in the record, and each attribute was associated with a standard value. The degree of anomaly of clinic visits was calculated according to the summation of weighted differences between the attribute values of the standard pathway and the visits. Xu et al. [47] devised a multi-intelligent agent framework to collect and analyze data for irregularity identification to improve the outbound logistics service. Bhuyan et al. [8] proposed the construction of a decision tree for selecting features and judging the normality of network-wide traffic; the new traffic was then compared with the tree to decide whether it was normal.

Although time is an essential aspect in business processes, only a few studies have attempted to detect process anomalies using temporal information. Di Ciccioet al. [17] established a prediction model that distinguished flight tracjectory as an anomaly based on detection of an airplane's behavior between two scheduled times. Jakkula et al. [22] used 13 temporal relationships between events developed by Allen in 1994 to describe the sequences of events and detect outliers when a sequence conflicted with the recorded order. Bruno and Garza [12] proposed merging events occurring within the same time window into mega tuples, which were then used to identify association rules comprising pairs of attributes and values. The rules could be used to build quasifunctional dependency among attributes and an outlier could be detected when a mega tuple broke one of the dependencies. Martínez-Rego et al. [32] modified a passive-aggressive kernel one-class classification algorithm incorporated with a Bernoulli CUSUM chart to identify the time when continuous data streams, collected from censors in the process, departed notably from normal patterns.

Thus, many studies are related to process modeling and contextual information description in supply chain management. However, to the best of our knowledge, only the works of Needleman and Wunsch [33], Chuang et al. [15], Wang et al. [44], Bouarfa and Dankelman [10], Huang et al. [21], Kim et al. [25], Bezerra and Wainer [6], Jakkula et al. [22], Xu et al. [46], Bhuyan et al. [8], Martínez-Rego et al. [32], and Kang et al. [24] have discussed process outlier detections. Furthermore, only the studies of Kim et al. [25], Kang et al. [24], and Chuang et al. [15] have addressed irregular duration.

This study is the first to investigate activity level intervals for detecting process outliers; therefore, activities that are completed too rapidly or too slowly might indicate trouble.

2.3 Time Management

Alotaibi and Liu [3] pronounced that business time management determines the satisfaction of customers

and the process cycle time is a vital competitive advantage that affects the success or failure of businesses. A firm should measure and manage business process time to achieve optimal process durations that satisfy customers. Substantial research has been conducted on the critical topic of time management in business operations [4, 26, 29, 48]. Previous studies have integrated process management with durations to plan execution times according to time constraints and deadlines [26, 29], determine critical paths according to time constraints [29, 50], construct simulation models using a critical path network [31], calculate the time required to extend projects [9], shorten the customer service and internal development time [3], and identify abnormal process terminations [24]. Rescue time optimization is also crucial in cases of catastrophic disaster, which could suddenly occur in business communities. Hu and Sheng [20] proposed an integrated mechanism to make the most appropriate decisions regarding rescue deploying resource networks and minimizing rescue times. Pluta and Wójcik [35] explored and concluded that the time management skills of employees can improve value for the client organization.

The aforementioned studies had focused on planning, allocating, and managing the temporal aspects of processes by controlling activity execution at run-time to avoid deadline violation. By contrast, the study focused on identifying irregular process instances by applying activity-level durations.

2.4 Durations in business process studies

This study adopted an agent-centric perspective of the business process; namely, agents must be assigned to activities before the activities can begin. The activity level duration includes the time required for transferring information and possibly physical objects to agents involved in subsequent activities (termed as the transmission duration); the time spent waiting until the agents are released from other activities, if any (termed as the queue duration); the time spent in preparation for the activity (termed as the procrastination duration); and the time the agent actually spends in executing the activity (termed as the execution duration). Execution, transmission, queue, and procrastination durations have all been proposed in previous studies, albeit not for the purposes of identifying irregular instances [9, 26, 29, 50].

Execution Duration: This common term has been adopted for various research purposes; therefore, it is characterized by confusing and occasionally conflicting definitions. The defined interval of execution duration has ranged from the time required to complete a single activity to the time required to execute an entire business process [9, 20, 21, 23, 26, 29, 50]. This duration, which has been referred to under various names,

such as activity duration, activity completion time, and workflow process duration, can be calculated according to previous execution or estimated by specialists according to their experience and expectations. In several studies, execution duration has been further categorized into minimum, maximum, and average durations. In this study, execution duration was defined as the amount of time required to complete an activity. A longer duration potentially indicates less efficiency.

Transmission Duration: This term, originally used to identify critical paths in processes [50], has been defined as the temporal interval between the executions of two consecutive activities [29]. In this study, transmission duration was defined as the interval between the completion time of an activity and the time of notification for the next activity. In complex cases, completing the transmission required transferring related data and physical objects [41]. The time required for performing such a transmission was regarded as part of the transmission duration. Therefore, agents who spend considerable time transmitting tasks can potentially reduce process efficiency.

Queue Duration: In this study, this term described the time spent waiting for resources that were required to complete an activity [23]; the resources were assumed to be the agents. Moreover, agents that caused an excessive queue duration warranted notice because they may have been assigned excess work or may require assistance to increase working efficiency.

Procrastination Duration: This term described the time agents spent preparing to execute assigned activities after the agents became available. Procrastination duration began when an agent received notification to begin an activity but postponed the execution for no apparent reason [9]. Agents who spent considerable time procrastinating can reduce process efficiency.

Table 1 presents a summary of the previous studies that have applied different combinations of activity-level durations.

Table 1
Previous studies on durations in business processes.

No	Researchers	Execution Duration	Transmission Duration	Queue Duration	Procrastination Duration
1.	Son and Kim (2001) Jiang and Nie (2011)	✔		✔	
2.	Zhuge et al. (2001) Li and Fan (2009) Sun and Zhao (2013)	✔	✔		
3.	Bierbaumer et al. (2005)	✔			✔
4.	Eder et al. (1999) Eder and Panagos (2001) Wang and Zeng (2008) Zeng et al.(2013)	✔			
5.	This study	✔	✔	✔	✔

This study differed from previous studies in that all four activity-level durations were considered and irregularities in process instances were identified.

2.5 Contextual information utilized in supply chain management

The study utilized agents and customers as contextual information to describe the circumstances under which a particular process instance was executed in an organization. Since supply chain management investigates the collaborating and competing issues among organizations in an ecosystem [26], this literature review provides much insight into defining contextual information of business processes. Xu et al. [47] designed a multi-perspective ontology framework to monitor outbound logistics exceptions. The contextual information discussed in the ontology included delivery, warehouse/storage, order/forecasting, planning, and transportation. Comuzzi et al. [16] proposed a flexible optimized monitoring framework and the implementation comprised the contextual information for monitoring stakeholders, which included customers in the cross-organizational business processes or any collaborating partners such as third-party auditors or public agencies. Guo and Lu [18] developed a personalized recommendation model by considering e-commerce user's personal features and contextual information including location, culture, and social and environmental factors that affected their behaviors. They believed that the contextual information determined the degree to which a user's particular information needs were satisfied. Chen and Kamara [14] built an information management framework to facilitate mobile computing system design. According to them, supplying contextual information that included the user, construction information, and construction site and

personalized construction information to on-site workers could advance the speed and correctness of information transmitting, thereby increasing workers' productivity. Singh and Best [40] used transaction-related contextual meta-data, such as client id, user id, date, time, object id, transaction code, and vendor bank account change number, to identify and report anomalies, which could not be identified using traditional methods.

In the aforementioned studies, contextual information has been considered a feature to describe the supply chain processes or the environment to satisfy customer needs. This study used employee and customer information to describe the environment where a process instance was executed to discover outliers. The study selected employee and customer information because of the availability of the data; most enterprise information systems stored the required data. However, the supply chain information may be stored by major players and may not be openly accessible. Nonetheless, the proposed method should be applicable when the supply chain contextual information is available.

3. Methodology

3.1 Data derived from e-business systems embedded with workflow modules

Traditional workflow logs include only agent id, process, activity, and activity starting and ending time data. However, because many workflow systems have been embedded in e-business systems, such as ERP systems, the information available in systems now includes diverse master and transaction data, including those on customers, products, materials, sales orders, purchase orders, and production orders. Consequently, the contextual information derived from master and transaction data can be used to describe the environments wherein activities are executed. For example, the number of orders processed by agents and placed by customers can be viewed as contextual information used to describe agent experience and customer familiarity.

Therefore, in this study, the following information was assumed to be stored in the same system: processes; process instance id; activity id; and the corresponding notification, beginning, and end times; agents who perform each task; and contextual information that can be derived from transaction and master data.

3.2 Outlier detection based on activity level durations

This section proposes a basic method to perform outlier detection based on activity level duration: The Outlier Detection with Activity Level (*ODAL*) durations comprises three steps, namely, calculating

activity-level durations according to differences in durations, identifying the k^{th} nearest neighbor for each instance, and identifying outlier instances according to the distance between the instance and its k^{th} neighbor.

A process instance is a record of the activities that is performed during a process execution. Some activities documented in a business process may not be performed at a particular instance when they are in a path branch that is not selected. In addition to activities, process instances record the dependency relationships among the executed activities and the notification, beginning and end time of activities, and agents in charge. Process instances are formally defined in Definition 1 based on graph theory because graphs are typically used to describe the partial orders among activities, and have been used in numerous studies to portray business processes [37, 38, 49].

Definition 1

Assume $G = \{g_1, \ldots, g_n\}$ as a set of agents.

a. $A = \{a_1, \ldots, a_n\}$ is a set of activities that have been performed for a particular execution of a business process.

b. $E \subseteq A \times A$, is a set of edges that forms an acyclic graph and depicts the operational dependency between activities. The relationships are inherited from the corresponding EPCs.

c. $T_N(a_i, g_j), T_S(a_i, g_j)$, and $T_F(a_i, g_j)$ denote the time that agent g_j is notified of a_i, the time a_i is started, and the time a_i is completed, respectively.

d. A process instance $L =< A, E, T_N, T_S, T_F >$.

e. "." is a dereference operator of process instances. For example, $L.A, L.E$, and $L.T_N$ denote the activities, edges between acvtivities, and the notification time of activities in L, respectively.

In this study, four types of activity-level durations were defined, namely execution duration (D_E), transmission duration (D_T), queue duration (D_Q) and procrastination duration (D_P). Such durations are defined in Definitions 2 and 3.

Definition 2

Assume L as a process instance and $a \in L.A,\ g, \hat{g} \in G$.

a. The execution duration is the duration between the statuses of Started and Finished and is defined as

$$D_E(L, a) = L.T_F(a, g) - L.T_S(a, g).$$

b. The transmission duration is the duration between the finish time of activities, which must be completed

before a according to $L.E$, and the time that g is notified of a.

$$D_T(L, a) = L.T_N(a, g) - max_{(<a_j, a> \in L.E)} L.T_F(a_j, \hat{g}).$$

A notification is sent for an activity only when all previous activities have been completed. If an activity has to wait for the completion of several other activities, then the transmission duration is defined as the elapse between the finish time of the last activity and the notification time of the target activity because the elapse time is regarded as the amount of time required to transfer data and physical objects used in previous activities to the site of the target activity. Furthermore, the definition of execution duration is self-explanatory and, therefore, is omitted.

Table 2 presents a set of logs in which $L_1, .., L_5$ are instances of one business process and L_{50} and L_{51} are instances of another business process. Agent g_1 participates in a_1 and a_2 of L_1 and a_1 of L_4. The a_a of L_{50}, a_2 of L_2, and a_1 of L_3 compete for the availability of g_3.

Table 2
Instances of two processes.

	a_1 Customer Order Handling				a_2 Production Order Handling			
	T_N	T_S	T_F	*Agent*	T_N	T_S	T_F	*Agent*
L_1	*07:50*	*08:00*	*08:30*	g_1	*09:30*	*10:00*	*10:30*	g_1
L_2	*08:00*	*10:00*	*11:00*	g_2	*11:30*	*13:00*	*14:00*	g_3
L_3	*13:00*	*14:10*	*15:20*	g_3	*15:30*	*16:30*	*17:00*	g_3
L_4	*14:40*	*15:10*	*15:30*	g_1	*15:35*	*15:41*	*16:05*	g_5
L_5	*16:10*	*18:20*	*18:30*	g_5	*19:00*	*19:15*	*19:30*	g_4
	a_a Accounts Receivable Handling				a_b Note Receivable Handling			
	T_N	T_S	T_F	*Agent*	T_N	T_S	T_F	*Agent*
L_{50}	*11:30*	*12:00*	*13:00*	g_3	*14:00*	*15:00*	*16:00*	g_7
L_{51}	*16:30*	*16:50*	*17:00*	g_9	*18:00*	*19:00*	*19:30*	g_8

Example 1

According to Table 2 and assuming that a_1 is before a_2 in corresponding business processes,

$$D_E(L_1, a_1) = L_1.T_F(a_1, g_1) - L_1.T_S(a_1, g_1) = 8:30 - 8:00 = 30$$

$$D_T(L_1, a_2) = L_1.T_N(a_2, g_1) - L_1.T_F(a_1, g_1) = 9:30 - 8:30 = 60$$

The definitions of queue and procrastination durations refer to other process instances in the system log because an agent may be required for more than one process instance simultaneously. Therefore, the time spent

waiting for the agent to become available or to start an activity is affected by other instances. Activities should be executed in a first-come-first-serve (FCFS) manner; in other words, activities are executed in a sequence of notifications when competing for the same agents. The execution of an activity is processed in a nonpreemptive mode; in other words, after an activity has begun, the agent will not engage in other activities until the current activity is completed.

Definition 3

Assume that LD is a set of process instances, $L \in LD$, $a \in L.A$, and $g \in G$.

a. The queue duration is the time spent waiting for a busy agent to become available; the agent may have been occupied by activities of other process instances and is defined as

$$D_Q(L,a) = \max\left(0, \max\left(\left\{L_h.T_F(a_j,g) \middle| L_h \in LD, L_h \neq L, a_j \in L_h.A, L_h.T_N(a_j,g) < L.T_N(a,\ g)\right\}\right) - L.T_N(a,\ g)\right).$$

b. The procrastination duration is the duration of delay caused by the procrastination of the assigned agent and is defined as

$$D_P(L,a) = L.T_S(a,\ g) - \max\left(\left\{L_h.T_F(a_j,g) \middle| L_h \in LD, L_h \neq L, a_j \in L_h.A, L_h.T_N(a_j,g) < L.T_N(a,g)\right\} \cup \{L.T_N(a,g)\}\right).$$

The expression of $L_h.T_N(a_j,g) < L.T_N(a,g)$ is designed to determine the activities that are scheduled before activity a and that compete for the same agent as that of a. These activities may or may not belong to the same process instance as a. According to the assumption of FCFS and the nonpreemptive mode, these activities must be completed before activity a can start. With the max operator surrounding the set of the finish time of the identified activities, the finish time of the activity scheduled immediately before activity a is returned.

The queue duration is defined as the amount of time spent waiting for the required agent. The agent can start work on an activity only after receiving the notification. Therefore, the amount of time after receiving notification and before the finish time of the activity scheduled immediately preceding activity a is the queue duration. If the amount is negative, then the queue duration is zero because no time has been spent in waiting.

The procrastination duration of an activity is defined as the time spent waiting for a free agent to start the activity. An agent is free only when all the assigned activities have been completed. Therefore, procrastination occurs in two situations. The first situation occurs when the agent is notified of activity a before he or she has completed the previous activity. In this situation, the procrastination duration is the difference between the

finish time of the activity scheduled immediately preceding activity a and the start time of activity a. The second situation occurs when the agent is free when the notification arrives. In this situation, the procrastination duration is the difference between the notification time and the start time.

Example 2

According to Table 2,

$D_Q(L_3, a_1)$ = *max (0, max (*$L_2.T_F(a_2,g_3)$*,*$L_{50}.T_F(a_1,g_3)$*) –* $L_3.T_N\ (a_1, g_3)$*)*

= *max (0, max (14:00, 13:00) –13:00)*

= *max (0, 60) = 60*

$D_Q(L_4, a_1)$ = *max (0, max (*$L_1.T_F(a_1,\ g_1)$*,* $L_1.T_F(a_2,\ g_1)$*) –*$L_4.T_N(a_1, g_1)$*)*

= *max (0, max (8:30, 10:30) – 14:40)*

= *max (0, –250) = 0*

$D_P(L_5, a_1)$ = $L_5.T_S(a_1,g_5)$ *–max* $(L_4.T_F\ (a_2,g_5),\ L_5.T_N(a_1,g_5))$

= *18:20 – max (16:05, 16:10)*

= *18:20 – 16:10 = 130*

3.3 Identifying irregular process instances

This study adopted the *k*-NN algorithm [24] to identify irregular process instances using the four durations. The *k*-NN algorithm is used to identify the distance of the k^{th} nearest neighbor for each instance. Instances that are exceptionally distant from their neighbors were considered irregular. This study applied Euclidean distance as the distance metric.

Definition 4

Given two process instances, L_m *and* L_{n} and $L_m.A = L_n.A$,

a. The distance of L_m and L_{n} at a is defined as

$$\text{Dist}(L_{\text{m}}, L_{\text{n}}, a)=\left(D_{\text{E}}(L_{\text{m}}, a)\text{-}D_{\text{E}}(L_{\text{n}}, a)\right)^2 + (D_{\text{T}}(L_{\text{m}}, a)\text{-}\text{D}_{\text{T}}(L_{\text{n}}, a))^{\ 2} +$$

$$(D_Q(L_m, a) - D_Q(L_n, a))^{\ 2} + (D_{\text{P}}(L_m, a)\text{-}D_{\text{P}}(L_n, a))^{\ 2}.$$

b. The duration distance (DISTANCE) of L_m and L_n is defined as

$$\text{DISTANCE}(L_m, L_{\text{n}}) = \sum_{\forall a \in L_m.A \cap L_n.A} \sqrt{\text{Dist}(L_m, L_n, a)}.$$

After calculation of the DISTANCE between each pair of instances that share the same activities, the

distance of the *k*-NN for each instance can be obtained. The instances with values located outside of n standard deviations to the mean are defined as irregular. Fig. 2 presents the algorithm of the *ODAL*.

Input: LD /* a set of process instances */
k /* the kth nearest neighbor */
n /* the nth standard deviation */
Output: Irregular_Instances
{
Irregular_instacnes = ϕ
For each $L \in LD$ {
Neighbor(L) = $\{L_m | L_m \in LD,\ L.A = L_m.A\}$
Neighbor$_k$(L)=arg$_{L_m \epsilon \text{Neighbor(L)}}$ kth DISTANCE(L, L_m)
$D = \{\text{DISTANCE}(L, L_m) | L_m \in \text{Neighbor}(L)\}$
If DISTANCE(L, Neighbor$_k$(L)) $> mean(D) + n * \sigma(D)$ then
Irregular_instances += L
}
Return Irregular_instances
}

Fig. 2. Algorithm of the *ODAL*.

Because business processes include various branching paths, not all activities are executed in all paths. Comparison among instances executing different activities can produce misleading results. Therefore, the *ODAL* associates each instance with instances that execute exactly the same activities as those of the neighbors before identifying the k^{th} nearest neighbor. In this algorithm, the mean and standard deviation are that of the distances among neighbors.

4. Estimating activity-level durations using contextual information

In the aforementioned discussion, instances with the same activities were assumed to have similar expected durations. However, such an assumption may not be entirely practical because the expected durations may differ under dissimilar circumstances. For example, compared with skilled agents, new agents are expected to require more time to process an order because new agents are unfamiliar with the processes and systems. Therefore, notifying managers of the new agents' requirement of extra time for processing an order is

irrelevant when the time spent by the new agents is not substantially longer than that spent by skilled agents. A similar situation occurs when new orders are processed from walk-in customers. The time required is expected to exceed that required for returning customers because the agents and new customers must negotiate specifications and become acquainted with each other's communication styles. Similarly, notifying managers of such situations is pointless and not beneficial to the organization. The contextual information includes skills of sales representatives and the frequency of customer visits.

According to the aforementioned observations, the distinct characteristic of contextual information was incorporated into the data model. The durations of instances were compared with and contrasted to the durations of instances sharing similar characteristics.

4.1 Contextual variables and related fuzzy functions

The instances can be categorized on the basis of several attributes, such as the skill level of involved agents and customer visit frequency; each attribute is described using a contextual variable.

Linguistic fuzzy sets represent certain values of the contextual variables. For example, the skill level of an agent might be described as Novice, Skilled, and Adept.

Definition 5

a. A linguistic fuzzy set z in the universe of discourse X, the domain of a contextual variable, is characterized by a membership function $\mu_z(x)$, which associates each element $x \in X$ with a real number in the interval [0, 1].

b. A fuzzy set Z is referred to as a triangular fuzzy number with a peak (or center) c, a left width $\alpha > 0$, and a right width $\beta > 0$, if its membership function $\mu_Z(x)$ is defined as

$$\mu_Z(x) = \begin{cases} \frac{\alpha - (c - x)}{\alpha} & if\ c - \alpha \le x \le c \\ \frac{\beta - (x - c)}{\beta} & if\ c \le x \le c + \beta \\ 0 & otherwise. \end{cases}$$

To determine α and β for each fuzzy set, the universe of discourse must be divided for each contextual variable. The most commonly adopted discretization methods are the equal-width and equal-frequency methods [39]. In the equal-width method, data within a target range is grouped into a predefined number of intervals, whereas in the equal-frequency method, all intervals contain the same number of data points. For a skewed data distribution, because the data points are more evenly distributed in the equal-frequency

discretization method, it is more suitable than the equal-width discretization method. Fig. 3 presents the distribution of the number of transactions conducted between a sample company and its customers.

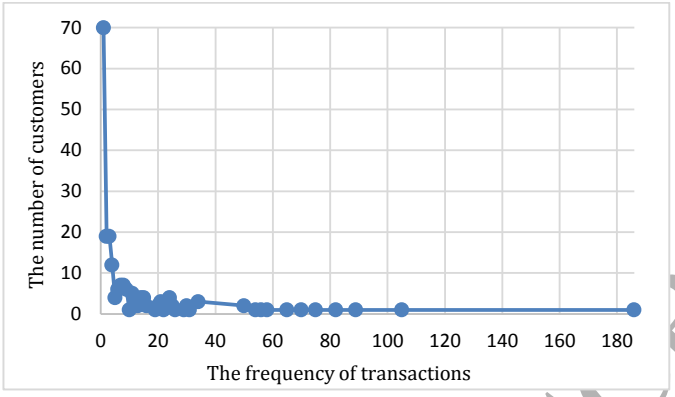

Fig. 3. Data distribution of customer transaction frequency.

The data distribution is extremely skewed. The number of transactions conducted between the company and its customers range from 1 to 186 within 1 year. Because most contextual data is skewed toward one end of the universe of discourse, the equal-frequency method is adopted to perform discretization. However, the proposed method can be applied to the linguistic fuzzy sets derived using the equal-width method.

The center points of the three intervals of data shown in Fig. 4 are 1, 30, and 186. According to the three center points, three fuzzy sets are formed with (c, α, β) of (1, 0, 30), (30, 3, 186), and (186, 30, 0).

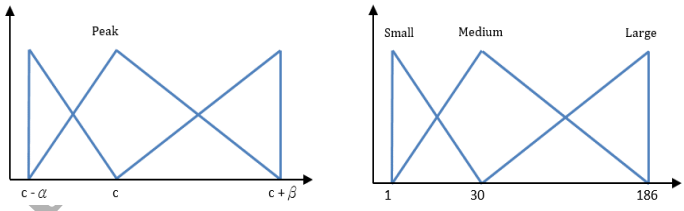

Fig. 4. Degree of membership function $\mu(x)$ of customer transaction frequency.

When more than one fuzzy variable is used to describe contextual information, a joined fuzzy set is applied, which is the combination of individual linguistic fuzzy sets, in which each set represents a value for a contextual variable. In other words, the joined fuzzy set is the co-occurrence of multiple events, each represented by a linguistic fuzzy set. Because most events are mutually independent, the membership function of a joined fuzzy set is defined as the product of all related linguistic fuzzy sets.

Definition 6

Assume z_i as a fuzzy set in the universe of discourse X_i and $x_i \in X_i$; the joined membership function of the fuzzy set $(z_1 \ldots z_m)$ is

$$\mu_{z_1 \ldots z_m}(x_1, \ldots, x_m) = \prod_{i=1}^{m} \mu_{z_i}(x_i).$$

4.2 Estimating durations using joined fuzzy sets

DEFINITION 7

Assume $X_1, \ldots, X_m$ as the universes of discourse for fuzzy sets $z_1, \ldots, z_m$, and $\bar{x} = (x_1, \ldots x_m)$, where $x_i \in X_i$.

a. A context annotated process instance $L^C = < A, E, T_N, T_S, T_F, \bar{x} >$.

b. Assume $D(L^C, a)$ as any one of the four durations defined in Section 3 and LD^C as a set of context annotated process instances. Thus, the expected durations for activity a under the contextual fuzzy sets $z_1 \ldots z_m$ are

$$ED_{z_1 \ldots z_m} \quad (a) = \frac{\sum_{L^C \in LD^C} D \quad (L^C, a) * \mu_{z_1 \ldots z_m}(L^C.\bar{x})}{\sum_{L^C \in LD^C} \mu_{z_1 \ldots z_m}(L^C.\bar{x})}.$$

c. Assume C_i as a set of fuzzy sets defined in the universe of discourse X_i; the fuzzy adjusted duration for $D(L^C, a)$ is

$$FD\left(L^C, a\right) = \sum_{z_1 \in C_1 \ldots z_m \in C_m,} \mu_{z_1 \ldots z_m}\left(L^C.\bar{x}\right) * ED_{z_1 \ldots z_m}(a).$$

The contextual *ODAL* (*CODAL*) is the same as *ODAL* except that the duration functions in Dist are replaced with fuzzy adjusted duration functions.

5. Case application

To verify whether the irregular instances identified using the proposed algorithm were abnormal, this study applied the data and process execution logs of an ERP system operating in a medium-sized manufacturing and logistics services company in the stone industry.

The 30-year old company designs, manufactures, imports and exports more than 100 types of natural stone products, such as engravings and sculptures, lanterns, fountains and cascades, benches and tables, and landscape and garden products. More than 50 materials, including various types of marble, granite, limestone,

sandstone, and quartzite, are used to produce the finished products. It receives orders from around the world and purchases materials from rock mines across China. After being processed in factories, the finished products are exported to Germany, Spain, USA, Turkey, Japan, Korea, and other countries. Headquartered in Taiwan, the company has subsidiaries in mainland China, Japan, and Germany; the number of transactions completed annually may not be high but the company has considerably high value in dollars. The lead time to process the transactions can be lengthy and occasionally require nearly a year to complete. After receiving client orders, the company confirms, modifies, and reconfirms the blueprint and then disassembles it into construction layouts that involve thousands or even tens of thousands of building stones.

The case company's ERP software was specifically developed for the construction industry with the aim to integrate its project seamlessly with subcontracting, finance, and client management system. The data flows directly from the valuation, specification confirmation, selection, price negotiation, order changing, budgeting, job scheduling, and warranty steps. Because it is a Web-based platform, this system can be accessed anywhere with a computer or mobile device. The log information used in this study contains ERP system's three modules, namely sales and distribution (SD), material management (MM), and production planning (PP). The data extracted from this company's transaction logs included process id, instance id, activity id, and the corresponding notification, start, and finish times; the information related to customer transactions, sales, and product managers was obtained from sales and production modules. The log level information was used to calculate durations, whereas the transaction and master-data level information was used as values for contextual variables. The study was based on the following assumptions: (1) all processes were successfully completed and (2) had no loop in the processes. The instances violating the assumption were deleted. Consequently, the data set comprised 2,169 process instances between January 2013 and December 2013. The outliers nominated using the algorithm were verified by five domain experts who, on average, had worked in the company for 14 years. This approach was similar to that adopted by Angiulli and Fassetti [4].

The processes comprised five main steps: sales order dispatching, factory capacity planning, manufacturing, production outsourcing, and inspection and shipping. Fig. 5 illustrates the business processes through event action chain notation [44, 50]. After receiving customer orders, the company identifies suppliers, imports raw materials (e.g., granite and marble blocks) from suppliers (e.g., quarry miners in various countries), and transforms the raw materials into end products, such as granite tiles by cutting blocks into slabs and polishing the slabs. When the factory capacity is not sufficient for processing customer orders, some of the

orders are outsourced to other partners. In the final stage, the end products are shipped to ports and transported to customer construction sites.

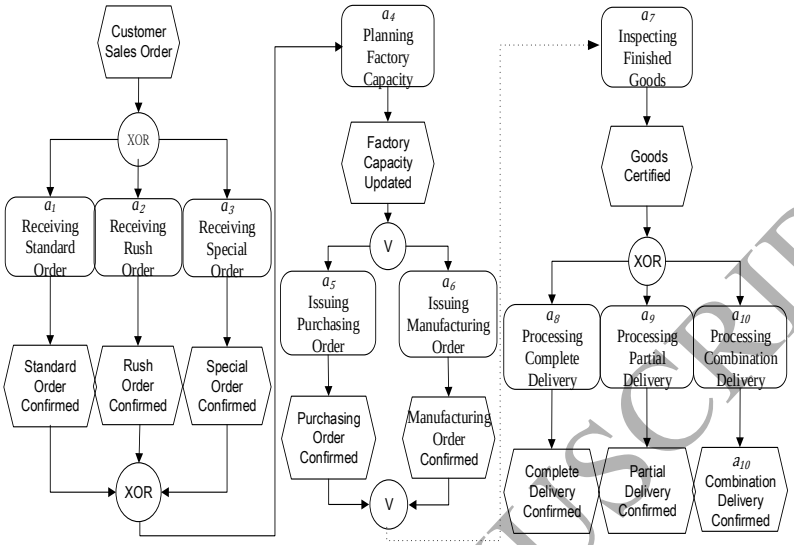

Fig. 5. Administrative process from order receipt to delivery.

1. Receiving orders (a_1–a_3): Orders are classified into Standard, Rush, and Special. When an order arrives, a sales agent receives the order and processes it according to the order type. The agent negotiates prices, checks the warehouse for raw materials, and examines possible suppliers for materials that must be purchased. Special orders are fulfilled by outsourcing to other companies.

2. Planning factory capacity (a_4): Standard and rush orders are fulfilled by the company or outsourced to other suppliers, depending on the company's manufacturing capacity. The availability and capacity of manufacturing lines are summarized and examined using ERP systems. Thereafter, a production plan is developed and executed according to the available capacity.

3. Issuing outsourcing and manufacturing orders (a_5–a_6): For outsourced orders, the company confirms the product price, quantity, delivery time, and quality requirements with suppliers. For in-house production, the agents issue a series of working orders and allocate work center schedules and the arrival times of raw materials and semifinished goods to each work center, thereby fully utilizing the planned capacity.

4. Inspection (a_7): Regardless of how the finished goods are manufactured, the goods must undergo quantity and quality inspection before being shipped to customers.

5. Shipping arrangement (a_8–a_{10}): Containers loaded with products are shipped to ports according to shipping schedules. The shipments can be classified into complete (a_8), partial (a_9), and combination (a_{10}) shipments. For a complete shipment, all goods purchased in a sales order are shipped in a single load. For multiple shipments, goods are separated into several loads, whereas for combination shipments, goods from several orders are combined into one load.

Fig. 5 presents the administrative process. The dotted line between a_5/a_6 and a_7 indicates that the actual manufacturing activities are omitted because the characteristics of such activities differ markedly from those of the administrative process and require distinct sets of contextual variables.

Table 3 displays the average durations of the ten activities. The transmission durations of a_1, a_2, and a_3 are zero because they are the starting activities that do not wait for any documents or materials. The transmission time of a_5 and a_6 is also zero because the ERP system automatically notified sales and production managers of the new working orders.

Table 3
Average durations (hours) of the ten activities.

	Activity	D_E	D_T	D_Q	D_P
a_1	Receiving Standard Order	35	0	65	72
a_2	Receiving Rush Order	38	0	59	68
a_3	Receiving Special Order	34	0	75	83
a_4	Planning Factory Capacity	83	43	59	34
a_5	Issuing Purchasing Order	67	0	15	16
a_6	Issuing Manufacturing Order	57	0	13	14
a_7	Inspecting Finished Goods	95	61	25	44
a_8	Processing Complete Delivery	24	16	51	65
a_9	Processing Partial Delivery	29	16	59	73
a_{10}	Processing Combination Delivery	22	16	47	62

5.1 Contextual variables

Three contextual variables were used in this study. The skills of sales agents and production managers and the ordering frequencies of customers all substantially affected process durations. Skilled agents can complete tasks considerably faster than can unskilled agents. Customers who have frequently placed orders can be matched with suppliers who have served them before, and thus, the order dispatching and manufacturing durations can be reduced.

Agent skill and transaction history were represented as fuzzy sets. Each variable was valued with three fuzzy sets. For skills, the fuzzy values were Novice, Skilled, and Adept. For customer transaction frequency, the fuzzy values were Irregular, Medium, and Frequent. Skill was determined by the number of orders processed in 1 year (2–329 orders). Customer visiting frequency was determined according to the number of orders placed by each customer (1–186 orders). In total, 31 sales agents, 17 production managers, and 203 customers were investigated in this study. Table 4 presents the fuzzy values.

Table 4
Fuzzy values corresponding to linguistic fuzzy sets.

Linguistic fuzzy sets	Contextual variables		
	Customer Visiting Frequency(CVF)	Sales Agent Skill (SAS)	Production Manager Skill (PMS)
Irregular / Novice	{ 0, 1, 30}	{ 0, 2, 234}	{ 0, 1, 129}
Medium / Skilled	{ 1, 30, 186}	{ 2, 234, 329}	{ 1, 129, 334}
Frequent / Adept	{ 30, 186, 0 }	{ 234, 329, 0}	{ 129, 334, 0}

As shown in Table 5, Customer Visiting Frequency and Sales Agent Skill were used as contextual variables from a_1 to a_3; Agent Skill and Production Manager Skill were used as the contextual variables from a_4 to a_7; and Customer Visiting Frequency and Production Manager Skill were used as the contextual variables from a_8 to a_{10}.

Table 5
Contextual variables for each type of activity.

	Activity	Customer Visiting Frequency (CVF)	Sales Agent Skill (SAS)	Production Manager Skill (PMS)
a_1	Receiving Standard Order	✔	✔	
a_2	Receiving Rush Order	✔	✔	
a_3	Receiving Special Order	✔	✔	
a_4	Planning Factory Capacity		✔	✔
a_5	Issuing Purchasing Order		✔	✔
a_6	Issuing Manufacturing Order		✔	✔
a_7	Inspecting Finished Goods		✔	✔
a_8	Processing Complete Delivery	✔		✔
a_9	Processing Partial Delivery	✔		✔
a_{10}	Processing Combination Delivery	✔		✔

5.2 Evaluation metrics

Five experts were consulted to verify the instances identified as irregular. The instances confirmed and refuted by experts to be irregular were considered as "abnormal" and "false alarms," respectively.

To measure and compare the effectiveness of different approaches, two metrics were employed: the Discovery Rate (DR) and Hit Rate (HR).

The DR(method) represents the ratio of irregularities identified by a particular method and is defined as

$$\mathrm{DR(method)}=\frac{|\text{confirmed abnormal instances identified by the method}|}{|\text{confirmed abnormal instances }|}. \quad (1)$$

Because requesting experts to review all process instances in advance was impractical, a confirmed abnormal instance was considered to be irregular when it was identified as using at least one method described in the experimental section and verified by experts to be abnormal.

The HR represents the percentage of identified irregularities confirmed by experts and is defined as

$$\mathrm{HR(method)}=\frac{|\text{confirmed abnormal instances identified by the method}|}{|\text{ instances nominated as irregular by the method}|}. \quad (2)$$

5.3 Experimental results

In addition to applying the *ODAL* and *CODAL*, this study included a common conventional method, the LOF, as a benchmark. The LOF is one of the algorithms used to identify outliers and has been applied to detect process anomalies [7, 11, 24, 30]. For example, Kang et al. [24] and Ma et al. [30] have adopted the LOF to detect abnormal process termination.

In the study, the LOF was used to predict whether an instance was terminated in a reasonable amount of time; in other words, the LOF was used to compute the degree of abnormality of instances based on durations between the notification time of the first activity and the finish time of the last activity. For each process instance, the LOF algorithm computes the degree to which the instance is an anomaly, called the *LOF*, according to how isolated the instance is compared with surrounding instances. A higher LOF value indicates that the instance is in an area with fewer surrounding instances and, therefore, is more likely to be an outlier instance. The LOF defined in this section is a revision of that described by Breunig et al. [11] and Kang et al. [24].

The LOF value for an instance L_q is computed as follows.

1. For each pair of instance $L_i, L_j \in LD$, compute the duration difference in instance level $d(L_i, L_j)$.
2. For each instance $L_i \in LD$, compute the distance to the k^{th} nearest neighbor of L_i, $\mathrm{d}_k(L_i)$.
3. For each instance $L_i, L_i \in LD$, compute a reachability distance from L_q, as $\mathrm{rd}_k(L_q, L_i) = \max\{d(L_q, L_i), d_k(L_i)\}$.
4. $N_k(L_q)$ is a set of k nearest neighbors of L_q, and the local reachability density of L_q is defined as follows:

$$lrd_k(L_q) = 1/\left(\frac{\sum_{L_i \in N_k(L_q)} \mathrm{rd}_k(L_q, L_i)}{k}\right).$$

5. Compute the LOF of L_q, which is defined as the average of the ratio of the lrd_k of L_q to that of its k nearest neighbors, as follows:

$$LOF\ \ (L_q) = \frac{\sum_{L_i \in N_k(L_q)} \frac{lrd_k(L_i)}{lrd_k(L_q)}}{k}.$$

For a normal instance, the local density should be similar to that of surrounding neighbors. Consequently, the value of the LOF is close to 1. However, for an outlier instance, which is distant from normal instances, the

local density is smaller than that of other instances, causing the LOF to exceed 1. Therefore, the greater the LOF value is, the greater is the likelihood of the instance being abnormal.

Ahmed et at. [1] examined, compared, and used synthetic data to verify various outlier detection methods that attempted to identify financial fraudulent activities; they believed that no comprehensive techniques exist to spot anomaly conditions. In addition to utilizing traditional *LOF* to detect outliers, the experiment included a revised *LOF* based on kd-trees (*KDT*) to organize points and simplify the selection of neighbor points for detecting outliers [25]. For comparison, the outlier detection based on execution intervals of the outlier detection with statistics (*ODS*) [15, 45] was also included. The *ODS* utilizes the entire duration to detect outlier processes and is very different from the proposed methods. In summary, all three of the methods consider the entire duration of process execution. However, only the *CODAL* and *ODAL* proposed in this study explored activity level duration.

All methods—the *ODAL*, *CODAL*, *ODS*, *KDT*, and *LOF*—require threshold values for outlier identification. Hawkins and Douglas [19] proposed separating outliers into two types, namely, mild and extreme outliers. Mild outliers are those whose distance to the centroid is between $\mu+0.6745\sigma$ and $\mu+2.698\sigma$, and extreme outliers are those whose distance to the centroid exceeds $\mu+2.698\sigma$, where μ is the average distance and σ is the standard deviation. In this study, considering the available time of the experts, only extreme outliers were presented for confirmation.

The examined threshold values were $\mu+2.5\sigma$, $\mu+2.698\sigma$, and $\mu+3\sigma$. Moreover, all five methods required setting the number of nearest neighbors (i.e., k) to decide the area of comparisons. The most typical method for selecting k is to evaluate the log-likelihood of the data according to a study by Terzakis [42], who proposed estimating k using the formula $\log(|\mathrm{LD}|)$ [48]. In this study, because the database contained 2,169 records and $\log(2{,}169) \doteq 3$, the examined values of k were set as 2–5.

Figs. 6, 8, and 10 present the HRs of the five methods estimated using different k, namely 2–5. The threshold values were $\mu+2.5\sigma$, $\mu+2.698\sigma$, and $\mu+3\sigma$, respectively. The figures revealed that the LOF-based methods, the *LOF* and *KDT*, performed the least favorably under all settings; the average HR of the *LOF* was 10% and that of the *KDT* was 22%. By contrast, the *CODAL* performed the most favorably, except when $k = 5$ and the threshold value was $\mu+2.5\sigma$ and $\mu+3\sigma$. In optimal situations, the HR of the *CODAL* can exceed 80% when $k = 3$ with threshold values $\mu+2.698\sigma$ and $\mu+3\sigma$. As Table 6 presents, the average HR of the *CODAL* was 26% higher than that of the *ODAL*, 56% higher than that of the *ODS*, 205% higher than that of the *KDT*, and 570% higher than that of the *LOF*.

Figs. 7, 9, and 11 present the DRs of the five methods under the same settings of k and thresholds. The figures again confirm that the *LOF* method performed the least favorably, with an average DR of 15%. As Table 6 illustrates, the *CODAL* again outperformed the other methods in all settings, with an average DR of 50%, which is 61% higher than that of the *ODAL*, 79% higher than that of the *ODS*, 52% higher than that of the *KDT*, and 233% higher than that of the *LOF*. The optimal DR of the *CODAL* can reach 63% when k=5 and the threshold values are $\mu+2.5\sigma$.

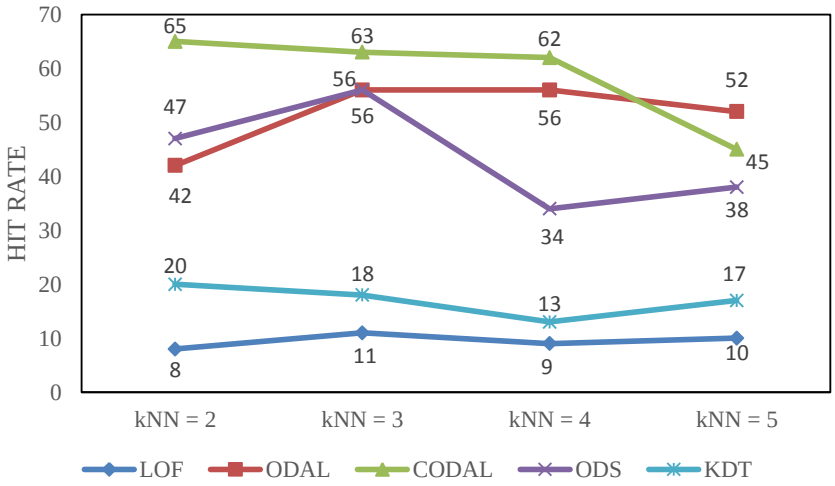

Fig. 6. HRs with threshold = μ + 2.5σ.

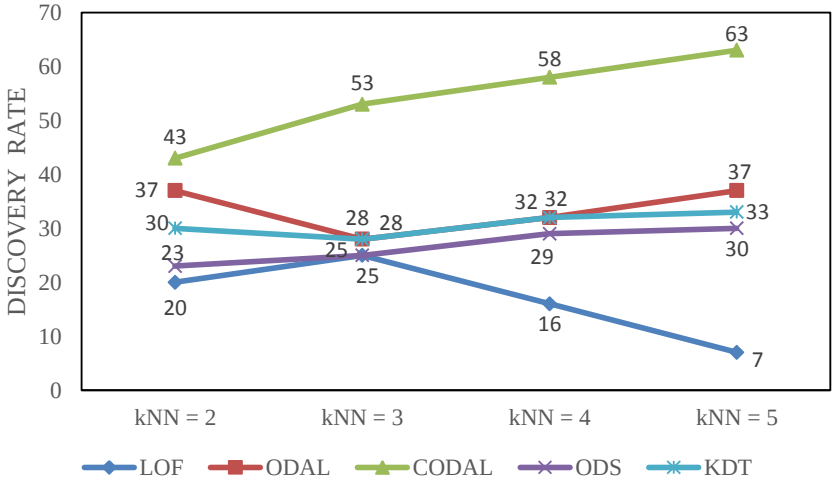

Fig. 7. DRs with threshold = μ + 2.5σ.

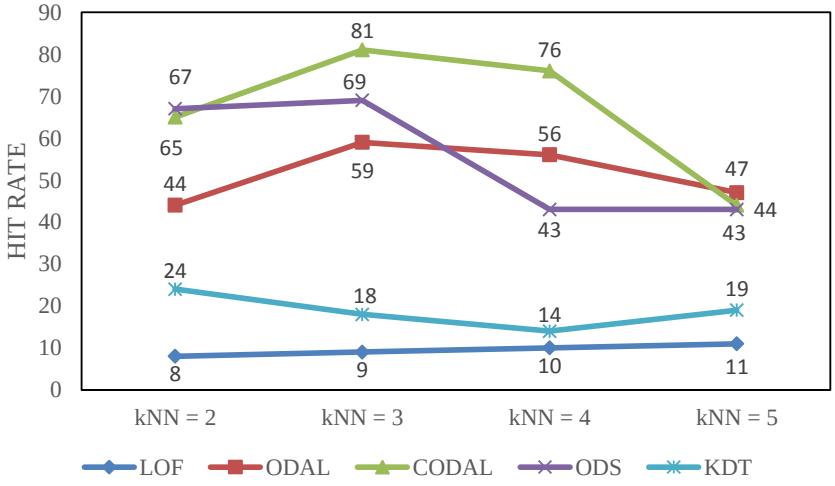

Fig. 8. HRs with threshold = $\mu + 2.698\sigma$.

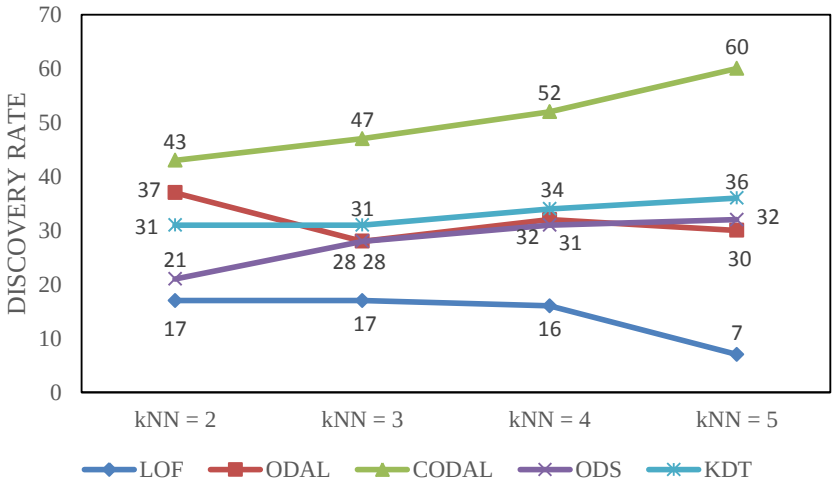

Fig. 9. DRs with threshold = $\mu + 2.698\sigma$.

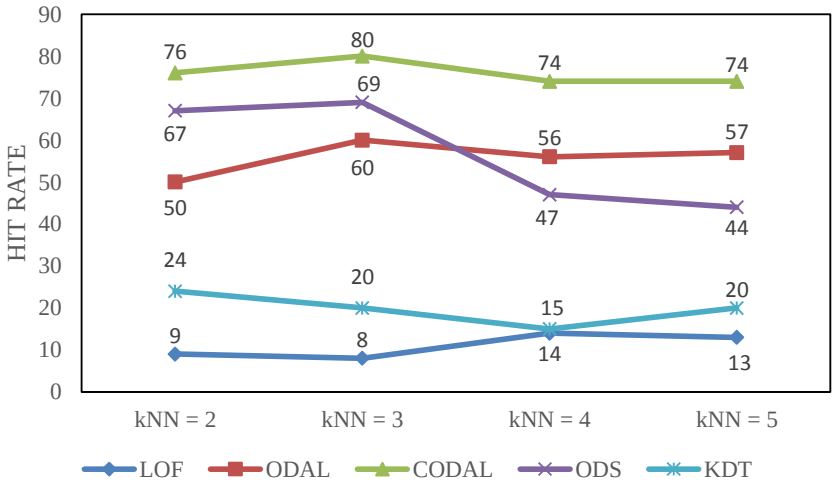

Fig. 10. HRs with threshold = $\mu + 3\sigma$.

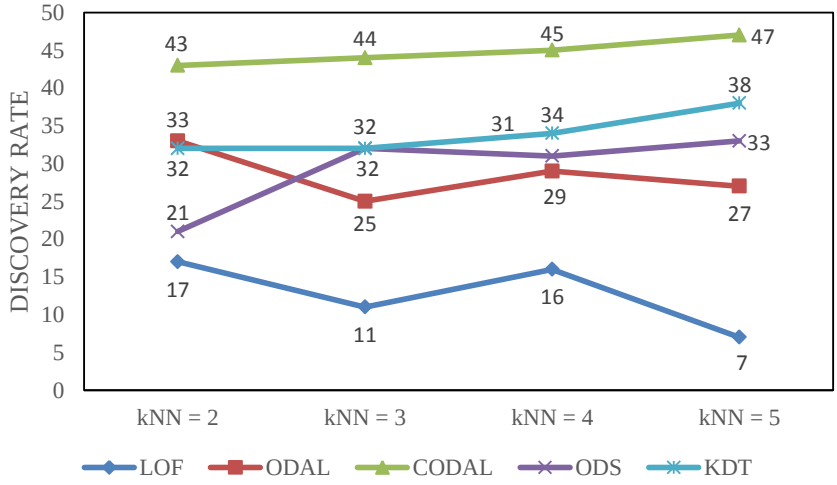

Fig. 11. DRs with threshold = $\mu + 3\sigma$.

Table 6
The average HR and DR for each approach.

	Average	
	HR (%)	DR (%)
CODAL	67	50
ODAL	53	31
ODS	43	28
KDT	22	33
LOF	10	15

Because the proposed methods tend to perform adequately when k =3, Figs. 12 and 13 present the HRs and DRs of the five methods when k=3 and under different threshold values, namely $\mu+2.5\sigma$, $\mu+2.698\sigma$, and

$\mu+3\sigma$. In optimal situations, the HR and DR of the *CODAL* can reach 81% and 53%, respectively.

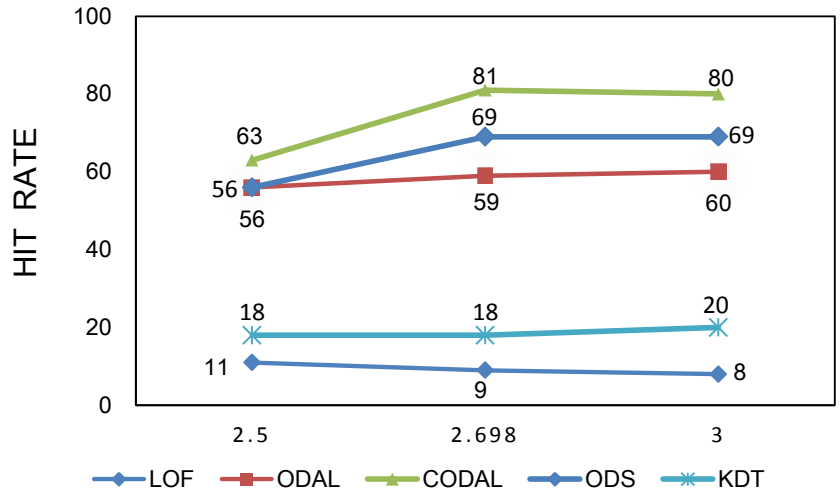

Fig. 12. HRs with *k*-NN = 3.

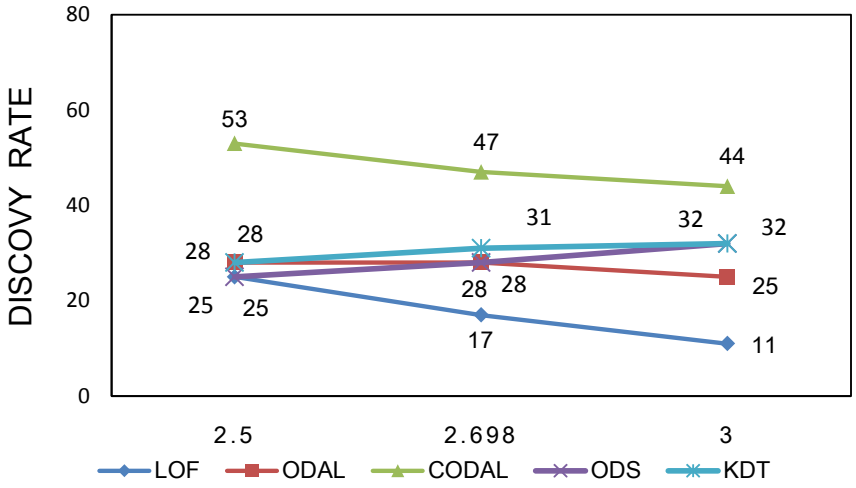

Fig. 13. DRs with *k*-NN = 3.

In summary, the *CODAL* and *ODAL* outperformed the other three competing methods and in the majority of cases, the *CODAL* outperformed the *ODAL*.

6. Conclusion and future research directions

To maintain competitiveness in uncertain and changing business environments, an organization must adapt and respond quickly to changes. Consequently, organizations must continuously improve their business processes to achieve operational excellence.

One crucial area for process improvement is time management, which can provide the competitive

advantages of agility and flexibility. Irregular execution durations in business processes can reduce an organization's agility. Ideally, managers should focus particularly on instances exhibiting irregular execution durations. However, as most managers can attest, daily workloads consume most managerial time and energy. Thus, they have no time to monitor business instances and identify irregularities.

Even when automation tools are used, according to the experimental section, the conventional method for identifying irregular instances by only measuring the entire instance execution time is ineffective. Thus, this paper proposes identifying business process irregularities by segmenting time intervals into four durations at the activity level. The durations include execution, transmission, queue, and procrastination durations. The *ODAL* identifies irregularities using these activity-level durations, whereas the *CODAL* integrates these durations with fuzzy-annotated contextual information for detection.

To verify the effectiveness of the proposed approaches, a data set was collected from a medium-sized logistics company; 2,169 process instances were collected and 36 of such instances were confirmed to be problematic by experts. Among the five methods compared in this study, the *CODAL* exhibited the most favorable performance and reached 81% and 47% of HR and DR, respectively, when k = 3 and the threshold value was $\mu+2.698\sigma$. Under the same circumstances, the *ODAL* reached 59% and 28% of HR and DR, respectively. The *ODS* recorded 69% and 28%, respectively, and the *KDT* attained 18% and 31%, respectively. The *LOF* achieved only 9% and 17%, respectively. Overall, the performance of the *CODAL* was substantially higher than that of the *ODAL*, *ODS*, *KDT*, and *LOF*.

This study primarily applied temporal information to determine process outliers, whereas most previous studies have determined outliers using activity execution sequences [6, 10, 15, 24, 25, 45]. Future studies can combine both approaches to propose a comprehensive method for identifying process outliers using both irregular activity sequences and execution durations. If supply chain information is available, more contextual information, such as warehouse distance, transportation time, and quality of supplied parts, can be experimented with to discover outlier process instances between organizations in the supply chains.

7. References

[1] M. Ahmed, A.N. Mahmood, M.R. Islam, A survey of anomaly detection techniques in financial domain, Future Generation Computer Systems, 55 (2016) 278-288.

[2] G. Alonso, D. Agrawal, A. El Abbadi, C. Mohan, Functionality and limitations of current workflow management systems, IEEE Expert, 12(5) (1997) 105-111.

[3] Y. Alotaibi, F. Liu, A novel secure business process modeling approach and its impact on business performance, Information Sciences, 277 (2014) 375-395.

[4] F. Angiulli, F. Fassetti, Exploiting domain knowledge to detect outliers, Data Mining and Knowledge Discovery, 28(2) (2014) 519-568.

[5] H. Bae, S. Lee, I. Moon, Planning of business process execution in business process management environments, Information Sciences, 268 (2014) 357-369.

[6] F. Bezerra, J. Wainer, Anomaly detection algorithms in business process logs, In: Proceedings of the 10th International Conference on Enterprise Information Systems (ICEIS), volume AIDSS, Barcelona, Spain, 2008, pp. 11-18.

[7] V. Bhatt, K.G. Sharma, A. Ram, An enhanced approach for LOF in data mining, In: IEEE International Conference on Green High Performance Computing (ICGHPC), 2013, pp. 1-3.

[8] M.H. Bhuyan, D.K. Bhattacharyya, J.K. Kalita, A multi-step outlier-based anomaly detection approach to network-wide traffic, Information Sciences, 348 (2016) 243-271.

[9] M. Bierbaumer, J. Eder, H. Pichler, Calculation of delay times for workflows with fixed-date constraints, In: Proceedings of 7th IEEE International Conference on E-Commerce Technology, 2005, pp. 544-547.

[10] L. Bouarfa, J. Dankelman, Workflow mining and outlier detection from clinical activity logs, Journal of Biomedical Informatics, 45(6) (2012) 1185-1190.

[11] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, LOF: Identifying density-based local outliers, In: ACM sigmod record, 29(2) (2000) 93-104.

[12] G. Bruno, P. Garza, Temporal outlier detection by using quasi-functional temporal dependencies, Data & Knowledge Engineering, 69(6) (2010) 619-639.

[13] A. Burattin , M. Cimitile, F.M. Maggi, A. Sperduti, Online discovery of declarative process models from event streams, IEEE Transactions on Services Computing, 8(6) (2015) 833-846.

[14] Y. Chen, J.M. Kamara, A framework for using mobile computing for information management on construction sites, Automation in Construction, 20(7) (2011) 776-788.

[15] Y.-C. Chuang, P.-Y. Hsu, M.-T. Wang,S.-C. Chen, A frequency-based algorithm for workflow outlier mining, In: Future Generation Information Technology, Springer Berlin Heidelberg, 2010, pp.191-207.

[16] M. Comuzzi, I. Vanderfeesten, T. Wang, Optimized cross-organizational business process monitoring: Design and

enactment, Information Sciences, 244 (2013) 107-118.

[17] C. Di Ciccio, H. van der Aa, C. Cabanillas, J. Mendling, J. Prescher, Detecting flight trajectory anomalies and predicting diversions in freight transportation, Decision Support Systems (2016).

[18] F. Guo, Q. Lu, A novel contextual information recommendation model and its application in e-commerce customer satisfaction management, Discrete Dynamics in Nature and Society, 2015.

[19] D.M. Hawkins, M. Douglas, Identification of outliers, Chapman and Hall, London, 1980.

[20] Z.H. Hu, Z.H. Sheng, Disaster spread simulation and rescue time optimization in a resource network, Information Sciences, 298 (2015) 118-135.

[21] Z. Huang, X. Lu, H. Duan, W. Fan, Summarizing clinical pathways from event logs, Journal of Biomedical Informatics, 46(1) (2013) 111-127.

[22] V.R. Jakkula, A.S. Crandall, D.J. Cook, Enhancing anomaly detection using temporal pattern discovery, In: Advanced Intelligent Environments, Springer, US, 2009, pp. 175-194.

[23] X.H. Jiang, Z.X. Nie, Load-related completion time estimation for business process instances, Computer Integrated Manufacturing Systems, 17(8) (2013) 1640-1646.

[24] B. Kang, D. Kim, S.H. Kang, Real-time business process monitoring method for prediction of abnormal termination using KNNI-based LOF prediction, Expert Systems with Applications, 39(5) (2012) 6061-6068.

[25] S. Kim, N.W. Cho, Y.J. Lee, S.H. Kang, T. Kim, H. Hwang, D. Mun, Application of density-based outlier detection to database activity monitoring, Information Systems Frontiers, 15(1) (2010) 55-65.

[26] R.K.L. Ko, A computer scientist's introductory guide to business process management (BPM), Crossroads, 15(4) (2009) 11-18.

[27] H.D. Kuna, R. García-Martinez, F.R. Villatoro, Outlier detection in audit logs for application systems, Information Systems, 44 (2014) 22-33.

[28] S.-K. Lee, B. Kim, M. Huh, S. Cho, S. Park, D. Lee, Mining transportation logs for understanding the after-assembly block manufacturing process in the shipbuilding industry, Expert Systems with Applications, 40(1) (2013) 83-95.

[29] W. Li, Y. Fan, A time management method in workflow management system, In: IEEE Workshops at the Grid and Pervasive Computing Conference (GPC'09), 2009, pp. 3-10.

[30] Y. Ma, H. Shi, H. Ma, M. Wang, Dynamic process monitoring using adaptive local outlier factor, Chemometrics and Intelligent Laboratory Systems, 127 (2013) 89-101.

[31] R.S. Mans, N.C. Russell, W. van der Aalst, P.J. Bakker, A.J. Moleman, Simulation to analyze the impact of a

schedule-aware workflow management system, Simulation, 86(8-9) (2010) 519-541.

[32] D. Martínez-Rego, D. Fernández-Francos, O. Fontenla-Romero, A. Alonso-Betanzos, Stream change detection via passive-aggressive classification and Bernoulli CUSUM, Information Sciences, 305 (2015) 130-145.

[33] S.B. Needleman, C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, Journal of molecular biology, 48(3) (1970) 443-453.

[34] A. Pika, , W.M.P. van der Aalst, M.T. Wynn, C.J. Fidge, A.H.M. Ter Hofstede, Evaluating and predicting overall process risk using event logs, Information Sciences, 352-353 (2016) 98-120.

[35] A. Pluta, G.P. Wójcik, Research on time–management skills of employees in the process of creating value for the customer, International Journal of Business Performance Management, 16(2-3) (2015) 246-261.

[36] S. Roy, A.S.M. Sajeev, S. Bihary, A. Ranjan, An empirical study of error patterns in industrial business process models, IEEE Transactions on Services Computing, 7(2) (2014) 140-153.

[37] R. Sarno, B.A. Sanjoyo, I. Mukhlash, H.M. Astuti, Petri net model of ERP business process variation for small and medium enterprises, Journal of Theoretical & Applied Information Technology, 54(1) (2013) 31-38.

[38] W.J. Sawaya, S. Pathak, J.M. Day, M.M. Kristal, Sensing abnormal resource flow using adaptive limit process charts in a complex supply network, Decision Sciences, 46(5) (2015) 961-979.

[39] O. Shafiq, R. Alhajj, J. Rokne, Log based business process engineering using fuzzy web service discovery, Knowledge-Based Systems, 60 (2014) 1-9.

[40] K. Singh, P.J. Best, Design and implementation of continuous monitoring and auditing in SAP enterprise resource planning, International Journal of Auditing, 19(3) (2015) 307-317.

[41] S.X. Sun, J.L. Zhao, Formal workflow design analytics using data flow modeling, Decision Support Systems, 55(1) (2013) 270-283.

[42] G. Terzakis, How can we find the k in kNN. http://www.researchgate.net/, 2014 (accessed July 2015).

[43] W.M.P. van der Aalst, Business process management as the "Killer App" for Petri nets, Software & Systems Modeling, 14(2) (2015) 685-691.

[44] W.M.P. van der Aalst, Formalization and verification of event-driven process chains, Information and Software Technology, 41(10) (1999) 639-650.

[45] M.-T. Wang, P.-Y. Hsu, Y.-C. Chuang, Mining workflow outlier with a frequency-based algorithm, International Journal of Control and Automation, 4 (2) (2011) 1-22.

[46] M. Werner, N. Gehrke, Multilevel Process Mining for Financial Audits, IEEE Transactions on Services

Computing, 8(6) (2015) 820-832.

[47] D. Xu, C. Wijesooriya, Y.G. Wang, G. Beydoun, Outbound logistics exception monitoring: A multi-perspective ontologies' approach with intelligent agents, Expert Systems with Applications, 38(11) (2011) 13604-13611.

[48] H. Yildiz, J. Yoon, S. Talluri, W. Ho, Reliable supply chain network design, Decision Sciences, 2015.

[49] W.Y. Yu, C.G. Yan, Z.J. Ding, C.J. Jiang, M.C. Zhou, Modeling and validating e-commerce business process based on petri nets, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 44(3) (2014) 327-341.

[50] H. Zhuge, T. Cheung, H.K. Pung, A timed workflow process model, Journal of Systems and Software, 55(3) (2001) 231-243.

Ping-Yu Hsu is a professor of Business Administration at National Central University in Jhongli, Taiwan. Hsu received his PhD in Computer Science from University of California Los Angeles, USA. He is also Secretary-in-Chief of the Chinese ERP Association. His research interests are business data applications, including data modeling, data warehousing, data mining, and ERP applications in business domains. His papers have been published in IEEE Transactions on Software Engineering, Information Systems, Information Sciences, Computers and Operations Research, and various other journals.
pyhsu@mgt.ncu.edu.tw

Yu-Cheng Chuang received his master's degree from the Computer Science and Information Engineering Department at Fu Jen Catholic University in 2000. Chuang received his PhD in Business Administration from National Central University, Taiwan. He is also an ERP consultant. His research interests are business data applications, including data mining, process mining, and ERP applications in business domains.
tocasper@hotmail.com

Yao-Chung Lo received his master's degree from the Department of Business Administration of National ChengChi University, Taiwan. He is a doctoral student in the Business Administration Department at National Central University, Taiwan. His research interests focus on the developments of Web, including the trends of social networking.
andrewlo0623@gmail.com

Shuang-Chuan He received his master's degree from the Business Administration Institute of National Central University in 2010. He is an engineer in the ERP Business Application Department. His research interests are data mining, process mining, and ERP applications in business domains.
974201058@cc.ncu.edu.tw