J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

Discovering Temporal Work Transference Networks from Workflow Execution Logs

Dinh-Lam Pham[1] Hyun Ahn[1] Kwanghoon Pio Kim[1*]

ABSTRACT

Workflow management systems (WfMSs) automate and manage workflows, which are implementations of organizational processes operated in process-centric organizations. In this paper, wepropose an algorithm to discover temporal work transference networks from workflow execution logs. The temporal work transference network is a special type of enterprise social networks that consists of workflow performers, and relationships among them that are formed by work transferences between performers who are responsible in performing precedent and succeeding activities in a workflow process. In terms of analysis, the temporal work transference network is an analytical property that has significant value to be analyzed to discover organizational knowledge for human resource management and related decision-making steps for process-centric organizations. Also, the beginning point of implementinga human-centered workflow intelligence framework dealing with work transference networks is to develop an algorithm for discovering temporal work transference cases on workflow execution logs. To this end, we first formalize a concept of temporal work transference network, and next, we present a discovery algorithm which is for the construction of temporal work transference network from workflow execution logs. Then, as a verification of the proposed algorithm, we apply the algorithm to an XES-formatted log dataset that was released by the process mining research group and finally summarize the discovery result.

☞ keyword : Workflow management system, work transference network, execution logs, temporal workcases

1. Introduction

Workflow is a sequence of steps, tasks, events, and those interactions that implement an organizational process for performing repetitive and standardized procedural work in an organization. Such organizational processes may contain many parts and individuals involved. For example, a simple online shopping workflow involves the following steps: First, a customer looks for the product and price, and if it is satisfied, then decide to buy the product. The customer fills out personal information into an input document for the payment and finally make payment will be completed. Likewise, for enterprise information systems, such as business process management (BPM), customer relationship management (CRM) and enterprise resource planning (ERP) systems, a workflow management system (WfMS), which is capable of automating workflow processes, can be a backbone platform of those systems.

In recent, exploiting and analyzing the historical data of workflow operations have been paid much attention [1-3]. It enables us to evaluate the performance of workflow processes so that effective adjustments can be made timely to achieve better process performance. In particular, social network modeling and analyzing techniques [4-8] can be applied to workflow processes to uncover social interaction between process workers (performers) and evaluatethe performance from the organizational aspect.

In this paper, we propose a technique for discovering temporal work transference network from workflow execution logs. Temporal work transference networks represent directional relationships between performers who are responsible for certain preceding and its succeeding activities (a unit of work) in a certain execution case of the workflow process (workcase). These relationships have significant meanings of interactions and collaborations between performers in workflow executions and can be

1 Div. of Computer Science and Engineering, Kyonggi University, Gyeonggi, 16227, Republic of Korea
* Corresponding author (kwang@kgu.ac.kr)
[Received 30 December 2018, Reviewed 23 January 2019, Accepted 20 March 2019]
☆ Apreliminary version of this paper was presented at APIC-IST 2018.
☆ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (Grant No. 2017R1A2B2010697). This work also was partially supported by Kyonggi University's Graduate Research Assistantship 2018.

http://dx.doi.org/10.7472/jksii.2019.20.2.101

applied to a variety of analytics tasks from the organizational aspect. To this end, we use an execution log dataset formatted in the XES (eXtensible Event Streams) standard [9]. This log format is based on the grammar for tag-based languages aiming at providing designers of information systems with a unified and extensible methodology for capturing systems behaviors by a means of event logs and event streams that are defined in the XES standard.

During executions of workflow processes, tremendous historical data is logged into the log database with detailed information, such as activity identifier, performers identifier, organization unit name, relative applications, and timestamp. This information describes the details of the system operations as well as the events that took place out with related entities. The performer is one of the primary entity types that participate in workflow processes and carry out activities in such processes. A temporal work transference network consists of nodes of performers and their work transference relationships that can be detected by analyzing execution logs of the corresponding workcase.

With this context, the remainder of this paper is organized as follows. In Section 2, aformal definition of work transference network model that is discovered from the planned workflow process in build-time will be presented. Then, Section 3 provides a formal definition of temporal work transference network model that is discovered from the execution logs of the corresponding workcase. In Section 4, a discovery algorithm of temporal work transference network model will be presented. To validate this work, Section 5 provides experimental results of the discovery algorithm by exploiting an execution log provided by a real company. Finally, this work will be concluded with future works in Section 6.

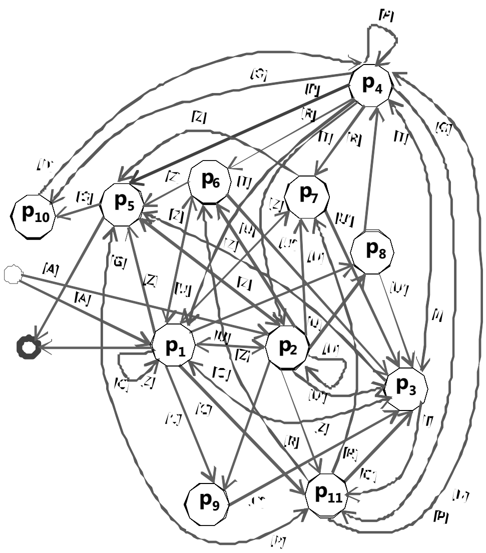
2. Work Transference Network Model

In this section, we present a formal definition of the work transference network [8] that is formed through the workflow process execution. Basically, procedural activities on a workflow process trigger off work transferences among the performers who are involved in the workflow process. Analyzing work transference network is to acquire human-centered knowledge of workflow executions, and such knowledge can be utilized in essential planning activities in human resource management and decision-making steps.

As a representation of such knowledge, a work transference network model of build-time is defined as Definition 1. A work transference network is represented by a directed graph (or digraph) model to express relationships of work transferences and their associated works among performers who are involved in a corresponding workflow process. Each vertex represents a performer, and each directed edge represents a work transference relationship.

Definition 1 (Work Transference Network Model). *A work transference network model of build-time is formally defined as* $\Lambda^B = (\sigma, \psi, F_r^B, T_0^B)$, *over a set* ***P*** *of performers, and a set* ***A*** *of activities in a workflow model, where:*

- F_r^B *is a finite set of coordinators or coordinator-groups connected from an external buildtime model of the work transferene network.*
- T_0^B *is a finite set of coordinators or coordinator-groups connected to an external buildtime model of the work transference network.*
- $\sigma = \sigma_i \cup \sigma_o$ */*Work (Activity) Transferences */*
  - $\sigma_o : P \rightarrow P(P)$ *is a multi-valued function mapping a performer to its set of (immediate) work (activity) transferrers.*
  - $\sigma_i : P \rightarrow P(P)$ *is a multi-valued function mapping a performer to its set of (immediate) work (activity) receivers.*
- $\psi = \psi_i \cup \psi_o$ */* Work (Activity) Associations */*
  - $\psi_i : (P \times P) \rightarrow P(A)$ *is a multi-valued function returning a set of receiving works (activities) on ordered pairs of performers,* $(\sigma_i(o), o), o \in P$, *from* $\sigma_i(o)$ *to* o.
  - $\psi_o : (P \times P) \rightarrow P(A)$ *is a multi-valued function returning a set of transferring works (activities) on ordered pairs of performers,* $(o, \sigma_o(o)), o \in P$, *from* o *to* $\sigma_o(o)$.



(Figure 1) A workflow transference network of buildtime from the library book acquisition workflow model

Figure 1 shows a graphical representation of work transference network at build-time from the library book acquisition workflow model. It is composed of eleven performers like $p_1 \sim p_{11}$ and their ordered pairs labeled with the associated works in the activity set, $A = \{A, C, C', U, U', I, P, G, D, R, T, Z\}$, of the workflow procedure.

3. Temporal Work Transference Network Model

During executions of workflow processes, the logging and auditing components in the workflow execution engine record an event on a log database whenever a state of the activity execution changes.

Those logged events are arranged in a form of a temporal sequence of events. This execution sequence of a workcase forms a *workcase event trace* and it is also involved with a performing sequence of the performers (*workcase performer trace*) who are in charge of the executions of the activities in the corresponding workcase. We can also extract a *workcase performer trace* from a *workcase event trace*.

Definition 2 (Workflow Activity Event). *Let* $we = (wi, pc, wf, wc, ac, p^*, t, s)$ *be an event of activity execution stored as log, where:*

- wi *is a workitem (activity instance) identifier.*
- pc *is a workflow package identifier.*
- wf *is a workflow process identifier.*
- wc *is a workcase identifier.*
- ac *is an activity identifier.*
- p^* *is a performer identifier.*
- t *is a timestamp.*
- s *is a workitem's current state, which is one of the states such as ready, assigned, reserved, completed, and cancelled.*

Definition 3 (Workcase Event Trace). *Let* $WT(c)$ *be the workcase event trace of a workcase, c, where* $WT(c) = (we_1, ..., we_n)$, *where* $we_i | we_i.wc = c \wedge we_i.t \le we_j.t \wedge we_i.pc = we_j.pc \wedge we_i.wf = we_j.wf \wedge we_i.wc = we_j.wc \wedge i < j \wedge 1 \le i, j \le n$, *which formally represents a temporally ordered activity event sequence of a workcase, which is built through preprocessing the workflow activity events by considering the timestamp and the state attributed.*

Based on the concepts above, to discover temporal work transferences, we define four types of temporal properties withholding one of the timestamp-origins.

- The *scheduled* point of time: the event's timestamp is taken at when the state of a workitem is changed from READY to ASSIGNED. A workflow activity event with a *scheduled* timestamp, $we^{(t.s)} \Rightarrow (t = we.t \wedge s = we.s. \wedge s = ASSIGNED)$.
- The *accessed* point of time: the event's timestamp is taken at when the state of a workitem is changed from ASSIGNED to RESERVED. A workflow activity event with a *accessed* timestamp, $we^{(t.e)} \Rightarrow (t = we.t \wedge e. = we.s \wedge e = RESERVED)$
- The *started* point of time: the event's timestamp is taken at when the state of a workitem is changed from RESERVED to RUNNING. A workflow activity event with a *running* timestamp, $we^{(t.u)} \Rightarrow (t = we.t \wedge u = we.s \wedge u = RUNNING)$.
- The *completed* point of time: the event's timestamp is

taken at when the state of a workitem is changed from RUNNING to COMPLETED. A workflow activity event with a *completed* timestamp, $we^{(t.o)} \Rightarrow (t = we.t \wedge o = we.s \wedge o = COMPLETED)..$

Based on the temporal properties, we can organize a temporal work transference, TWT(*c*), by taking the performer identifier from each of the workflow activity events making up the corresponding workcase event trace. The temporal work transference is transformed into a temporal work transference model, TWTM(*c*).

Definition 4 (Temporal Work Transference). *Let* TWT(*c*) *be the workcase performer trace of a workcase,* c*:*

- $TWT(c) = (we^{\tau[.\phi]}_{(p_1)}, \ldots, we^{\tau[.\phi]}_{(p_m)})$, *where*

  $we_p^{\tau[.\phi]} | p = we.p \wedge \tau = we.t \wedge \phi \in e,u,o \wedge we_p^{\tau[.\phi]}.$

  $wc = c \wedge we^{\tau}_{(p_i)} < we^{\tau}_{(p_j)} \wedge \tau_i < \tau_j \wedge i < j \wedge 1 \le i.$

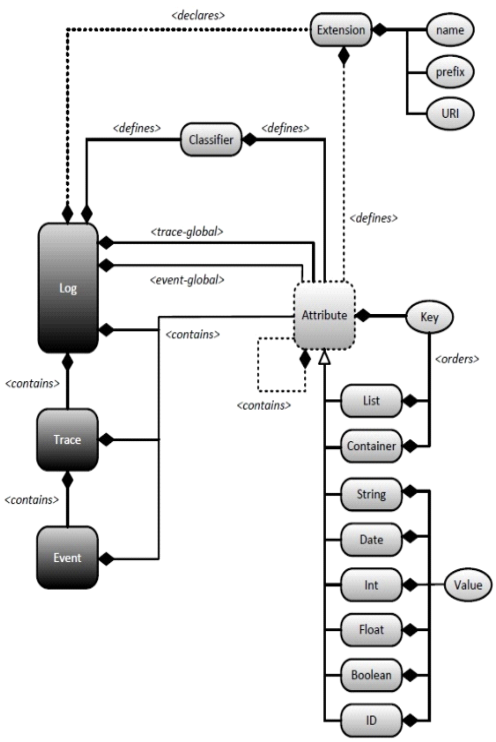
Definition 5 (Temporal Work Transference Network Model, TWTM). *A work transference network model of runtime is formally defined as* $\Lambda^R = (\sigma, \psi, F_r^R, T_o^R)$, *over a set* ***P*** *of performer and a set* ***A*** *of activities in a bunch of workcase event traces logged from enacting a specific workflow process, where:*

- F_r^B *is a coordinators or a coordinator-group linked from some external work transference networks of runtime.*
- T_0^B *is a coordinators or a coordinator-group linked to some external work transference networks of runtime.*
- $\sigma = \sigma_i \cup \sigma_o$ */* Work (workitem) Transferences */*
  - $\sigma_o : P \rightarrow P(P)$ *is a multi-valued function mapping a performer to its set of (immediate) work (workitem) transferrers.*
  - $\sigma_i : P \rightarrow P(P)$ *is a multi-valued function mapping a performer to its set of (immediate) work (workitem) receivers.*
- $\psi = \psi_i \cup \psi_o$ */* Work (workitem) Associations */*
  - $\psi_i : (P \times P) \rightarrow P(A)$ *is a multi-valued function returning a bag of receiving works (workitems) on ordered pairs of performers,* $(\sigma_i(o), o), o \in P$*, from* $\sigma_i(p)$ *to* p*.*
  - $\psi_o : (P \times P) \rightarrow P(A)$ *is a multi-valued function returning a bag of transferring works (workitems) on ordered pairs of performers,* $(p, \sigma_o(p)), p \in P,$*, from* p *to* $\sigma_o(p)$*.*

4. TWTM Discovery Algorithm

Based upon the definition of TWTM, we develop an algorithm that is able to discover a TWTM from a workcase event trace of a single workcase. The algorithm takes XES-formatted execution logs as input data. The XES standard includes an XML schema describing the structure of an XES event log/stream. Moreover, a basic collection of so-called XES extension prototypes that provide semantics to certain attributes as recorded in the event log/stream is included in this standard. Figure 2 represents an XML schema of the XES standard and its core elements are as follows:

- <*log*>:Root element of an XES log file.



(Figure 2) The XML schema of the XES standard

- <*trace*>: Tag element for a trace object included in a <*log*>element. It corresponds to a certain workcase and contains information pertaining to the execution of the workcase from its starting to termination.
- <*event*>: Tag element for the event object included in a specific <*trace*> tag element. Each <*event*> element contains detailed information including activity and performer identifiers, and its timestamp.

Overall procedure of the discovery algorithm is as follows: First, the input XES log file is put into the XES log repository. The control module reads the input data, passing it to the analysis module. Based on the XES standard structure and data is transferred, the module performs the analysis and disassembly tasks and then transfers intermediate results to the statistic module. Next, the statistic module performs statistical analysis (e.g., frequencies of work trace participation of performers). Finally, the visualization module relies on data generated by previous modules to visualize the temporal work transference network. To implement this algorithm, we split the job into two sub-steps:

- Step 1: Finding individual performers. At this step, the system performs the parsing data from the XES file, extract each temporal workcase and save it to the system. From each workcase, remove the duplicated performers then add it to a list.
- Step 2: Discovering a temporal work transference network. With the data that has been initialized from step 1, the systems constructs relationships between performers then finally visualize the discovered temporal transference network.

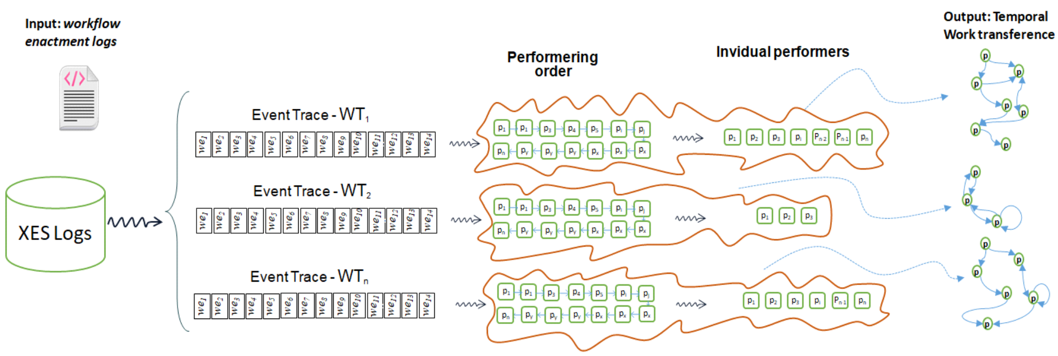
These steps and algorithms can be summarized in Figure 3.

Algorithm: Finding individual performers

```
Input: A Workflow execution log file (ωε.
Output: A list of individual performers (γ).
Begin
Open the execution log.
        Initialize all worcase β[], performers[] to ∅
        Set Number of Trace μ= 0
        For (∀race[i] ∈(ωε)
            μ= μ+1
          For (∀vent[j] ∈trace[i])
              add (event[j]   β [ μ].event.next)
          Endfor
        Endfor
        For (∀vent[j] ∈ β [])
            If(not event[j].performer ∈performers[])
        Add(event[j].performer   performers[]
        Endfor
        γ  performers[]
        Close the execution log
End
```

Algorithm: TWTM discovery

```
Input: A List of individual performers (γ), a set of workcases β[]
and a number of trace μ
Output: A temporal work transference network model ω
Begin
        For (∀=0   μ-1)
            Current_performer = β[i]performer
            Next_performer = β[i+1] performer
            Find j where γ[j] = Current_performer
            Find k where γ[k] = Next_performer
            γ[j]next = γ[k]
        Endfor
        ω←γ
End
```



(Figure 3) Overall procedure of the TWTM discovery

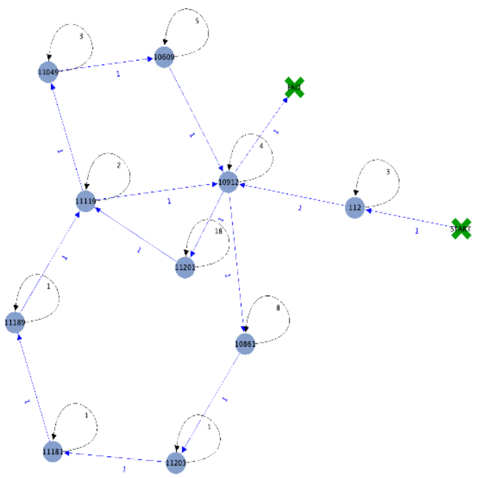
5. Experimental Results

In order to validate the TWTM discovery algorithm, we exploit a real log dataset, provided by 4TU [10] and containing execution logs of a loan application process in the anonymous bank company. The summary of the log dataset is presented in Table 1.

(Table 1) Summary of the log dataset.

Workflow process name	Loan application process
Start time	2011-10-01
End time	2012-03-14
Num. of activities	24
Num. of performers	68
Num. of traces	13087
Num. of events	262200

As an operational example, we applyour discovery algorithm to the dataset and Figure 4 is a visualization of the discovered TWTM of the trace-2 in the dataset.



(Figure 4) Temporal work transference network model discovered from trace-2

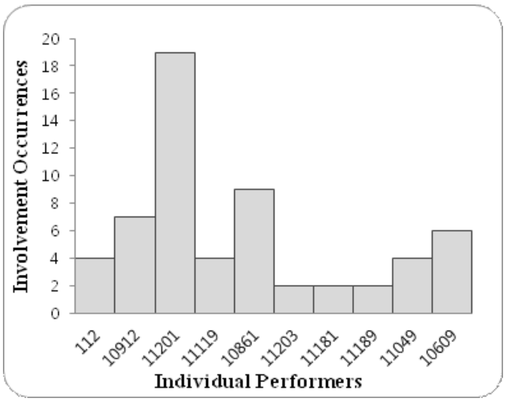
For the trace-2, there are 59 events and 10 performers participated in the corresponding workcase. As shown in Figure 4, there is a chordless cycle [11] between performers: *10912, 10861, 11203, 11181, 11189*, and *11119*.

(Table 2) Involvement occurrences of performers for the trace-2

Performer ID	Involvement occurrences
11203	2
11181	2
11189	2
112	4
11119	4
11049	4
10609	6
10912	7
10861	9
11201	19

- The entire order of performers attached to its event: *112, 112, 112, 112, 10912, 10912, 10912, 10912, 11201, 11201, 11201, 11201, 11201, 11201, 11201, 11201, 11201, 11201, 11201, 11201, 11201, 11201, 11201, 11201, 11201, 11201, 11201, 11119, 11119, 10912, 10912, 10861, 10861, 10861, 10861, 10861, 10861, 10861, 10861, 10861, 11203, 11203, 11181, 11181, 11189, 11189, 11119, 11119, 11049, 11049, 11049, 11049, 10609, 10609, 10609, 10609, 10609, 10609, 10912.*

The number of turns involved in the performer's event is presented in Table 2 and Figure 5. The workcase (trace-2) was started by the performer 112 and was finished by the performer 10912. For the performer 11201, he/she have participated in the 19 workitems.



(Figure 5) Involvement occurrences of the performers in trace-2

6. Conclusion

The significance of analyzing workflow execution logs for improving workflow processes has been growing in recent. In this regard, we define the temporal work transference network model (TWTM) that reflects interactions between performers in workflow executions and aims to enable workflow process analyses and improvements from the organizational aspect. Next, we present an algorithm to discover a TWTM from workflow execution logs.

To verify the proposed algorithm, we implementedthe discovery algorithm and applied it to the execution log dataset of the loan application process [8]. Through the demo system, we confirm that the proposed algorithm is feasible and can discover TWTM from given execution logs.

However, the correctnessof discovered TWTMs remains as a challenge. Accordingly, to discover exact or more precise TWTMs, we are planning to study the following in the future:

- Handling complicated control-flow patterns (e.g., loop).
- Developing a comprehensive analytics system for visualizing TWTMs and providing those statistics results.
- Proving the effectiveness and utility of the TWTM discovery and analytics with applying to real-world workflow processes.

References

[1] W. M. P. van der Aalst, et al., "Workflow Mining: A Survey of Issues and Approaches," *Data & Knowledge Engineering*, Vol. 47, No. 2, pp. 237-267, 2003. https://doi.org/10.1016/S0169-023X(03)00066-1

[2] K. Kim, M. Yeon, B. Jeong and K. Kim, "A Conceptual Approach for Discovering Proportions of Disjunctive Routing Patterns in a Business Process Model," *KSII Transactions on Internet and Information Systems*, Vol. 11, No. 2, pp. 1148-1161, 2017. https://dx.doi.org/10.3837/tiis.2017.02.030

[3] K. Kim, Y.-K. Lee, H. Ahn and K. P. Kim, "An Experimental Mining and Analytics for Discovering Proportional Process Patterns from Workflow Enactment Event Logs," *Wireless Networks*, pp. 1-8, Online published, 2018. https://dx.doi.org/10.1007/s11276-018-01899-z

[4] M. Park, H. Ahn and K. P. Kim, "Workflow-Supported Social Networks: Discovery, Analyses, and System," *Journal of Network and Computer Applications*, Vol. 75, pp. 355-373, 2016. https://doi.org/10.1016/j.jnca.2016.08.014

[5] J. Kim, et al., "An Estimated Closeness Centrality Ranking Algorithm and Its Performance Analysis in Large-Scale Workflow-supported Social Networks," *KSII Transactions on Internet and Information Systems*, Vol. 10, No. 3, pp. 1454-1466, 2016. https://dx.doi.org/10.3837/tiis.2016.03.031

[6] M.-J. Kim, H. Ahn, M.-J. Park, "A Theoretical Framework for Closeness Centralization Measurements in a Workflow-Supported Organization," *KSII Transactions on Internet and Information Systems*, Vol. 9, No. 9, pp. 3611-3634, 2015. https://dx.doi.org/10.3837/tiis.2015.09.018

[7] M.-J. Kim, H. Ahn, M.-J. Park, "A GraphML-based Visualization Framework for Workflow-Performers' Closeness Centrality Measurements," *KSII Transactions on Internet and Information Systems*, Vol. 9, No. 8, pp. 3216-3230, 2015. https://dx.doi.org/10.3837/tiis.2015.08.028

[8] K. Kim, M. Jin, H. Ahn and K. P. Kim, "Discovering Work Transference Networks on Workflows," In *Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services*, pp. 568-572, 2017. https://dx.doi.org/10.1145/3151759.3156473

[9] eXtensible Event Streams (XES) Definition. http://www.xes-standard.org

[10] 4TU, https://data.4tu.nl/repository

[11] E. W. Weisstein, "ChordlessCycle," Wolfram MathWorld, accessed 12-29-2018, http://mathworld.wolfram.com/ChordlessCycle.html

◐ 저 자 소 개 ◑

Dinh-Lam Pham

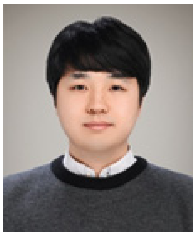
2010 M.S. in Computer Science, Thai Nguyen University

2016~Present, Researcher of the Information Technology Institute at Vietnam National University.

2018~Present, Ph.D. course in Computer Science, Kyonggi University

Research Interests : process mining, enterprise social network analysis.

E-mail : phamdinhlam@kgu.ac.kr



Hyun Ahn

2011 B.S. in Computer Science, Kyonggi University

2013 M.S. in Computer Science, Kyonggi University

2017 Ph.D. in Computer Science, Kyonggi University

2018 ~ Present, Assistant Professor of the Dept. of Computer Science and Engineering at Kyonggi University

Research Interests : Business Process Management, business process intelligence, process mining.

E-mail : hahn@kgu.ac.kr

Kwanghoon Pio Kim

1984 B.S in Computer Science, Kyonggi University

1986 M.S. in Computer Science, Chungang University

1994 M.S. in Computer Science, University of Colorado at Boulder

1998 Ph.D. in Computer Science, University of Colorado at Boulder

1998 ~ Present, Professor of the Dept. of Computer Science and Engineering at Kyonggi University

Research Interests : CSCW, workflow systems, Business Process Management, process mining, enterprise social network analysis.

E-mail : kwang@kgu.ac.kr