

Process-aware Event Log Datacubes for Workflow Process and Knowledge Mining, Predicting and Analyzing Frameworks

Seong-Hun Ham

Div. of Comp. Sci. & Eng.
KYONGGI UNIVERSITY
Suwon, South Korea
shham9@kgu.ac.kr

Dihn-Lam Pham

Div. of Comp. Sci. & Eng.
KYONGGI UNIVERSITY
Suwon, South Korea
pdlam@kgu.ac.kr

Kyoung-Sook Kim

Div. of Comp. Sci. & Eng.
KYONGGI UNIVERSITY
Suwon, South Korea
khmjmc@kgu.ac.kr

Kwanghoon Pio Kim

Div. of Comp. Sci. & Eng.
KYONGGI UNIVERSITY
Suwon, South Korea
kwang@kgu.ac.kr

Abstract—The issue of workflow process mining and analytics is beginning to make its appearance in the workflow-supported enterprise intelligence and systems literature. In order to improve the quality of workflow process intelligence, it is essential for an efficient and effective data center storing workflow enactment event logs to be provisioned in carrying out the workflow process mining and analytics. In this paper, we propose a three-dimensional datacube, which is named as a *process-aware datacube*, for organizing workflow-supported enterprise data centers to efficiently as well as effectively store the workflow process enactment event logs in the IEEE XES format, and also we carry out an experimental process mining and analytics to show how much perfectly the process-aware datacubes are suitable for discovering workflow process patterns and its analytical knowledge, like enacted proportions and enacted work transferences, from the workflow process enactment event histories. From the process-aware datacubes, the workflow process mining must be able to properly discover all the workflow process patterns based upon the four types of control-flow primitives such as linear (sequential) routing, disjunctive (selective) routing, conjunctive (parallel) routing, and iterative (loop) routing patterns, whereas the workflow process analytics is to discover the enacted proportions of each of the process patterns and the enacted work transferences of each of the workflow performers.

Index Terms—process-aware datacubes, workflow process mining, workflow process analytics, temporal workcase, temporal worktransference, XES format, workflow-supported enterprise data center

I. INTRODUCTION

In recent, the workflow (or business process) automation technologies are swiftly growing and increasingly used by many newly formed workflow-supported enterprises. Accordingly, a new branch of requirements and demands has been raised, which is surely a novel concept of workflow fidelity. It is eventually related with reengineering and redesigning those deployed workflow procedures and their models running on the process-aware organizations. In order to improve the quality of re-engineering and redesigning a workflow process model, we need its enactment history to measure how much

it is faithful to the original workflow model. This measuring activity is defined as a novel concept of workflow fidelity, which can be concretized by comparing a workflow process model discovered on Buildtime to the workflow process model rediscovered on Runtime. The former is concerning about planning and defining the workflow process model, whereas the latter is concerning about mining the enacted workflow process model from its enactment event histories and logs, which is so-called the workflow process mining issue. Many experiments so far have been devoting considerable attention to the workflow process mining issue, in particular, that is related with devising the workflow process mining algorithms and architectures. The most essential architectural element ought to be the workflow event log datasets provisioned in data centers of workflow-supported enterprises. In this paper, we propose a formal approach for organizing three-dimensional datacubes, which is named as process-aware datacubes, and implement the approach to be used for workflow event log data centers. Afterward, we carry out an experimental analytics based upon a process-aware datacube that is provisioned from a real dataset that was released to the public at the BPI Challenges website of 4TU.Centre for Research Data [10]. In other words, in order to carry out the experimental analytics, we apply the proposed approach into a single dataset chosen out of the 10 workflow enactment event log datasets, which was recorded from enacting the Large Bank Transaction Process Model, in particular.

II. RELATED WORKS

In order to analyze the performative histories and goals via the business activity monitoring and analytics in a workflow-supported enterprise, it is important to store away all the behavioral records and auditing information to a specific form of enterprise-wide data centers. The workflow-supported behaviors are represented by the workflow process models and packages, realized by enacting the model, and recorded by the workflow enactment engine. The logging and auditing component of the workflow enactment engine records workflow instances' and workitems' execution events on its log data

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (Grant No. 2017R1A2B2010697).

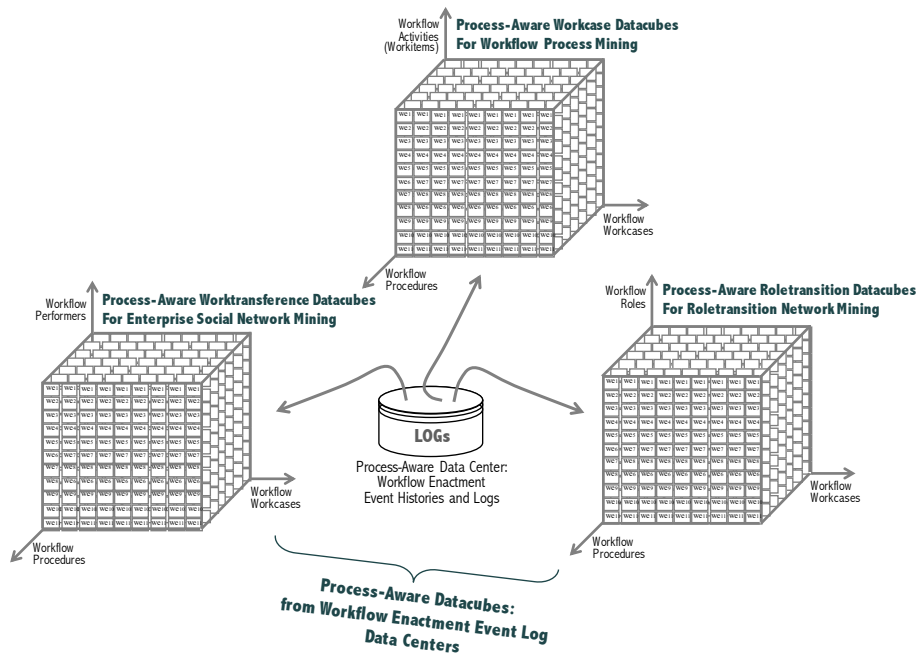


Fig. 1. A Conceptual Formation of Process-Aware Datacubes from Workflow Enactment Event Log Data Centers

centers, and those logged events are arranged in a form of three dimensional datacubes — *workflow process models*, *their workflow instances' event traces* and *their workitems' event traces*. So far, there are several research works considering the log structures and formats of workflow enactment event log data centers. All of them consider the workflow enactment event logs being stored in a certain format of tag-based languages. The typical outcomes are XWELL, BPAF, and IEEE XES. An XML-based workflow enactment event log format, XWELL¹ [13], for the purpose of workflow mining was studied and proposed by the authors' research group. The WfMC, which stands for workflow management coalition as an international standardization organization of the workflow technologies, had released the standardized audit and log format specification, BPAF² [17]. In recent, IEEE has released a standard tag-based language, XES³ [18], whose aim is to provide designers of information systems with a unified and extensible methodology for capturing systems' behaviors by means of event logs and event streams. As the format of workflow enactment event log data centers, the IEEE XES ought to be now the most appropriate format. All of the event history datasets, which are now available in the literature, are based upon the "XES Schema [18]" describing the structure of an XES event log/stream and the "XES extension" describing the structure of an extension of such a log/stream. Based upon

the format of the XES schema, we propose a structure of datacubes for workflow enactment event log data centers in this paper, and we try to apply to all those event history and log datasets.

III. PROCESS-AWARE DATACUBES

According as a workflow instance is executed, the logging and auditing component of the workflow enactment engine records its workitem (activity) execution events onto a log repository. We assume that each of those logged events are formatted in the IEEE XES standard XML-tag language. In order for those logged events to be analyzed by a certain type of workflow process mining and analytics tools, we need to arrange all the logged events in various orders of the properties like the workflow model identifier, the workflow instance identifier, the type of time-stamps, and the time-stamp, itself. Through these arranged logged-events, we are able to construct various types of workflow event log data centers, each of which is organized in a process-aware datacube. Fig. ?? illustrates a conceptual formation of organizing various types of process-aware datacubes from the workflow event log data centers. In this section, we theoretically formalize both an event log format for organizing workflow enactment event log data centers and a cubic structure for forming process-aware datacubes, and practically build a typical type of process-aware datacubes, which is called a process-aware workcase datacube, being organized from the BPI challenge datasets [10], as a concrete example.

¹XWELL stands for XML-based workflow execution logging mechanism and language.

²BPAF stands for business process audit format, which is supported by the Workflow Management Coalition (WfMC).

³XES stands for extensible event stream, which is supported by the extensible event stream working group of the IEEE standard committee.

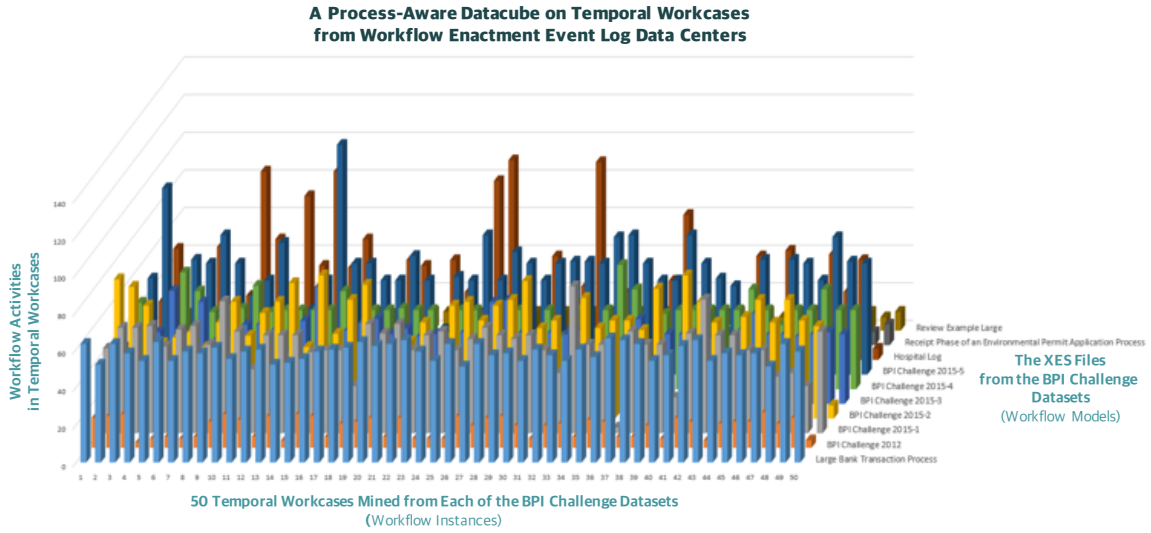


Fig. 2. A Process-Aware Datacube on Temporal Workcases from the BPI Challenge Datasets

A. Workflow Enactment Event Log Data Centers

Whenever a workflow enactment engine instantiates a workflow process model and enacts its instances, each of which is called a *workcase*, a logging functional component of the engine records all of their activities' execution (workitems) event histories onto a certain type of event log repositories. Here, we start to formally describe the conceptual formation from defining a formal format of activities' execution event histories as the name of the workflow workitem event log as shown in the following *Definition III-A*. All the event logs of the workflow packages and models enacted and controlled by the engine are accumulated onto a process-aware data center, and from which we are able to build various types of process-aware datacubes, such as process-aware workcase datacube, process-aware worktransference datacube and process-aware roletransition datacube, as depicted in the conceptual formation of Fig. ??.

[Definition] III-A: *Workflow Workitem Event Log*. Let $we = (\alpha, pc, wf, wc, ac, p^*, t, s)$ be a workflow work-item event stored as log, where

- α is a work-item (activity instance) identifier,
- pc is a package identifier,
- wf is a workflow process identifier,
- wc is a workflow instance (workcase) identifier,
- ac is an activity identifier,
- p^4 is a participant (or performer) identifier,
- t is a timestamp, and
- s is a work-item's current state, which is one of the states such as *ready*, *assigned*, *reserved*, *running*, *completed*, and *cancelled*.

In terms of the concrete log format of the workflow work-item event log, we use the XES standard format [18] as we mentioned in the previous section. Based upon the description

⁴Note that * indicates multiplicity.

of the standard format, we simply define the essential attributes in the workflow work-item event log, as follows:

- The **EVENT** attribute is used to specify an event identifier, which is assigned by the workflow enactment engine.
- The **WORKITEM** attribute of a workflow activity event represents a work-item identifier that is uniquely assigned by using those combined identifiers, such as PACKAGE ID, WORKFLOW ID, ACTIVITY ID, and INSTANCE ID.
- The **PARTICIPANT** attribute is used to specify the performer who is in charge of enacting the work-item.
- The **TIMESTAMP** attribute specifies the time stamp of the event occurred.
- Finally, the **STATE** attribute represents the work-item's runtime state maintained by the engine. Whenever the work-item's state is changed, it is logged with the event-code of *WMChanged Work-itemState*. It ought to be one of those states such as *READY*, *ASSIGNED*, *RESERVED*, *RUNNING*, *COMPLETED*, and *CANCELLED*.

B. Process-Aware Workcase Datacubes

From a process-aware data center accumulating their event histories and logs while the engine enacts all the workflow packages and models, we are able to build a cubic structure of process-aware workcases, which is named as a process-aware workcase datacube that has three dimensions of workflow model dimension, workflow instance dimension and workflow activity dimension. The essential dataset of the datacube is a group of temporal workcases, and each of which is determined by a *workflow instance activity-event trace*. That is, a temporal sequence of the activities' execution of each workflow instance is forming a *workflow instance activity-event trace* by extracting its workitem event logs from the process-aware data center, and it is formally represented by a model of *temporal workcases*. The *workflow instance activity-event trace* and the

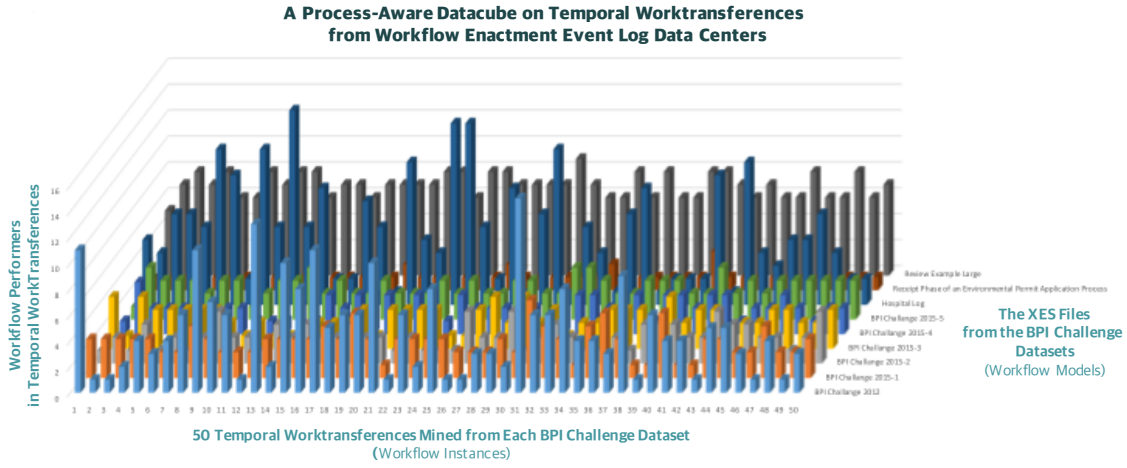


Fig. 3. A Process-Aware Datacube on Temporal Worktransferences from the BPI Challenge Datasets

temporal workcases are formally defined in *Definition III-B* and *Definition III-B*, respectively.

[Definition] III-B: Workflow Instance Activity-Event Trace. Let $WT(c)$ be the workflow instance event trace of a work-case, c , where $WT(c) = (we_1, \dots, we_n)$, where $\{we_i \mid we_i.wc = c \wedge we_i.t \leq we_j.t \wedge we_i.pc = we_j.pc \wedge we_i.wf = we_j.wf \wedge we_i.wc = we_j.wc \wedge i < j \wedge 1 \leq i, j \leq n\}$, which formally represents a temporally ordered workflow activity event sequence of a specific work-case, which is built through preprocessing the workflow instance event logs by considering the *TIMESTAMP* and the *STATE* attributes.

Basically, from the formal model of a workflow instance event trace, we build a temporal work-case model, $TWC(c)$, as well as a temporal work-transference model, $TWT(c)$, as shown in Fig. ?? At this moment, we have to remind that the meaningful temporal order in managing workflow instances (work-cases) ought to be based upon one of the following instantaneous points of time, each of which is named as a *timestamp-origin*. Accordingly, we necessarily discover a series of workflow instance event traces from the workflow event log histories, and from each of the discovered traces, we build four species of temporal work-cases and temporal work-transferences with holding one of the timestamp-origins.

- The Scheduled Point of Time: the event's timestamp is taken at when the state of a work-item is changed from *READY*⁵ to *ASSIGNED*⁶: a workflow activity event log with *scheduledTimestamp*, $we^{t.s} \Rightarrow (t = we.t \wedge s = we.s \wedge s = 'assigned')$
- The Assessed Point of Time: the event's timestamp is taken at when the state of a work-item is changed from

ASSIGNED to *RESERVED*⁷: a workflow activity event log with *assessedTimestamp*, $we^{t.e} \Rightarrow (t = we.t \wedge e = we.s \wedge e = 'reserved')$

- The Started Point of Time: the event's timestamp is taken at when the state of a work-item is changed from *RESERVED* to *RUNNING*⁸: a workflow activity event log with *runningTimestamp*, $we^{t.u} \Rightarrow (t = we.t \wedge u = we.s \wedge u = 'running')$
- THE Completed Point of Time: the event's timestamp is taken at when the state of a worktime is changed from *RUNNING* to *COMPLETED*⁹: a workflow activity event log with *completedTimestamp*, $we^{t.o} \Rightarrow (t = we.t \wedge o = we.s \wedge o = 'completed')$

[Definition] Temporal Workcase. Let $TWC(c)$ be the workflow instance activity trace of a work-case (temporal work-case), c :

- $TWC(c) = (we_{\alpha_1}^{\tau[\cdot\phi]}, \dots, we_{\alpha_m}^{\tau[\cdot\phi]})$, where $\{we_{\alpha}^{\tau[\cdot\phi]} \mid \alpha = we.ac \wedge \tau = we.t \wedge \phi \in \{s, e, u, o\} \wedge we_{\alpha}.wc = c \wedge (we_{\alpha_i}^{\tau_i} \prec we_{\alpha_j}^{\tau_j})^{10} \wedge \tau_i < \tau_j \wedge i < j \wedge 1 \leq i, j \leq m\}$,

which is a temporally ordered workflow activity event sequence along with the timestamp-origins. Especially, a temporal work-case is formally defined as a temporal work-case model, and it is assumed that all the work-items in the corresponding work-case are successfully completed, and their executions are running without being suspended, as well.

Based on *Definition III-B*, we can interpret the formal definition as the conceptual implication that all of the workflow activity events holding an identical *INSTANCE ID* are lined

⁷The *RESERVED* state of a work-item implies that the work-item has been assigned to a named user (a single participant), but work has not started yet.

⁸The *RUNNING* state of a work-item implies that the work-item is actively being worked on, and time spent in this state would be recorded as processing time or work time.

⁹The *COMPLETED* state of a work-item implies that the work-item has been fully executed and completed with either success or failure.

¹⁰ $we_{\alpha_i}^{\tau_i}$ is the predecessor of $we_{\alpha_j}^{\tau_j}$ in the list of a temporal work-case.

⁵The *READY* state of a work-item implies that the work-item is ready to be processed but has not been assigned to a particular participant.

⁶The *ASSIGNED* state of a work-item implies that the work-item has been assigned to a role (potentially a group of participants), but work has not started yet.

up in a workflow instance activity trace along with the timestamp-origins. Consequently, from a workflow instance event trace, we produce a workflow instance activity trace by extracting the activity identifiers and their timestamps, and we name it as a temporal work-case along with the timestamp-origins like the scheduled time, assessed time, started time, and completed time. For the sake of the formal representation, a temporal work-case is defined as a temporal work-case model as formally described in Definition ?? . Note that all of the event logs in a workflow instance activity trace may have an identical timestamp-origin. Accordingly, there possibly exist four species of the temporal work-case and their models, as follows:

- ScheduledTime Temporal Work-case Species and Model
- AssessedTime Temporal Work-case Species and Model
- StartedTime Temporal Work-case Species and Model
- CompletedTime Temporal Work-case Species and Model

C. Process-Aware Datacubes on Temporal Work-transferences

The activities' execution sequence of a workflow instance is involved with a performing sequence of the performers, which is called a *workflow instance performer-event trace*, who are in charge of the executions of the workitems in the corresponding workflow instance. We can also extract a *workflow instance performer-event trace* from the repositories and it is formally represented by a model of *temporal work-transferences*.

[Definition] *Temporal Work-transference*. Let $TWT(c)$ be the workflow instance performer trace of a work-case (temporal work-transference), c :

- $TWT(c) = (we_{p_1}^{\tau[\phi]}, \dots, we_{p_m}^{\tau[\phi]}),$
 where $\{we_p^{\tau[\phi]} \mid p = we.p \wedge \tau = we.t \wedge \phi \in \{e, u, o\} \wedge we_p^{\tau[\phi]}.wc = c \wedge (we_{p_i}^{\tau} \prec we_{p_j}^{\tau})^{11} \wedge \tau_i < \tau_j \wedge i < j \wedge 1 \leq i, j \leq m\}$;

which is a temporally ordered workflow performer event sequence along with the timestamp-origins. Note that the ScheduledTimestamp origin is not available in the workflow performer event sequence. Especially, a temporal work-transference is formally defined by a temporal work-transference model, and it is assumed that all the work-items in the corresponding work-case are successfully completed, and their executions are running without being suspended, as well.

Based on Definition III-C, we can interpret the formal definition as the conceptual implication that all of the workflow activity events holding an identical INSTANCE ID are lined up in a workflow instance performer trace along with the timestamp-origins. Consequently, from a workflow instance event trace, we produce a workflow instance performer trace by extracting the performer identifiers and their timestamps, and we name it as a temporal work-transference along with the timestamp-origins like the assessed time, started time, and completed time. For the sake of the formal representation, a temporal work-case is defined as a temporal work-case model

¹¹ $we_{p_i}^{\tau_i}$ is the predecessor of $we_{p_j}^{\tau_j}$ in the list of a temporal work-transference.

as formally described in Definition ?? . Note that all of the event logs in a workflow instance performer trace may have an identical timestamp-origin. Accordingly, there possibly exist three species of the temporal work-transference and their models, as follows:

- AssessedTime Temporal Work-transference Species and Model
- StartedTime Temporal Work-transference Species and Model
- CompletedTime Temporal Work-transference Species and Model

In summarily completing this subsection, we know that so-called *workflow warehouses*, which are based on the temporal work-cases and the temporal work-transferences, are needed to build a workflow mining system for discovering the work-transference networking knowledge from the workflow enactment event logs piled up on an enterprise big data storage. A workflow warehouse that ought to be shaped in a cube with three axes: workflow procedures (*workflow models*), workflow instances (*temporal work-cases* and *temporal work-transferences*), and workflow work-items (*workflow activity event logs*). In order to build a workflow warehouse we have to fulfill the preprocessing phase that extracts a series of temporal work-cases with temporal work-transferences from a pile of workflow enactment event logs, and forms two cubes, one of which is for temporal work-cases, and the other of which is for temporal work-transferences. These two cubes eventually become the input of a workflow work-transference network rediscovery algorithm to be developed in the next section. We have already implemented a mining function that is able to build a workflow warehouse based upon the temporal work-case model and the temporal work-transference model. Fig. ?? shows a workflow warehouse composed of 50 event traces mined from each of the 10 workflow enactment event history files provided in the BPI Challenge dataset [10].

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (Grant No. 2017R1A2B2010697).

REFERENCES

- [1] Fabio Casati, et al, "Business Process Intelligence," Technical Report, HPL-2002-119, HP Laboratories Palo Alto, 2002.
- [2] Daniela Grigori, Fabio Casati, Malu Castellanos, Umeshwar Dayal, Mehmet Sayal and Ming-Chien Shan, "Business Process Intelligence," Journal of Computers in Industry, Vol. 53, Issue 3, 2004.
- [3] W. M. P. van der Aalst, B. F. van Dongena; J. Herbst, L. Marustera, G. Schimm and A. J. M. M. Weijters, "Workflow mining: A survey of issues and approaches," Journal of Data & Knowledge Engineering, Vol. 47, Issue 2, pp. 237-267, 2003.
- [4] W. M. P. van der Aalst and A. J. M. M. Weijters, "Process mining: a research agenda," Journal of Computers in Industry, Vol. 53, Issue 3, 2004.
- [5] Aalst, W.P.M., Alves de Medeiros, A.K., and Weijters, A.J.M.M., "Process Equivalence: Comparing Two Process Models Based on Observed Behavior", BPM2006, Lecture Notes in Computer Science, Vol. 4102, pp. 129-144, 2006.

- [6] Kwang-Hoon Kim, "Control-path Oriented Workflow Intelligence Analysis on Enterprise Workflow Grids," Proceedings of the International Conference on Semantics, Knowledge, and Grid, Beijing, China, 2005.
- [7] Kwang-Hoon Kim and Clarence A. Ellis, "Workflow Reduction for Reachable-path Rediscovery in Workflow Mining," Series of Studies in Computational Intelligence: the Foundations and Novel Approaches in Data Mining, Vol. 9, pp.289-310, Springer, 2006.
- [8] Minjae Park and Kwanghoon Kim, "Control-path Oriented Workflow Intelligence Analyses," JOURNAL OF INFORMATION SCIENCE AND ENGINEERING, Vol. 24, pp. 343-359, 2008.
- [9] Kyoungsook Kim, Moonsuk Yeon, Byeongsoo Jeong, Kwanghoon 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.
- [10] BPI Challenge 2012, 2013, 2014, 2015, 2016, 2017, 2018, 4TU.Centre for Research Data, <https://data.4tu.nl/repository/collection:event-logs-real>.
- [11] Ellis, Clarence A., Kim, K., Rembert, A., Wainer, J., "Investigations on Stochastic Information Control Nets," INFORMATION SCIENCES, Vol. 194, pp. 120-137, 2012.
- [12] Kim, Kwanghoon, "A XML-Based Workflow Event Logging Mechanism for Workflow Mining," The Proceedings of the International Workshop on APWeb, pages 132-136, 2006.
- [13] Minjae Park and Kwanghoon Kim, "XWELL: A XML-Based Workflow Event Logging Mechanism and Language for Workflow Mining Systems," Lecture Notes in Computer Science, Vol. 4707, pp. 900-909, 2007.
- [14] Kim, Kwanghoon and Ellis, Clarence A., " σ -Algorithm: Structured Workflow Process Mining Through Amalgamating Temporal Workcases," The Proceedings of PAKDD2007, Advances in Knowledge Discovery and Data Mining, Lecture Notes in Artificial Intelligence, Vol. 4426, pp. 119-130, 2007.
- [15] Ahn, H. and Kim, K., "A Stochastic Activity-to-Performer Affiliation Binding Formalism in ICN-based Workflow Models," ICICI Express Letters, Vol. 9, No. 12, 2015.
- [16] Kim, H., Ahn, H., and Kim, K., "Modeling, Discovering, and Visualizing Workflow Performer-Role Affiliation Networking Knowledge," KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS, Vol. 8, No. 2, pp. 691-708, 2014.
- [17] Michael zur Muehlen and Keith D. Swenson, "BPAF: A Standard for the Interchange of Process Analytics Data," Lecture Notes in Business Information Processing, Vol. 66, pp. 170-181, 2011.
- [18] IEEE, "IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams," IEEE 1849-2016, 2016. 10.1109/IEEESTD.2016.7740858
- [19] Kim, Kwanghoon Pio, "Discovering Activity-Performer Affiliation Knowledge on ICN-based Workflow Models," JOURNAL OF INFORMATION SCIENCE AND ENGINEERING, Vol. 29, No. 1, pp. 79-97, 2013.
- [20] Kim, Kwanghoon Pio, "Human-Centered Workflow Intelligence," Tutorial at the International Conference on Advanced Communications Technology, Phoenix Park, Pyeongchang, South Korea, Feb. 2014.
- [21] Kim, Kwanghoon Pio, "A Stochastic Workflow Performer-to-Activity Affiliation Network Model," The Proceedings of the 28th Annual Conference of Biomedical Fuzzy System Association (BMFSA2015), Kumamoto, Japan, 21-22 Nov. 2015.
- [22] Kim, Kwanghoon and Ellis, Clarence A., "Section II / Chapter VII. An ICN-based Workflow Model and Its Advances," HANDBOOK OF RESEARCH ON BUSINESS PROCESS MODELING, pp. 142-172, IGI Global, ISR, 2009.
- [23] Snijders, Tom A.B. , Bunt, Gerhard G. van de, Steglich, Christian E.G., "Introduction to Stochastic Actor-Based Models for Network Dynamics," SOCIAL NETWORKS, Vol. 32, pp. 44-60, 2010.



Seong-Hun Ham is a full-time M.S. student in the division of computer science and engineering as well as a research member of the data and process engineering research laboratory at Kyonggi University, South Korea. He received B.S. degree in computer science from Kyonggi University in 2019. His research interests include predictive process monitoring, process mining, Workflow System and process model discovery and analysis, process-aware Internet of Things architecture and services, and deep-learning-based process predicting.



Dinh-Lam Pham is a full-time PhD. student in Kyonggi University, South Korea since March, 2018. He got the MSc in computer science, Thainguyen University, Vietnam (2010). He worked in Thainguyen University from 2008 to 2016 as an IT engineering. From 2016, he works in the Information Technology Institute, Vietnam National University, Hanoi. His research interests include process mining, workflow management system, social network, intelligent process, information control net.



Kyoung-Sook Kim is a part-time professor in the division of computer science and engineering as well as a research member of the data and process engineering research laboratory at Kyonggi University, South Korea. She received B.S. degree in computer science from Kyonggi University in 1984. She also received M.S. degree and Ph.D. degree in computer engineering from Kyunghee University in 2001 and 2019, respectively. She had worked as a researcher and developer at FAST Systems Inc. and OSI Corp. in South Korea. Her research interests include large-scale database systems and applications, enterprise information systems, the Internet of Things collaboration architectures and services, process-aware knowledge discovery, process mining, and process monitoring and predictive monitoring.



Kwanghoon Pio Kim is a full professor of computer science department and the founder and supervisor of the data and process engineering research laboratory at Kyonggi University, South Korea. He received B.S. degree in computer science from Kyonggi University in 1984. And he received M.S. degree in computer science from Chungang University in 1986. He also received his M.S. and Ph.D. degrees from the computer science department at University of Colorado Boulder, in 1994 and 1998, respectively. At Kyonggi University, he is Dean of the Computerization and Informatics Institute, and the director of the contents convergence software research center, as well. He had worked as researcher and developer at Aztek Engineering, American Educational Products Inc., and IBM in USA, as well as at Electronics and Telecommunications Research Institute (ETRI) in South Korea. In present, he is a vice-chair of the BPM Korea Forum. He has been in charge of a country-chair (Korea) and ERC vice-chair of the Workflow Management Coalition. He has also been on the editorial board of the journal of KSII, and the committee member of the several conferences and workshops. His research interests include groupware, workflow systems, BPM, CSCW, collaboration theory, Grid/P2P distributed systems, process warehousing and mining, workflow-supported social networks discovery and analysis, process-aware information systems, data intensive workflows, process-aware Internet of Things, deep-learning frameworks, and predictive process monitoring and modeling.