An ICN-Process Graph Pattern Mining Algorithm*

Kyoungsook Kim†
Kyonggi University
Suwon-si Gyeonggi-do, South Korea
khmjmc@kgu.ac.kr

Dinh-Lam Pham‡
Kyonggi University
Suwon-si Gyeonggi-do, South Korea
phamdinhlam@kgu.ac.kr

Kwanghoon Pio Kim§
Kyonggi University
Suwonsi Gyeonggido, South Korea
kwang@kgu.ac.kr

ABSTRACT

This paper develops an algorithm that is able to discover the process graph patterns of the information control nets from process enactment event logs, implements the algorithm as a process graph mining system, and carries out a couple of operational experiments with process event log datasets. The core is the ρ-Algorithm that ought to be a novel approach for rediscovering all the structured process graph patterns, such as linear (sequential), disjunctive (selective-OR), conjunctive (parallel-AND), and repetitive (iterative-LOOP) process graph patterns, of the information control nets from an XES-formatted process enactment event log dataset. We prove that the proposed process graph pattern mining algorithm is able to complete the information control net process rediscovery functionalities successfully through fulfilling a series of experimental studies. Additionally, we discuss those future issues of the process graph pattern discovery and rediscovery in the workflow and business process intelligence literature.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

Workflow and Business Process Models; Structured Information Control Nets; Process Graph Patterns; Process Discovery; Process Rediscovery; Process Graph Pattern Mining Algorithms

ACM Reference Format:
Kyoungsook Kim, Dinh-Lam Pham, and Kwanghoon Pio Kim. 2019. An ICN-Process Graph Pattern Mining Algorithm. In *Proceedings of ACM international workshop on mining and learning with graphs (MLG'19).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.475/123_4

*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).
†Adjunct Professor in the Division of Computer Science and Engineering
‡Ph. D. Student in the Division of Computer Science and Engineering
§Corresponding Author, Professor in the Division of Computer Science and Engineering

1 INTRODUCTION

A workflow process[1] management system is able to support two fundamental functionalities|modeling functionality and enacting functionality. The modeling functionality allows modelers to define, analyze and maintain processes by hooking all the essential entities, such as activities, roles, performers, relevant data and invoked applications, on the corresponding procedures, whereas the enacting functionality supports performers to play the essential roles of invoking, executing and monitoring all the instances of the processes. As we all know, the logical foundation of such a process management system is based upon its underlying process modeling methodology, which implies that the system is able to automate definition, creation, execution, and management of processes according to the internal principle and structure of the process graph patterns. Until now, several process modeling methodologies [7][2] have been proposed in the workflow and business process literature, and almost all of them employ the five essential entity-types, such as activity, role, performer, repository and application entity-types, to represent organizational works and their procedural collaborations. We turn our attention to the activity entities and their temporal orderings with process graph patterns in this paper.

There possiblly exist two main branches of the research issues on the process graph patterns and models. One is so-called *discovery* [1] issues, the other is so-called *rediscovery* [6][10] issues. The former is to discover a process graph model through analyzing and mining activity entities and their temporal orderings from the specific event log histories of the legacy information systems, whereas the latter is concerned with mining a process graph model from the process enactment event log histories under the control of a specific workflow and business process management system, the topic of which is named as ***rediscovery***. More specifically we would differentiate the former from the latter; the former is to explore process patterns and models as a process planning and defining activity, while on the other hand the latter is to explore process patterns and models as a reengineering and redesigning activity. The paper is directly related with the process patterns and models rediscovery issue, which means that the proposed algorithm is aiming at mining a process graph model of information control nets from a specific dataset of process enactment event log histories. Conclusively, the purpose of this paper is to originate a fundamental mining and visualizing principle for rediscovering process graph patterns and models from those datasets recorded as process enactment event logs formatted in the IEEE XES [5] standardized XML event streaming language.

[1]In terms of the terminological usage, the term, workflow process, can be interchangeably used with the term, business process. We prefer to use the term, process, simply in this paper.

2 RELATED WORKS

The main research challenge of this paper is to devise an algorithmic process mining framework for rediscovering a model of structured information control nets from a workflow process enactment event log dataset. Therefore, the literature surveys of the challenge are summarized in this chapter of related works. The most popular approaches of the theoretical modeling methodologies to formally define and graphically represent business process models are the Petri-net process graph model [11] of process modeling methodology and the information control net process graph model [2] of process modeling methodology. Both of them are based upon the mathematical representation and the graphical representation at the same time. The Petri-net process graph model has a strong advantage in terms of the mathematical and analytical power, whereas the information control net process graph model has a much stronger merit in terms of the expressiveness of the workflow process domain. So far, there have been several process graph mining algorithms in the literature. One of the typical process graph mining algorithms for rediscovering the Petri-net process graph models is the α(alpha)-Algorithm [10], whereas the typical process graph mining algorithm for rediscovering the information control net process graph models is the σ(sigma)-Algorithm [6]. Note that the name of the process graph mining algorithm proposed in the paper is the ρ(rho)-Algorithm, and the naming reason of the ρ-Algorithm will be explained later. The crucial idea and characteristics of these algorithms and their comparisons with the ρ-Algorithm are arranged in this section.

The essential goal of the paper is concerned with the concept of process graph mining algorithm that is able to completely rediscover a structured information control net process model with process graph patterns from the process enactment event logs. For the sake of eventually supporting the model-log comparison [4], which is the process fidelity issue, the literature has produced so far the two concepts and theories; one is the Petri-net process graph modeling methodology and the other is the information control net process graph modeling methodology. Also, the literature has published the α-Algorithm [10] and the σ-Algorithm [6]for discovering Petri-net process graph models and information control net process graph models, respectively. However, all of these rediscovery approaches have the limitations that they are able to deal with only sequential, parallel and selective process graph patterns out of all the types of the process graph patterns such as sequential, parallel, selective and repetitive process graph patterns. In other words, both of the rediscovery algorithms have a limitation in dealing with rediscovering the repetitive-LOOP process graph patterns. Note that the α-Algorithm is implicitly dealing with the repetitive-LOOP process graph pattern as the selective process graph pattern. Conclusively speaking, the algorithm (named as the ρ-Algorithm) proposed in the paper is able to explicitly deal with rediscovering not only the three types of the primitive process graph patterns supported by the conventional algorithms but also the most challenging type of the repetitive-LOOP process graph pattern.

3 THEORETICAL BACKGROUND: THE PROCESS GRAPH MODEL OF STRUCTURED INFORMATION CONTROL NETS

The theoretical background is the information control net methodology [7] that is a typical workflow modeling approach supporting graphical and formal representations. In defining a workflow procedure, the methodology uses the basic workflow entity types–activity, role, actor/performer, invoked application and transition condition–to represent the procedural properties of workflow, such as control-flow and data-flow, as well as the associative properties of workflow such as activity-to-role, role-to-performer, activity-to-condition/rule, and activity-to-application associations. In this section, we define only the control-flow aspect of the information control net of a workflow process model through the following [*Definition 3.1*] for formally representing the process graph patterns and models.

Definition 3.1. Information Control Net (ICN) for formally defining a workflow model. A basic **ICN** is 8-tuple $\Gamma = (\delta, \rho, \lambda, \varepsilon, \pi, \kappa, \mathbf{I}, \mathbf{O})$ over a set of **A** activities (including a set of group activities), a set **T** of transition conditions, a set **D** of data repositories, a set **G** of invoked application programs, a set **R** of roles, and a set **P** of performers (including a set of performer groups), where

(1) External Data Properties
- **I** is a finite set of initial input repositories, which is assumed to be loaded from some external workflow before execution;
- **O** is a finite set of final output repositories, which is assumed to be transferred to some external workflow after execution;

(2) Procedural Properties
- $\delta = \delta_i \cup \delta_o$
  where, $\delta_o : \mathbf{A} \longrightarrow \wp(\mathbf{A})$[2] is a multi-valued function mapping an activity to its sets of (immediate) successors, and $\delta_i : \mathbf{A} \longrightarrow \wp(\mathbf{A})$ is a multi-valued function mapping an activity to its sets of (immediate) predecessors;
- $\kappa = \kappa_i \cup \kappa_o$
  where $\kappa_i : \mathbf{A} \longrightarrow \wp(\mathbf{T})$ is a multi-valued function mapping a set of control-transition conditions, **T**, on directed arcs, $(\delta_i(\alpha), \alpha \in \mathbf{A})$ between $\delta_i(\alpha)$ and α; and $\kappa_o : \mathbf{A} \longrightarrow \wp(\mathbf{T})$ is a multi-valued function mapping a set of control-transition conditions, **T**, on directed arcs, $(\alpha \in \mathbf{A}, \delta_o(\alpha))$ between α and $\delta_o(\alpha)$;

Starting and Terminating Event Nodes. Additionally, the execution of a workflow model commences with a single χ transition-condition. So, we always assume without loss of generality that there is a single starting event node (∇:α_I). At the commencement, it is also assumed that all the input repositories in the set $\boldsymbol{I}$ have been initialized with relevant data from its external workflow model:

$$\exists \alpha_I \in \boldsymbol{A} \mid \delta_i(\alpha_I) = \{\emptyset\} \land \kappa_o(\alpha_I) = \{\{\chi\}\}.$$

The execution is terminated with any one λ output transition-condition. Also we assume without loss of generality that there is a

[2] $\wp(\mathbf{A})$ is the powerset of **A**.

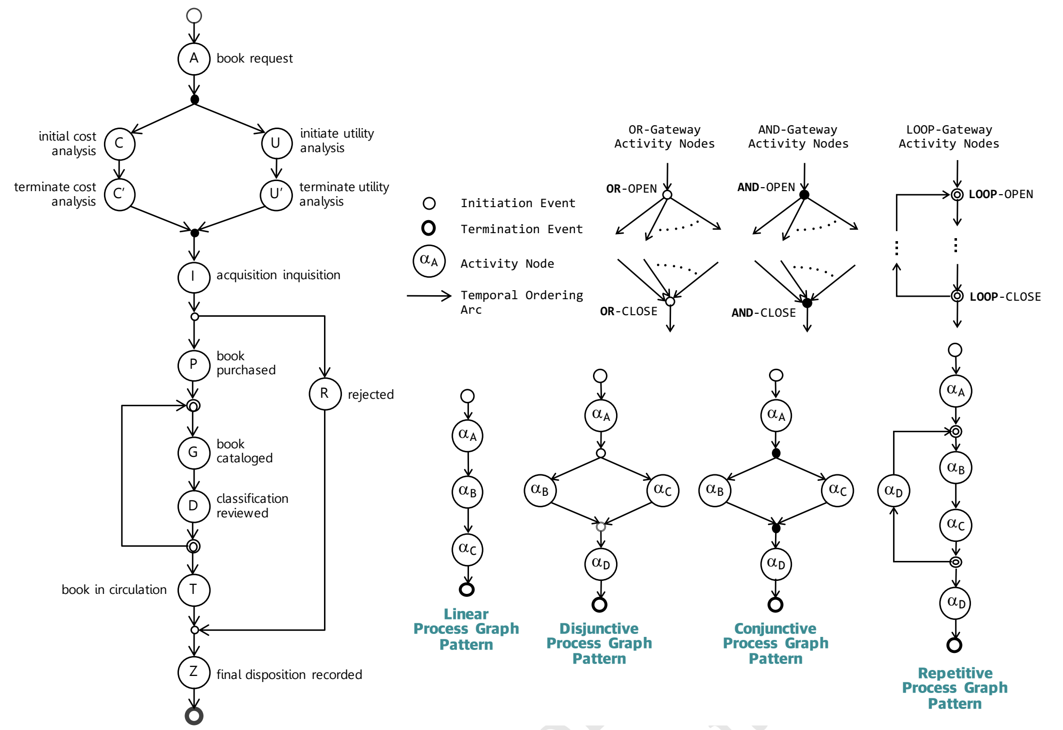
A book request
C initial cost analysis
U initiate utility analysis
C' terminate cost analysis
U' terminate utility analysis
I acquisition inquisition
P book purchased
R rejected
G book cataloged
D classification reviewed
T book in circulation
Z final disposition recorded
Initiation Event
Termination Event
Activity Node
Temporal Ordering Arc
OR-Gateway Activity Nodes
OR-OPEN
OR-CLOSE
AND-Gateway Activity Nodes
AND-OPEN
AND-CLOSE
LOOP-Gateway Activity Nodes
LOOP-OPEN
LOOP-CLOSE
Linear Process Graph Pattern
Disjunctive Process Graph Pattern
Conjunctive Process Graph Pattern
Repetitive Process Graph Pattern

Figure 1: Process Graph Patterns of Structured Information Control Net: the Library Book Acquisition Process Model

single terminating event node ($\triangle{:}\alpha_F$). A set of the output repositories $\boldsymbol{O}$ is a group of data holders that is transferred to the external workflow model after after termination:

$$\exists \alpha_F \in \boldsymbol{A} \mid \delta_o(\alpha_F) = \{\emptyset\} \wedge \kappa_i(\alpha_F) = \{\{\chi\}\}.$$

Process Graph Patterns with Gateway Activities: Temporal Orderings of Activities. Given a formal definition, the temporal ordering of activities in a process graph model can be formally represented as follows: For any activity α, in general

$$\delta(\alpha) = \{ \\ \{\beta_{11}, \beta_{12}, \ldots, \beta_{1m(1)}\}, \\ \{\beta_{21}, \beta_{22}, \ldots, \beta_{2m(2)}\}, \\ \ldots, \\ \{\beta_{n1}, \beta_{n2}, \ldots, \beta_{nm(n)}\} \\ \}$$

which can be interpreted as follows:

- Upon the completion of activity α with its single incoming transition, it simultaneously initiates all of the activities β_{i1} through $\beta_{im(i)}$, then all the initiated transitions are called parallel (conjunctive) transitions;
- After completion of activity α with its single incoming transition, only one value out of i ($1 \leq i \leq n$) is selected as the result of a decision made, then the selected transition is called a selective-decision (disjunctive or exclusive-OR) transition;
- After completion of activity α with its single incoming transition, if $n = 1 \wedge m(n) = 1$, then neither decision nor parallel is needed, and the transition is called a sequential transition;
- After completion of activity α with more than one incoming transition, if $n > 1 \wedge m(n) = 1$, then only one value out of i ($1 \leq i \leq n$) is selected as the result of a decision made, then the selected transition is called a loop-decision (repetitive) transition;
- Additionally stating to make sure, if $m(i) = 1$ for all i, then no parallel processing is initiated by completion of α.

Graphical Formation of the Process Graph Patterns. Based on the interpretation, we graphically define these primitive open-transition types. The conjunctive (or parallel) outgoing transitions are connected by a solid dot(•) with a single incoming transition, the disjunctive (or selective-decision) outgoing transitions are connected by a hollow dot(◦) with a single incoming transition, and the two repetitive incoming transitions are connected by a double-hollow dot with a single outgoing transition. For the primitive close-transition types, the graphical formations vice-versa. Besides, these special types of nodes are called gateway activity nodes in the workflow and business process literature. The starting and the terminating event nodes are medium-sized circle with thin line and medium-sized circle with thick line, respectively. Additionally, the gateway activity nodes need to be formed in a matched pair of open and close types. Also, multiple sets of the open-close gateway nodes should be kept in a properly nested form for supporting syntactically safeness and soundness. Summarily, Figure 1, as an exemplary process graph model, depicts a structured information control net of the Library Book Acquisition Workflow Procedure firstly introduced in [2]. The left-hand side of the figure is the control flow aspect of the information control net that consists of a

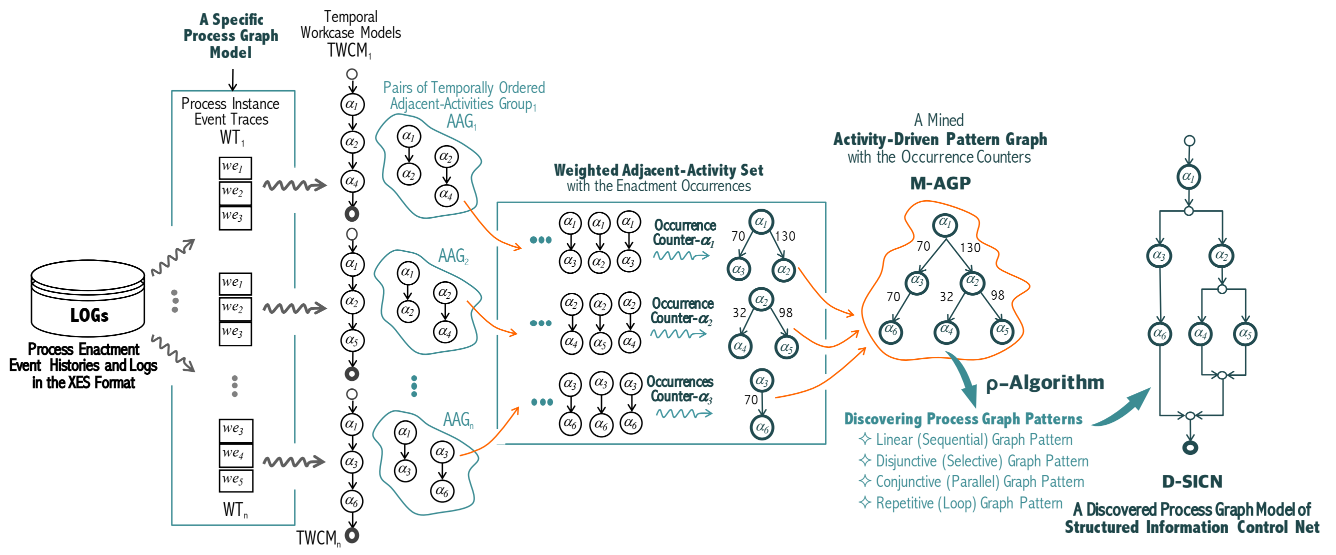
A Specific Process Graph Model
Temporal Workcase Models TWCM₁
Process Instance Event Traces WT₁
Pairs of Temporally Ordered Adjacent-Activities Group₁
AAG₁
AAG₂
AAGₙ
LOGs
Process Enactment Event Histories and Logs in the XES Format
WTₙ
TWCMₙ
Weighted Adjacent-Activity Set with the Enactment Occurrences
Occurrence Counter-α₁
Occurrence Counter-α₂
Occurrences Counter-α₃
A Mined Activity-Driven Pattern Graph with the Occurrence Counters
M-AGP
ρ-Algorithm
Discovering Process Graph Patterns
✧ Linear (Sequential) Graph Pattern
✧ Disjunctive (Selective) Graph Pattern
✧ Conjunctive (Parallel) Graph Pattern
✧ Repetitive (Loop) Graph Pattern
D-SICN
A Discovered Process Graph Model of Structured Information Control Net

Figure 2: A Stepwise Representation of the Process Graph Patterns Mining Algorithm

starting node, a terminating node, 12 activities, a pair of open and close parallel gateway nodes, a pair of open and close selective decision gateway nodes, and a pair of open and close repetitive gateway nodes.

4 A PROCESS GRAPH PATTERN MINING ALGORITHM

The overall algorithmic approach, as illustrated in Figure 2, is a stepwise mining procedure with the functional components to be used for discovering all the types of the primitive process graph patterns that constitute a structured information control net process model. The approach is named as ρ-Algorithm and supported by a series of stepwise transformation algorithms for discovering structured information control net process models from the process enactment event logs. The first transformation algorithm is to discover the enacted workcases from the event logs, each of which can be modeled into a temporal workcase model. At the same time, it is necessary to count the occurrence of each temporal workcase with its activities. The second transformation algorithm is to discover an activity-driven pattern graph by integrating all the members of the adjacent-activity set and calculating the occurrences of the temporal workcases. In terms of discovering the structured information control net process model from the corresponding activity-driven pattern graph, we develop an algorithm that is able to deal with an any combinational number of AND/OR/LOOP process graph patterns. In this figure, we show only the Exclusive-OR (disjunctive process graph pattern), as an example. Here, we start to formally describe the process graph pattern mining algorithm from defining a formal structure of the process enactment event logs as defined in the following definitions of *Definition 4.1* and *Definition 4.2*:

Definition 4.1. Workflow Process Workitem Enactment Event. Let $\boldsymbol{we} = (\alpha, pc, wf, wc, ac, p^*, t, s)$ be a workflow process workitem enactment event stored as logs, where

- α is a workitem (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[3] is a participant (or performer) identifier,
- t is a timestamp, and
- s is a workitem's current state, which is one of the states such as *ready, assigned, reserved, running, completed,* and *cancelled.*

Definition 4.2. Temporal Workcase Model. A temporal workcase model is formally defined through 3-tuple $TWCM = (\omega, F_r^c, T_o^c)$ over a set $\mathbf{A}$ of activity trace-nodes, $\forall\eta_\alpha^{\tau[.\phi]}$, on a temporal workcase, $TWC(\mathbf{c})$, of a workflow instance (workcase), $\mathbf{c}$, and a species $\mathbf{K}$ $(= \{s, e, u, o\})$ of the timestamp-origins, where

- F_r^c is an activity or an activity-group linked from an external temporal workcase model;
- T_o^c is an activity or an activity-group linked to an external temporal workcase model;
- $\omega = \omega_i \cup \omega_o$ on $\forall\eta_\alpha^{\tau[.\phi]} \in \mathbf{A}$,
  - $\omega_o : \mathbf{A} \longrightarrow \wp(\mathbf{A})$ is a single-valued mapping function of an activity trace-node, $\eta_\alpha^{\tau[.\phi]} = we_\alpha^{\tau[.\phi]} \land \phi \in \mathbf{K}$, to its (immediate) successor in a temporal workcase;
  - $\omega_i : \mathbf{A} \longrightarrow \wp(\mathbf{A})$ is a single-valued mapping function of an activity trace-node, $\eta_\alpha^{\tau[.\phi]} = we_\alpha^{\tau[.\phi]} \land \phi \in \mathbf{K}$, to its (immediate) predecessors in a temporal workcase.
- The species of temporal workcase models: $TWCM^\phi$
  - ScheduledTime Temporal Workcase Model: ϕ = ‘s‘ *in* $\forall\eta_\alpha^{\tau.\phi}$ of a temporal workcase model

[3]Note that * indicates multiplicity.

- AssessedTime Temporal Workcase Model: ϕ = 'e' in $\forall\eta_{\alpha}^{\tau.\phi}$ of a temporal workcase model
- StartedTime Temporal Workcase Model: ϕ = 'u' in $\forall\eta_{\alpha}^{\tau.\phi}$ of a temporal workcase model
- CompletedTime Temporal Workcase Model: ϕ = 'o' in $\forall\eta_{\alpha}^{\tau.\phi}$ of a temporal workcase model

STEP-1: Groups of Temporally Ordered Adjacent-Activity Pairs. From now on, the first step of the ρ-Algorithm is to mine a group of temporally ordered adjacent-activities pairs from temporal workcases (*Definition 4.2*) of the process instance event logs (*Definition 4.1*). Also, each of the temporal workcases is formally represented by one of the workcase model types introduced in the previous section. That is, a temporal workcase represents an ordered enactment sequence of activity event logs, each of which is formed with its activity identifier and its time-stamp extracted from its corresponding process enactment event log. The extracted temporal workcase is used for formally defining its workcase model ($TWCM_1$), from which the STEP-1 algorithm is able to mine a group (AAG_1) of temporally ordered adjacent-activity pairs belonging to a corresponding process instance event trace.

STEP-2: Weighted Adjacent-Activity Set and Activity-Driven Pattern Graph. The STEP-2 of the ρ-Algorithm is to build all th groups of temporally ordered adjacent-activity pairs, each of which corresponds to a process instance event trace. The eventual output of this algorithm is a weighted adjacent-activity set named as adjacencyList β. This set is built from all the groups of temporally ordered adjacent-activity pairs through an internal transformation procedure. Especially, the concept of weight on an edge with its connected activities of each pair implies the number of occurrences of the corresponding pair and its connected activities. Also, the set is used to produce an activity-driven pattern graph with all the edges' weights and their activities' weights through an internal transformation algorithm.

STEP-3: Discovering Process Graph Patterns of Structured Information Control Nets. The final step (STEP-3) of the ρ-Algorithm is to discover a process pattern graph of a structured information control net process model from the activity-driven pattern graph mined from all the groups of temporally ordered adjacent-activity pairs. The eventual goal of the ρ-Algorithm is accomplished through this step. Note that the structured information control net process model must be satisfied with the proper nesting as well as the matched pairing properties in forming gateway activities in each process graph pattern. Therefore, the STEP-3 algorithm is able to completely transform from an activity-driven pattern graph as the outcome of the STEP-2 algorithm to a structured information control net model with the occurrences on every branch of the associated gateway activities. The Algorithm 1 is the pseudo codes of the STEP-3 of the ρ-Algorithm. The STEP-3 algorithm performs two essential functions. One (STEP-3.1) is a graphical visualization function to visualize the activity-driven pattern graph in a form of graphical viewer using the Graph Stream Library. The other (STEP-3.2) is a visual transformation function to transform the activity-driven pattern graph into a graphical viewer of the structured information control net process model formed by a certain combination of the four types of process graph patterns. As described in the previous step, the activity-driven pattern graph is built by performing the internal transformations from adjacencyList β. In terms of visualizing the activity-driven patterns and its information control nets, the Graph Stream Library is used. The core part of the ρ-Algorithm is the STEP-3.2 function that decides open-gateways and close-gateways of each process graph pattern in the graph by using the concept of rho (ρ)[4].

Summarily, the eventual goal of the ρ-Algorithm proposed in this paper is to discover structured information control net process model being composed of a certain combination of all the process graph patterns from a dataset of process enactment event logs through deploying the ρ-Algorithm with a series of concepts and their related algorithms like STEP-1, STEP-2 and STEP-3 algorithms. Until now, the paper has simply described those conceptual and procedural formal definitions, such as the group of temporally ordered adjacent-activity pairs, the weighted adjacent-activity set, the activity-driven pattern graph and the structured information control net process model. Therefore, it is necessary to summarily show that the two goals of the paper have been achieved by not only discovering a structured information control net process model as one goal of the process graph mining aspect but also discovering its enactment occurrences on each of the process graph patterns as the other goal of the process-aware knowledge discovery aspect.

5 EXPERIMENTAL STUDIES

In this section, we carry out a couple of experimental analyses to verify the correctness of the process graph pattern mining algorithm based upon the implemented ρ-Algorithm. In the previous research [8], the authors' research group fulfilled a feasibility analysis of the proposed algorithm by applying the algorithmic steps to the exemplary dataset of very large scale process event logs. Through the operational feasibility analysis, we had shown that the functional correctness of the ρ-Algorithm ought to be reasonable and applicable. However, it is necessary for us to carry out an additional experimental analysis for the specialized dataset of a real process event logs and traces not only to verify the correctness of theρ-Algorithm, but also to see how the ρ-Algorithm operates and works on the specialized dataset. Due to the page limitation, we won't describe the details of the implemented process graph pattern mining system of the ρ-Algorithm.

(1) *Preparation of the Process Event Log Dataset.* According as a process instance is executed, the logging and auditing component of the process enactment engine records its workitem execution events on a log repository, and those logged events are arranged in a form of the temporal sequence of events. This execution sequence of a process instance is forming a process instance event trace, from which we can extract the process instance's workitem event trace and its formal representation is specified by a concept of temporal workcases and its workcase model as defined in the previous section. We already implemented the event trace mining algorithm (the STEP-2 algorithm of the ρ-Algorithm) by considering a formal structure of the process activity enactment (workitem) event log. In general, the log formats for the process enactment event log used to

[4] In the ρ-Algorithm, the symbol and name of rho (ρ) comes from the A programming language (APL) firstly released in 1960's. The function rho, coded like ρX in APL, implies that it gives the number of elements in X, from which the concept of ***mass*** comes. The central idea of the discovery algorithm of the framework is exactly same to the implication of the APL function, rho (ρ).

Algorithm 1 ρ-ALGORITHM

```
Require: An Activity-Driven Pattern Graph - $\mathbf{G}$
Ensure: A Structured Information Control Net Process Graph - $\mathbf{G}^{SICN}$
 1: procedure $\rho$-MAIN($\mathbf{G}$)                ▷ Mining the structured information control net process model
 2:     $\mathbf{G}$.removePhantomEdge();                   ▷ Removing the phantom edges on $\mathbf{G}$
 3:     for $\forall\alpha \in \mathbf{G}$ do                  ▷ $\alpha$ is a member node of the node-set in $\mathbf{G}$.
 4:         $\alpha$.listEdgeOutGoing ← $\mathbf{G}$.getEdgeOutGoing($\alpha$);
 5:         if $\alpha$.listEdgeOutGoing.size() > 1 then
 6:             $\mathbf{G}^{open}$ ← ProcessForTheOpenGate($\mathbf{G}$, $\alpha$, $\alpha$.listEdgeOutGoing);   ▷ Discovering the open-gateways in $\mathbf{G}$
 7:         end if
 8:         $\alpha$.listEdgeInComing ← $\mathbf{G}$.getEdgeInComing($\alpha$);
 9:         if $\alpha$.listEdgeInComing.size() > 1 then
10:             $\mathbf{G}^{close}$ ← ProcessForTheCloseGate($\mathbf{G}$, $\alpha$, $\alpha$.listEdgeInComing);   ▷ Discovering the close-gateways in $\mathbf{G}$
11:         end if
12:     end for
13:     $\mathbf{G}^{SICN}$ ← ProcessForTheLoopGate($\mathbf{G}^{open}$, $\mathbf{G}^{close}$);   ▷ Discovering the loop-gateways from $\mathbf{G}^{open}$ and $\mathbf{G}^{close}$
14:     return $\mathbf{G}^{SICN}$        ▷ Discovered a structured information control net process model, $\mathbf{G}^{SICN}$
15: end procedure
```

be a tag-based language like XWELL [9], BPAF, and XES. In recent, IEEE released a standard tag-based language, XES [5], 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. Conclusively, as the format of the process enactment event log structure, those event history files used the IEEE XES Schema 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 IEEE XES schema, the STEP-2 algorithm was developed, implemented and applied to all those event history data-sets. For the sake of the experimental study, we prepare a couple of process enactment event log datasets that are available in the 4TU.Centre for Research Data [3] and that are well-fitted and appropriate for carrying out the experimental study. First of all, it is necessary for the ρ-Algorithm to be checked up on the correctness of the basic discovery functionality supporting the primitive types of process graph patterns such as linear (sequential), disjunctive (exclusive-OR), conjunctive (parallel-AND), and repetitive (iterative-LOOP) process patterns.

(2) *Linear and Exclusive-OR Process Graph Pattern Discovery.* Figure 3 shows the 3 captured-screens of the experimental results being produced from the dataset of ETM-Configuration2.xes [3]. The dataset contains the total 70 process instance event traces (instance event traces, and the following) and the total 7 associated-activities in the underlying process model. The 3 captured-screens correspond to the Groups of Adjacent-Activity Pairs and Weights in the Adjacent-Activity Set, Activity-driven Pattern Graph, Structured Information Control Net Process Model with a matched pair of exclusive-OR open-gateway node and exclusive-OR close-gateway node, respectively.

(3) *Iterative-LOOP Process Graph Pattern Discovery.* Figure 4 shows a single captured-screen of the experimental results being produced from the dataset of Review-Example-Large.xes [3]. This experiment deploys the implemented system to only a single instance event trace of the dataset. It also verifies that the ρ-Algorithm is able to discover a structured information control net process model from not all the instance event traces but only a single instance event trace. The two captured-screens are overlapped and correspond to the Group of Adjacent-Activity Pairs for the instance event trace and the ultimate discovery result, a Structured Information Control Net Process Model of process graph patterns with 5 matched pairs of iterative-LOOP gateway-nodes, respectively.

(4) *Parallel-AND Process Graph Pattern Discovery.* Figure 5 shows the 4 captured-screens of the experimental results being produced from the dataset of ETM-Configuration1.xes [3]. The dataset contains the total 100 instance event traces and the total 7 associated-activities in the underlying business process model. The 4 captured-screens correspond to the Groups of Adjacent-Activity Pairs and Weights Adjacent-Activity Set, Activity-driven Pattern Graph after eliminating the phantom pairs, and Structured Information Control Net Process Model with nested and matched pairs of parallel-AND gateway-nodes and exclusive-OR gateway-nodes, respectively.

6 CONCLUSION

So far, this paper has proposed the process graph pattern mining algorithm and performed a series of experimental analyses based upon the real process enactment event log datasets. The theoretical background of the proposed algorithm stems from the conceptual rediscovery approach of rediscovering the process graph patterns such as linear, disjunctive, conjunctive and repetitive process patterns from the process-aware warehouses and datasets, whereas the implementable background of the proposed algorithm is supported by the algorithmic rediscovery approach of the STEP-1, STEP-2 and STEP-3 algorithms, named as the ρ-Algorithm, by discovering the structured information control net process models from the process enactment event log datasets. Based upon these theoretical and algorithmic approaches, the paper devised, implemented and developed these concepts, algorithms and systems, respectively. Based upon these implemented algorithms, the paper carried practically out a series of experimental studies by deploying them onto a series of process enactment event log data-sets provided by the 4TU.Centre for Research Data. As the future research of the paper, the process

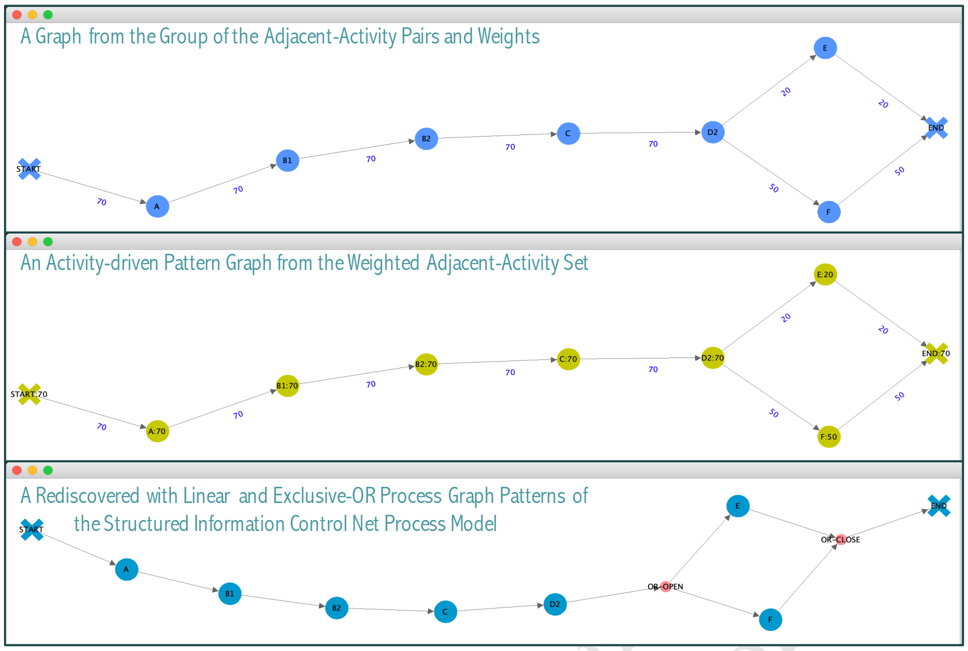
A Graph from the Group of the Adjacent-Activity Pairs and Weights
START
A
B1
B2
C
D2
E
F
END
70
20
50
An Activity-driven Pattern Graph from the Weighted Adjacent-Activity Set
START:70
A:70
B1:70
B2:70
C:70
D2:70
E:20
F:50
END:70
A Rediscovered with Linear and Exclusive-OR Process Graph Patterns of the Structured Information Control Net Process Model
OR-OPEN
OR-CLOSE

Figure 3: An Experimental Result (Sequential and Selective Process Graph Patterns) of the Process Graph Patterns Mining Algorithm

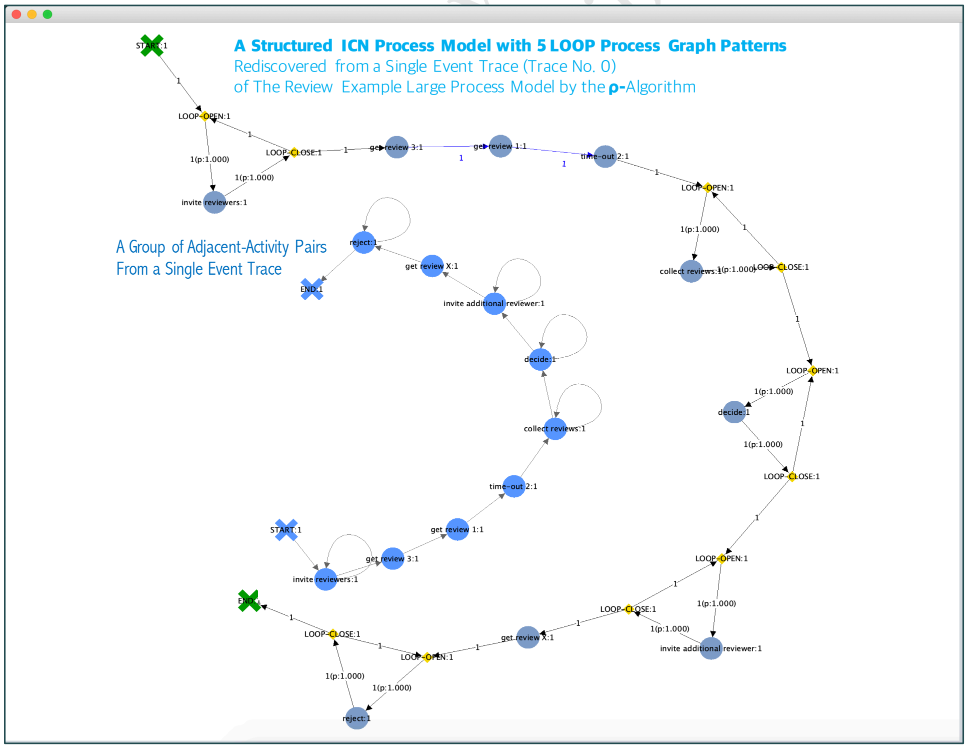
A Structured ICN Process Model with 5 LOOP Process Graph Patterns
Rediscovered from a Single Event Trace (Trace No. 0)
of The Review Example Large Process Model by the ρ-Algorithm
A Group of Adjacent-Activity Pairs
From a Single Event Trace
START:1
LOOP-OPEN:1
LOOP-CLOSE:1
invite reviewers:1
get review 3:1
get review 1:1
time-out 2:1
collect reviews:1
decide:1
get review X:1
invite additional reviewer:1
reject:1
END:1
1(p:1.000)
1

Figure 4: An An Experimental Result (Repetitive Process Graph Patterns) of the Process Graph Patterns Mining Algorithm

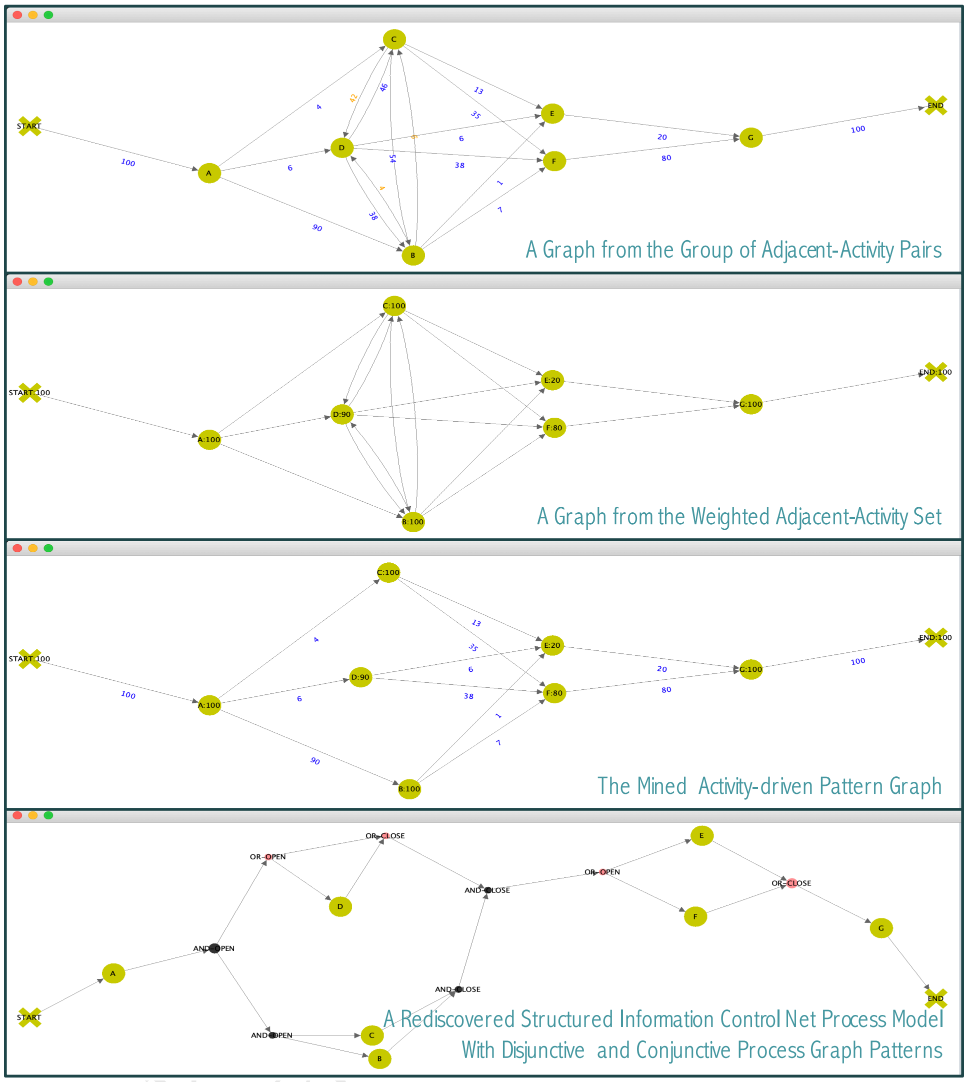
A Graph from the Group of Adjacent-Activity Pairs
A Graph from the Weighted Adjacent-Activity Set
The Mined Activity-driven Pattern Graph
A Rediscovered Structured Information Control Net Process Model
With Disjunctive and Conjunctive Process Graph Patterns
START
END
START:100
A:100
B:100
C:100
D:90
E:20
F:80
G:100
END:100
AND-OPEN
AND-CLOSE
OR-OPEN
OR-CLOSE

Figure 5: An An Experimental Result (Parallel Process Graph Patterns) of the Process Graph Patterns Mining Algorithm

graph pattern mining algorithm will be extended to discover all the work transference graph patterns of the performers' involvements, which will be especially the future research challenge of the paper.

REFERENCES

[1] Ellis C.A., Rembert A.J., Kim KH., and Wainer J. 2006. Beyond Workflow Mining. *In: Dustdar S., Fiadeiro J.L., Sheth A.P. (eds) Business Process Management. BPM 2006. Lecture Notes in Computer Sciences* 4102 (2006), 49–64. https://doi.org/10.1007/11841760_5

[2] Clarence A. Ellis, Kwanghoon Kim, Aubrey Rembert, and Jacques Wainer. 2012. Investigations on Stochastic Information Control Nets. *Information Sciences* 194 (July 2012), 120–137.

[3] 4TU Centre for Research Data. 2012, 2013, 2014, 2015, 2016, 2017, 2018. BPI Challenges. BPM. (2012, 2013, 2014, 2015, 2016, 2017, 2018).

[4] Daniela Grigori, Fabio Casati, Malu Castellanos, Umeshwar Dayal, Mehmet Sayal, and Ming-Chien Shan. 2004. Business Process Intelligence. *Journal of Computers in Industry* 53, 3 (2004).

[5] IEEE. 2106. IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams. IEEE 1849-2016. (2106).

[6] Kwanghoon Kim and Clarence A Ellis. 2007. σ-Algorithm: Structured Workflow Process Mining Through Amalgamating Temporal Workcases. *Zhou ZH., Li H., Yang Q. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2007. Lecture Notes in Computer Science* 4426 (2007), 119–130.

[7] Kwanghoon Kim and Clarence A. Ellis. 2009. *Section II / Chapter VII. An ICN-based Workflow Model and Its Advances, Handbook of Research on BP Modeling*. IGI Global, ISR, Hershey, Pennsylvania, USA.

[8] Kyoungsook Kim, Dinh-Lam Pham, Hyun Ahn, and Kwanghoon Kim. 2018. An experimental mining and analytics for discovering proportional process patterns from workflow enactment event logs. *WIRELESS NETWORKS* Online (2018), 1–8.

[9] Minjae Park and Kwanghoon Kim. 2007. XWELL: A XML-Based Workflow Event Logging Mechanism and Language for Workflow Mining Systems. *Lecture Notes in Computer Science* 4707 (2007). https://doi.org/10.1007/978-3-540-74484-9_76

[10] Wil Van der Aalst, Ton Weijters, and Laura Maruster. 2004. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge & Data Engineering* 9 (2004), 1128–1142.

[11] Boudewijn F Van Dongen, Ana Karla A de Medeiros, HMW Verbeek, AJMM Weijters, and Wil MP Van Der Aalst. 2005. The ProM framework: A new era in process mining tool support. In *International conference on application and theory of petri nets*. Springer, 444–454.