applied sciences

MDPI

Article

An Experimental Analytics on Discovering Work Transference Networks from Workflow Enactment Event Logs

Hyun Ahn, Dinh-Lam Pham and Kwanghoon Pio Kim *

Department of Computer Science and Engineering, Kyonggi University, Suwon 16227, Gyeonggi, Korea; hahn@kgu.ac.kr (H.A.); phamdinhlam@kgu.ac.kr (D.-L.P.)

* Correspondence: kwang@kgu.ac.kr; Tel.: +82-31-249-9670

Received: 21 May 2019; Accepted: 5 June 2019; Published: 10 June 2019

Abstract: Work transference network is a type of enterprise social network centered on the interactions among performers participating in the workflow processes. It is thought that the work transference networks hidden in workflow enactment histories are able to denote not only the structure of the enterprise social network among performers but also imply the degrees of relevancy and intensity between them. The purpose of this paper is to devise a framework that can discover and analyze work transference networks from workflow enactment event logs. The framework includes a series of conceptual definitions to formally describe the overall procedure of the network discovery. To support this conceptual framework, we implement a system that provides functionalities for the discovery, analysis and visualization steps. As a sanity check for the framework, we carry out a mining experiment on a dataset of real-life event logs by using the implemented system. The experiment results show that the framework is valid in discovering transference networks correctly and providing primitive knowledge pertaining to the discovered networks. Finally, we expect that the analytics of the work transference network facilitates assessing the workflow fidelity in human resource planning and its observed performance, and eventually enhances the workflow process from the organizational aspect.

Keywords: workflow management; work transference network; workflow enactment event logs; experimental analytics

1. Introduction

A workflow (or business process) management system can support two fundamental functionalities, namely, the modeling functionality and the enacting functionality. The modeling functionality allows modelers to define, analyze, and maintain the workflow processes by hooking all essential workflow entities, such as activities, roles, performers, relevant data, and invoked applications, on the corresponding procedures. By contrast, the enacting functionality supports performers to play the essential roles of invoking, executing, and monitoring all instances of the workflow processes. The logical foundation of such a workflow management system is based on its underlying workflow model, which implies that the system is able to automate the defining, creation, execution, and management of the workflow processes according to the internal principle and structure of the underlying workflow model. To date, several workflow models [1,2] have been proposed, almost all of which employ five essential entity types, namely, the activity, role, performer, data repository, and application, to represent organizational works and their procedural collaborations. In this study, we focused on the performer entity type.

In recent years, studies on workflow started focusing on people working in workflow-supported organizations because it is widely accepted for a workflow system to be a people system. By analyzing the interactive and collaborative behaviors among people involved in performing workflow processes, we are able to measure and estimate their overall performance in real businesses, as well as their work productivity. For more than a decade, workflow mining [3,4] has received significant attention as a key enabling technology for acquiring human-centered knowledge regarding workflow processes. In this regard, the authors' research group has proposed research and development issues when applying the concept of social networks and analysis methods into human-centered workflow knowledge discovery and analysis.

Under this context, we are particularly interested in the work transference network [5] among performers in a workflow-supported organization. More specifically, this network is established through work transference (or handover) relationships between two performers in charge of the preceding and following activities within the workflow process. As shown in Figure 1, there are two performers, $Jeff$ and $Adam$, in charge of the consecutive activities A and B, respectively. As the predecessor of this case, $Jeff$ will transfer the results of the execution of A to his successor $Adam$. Therefore, this type of relationship reflects the relevancy and intensity between performers in terms of working, and accordingly, can eventually be an important analytical property for the acquisition of human-centered workflow knowledge.

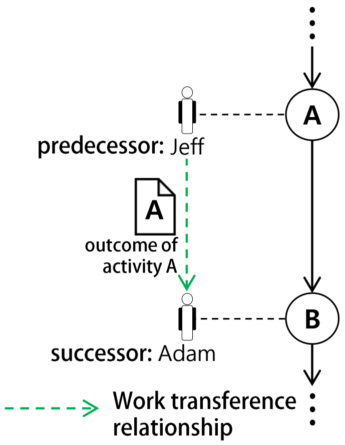


Figure 1. Concept of work transference relation.

In research on the work transference network, two main branches exist: discovery and rediscovery. The former is to discover a work transference network through analyzing a specific workflow model, whereas the latter is concerned with mining a work transference network from workflow event logs of the model. More specifically, we differentiate the former from the latter; the former is used to explore a planned work transference network [5], whereas, the latter is to explore an enacted work transference network. This paper is directly related to work transference network rediscovery. Ultimately, through these discovery and rediscovery concepts, it is possible to assess the workflow fidelity, which indicates how faithfully the observed work transference network (rediscovery) reflects the planned network (discovery). Through the generalization of workflow fidelity, we can answer the following managerial questions.

- Who is the performer most closely working with a particular performer?
- Based on work transferences, what is the highly recommended performer group in specific procedural working steps within the workflow process?
- How well the designed resource allocation plan and its resulting collaboration patterns be accomplished?

To actualize the workflow fidelity assessment, we should address the fact that there still lacks a common implementation that provides system supports to rediscover work transference network correctly and to analyze discrepancies between the discovery and rediscovery results. As a step forward to resolve this gap, this paper describes a framework for rediscovering work transference networks hidden in event logs. In addition, the framework is designed to handle heterogeneous event logs of different timestamp origins, such as the assessed, started and completed timestamp origins. To verify the framework, we conduct experimental analytics on the complete event log dataset of the BPI 2018 Challenge of a specific workflow model which deals with the handling of applications for direct EU payments for German farmers from the European Agricultural Guarantee Fund. Conclusively, the purpose of this paper is to originate a fundamental principle for rediscovering a work transference network from a specific workflow's event logs by fulfilling the experimental analytics and delivering the analytical results.

The remainder of this paper is outlined as follows. Section 2 describes related studies concerning workflow knowledge discovery. Section 3 describes a series of conceptual definitions and a discovery procedure of the framework. Section 4 describes the experimental analytics of mining work transference networks based on this framework. Finally, some concluding remarks are given in Section 5.

2. Related Works

Numerous emerging technologies and research issues can be found in the literature on workflow management. A few of the most recent issues to be highlighted are workflow mining and knowledge discovery issues, which are related with collecting runtime event log data into workflow logs, filtering out and forming workflow warehouses from the logs, and discovering knowledge from such workflow warehouses. Almost all recent workflow management systems provide their own logging mechanisms [6] for organizing such workflow logs and warehouses. In terms of collecting, filtering, and discovering activities with workflow logs and warehouses, to date, the related studies have mainly focused on two specialties of workflow discovery activities, namely, workflow process discovery [3,7–12] and workflow knowledge discovery [13–29]. The workflow process discovery is directly concerned with redesigning and reengineering the control-flow aspect of the workflow processes by discovering workflow models from event logs. To obtain desirable process models from event logs, diverse discovery techniques have been proposed in the literature, including alpha [3], amalgamation [7], heuristics-based [8], and inductive [9] mining algorithms.

By contrast, the workflow knowledge discovery is closely connected with replanning and realigning the resource aspect (e.g., the data-flow, performer, role, and program) of the workflow models by rediscovering enacted and binding histories (e.g., temporal work transferences) from the workflow logs. The eventual goal of this research area is measuring, evaluating, controlling, and predicting the degree of workflow fidelity in a workflow-supported organization. Through the novel concept of workflow fidelity, we can achieve the managerial decision-making goal to minimize the discrepancies between the planned workflow models as estimated on the build time and their enacted workflow models as actually performed at runtime. In this study, we focus experimentally on workflow knowledge discovery and analytics with particular attention paid to work transference networking knowledge.

The workflow knowledge discovery stems from the concepts of business process intelligence [13] and workflow intelligence [14]. In [13], the authors claimed that business process intelligence is a suite of integrated software tools aiming at managing the workflow execution quality by providing several features, such as analysis, prediction, monitoring, control, and optimization. Through a business process intelligence suite, we can accomplish a higher level of enhancement, such as the detection and prevention of nonconformances through auditing [15], refining workflow data preparations, and integrating other data mining techniques, in managing the quality of the workflow execution. The work in [13] is much closer to the conceptual contribution described herein, whereas the approach in [14] is much more concrete. In [14], the authors proposed a framework of control-path oriented workflow intelligence and quality improvement to achieve a higher degree of the workflow traceability and discoverability, and devised an efficient control-path analysis approach through the concept of a minimal workflow model. In particular, the discovered knowledge in the proposed framework is a quantitative measure of runtime enactments according to each control-path generated from a workflow model. The control paths and reachable paths with frequencies of their runtime enactments become valuable knowledge for redesigning and reengineering the corresponding workflow model.

From the viewpoint of workflow intelligence, in particular, workflow knowledge discovery needs to connect all perspectives such as the behavioral, temporal, organizational, and performance perspectives. From an organizational viewpoint, Kim [16] first triggered the initiation of the human-centered knowledge discovery issue on workflows by observing the collaborative behaviors among the workflow performers. The authors proposed a formal approach using an algorithm that can discover a procedural collaboration network among the workflow performers. Following this approach, the work in [17] became a major turning point in human-centered workflow knowledge discovery. In their study, the authors understood the workflow performers' network as the relationships of work transferences (which they call a handover) among the workflow performers. In other words, they interpreted the concept of a workflow performer network into the concept of a social network and analyzed them by using social network analysis (SNA) techniques. Furthermore, Song et al. [18] attempted to discover organizational models as well as workflow performer networks, and measured their performance. They defined this analytics based on event logs as a more comprehensive term, organizational mining. In addition, Park et al. [19] developed another approach and system for analyzing the social networks of workflow performers using process models. As such, SNA techniques facilitate the discovery of organizational structures among workflow performers and those applications for acquiring human-centered knowledge [17–19], including the measurement of employee contribuion [19], resource community detection [20,21], and recommendation [22,23].

In addition to these studies, results related to the integrated concept of workflows and social networks for dealing with the discovery of temporal workflow patterns [24], and mining techniques to be used for workflow resource allocation [25]. Beyond the discovery of human-centered knowledge, Hong et al. [26] presented a methodology for redesigning an organizational structure based on the results of social networks analyzed from workflow enactment event logs. In addition, to address a sustainable analysis, Appice et al. [27] proposed a technique for continually updating a social network discovered from event logs. Through their approach, it is possible to track changes to a social network and gain knowledge from its histories in terms of the dynamics. As an industrial case study, Aloini et al. [28] analyzed a social network among port logistics workers by using workflow mining and SNA techniques. The results from the study suggest that handover relationships of such workers affect the overall performance of the export process efficiency.

In the present paper, we focus on a new shape of human-centered networking knowledge hidden inside the workflow processes and their enactment event histories, which is called work transference networking knowledge and is formally represented as a work transference network model. Our research group has successfully developed a mining system that is able to discover work transference networks from workflow enactment event log datasets formatted in the extensible event stream (XES) standard [30]. Through the discovery of work transference networks, we are able to not only quantitatively measure the degree of work-sharing and relevancy among performers, but also qualitatively estimate the levels of work-intensity among workflow performers in a workflow-supported organization.

3. Conceptual Discovery Framework

In this section, we formally describe a conceptual framework that includes a series of conceptual definitions and a procedure to discover a workflow transference network from event logs. Figure 2 illustrates the framework and functional relationships of its core concepts. To deploy the formalization of the framework, we define a series of formal concepts including those from an event trace to a work transference network model. An event trace corresponds to an execution of a workflow instance, and therefore this aggregates all completed events temporally occurring during its execution. We introduced a formal definition of this concept (called a temporal trace) represented in [7].

In addition, each event trace has its own performer trace, which has been formally called a temporal transference model. Together with these concepts, we provide a formal definition of the work transference network model.

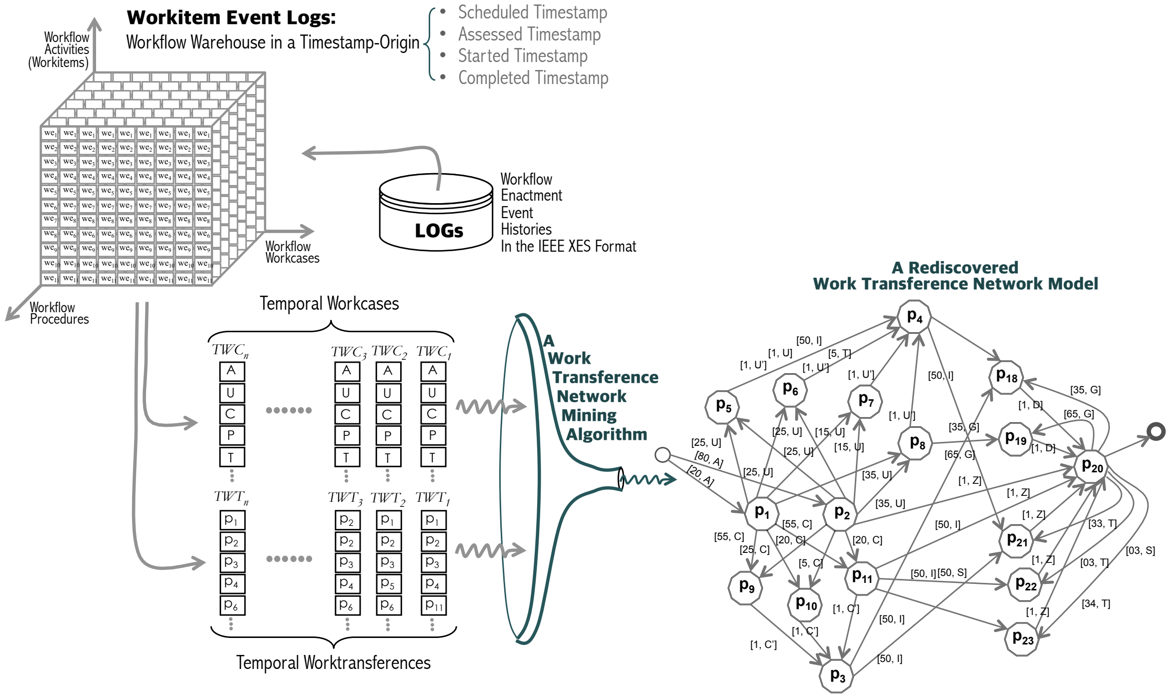

Workitem Event Logs:
Workflow Warehouse in a Timestamp-Origin
Scheduled Timestamp
Assessed Timestamp
Started Timestamp
Completed Timestamp
Workflow Activities (Workitems)
Workflow Workcases
Workflow Procedures
LOGs
Workflow Enactment Event Histories In the IEEE XES Format
Temporal Workcases
TWC_n
TWC_3
TWC_2
TWC_1
A
U
C
P
T
TWT_n
TWT_3
TWT_2
TWT_1
Temporal Worktransferences
A Work Transference Network Mining Algorithm
A Rediscovered Work Transference Network Model
p1
p2
p3
p4
p5
p6
p7
p8
p9
p10
p11
p18
p19
p20
p21
p22
p23

Figure 2. Conceptual discovery framework.

3.1. Workflow Enactment Event Logs

According to the workflow instance executed, the logging component of the workflow execution engine records its execution events in a log repository, and the logged events are arranged in the form of a temporal sequence of events. This sequence corresponding to a workflow instance is called an event trace, from which we can extract a activity trace, a formal representation of which is specified as a model of temporal trace. An event trace is also involved with a sequence of the performers (performer trace) participating in the executions of the work items in the corresponding workflow instance. We can also extract a performer trace from an event trace, the formal representation of which is specified as a model of temporal work transference. Here, we describe the discovery framework from formally defining an event, which is stored as a single record of log, as shown in the following.

Definition 1 (Event). *Let* $\boldsymbol{we} = (\alpha, pc, wf, wc, ac, p, t, s)$ *be an 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 identifier,*
- *ac is an activity identifier,*
- *p is a performer identifier,*
- *t is a timestamp, and*
- *s is a current state of the work item that is one of the states such as ready, assigned, reserved, running, completed, and cancelled.*

In terms of the log formats, we consider that event logs are stored in a tag-based language. An XML-based log format, XWELL [6], is proposed for the purpose of workflow mining at the academic level, and WfMC has released a standardized audit and log specification, namely, BPAF [31].

IEEE has recently released a standard for event log format, XES [30], whose aim is to provide designers of information systems with a unified and extensible methodology for capturing system behaviors by means of event logs. As a format for the event log structure, we can use the XES schema describing the structure of an XES event log/stream. Based upon the XES format, we simply summarize the essential attributes in the event log, as follows:

- The event attribute is used to specify an event identifier, which is assigned by the workflow execution engine.
- The work item attribute of an event represents a work item identifier that is uniquely assigned by using those combined identifiers, such as the package id, workflow id, activity id, and instance id.
- The performer attribute is used to specify a human-resource in charge of enacting the work item.
- The timestamp attribute specifies the time information of an occurred event.
- Finally, the state attribute represents the current state of the work item maintained by the engine. Whenever the state is changed, the resulting event will be logged. The state should be ready, assigned, reserved, running, completed, and cancelled.

3.2. Temporal Workcases and Work Transferences

Definition 2 (Event Trace)**.** *Let* $WT(\boldsymbol{c})$ *be an event trace of a workflow instance,* $\boldsymbol{c}$*, where* $WT(\boldsymbol{c})$ = $(we_1, \ldots, we_n)$*, where* $\{\forall we_i, we_j \in WT(\boldsymbol{c}) \mid we_i.wc = \boldsymbol{c} \wedge we_i.t \leq we_j.t \wedge we_i.pc = we_j.pc \wedge we_i.wf = we_j.wf \wedge we_i.wc = we_j.wc \wedge i < j \wedge 1 \leq i, j \leq n\}$*, which formally represents a temporally ordered event sequence of a specific workflow instance, which can be extracted from the event logs by considering the timestamp and the state attributes.*

From the formal definition of event trace, we build a temporal workcase model, TWC($\boldsymbol{c}$), along with a temporal work transference model, TWT($\boldsymbol{c}$), as shown in Figure 3. Note that the meaningful temporal order in managing the workflow instances must be one of the following instantaneous points in time, each of which is called a timestamp origin. Accordingly, we necessarily discover a series of event traces from numerous event logs, and from each of the discovered traces, we build four types of temporal workcases and temporal work transferences with one of the timestamp origins.

- The scheduled time: the event's timestamp is taken at when the state of a work item is changed from ready (the 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) to assigned (the assigned state of a work item implies that the work item has been assigned to a group of participants, but work has not started yet). An event with a timestamp of the scheduled time holds that, $we^{t.s} \Rightarrow (t = we.t \wedge s = we.s \wedge s = \text{'}assigned\text{'})$.
- The assessed time: the event's timestamp is taken at when the state of a work item is changed from assigned to reserved (the reserved state of a work item implies that the work item has been assigned to a single participant, but work has not started yet). An event with a timestamp of the assessed time holds that, $we^{t.e} \Rightarrow (t = we.t \wedge e = we.s \wedge e = \text{'}reserved\text{'})$.
- The started time: the event's timestamp is taken at when the state of a work item is changed from reserved to running (the 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). An event with a timestamp of the started time holds that, $we^{t.u} \Rightarrow (t = we.t \wedge u = we.s \wedge u = \text{'}running\text{'})$.
- The completed time: the event's timestamp is taken at when the state of a work item is changed from running to completed (the completed state of a work item implies that the work item has been fully executed and completed with either success or failure). An event with a timestamp of the completed time holds that, $we^{t.o} \Rightarrow (t = we.t \wedge o = we.s \wedge o = \text{'}completed\text{'})$

Definition 3 (Temporal workcase)**.** *Let TWC(**c**) be a temporal workcase of a workflow instance, **c**:*

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

*which is a temporally ordered activity sequence along with the specific timestamp origin. It is assumed that all the work items in **c** are successfully completed, and their executions are running without being suspended, as well.*

Based on Definition 3, we can interpret the formal definition as a conceptual implication in which all activities of the event trace of ***c*** that have the same instance id are lined up in an activity sequence along with the timestamp origin. Consequently, from an event trace, an activity sequence is constructed by extracting the activity identifiers and their timestamps, which we call a temporal workcase. Note that, according to the different types of timestamp origins, four types of temporal workcases potentially exist, where all activities ($we_{\alpha}^{\tau[.\phi]}$) are aligned with the specific timestamp origin that should be one of the scheduled ($\phi =$'s'), assessed ($\phi =$ 'e'), started ($\phi =$ 'u'), or completed ($\phi =$ 'o') timestamp origins.

Definition 4 (Temporal Work Transference)**.** *Let TWT(**c**) be a temporal work transference of a workflow instance, **c**:*

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

*which is a temporally ordered performer sequence along with the specific timestamp origin. Note that the scheduled timestamp origin is not available in performer sequences. Especially, a temporal work transference is formally defined by a temporal work transference model, and it is assumed that all the work items in **c** are successfully completed, and their executions are running without being suspended, as well.*

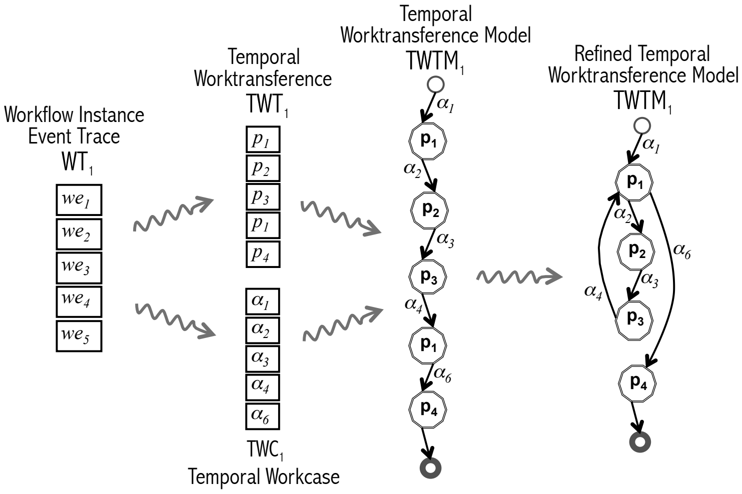


Figure 3. Conceptual steps in building a temporal work transference model from an event trace.

Based on Definition 4, we can interpret the formal definition as a conceptual implication that all performers of the event trace of ***c*** that have the same instance id are lined up in a performer sequence along with the timestamp origin. Consequently, from an event trace, a performer sequence is constructed by extracting the performer identifiers and their timestamps, which we call a temporal work transference. Owing to the inapplicability of the scheduled timestamp type, there may exist three types of temporal work transference where all performers are aligned with the specific timestamp origin, which should be an assessed, started, or completed timestamp origin.

Definition 5 (Temporal Work Transference Model)**.** *A temporal work transference model is formally defined as 3-tuple* $TWTM$ = (χ, F_r^t, T_o^t) *over a set* $\boldsymbol{P}$ *of performer nodes,* $\forall \eta_p^{\tau[\cdot\phi]}$, *on a temporal work transference,* $TWT(\boldsymbol{c})$, *of a workflow instance,* $\boldsymbol{c}$, *and a set* $\boldsymbol{K}$ $(= \{e, u, o\})$ *of the timestamp origins, where:*

- F_r^t *is a coordinator or a coordinator-group linked from an external temporal work transference model;*
- T_o^t *is a coordinator or a coordinator-group linked to an external temporal work transference model;*
- $\chi = \chi_i \cup \chi_o$ *on* $\forall \eta_p^{\tau[\cdot\phi]} \in \boldsymbol{P}$,
  - $\chi_o : \boldsymbol{P} \longrightarrow \wp(\boldsymbol{P})$ *is a single-valued mapping function of a performer node,* $\eta_p^{\tau[\cdot\phi]} = we_p^{\tau[\cdot\phi]} \wedge \phi \in \boldsymbol{K}$, *to its immediate successor in a temporal work transference;*
  - $\chi_i : \boldsymbol{P} \longrightarrow \wp(\boldsymbol{P})$ *is a single-valued mapping function of a performer node,* $\eta_p^{\tau[\cdot\phi]} = we_p^{\tau[\cdot\phi]} \wedge \phi \in \boldsymbol{K}$, *to its immediate predecessors in a temporal work transference.*

According to different timestamp origins, temporal work transference models, $TWTM^{\phi}$, are defined as follows:

- $TWTM^{\phi}$ with the assessed time: $\phi =$ 'e' *in* $\forall \eta_p^{\tau.\phi}$ of a temporal work transference model
- $TWTM^{\phi}$ with the started time: $\phi =$ 'u' *in* $\forall \eta_p^{\tau.\phi}$ of a temporal work transference model
- $TWTM^{\phi}$ with the completed time: $\phi =$ 'o' *in* $\forall \eta_p^{\tau.\phi}$ of a temporal work transference model.

3.3. Work Transference Network Model

As described above, we confirmed that a TWTM can be constructed for each workflow instance by combining the corresponding TWC and TWT. At this stage, one step remains for amalgamating all TWTMs into the workflow transference network model (WTNM) that we ultimately aim to discover. A WTNM reveals the real form of the work transference relationships among the performers after a lapse of a certain time since deploying the corresponding workflow model and it is discovered from the event logs shaped in the form of a workflow warehouse. As shown on the right side of Figure 2, as the final outcome of the framework, a WTNM has the formal and graphical structure of a directed graph (digraph) model to represent the performers and work transferences among them. Each node represents a performer, and each ordered pair of nodes (or a directed edge) represents a work transference relationship. Edges contain the intermediate activities and their occurrences; for example, the performer p_2 has completed the work items of activity C 20 times and has transferred those outcomes to the performer p_{11}.

As a formal representation of such knowledge regarding the work transferences, we define the WTNM through Definition 6. Using the formal notation of $\sigma(= \sigma_i \cup \sigma_o)$, we define the work transference relationships, in which the two nodes of a directed edge represent the predecessor and successor of the work items. In addition, we define the formal notation of $\psi(= \psi_i \cup \psi_o)$ for the work association relationships by labelling each directed edge with the activity names and occurrences corresponding the work items that are transferred to the successor and received from the predecessor of the corresponding work transference relationship concurrently.

Definition 6 (Work Transference Network Model)**.** *A work transference network model is formally defined as* Λ^R = (σ, ψ, F_r, T_o)*, over a set* $\boldsymbol{P}$ *of performer nodes,* $\eta_p^{\tau[\cdot\phi]}$, *and a set* $\boldsymbol{A}$ *of activity nodes,* $\eta_\alpha^{\tau[\cdot\phi]}$, *in a set of event traces logged from enacting the underlying workflow model, where:*

- F_r *is a coordinators or a coordinator-group linked from some external work transference networks;*
- T_o *is a coordinators or a coordinator-group linked to some external work transference networks;*
- $\sigma = \sigma_i \cup \sigma_o$ */* Work Transferences */*
  - $\sigma_i : \boldsymbol{P} \longrightarrow \wp(\boldsymbol{P})$ *is a multi-valued function mapping a performer to its set of immediate predecessors;*

  - $\sigma_o : \boldsymbol{P} \longrightarrow \wp(\boldsymbol{P})$ *is a multi-valued function mapping a performer to its set of immediate successors;*
- $\psi = \psi_i \cup \psi_o$ */* Work Associations */*
  - $\psi_i : (\boldsymbol{P} \times \boldsymbol{P}) \longrightarrow [\wp(\boldsymbol{A}), \mathbb{N}]$ *is a multi-valued function returning a paired list of receiving work items and their occurrences on ordered pairs of performers,* $(\sigma_i(p), p)$*,* $p \in \boldsymbol{P}$*, from* $\sigma_i(p)$ *to* p*;*
  - $\psi_o : (\boldsymbol{P} \times \boldsymbol{P}) \longrightarrow [\wp(\boldsymbol{A}), \mathbb{N}]$ *is a multi-valued function returning a paired list of transferring work items and their occurrences on ordered pairs of performers,* $(p, \sigma_o(p))$*,* $p \in \boldsymbol{P}$*, from* p *to* $\sigma_o(p)$*;*

4. Experimental Discovery and Analytics

Based on the formal models defined above, we describe an implementation of the framework that is designed for discovering a WTNM from XES-formatted event log datasets. As a running example, we carry out a mining experiment on the 2018 BPI Challenge dataset [32] by using the implemented mining system.

4.1. Implementation of the Framework

We try to construct the discovery framework and its related algorithms to mine a work transference network from an event log dataset. The essential point of the framework starts from mining all temporal workcases as well as all temporal work transferences, each of which can be found from a corresponding event trace based on the completed timestamps of its events in the temporal order of precedence. Figure 4 illustrates the algorithmic mining procedure. In the framework, there are two new concepts, namely, work transference pair-groups and a work transference lineal set, essential to implementing the WTNM discovery. A work transference pair-group is formed through the pairing of the immediate predecessor-successor (the completion time precedence; every event in an event trace has the timestamp property as its completed times of the work item) of the performers in the temporal order of the work transference, and by adding its related activity on the edge of the corresponding pair. This step is implemented by combining a TWC and a TWT for the workflow instance. For the temporal work transference, $TWT_1 = \{p_1, p_2, p_3, p_1, p_4\}$ and the temporal workcase, $TWC_1 = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_6\}$ in Figure 4, they are combined into the work transference pair-group, $WTPG_1 = \{((p_{start}, p_1), \alpha_1), ((p_1, p_2), \alpha_2), ((p_2, p_3), \alpha_3), ((p_3, p_1), \alpha_4), ((p_1, p_4), \alpha_6), ((p_4, p_{end}), \emptyset)\}$. By amalgamating all work transference pair-groups, we can make a work transference lineal set that serves as a building block of a WTNM. For example, by connecting all lineal sets in Figure 4, we can confirm the constructed $WTNM$ that includes the formal discovery results of Definition 6, such as $\sigma_i(p_4) = \{p_1, p_7\}$, $\sigma_o(p_5) = \{p_6, p_7\}$, $\psi_i(p_3, p_1) = \{\{\alpha_1, 18\}, \{\alpha_2, 1008\}\}$, and $\psi_o(p_1, p_5) = \{\{\alpha_7, 55\}\}$.

The details of the algorithmic mining procedure are not presented herein. However, based on these concepts and their related algorithms, we successfully implemented a work transference network mining system that can discover a WTNM from an event log dataset formatted in the XES standard. Figure 5 shows a captured screen of the implemented mining system. On the screen, dashboard and visualization panels of the system can be seen. The main items, such as choose log and analyze, and the boxes on the dashboard panel indicate that the 2018 BPI Challenge Dataset was chosen in this experiment, and the results of 43,809 event traces recorded by enacting 41 different types of activities conducted by 165 workflow performers in all. The graphical model shown in the visualization panel is a work transference network discovered from this experimental dataset.

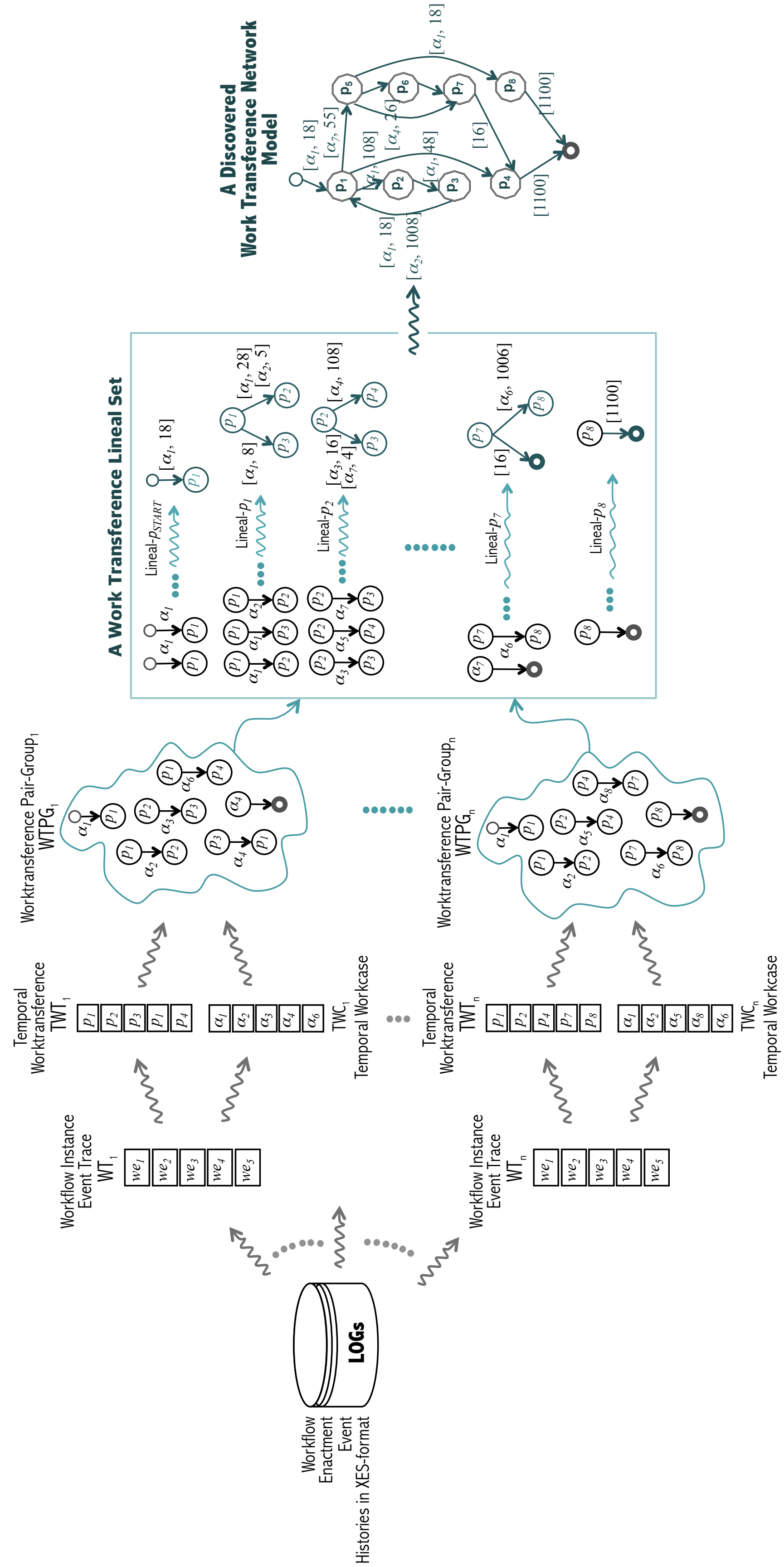
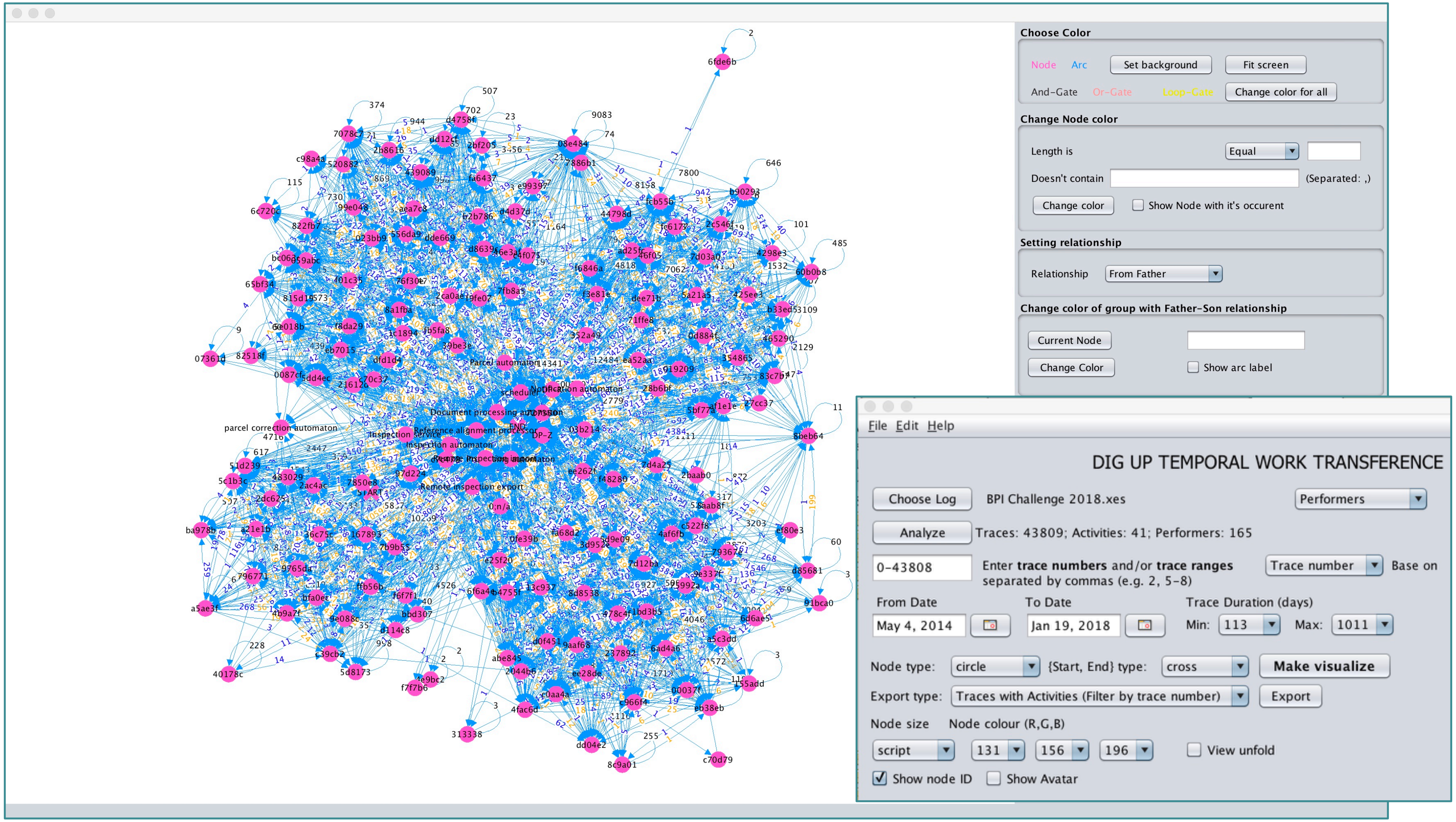

Workflow Enactment Event Histories in XES-format
LOGs
Workflow Instance Event Trace WT_1
Workflow Instance Event Trace WT_n
we_1 we_2 we_3 we_4 we_5
Temporal Worktransference TWT_1
Temporal Worktransference TWT_n
TWC_1 Temporal Workcase
TWC_n Temporal Workcase
Worktransference Pair-Group_1 WTPG_1
Worktransference Pair-Group_n WTPG_n
A Work Transference Lineal Set
Lineal-p_START
Lineal-p_1
Lineal-p_2
Lineal-p_7
Lineal-p_8
[α_1, 18]
[α_1, 8]
[α_1, 28] [α_2, 5]
[α_3, 16] [α_7, 4]
[α_4, 108]
[16]
[α_6, 1006]
[1100]
A Discovered Work Transference Network Model
[α_1, 18]
[α_7, 55]
[α_4, 26]
[α_1, 108]
[α_1, 48]
[16]
[1100]
[α_1, 18]
[α_2, 1008]
[1100]
[α_1, 18]

Figure 4. Concrete discovery framework.



Length is: Equal

Doesn't contain: (Separated: ,)

□ Show Node with it's occurent

Relationship: From Father

□ Show arc label

DIG UP TEMPORAL WORK TRANSFERENCE

BPI Challenge 2018.xes Performers

Traces: 43809; Activities: 41; Performers: 165

0-43808 Enter **trace numbers** and/or **trace ranges** separated by commas (e.g. 2, 5-8) Trace number Base on

From Date: May 4, 2014

To Date: Jan 19, 2018

Trace Duration (days) Min: 113 Max: 1011

Node type: circle {Start, End} type: cross

Export type: Traces with Activities (Filter by trace number)

Node size: script

Node colour (R,G,B): 131 156 196

□ View unfold

☑ Show node ID □ Show Avatar

Figure 5. Captured screen of the work transference network mining system.

4.2. Dataset Preparation

Using the discovery framework and its system, we carry out experimental analytics on a specific dataset of event logs from the BPI Challenge (BPIC) 2018 [32]. More precisely, the dataset of BPIC 2018, selected for the experiment contains event logs of the workflow process for handling applications for direct EU payments. Table 1 represents a basic information of the BPIC 2018 dataset.

Table 1. Summary of the BPI Challenge (BPIC) 2018 dataset.

Attribute	Value
Workflow process	Handling applications for EU direct payments
Start date	4 May 2014
End date	19 January 2019
# Event traces	43,809
# Events	2,514,266
# Activites	165
Avg. of events per trace	57 events
The shortest trace	24 events
The longest trace	2973 events

As a preparation step to enable the WTNM discovery, we built a workflow warehouse, as shown in Figure 2, which is a data cube of event logs made up of all temporal workcases and temporal work transferences. Figure 6 shows a partial summary of the data cube based on these temporal work transferences mined from a number of datasets collected from 4TU [32].

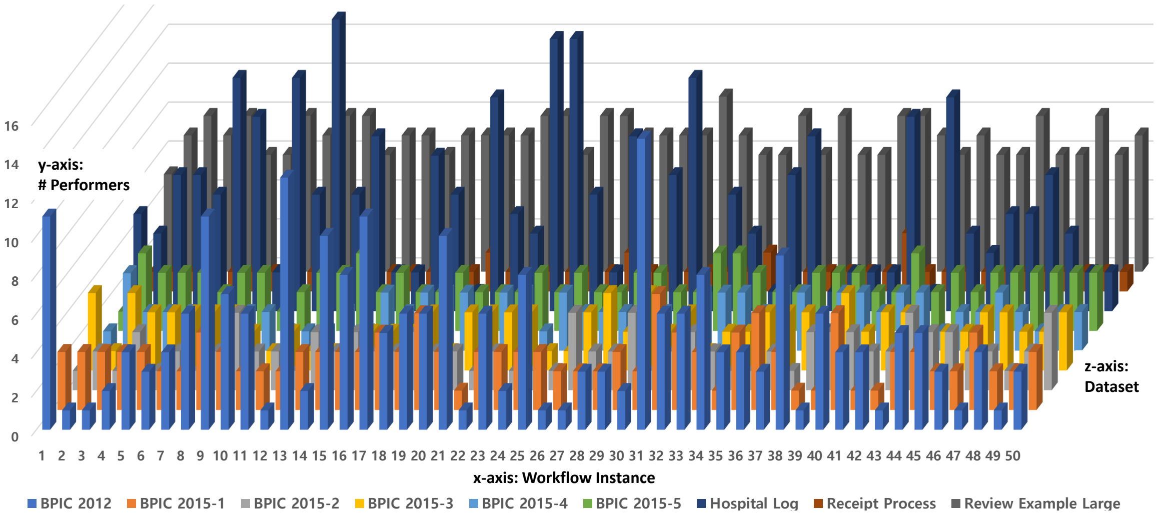


Figure 6. Summary of data cube of work transferences for each set of event traces mined from nine event log datasets.

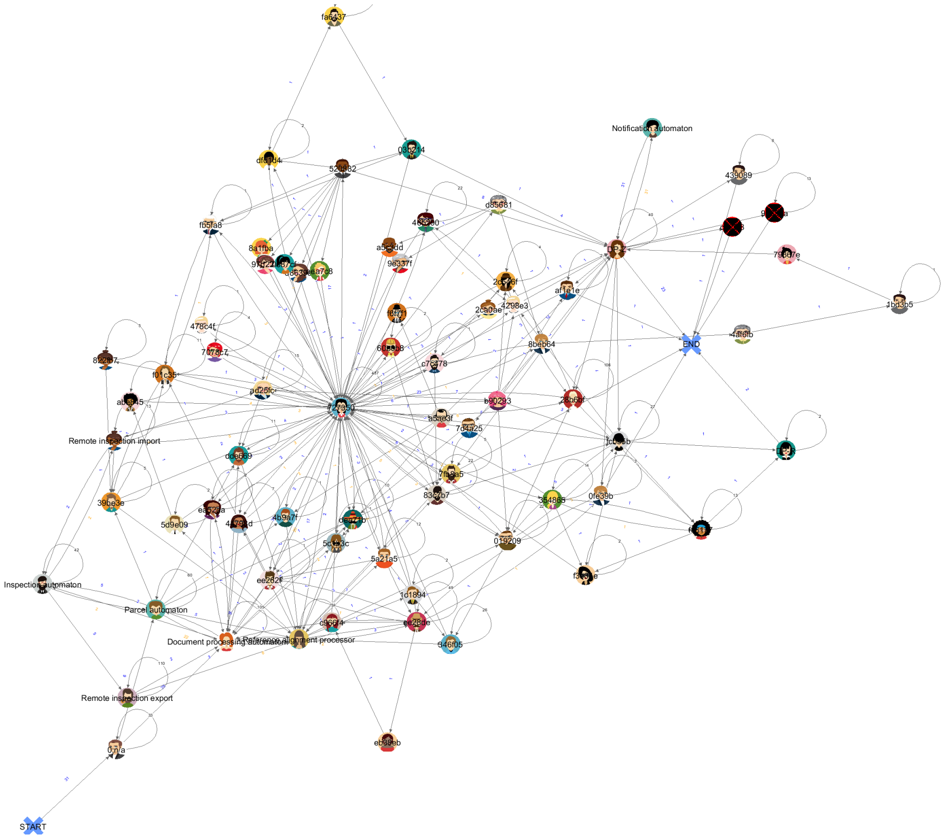
As shown in the figure, the data cube is constructed with three axes: the workflow processes, workflow instances, and work items (activities) or performers. To build a workflow warehouse, we need to fulfill the preprocessing phase, where a series of temporal workcases and work transferences is mined from event log datasets. Each dataset forms two types of data cube, one of which is based on temporal workcases, and the other on temporal work transferences. These two data cubes eventually become the input of the WTNM discovery framework.

4.3. Experimental Analytics

Out of the workflow warehouse, we select all temporal workcases and temporal work transferences belonging to the BPIC 2018 dataset. Each of the workflow instances corresponds to a German farmer's direct payment application for the European Agricultural Guarantee Fund. In this section, we describe an experiment carried out not only to discover a work transference network and its primitive knowledge but also to analyze the quantitative measurements and their interpretations. We describe the details of the experiment and analysis results in this subsection.

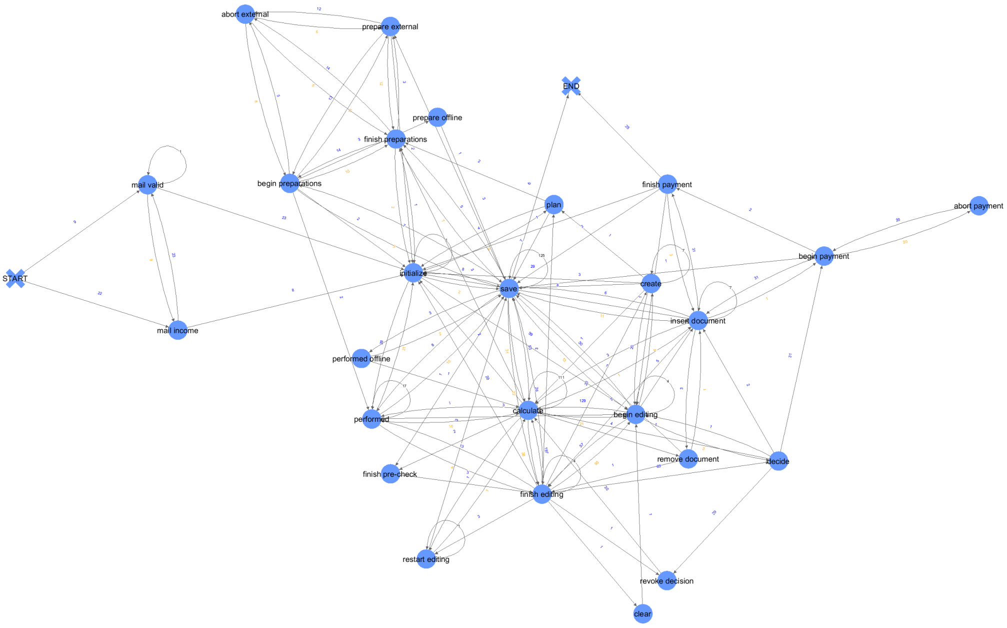
4.3.1. Discovered Work Transference Network.

First, Figure 5 shows two captured screens; the screen on the bottom right is the main dashboard used to control all experiment activities on the system; the main screen is the graphical representation of a work transference network finally discovered from mining all event traces among the 43,809 total workflow instances executed by the 165 performers handling 41 different types of activities. These numerical values from the analytical results can be recognized in the "analyze" item on the main dashboard of the system. The discovered work transference network shown on the visualization panel might be too complex to grasp its meaning and make a close analysis of the experimental results at a glance. We therefore select a small number of workflow instances (20,000–20,030th) and analyzed their event traces. The two networks in the upper part of Figure 7 visualize the work transference network among the performers and the work transition network among the activities mined from those event traces of the selected workflow instances, respectively. There are also two networks shown in Figure 8; one is to visualize a temporal work transference from the 20,001st event trace, whereas the other represents a temporal workcase from the same event trace. By combining these two types of network, we are eventually able to discover a work transference network. In addition, the implemented system has several functions that can generate several aspects of the measured knowledge in a textual format, such as the total numbers of performers, activities, and applications involved in the workflow process, and information regarding their occurrences.



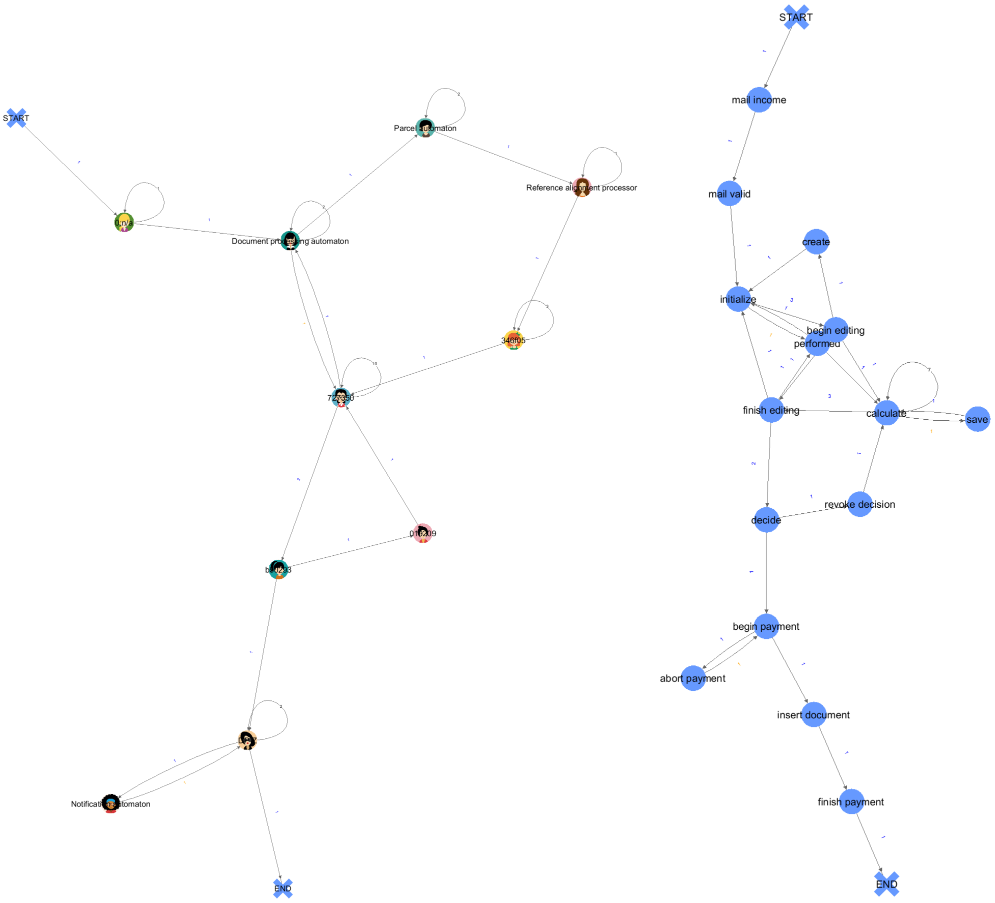
(**a**)

Figure 7. *Cont.*



(**b**)

Figure 7. (**a**) Discovered work transference network and (**b**) its temporal workcase among the activities from 31 event traces (20,000th–20,030th).



(**a**) (**b**)

Figure 8. (**a**) Discovered work transference network of the 20,001st event trace and (**b**) its temporal workcase.

4.3.2. Quantitative Measurements

In particular, the total numbers of performers and activities in the discovered work transference network model are 165 and 41, respectively, as shown in Figures 9 and 10. These figures also show the occurrences (which implies the number of work transferences that occurred for each performer) of the performers involved. We can estimate that not all performers are human because some identifiers have something to do with the titles of invoked applications, which indicates the use of automatically executed application programs without human interference. Note that the 16 automatic applications and their occurrences are shown in Figure 11. The document process automaton program is the most influential out of the 16 automatic application programs. The upper part of Figure 9 shows that only one performer, whose identifier is 727,350, recorded an extraordinarily large number of occurrences, namely, 748,950, whereas the lower bar chart shows a part of performers with moderate occurrences. In Figure 10, the top-five tasks (involved activities) transferred and received among the performers are calculate, finish editing, being editing, save and initialize, and their occurrences are 466,141, 405,691, 397,133, 288,902 and 205,082, respectively.

4.3.3. Work Transference Occurrences

Finally, the most important results of the experimental analytics, which are directly related to the main subject of this work, are depicted in Figures 12–14. We found numerous work transferences through which work was transferred to the performer itself. We therefore need to examine more closely into the dataset and obtain detailed analysis results, as shown in Figure 12. As the figure indicates, 22 performers transferred their work (10∼29 differently named activities) to themselves, and as an example, performer '97d224' transferred 14,116 work items to him/herself and was involved in performing 19 different types of activities in the underlying workflow model as well. The main analytics of this paper is to inspect the statistical measures and patterns of the work transferences in the event logs of the BPIC 2018 dataset. Based on the results analyzed using the implemented system, we observed that 5911 different cases of work transferences occurred during the period in which the underlying workflow model of the dataset was enacted. Table 2 shows the details of numerous involved activities for each of the different cases. Additionally, Figures 13 and 14 show the detailed number of work transferences among the performers, of which the numbers of their activities involved, as shown in the graphs, are nine and ten, respectively. As an instance from the graph in Figure 13, the performer fb5fa8 transferred 1497 times of works to performer 727,350, and nine differently named activities were involved. One more example of a work transference from the graph in Figure 14 is that of performer 019,209 to performer fcb55b. We can interpret this example as 019209 transferred work to fcb55b 735 times, where ten differently named activities were involved.

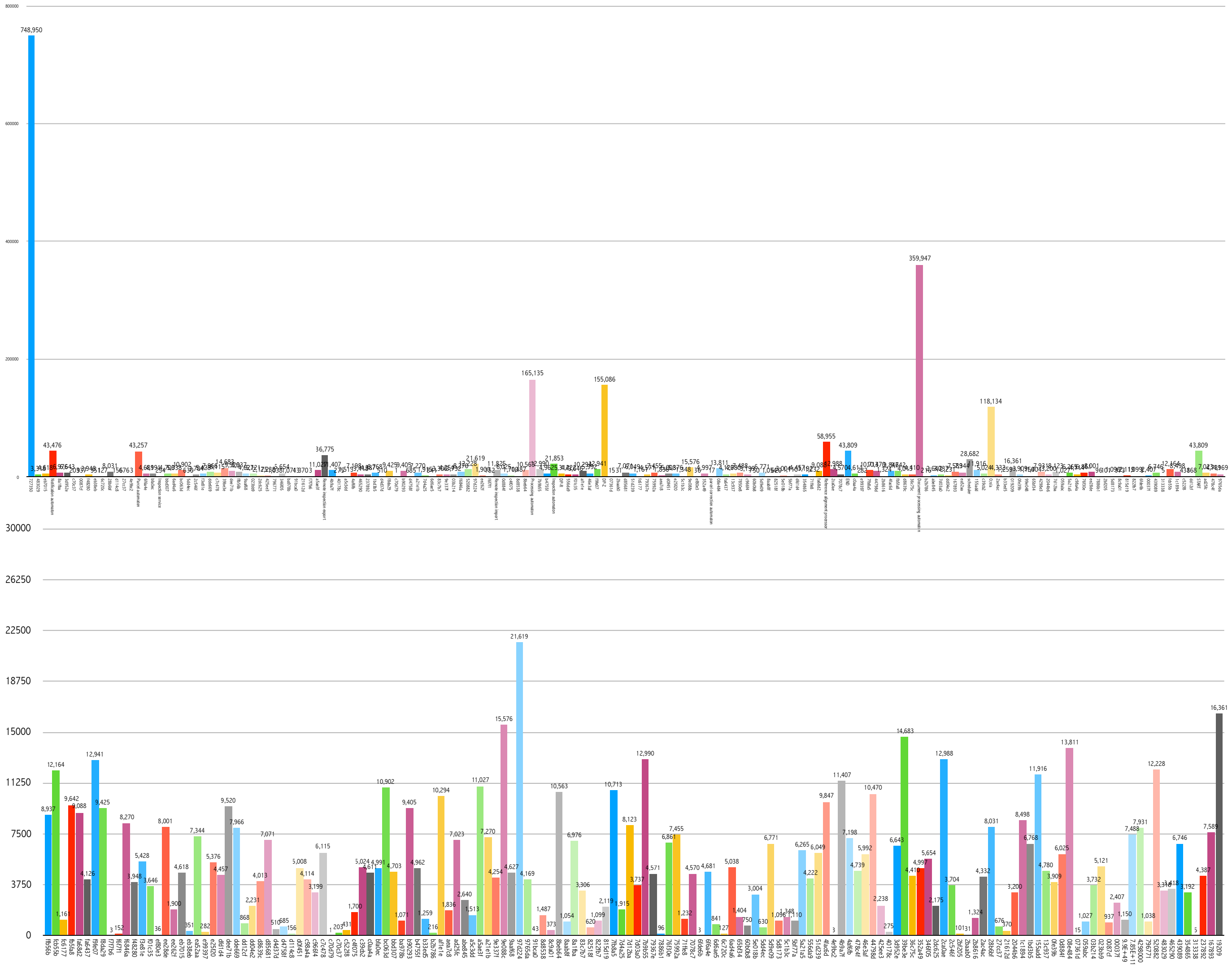


Figure 9. (**a**) Measured occurrences of all the performers and (**b**) the part of the measurement results, centered on the performer group with moderate occurrences.

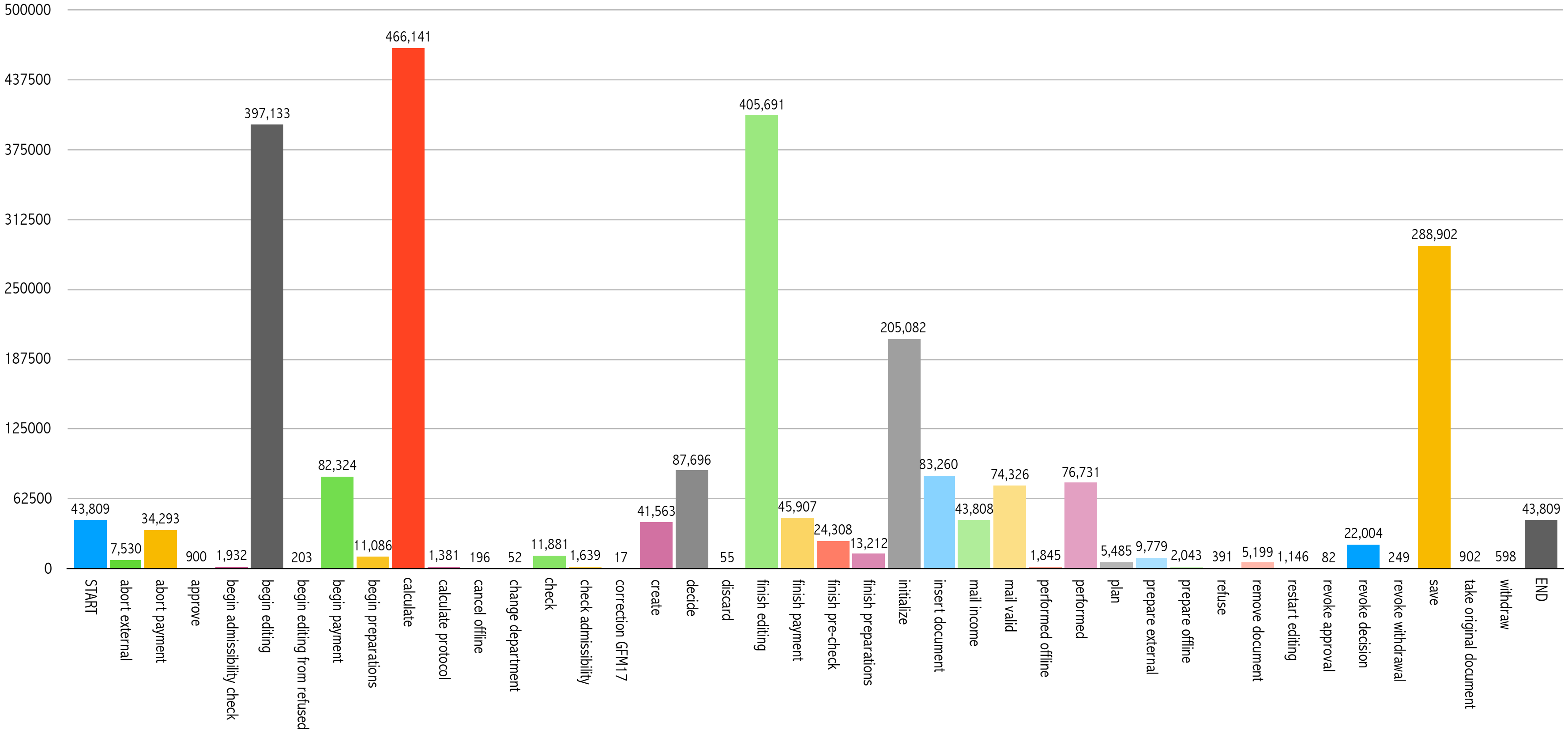


Figure 10. Activities and their enacted occurrences.

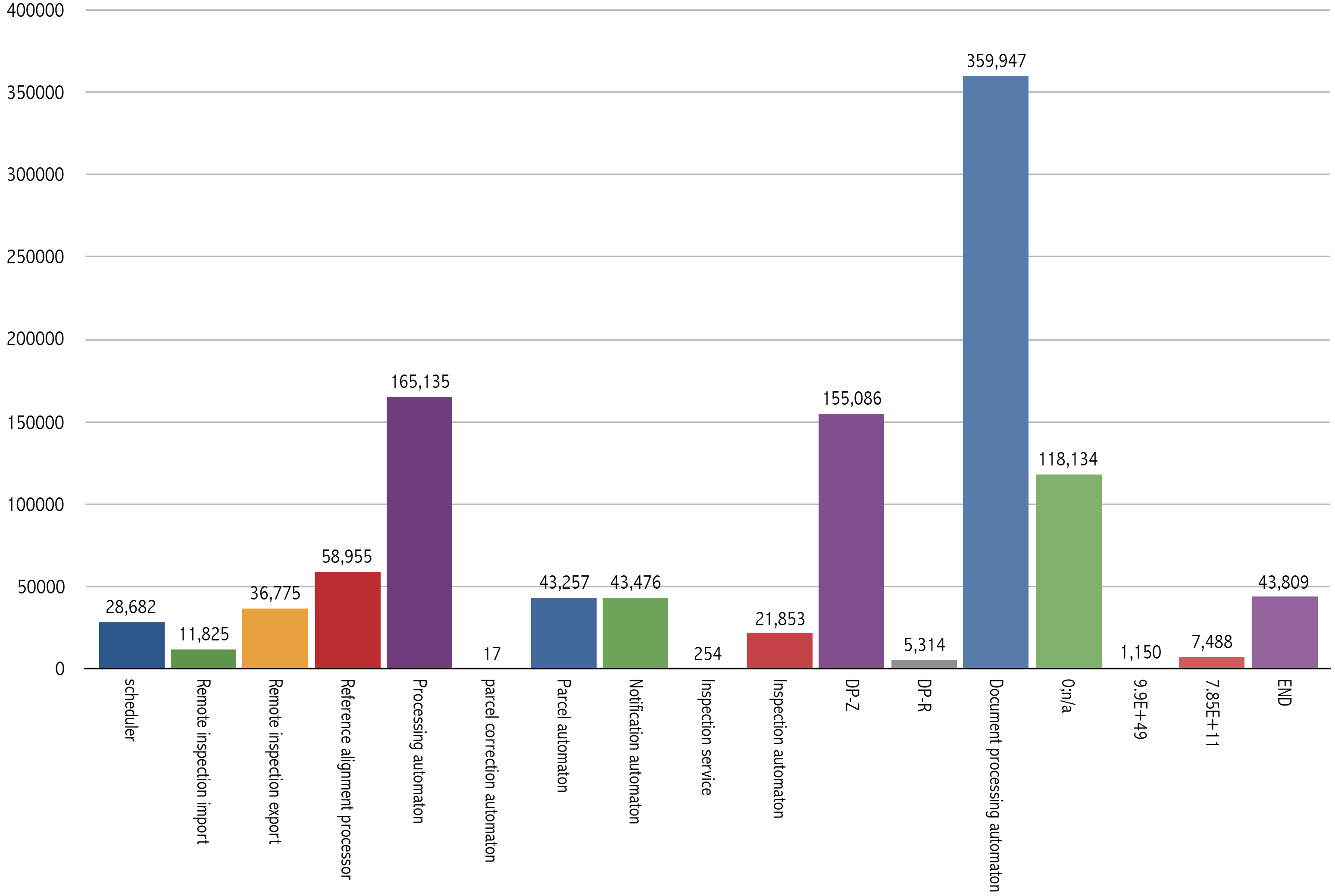


Figure 11. Automatic programs without human interference and their occurrences.

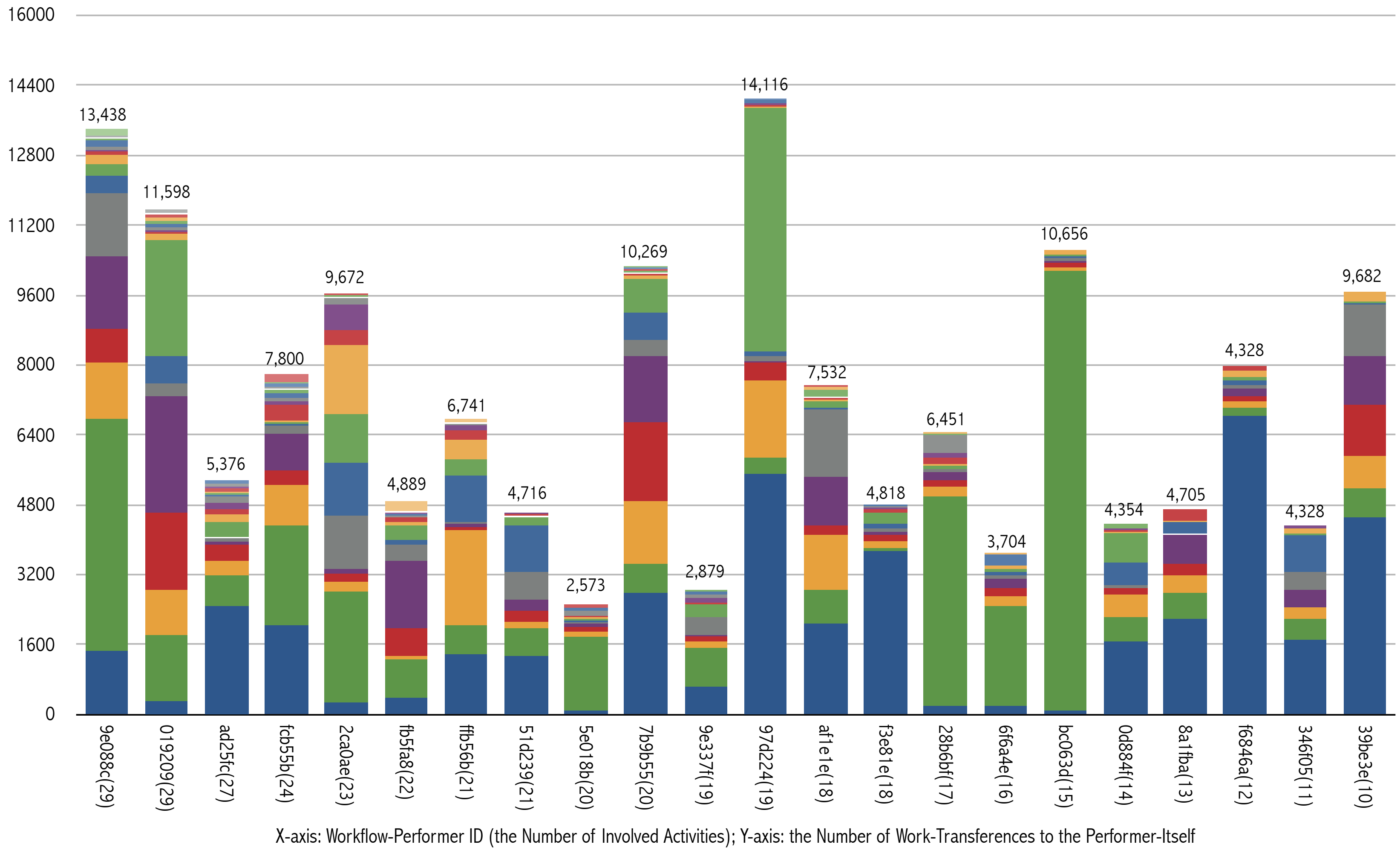


Figure 12. Performers with the number of transferring works to themselves (10 or more different involved activities).

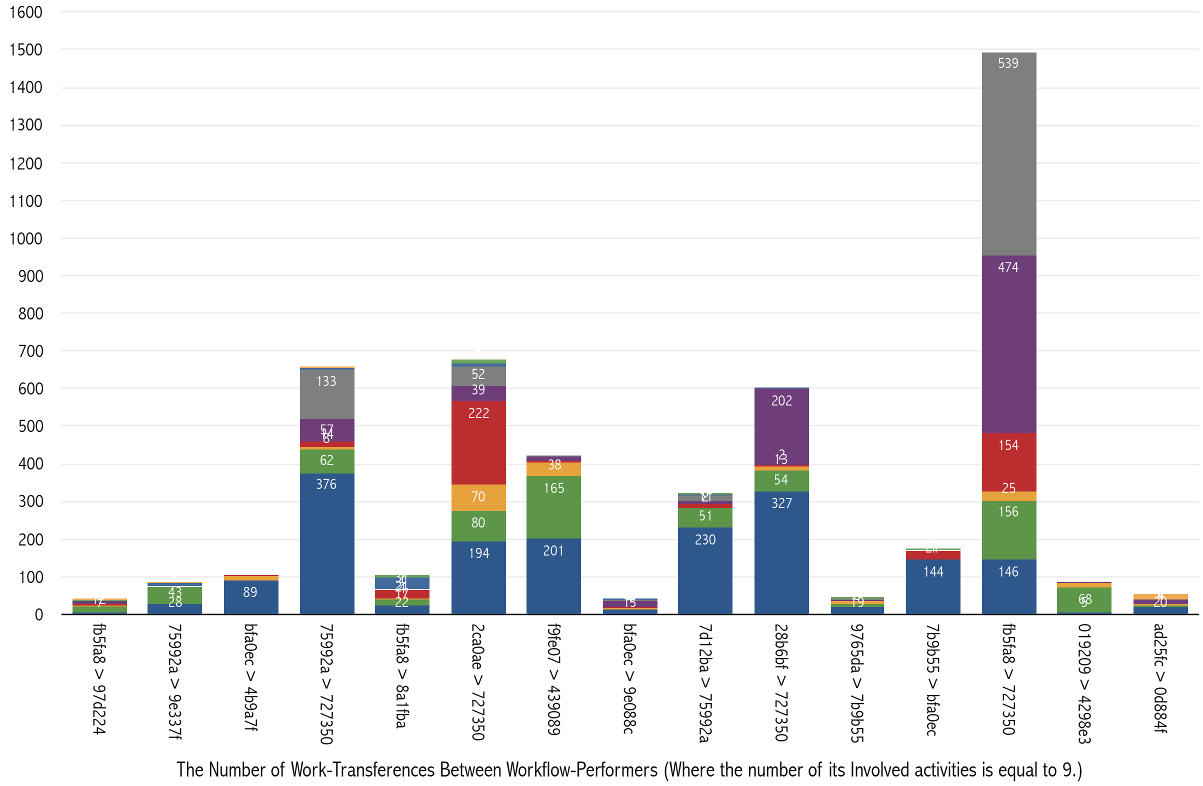


Figure 13. Performers with the number of transferring works to others (nine different involved activities).

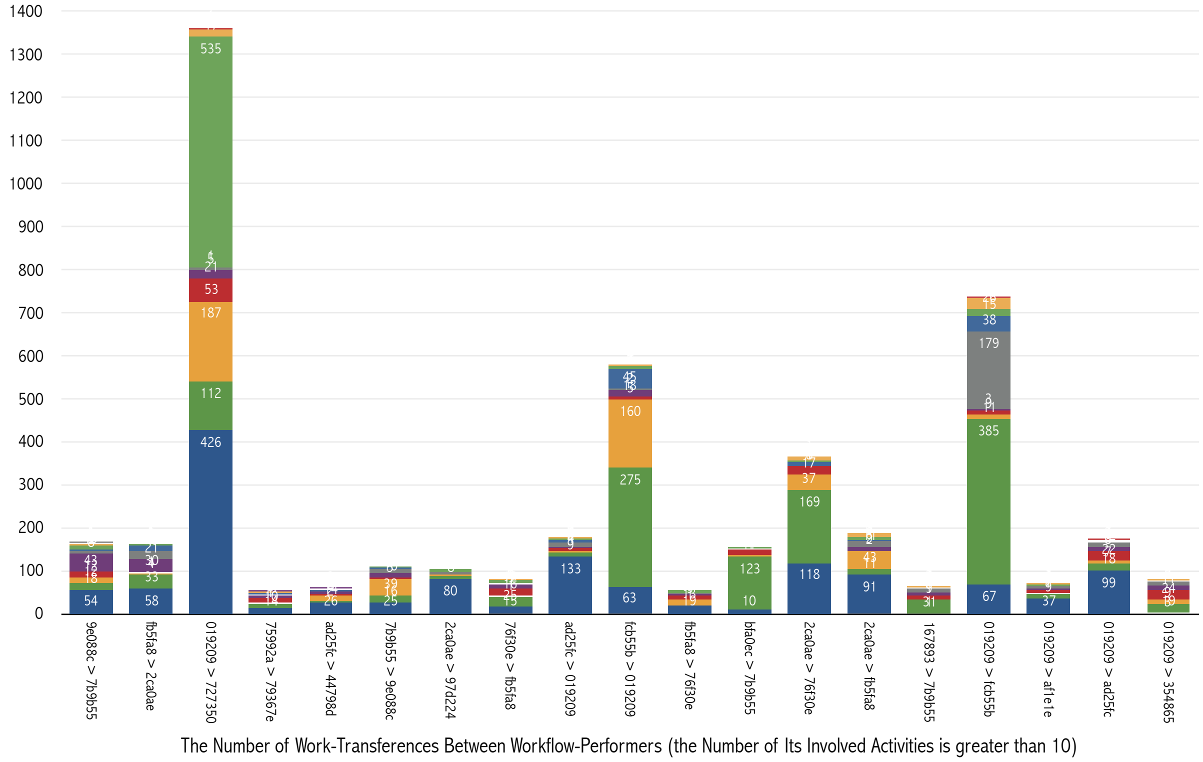


Figure 14. Performers with the number of transferring works to others (10 or more different involved activities).

Table 2. Numbers of different cases of work transferences and their involved activities.

Involved activities	29	27	24	23	22	21	20	19	18	17	16	15	14	
Work transferences	2	1	1	1	2	5	5	8	6	5	7	4	9	
Involved activities	13	12	11	10	9	8	7	6	5	4	3	2	1	Total
Work transferences	8	12	15	23	27	43	71	148	247	435	740	1560	2526	5911

To summarize, we obtained many different types of primitive knowledge pertaining to the discovered work transference network, but we did not introduce all of them. However, we paid close attention to several extraordinary facts discovered through this experimental analytics, including the fact that work transference behaviors occurred with performer 727,350, who performed activities 748,950 times, as well as the names of the activities and invoked applications that were enacted in extraordinary number during the period of enacting the underlying workflow model. We leave the interpretations and understanding of these phenomena and the usage of these experimental results on the reengineering and redesigning of the underlying workflow model as future work.

5. Conclusions

As a study focusing on the work transference network (WTN) for the workflow knowledge discovery, our contributions in this paper include the following:

- Formal definitions and procedure of the framework for discovering a WTN from event logs.
- Brief description of the implemented system of the framework that can handle event logs of different timestamps.
- Experimental analytics of the discovery of WTN using real-life event log dataset.

Based on the contributions above, we confirmed that the proposed framework and its implemented system are valid in discovering WTNs and acquiring primitive knowledge. To recap, the information and knowledge that the proposed framework can provide are listed as follows:

- A structure of the WTN observed during a long-term period (from accumulated event logs).
- A structure of the WTN for a single workflow instance (from an event trace).
- Total numbers of performers and activities associated with certain workflow processes and their occurrences.
- Patterns showing how work transference relationships are made (e.g., self-transferring).
- Degree of strength in terms of the work transference between two performers.

Conclusively, we consider this study as a meaningful step towards the workflow fidelity assessment from the organizational aspect that has not been supported with a systemic way. Despite the feasibility of the framework, our work is still lacking regarding the suggestion of possible and attractive applications for business analysts. Therefore, as future study, we need to improve our system to provide more sophisticated analytical capabilities to effectively discover and deliver more valuable knowledge. In addition, we have a plan to conduct a case study in which we will apply our system to massive real-life event logs to verify how beneficial the discovery framework is to workflow-supported organizations.

Author Contributions: H.A. and K.P.K. conceived and designed the framework. D.-L.P. implemented the mining system. K.P.K. and D.-L.P. performed experiments and analyzed the results. All authors have contributed in writing and proofreading the paper.

Acknowledgments: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT, Grant No. NRF-2018R1C1B5086414). Also, this work was partially supported by Basic Science Research Program through the NRF of Korea funded by the Ministry of Science, ICT & Future Planning (Grant No. 2017R1A2B2010697).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Van der Aalst, W.M.P.; ter Hofstede, A.H.M. YAWL: Yet another workflow language. *Inf. Syst.* **2005**, *30*, 245–275. [CrossRef]
2. Kim, K.; Ellis, C.A. ICN-based workflow model and its advances. In *Handbook of Research on Business Process Modeling*; Cardoso, J., van der Aalst, W., Eds.; IGI Global: Hershey, PA, USA, 2009; pp. 142–171.
3. Van der Aalst, W.; Weijters, T.; Maruster, L. Workflow mining: Discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1128–1142. [CrossRef]
4. Van der Aalst, W. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*; Springer: Berlin/Heidelberg, Germany, 2011.
5. Kim, K.; Jin, M.; Ahn, H.; Kim, K.P. Discovering work transference networks on workflows. In Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services, Salzburg, Austria, 4–6 December 2017; pp. 568–572.
6. Park, M.; Kim, K. XWELL: An XML-Based workflow event logging mechanism and language for workflow mining systems. In *Computational Science and Its Applications—ICCSA 2007*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4707, pp. 900–909.
7. Kim, K.; Ellis, C.A. σ-algorithm: Structured workflow process mining through amalgamating temporal workcases. In *Advances in Knowledge Discovery and Data Mining— PAKDD 2007*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4426, pp. 119–130.
8. Weijters, A.J.M.M.; Ribeiro, J.T.S. Flexible heuristics miner (FHM). In Proceedings of the 2011 IEEE Symposium on Computational Intelligence and Data Mining, Paris, France, 11–15 April 2011; pp. 310–317.
9. Leemans, S.J.; Fahland, D.; van der Aalst, W.M. Discovering block-structured process models from event logs-a constructive approach. In Proceedings of the International Conference on Applications and Theory of Petri Nets and Concurrency, Milan, Italy, 24–28 June 2013; pp. 311–329.
10. Arriagada-Benitez, M.; Sepulveda, M.; Munoz-Gama, J.; Buijs, J.C.A.M. Strategies to automatically derive a process model from a configurable process model based on event data. *Appl. Sci.* **2017**, *7*, 1023. [CrossRef]
11. Rojas, E.; Sepulveda, M.; Munoz-Gama, J.; Capurro, D.; Traver, V.; Fernandez-Llatas, C. Question-driven methodology for analyzing emergency room processes using process mining. *Appl. Sci.* **2017**, *7*, 302. [CrossRef]
12. Wisniewski, P.; Kluza, K.; Ligeza, A. An approach to participatory business process modeling: BPMN model generation using constraint programming and graph composition. *Appl. Sci.* **2018**, *8*, 1428. [CrossRef]
13. Grigori, D.; Casati, F.; Castellanos, M.; Dayal, U.; Sayal, M.; Shan, M.-C. Business process intelligence. *Comput. Ind.* **2004**, *53*, 321–343. [CrossRef]
14. Park, M.; Kim, K. Control-path oriented workflow intelligence analyses. *J. Inf. Sci. Eng.* **2008**, *24*, 343–359.
15. Zerbino, P.; Aloini, D.; Dulmin, R.; Mininno, V. Process-mining-enabled audit of information systems: Methodology and an application. *Expert Syst. Appl.* **2018**, *110*, 80–92. [CrossRef]
16. Kim, K. Actor-oriented workflow model. In Proceedings of the 2nd International Symposium on Cooperative Database Systems for Advanced Applications, Wollongong, Australia, 27–28 March 1999; pp. 150–164.
17. Van der Aalst, W.M.P.; Reijers, H.A.; Song, M. Discovering social networks from event logs. *Comput. Support. Coop. Work.* **2005**, *14*, 549–593. [CrossRef]
18. Song, M.; van der Aalst, W.M.P. Towards comprehensive support for organizational mining. *Decis. Support Syst.* **2008**, *46*, 300–317. [CrossRef]
19. Park, M.; Ahn, H.; Kim, K.P. Workflow-supported social networks: Discovery, analyses, and system. *J. Netw. Comput. Appl.* **2016**, *75*, 355–373. [CrossRef]
20. Ye, J.; Li, Z.; YI, K.; Al-Ahmari, A. Mining resource community and resource role network from event logs. *IEEE Access* **2018**, *6*, 77685–77694. [CrossRef]
21. Kopka, M.; Kudělka, M. Analysis of SAP log data based on network community decomposition. *Information* **2019**, *10*, 92. [CrossRef]
22. Lin, S.; Luo, Z.; Yu, Y.; Pan, M. Effective team formation in workflow process context. In Proceedings of the IEEE International Conference on Cloud and Green Computing, Karlsruhe, Germany, 30 September–2 October 2013; pp. 508–513.

23. Mezzanzanica, M.; Mercorio, F.; Cesarini, M.; Moscato, V.; Picariello, A. GraphDBLP: A system for analysing networks of computer scientists through graph databases. *Multimed. Tools Appl.* **2018**, *77*, 18657–18688. [CrossRef]
24. Hwang, S.-Y.; Wei, C.-P.; Yang, W.-S. Discovery of temporal patterns from process instances. *Comput. Ind.* **2004**, *53*, 345–364. [CrossRef]
25. Liu, T.; Cheng, Y.; Ni, Z. Mining event logs to support workflow resource allocation. *Knowl.-Based Syst.* **2012**, *35*, 320–331. [CrossRef]
26. Hong, S.; Lee, Y.; Kim, J.; Choi, I. A methodology for redesigning an organizational structure based on business process models using SNA techniques. *Int. J. Innov. Comput. Inf. Control* **2012**, *8*, 5411–5424.
27. Appice, A.; Di Pietro, M.; Greco, C.; Malerba, D. Discovering and tracking organizational structures in event logs. In Proceedings of the International Workshop on New Frontiers in Mining Complex Patterns, Porto, Portugal, 7 September 2015; pp. 46–60.
28. Aloini, D.; Benevento, E.; Stefanini, A.; Zerbino, P. Process fragmentation and port performance: Merging SNA and text mining. *Int. J. Inf. Manag.* **2019**, online published. [CrossRef]
29. Stefanini, A.; Aloini, D.; Benevento, E.; Dulmin, R.; Mininno, V. Performance analysis in emergency departments: A data-driven approach. *Meas. Bus. Excell.* **2018**, *22*, 130–145. [CrossRef]
30. IEEE. *IEEE Standard for Extensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams*; IEEE: Piscataway Township, NJ, USA, 1849–2016.
31. Zur Muehlen, M.; Swenson, K.D. BPAF: A standard for the interchange of process analytics data. In *Business Process Management Workshops—BPM 2010*; Lecture Notes in Business Information Processing; Springer: Berlin/Heidelberg, 2011; Volume 66, pp. 170–181.
32. 4TU Centre for Research Data. Available online: https://data.4tu.nl/repository/collection:event_logs (accessed on 10 June 2019).