Information Sciences 515 (2020) 1–25

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Formal approach for discovering work transference networks from workflow logs

Hyun Ahn, Kwanghoon Pio Kim*

Department of Computer Science and Engineering, Kyonggi University, Seoul, South Korea

ARTICLE INFO

Article history:
Received 12 February 2018
Revised 25 September 2019
Accepted 24 November 2019
Available online 6 December 2019

Keywords:
Workflow model
Information control net
Workflow intelligence
Workflow event log
Workflow discovery
Workflow rediscovery
Work transference network
Workflow-supported organizational social network

ABSTRACT

This paper proposes a formal principle for discovering work transference networks of workflow-supported organizational employees from workflow enactment histories contained in event logs. This originates from the strong belief that those work transference networks, hidden in workflow enactment activities and histories, can both connote the degree of work sharing and work relevancy between workflow performers and denote their degree of work intensity. In this paper, we devise a series of formal definitions and algorithms for discovering a work transference network from a workflow procedure, and from its enactment histories in event logs. The final goal of the paper is to theoretically build the principle of fidelity into workflow human resource planning and its performance, via a novel concept of work transference networks that can be discovered from a workflow model and, moreover, rediscovered from its enactment history. In deploying the proposed principle, we base the formal representation on information control net theory, which can graphically and mathematically represent workflow procedures. We apply directed graph theory to formally and graphically define the work transference network model proposed in this paper. For the sake of verifying and validating the proposed concepts and algorithms, we implement a work transference network rediscovery system, and apply it to a workflow enactment event log dataset, in an experimental study. Finally, we describe the implications of discovering and rediscovering work transference networks, as a human resource management and evaluation technique, in workflow-supported enterprises and organizations.

1. Introduction

A workflow[1] (or business process) management system is able to support two fundamental functionalities|modeling functionality and enacting functionality. The modeling functionality allows modelers to define, analyze, and maintain workflow procedures by associating all entities essential to workflow—such as activities, roles, performers, relevant data, and invoked applications—with the corresponding procedures. In contrast, the enacting functionality enables performers to play the essential roles of invoking, executing, and monitoring the instances of the workflow procedures. The logical foundation of such a workflow management system is based upon its underlying workflow model, which implies that the system is able to automate the definition, creation, execution, and management of workflow procedures according to the internal principles and

* Corresponding author.
E-mail address: kwang@kgu.ac.kr (K.P. Kim).

[1] The term *w*orkflow can be used interchangeably with the term *b*usiness process. We prefer the former to the latter in this paper.

https://doi.org/10.1016/j.ins.2019.11.036

structure of the underlying workflow model. Several workflow models [1] have been proposed in the workflow literature, and almost all of them employ the five essential entity types—such as activity, role, performer, repository, and application—to represent organizational workers and their procedural collaborations. We turn our attention to the performer entity type in this paper.

In recent years, the workflow literature has started to be focused on "people" working in workflow-supported organizations, because it is widely accepted for a workflow system to be a "people system". By analyzing the social relationships and collaborative behaviors between the people who are involved in enacting workflow models, we are able to measure and estimate their overall performance in the real organization, and their working productivity as well. The authors' research group has proposed the research and development issues of applying the concept of social networks, and their analysis methods, to human-centered workflow knowledge discovery [2,3] and analysis [4–6]. In this paper, we are particularly interested in the work transference relationships between the performers who participate in the enactment of a specific workflow procedure, which can be formally and graphically represented by a work transference network model.

There exist two main branches of research on the work transference network model. One branch involves *discovery* issues, and the other involves *rediscovery* issues. The former aims to discover a work transference network by analyzing a specific workflow model, whereas the latter is concerned with mining a work transference network from workflow event logs from the model. More specifically, we can differentiate them as follows: the former is to construct a planned work transference network, whereas the latter is to construct an enacted work transference network. This paper directly concerns the work transference network rediscovery issue, which means that the proposed approach can discover an enacted work transference network from a specific workflow's enactment event logs. In conclusion, the purpose of this paper is to originate a fundamental principle for rediscovering a work transference network from a specific workflow's event logs, through which we are able to graphically visualize, and mathematically measure, the human-centered organizational work transference relationships and their intensities that are formed by enacting the corresponding workflow model.

2. Information control nets

The theoretical background is the information control net (ICN) methodology [1], which 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 the activity-to-role, role-to-performer, activity-to-condition/rule, and activity-to-application associations. In this section, we define the formal representation of the ICN of a workflow model, through Definition 1.

Definition 1. *ICN.* A basic **ICN** is an 8-tuple $\Gamma = (\delta, \rho, \lambda, \varepsilon, \pi, \kappa, \mathbf{I}, \mathbf{O})$ over a set **A** of activities[2] (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 the following hold.

- 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.
- Procedural Properties
  - $\delta = \delta_i \cup \delta_o$,
    where δ_o: $\mathbf{A} \longrightarrow \wp(\mathbf{A})$[3] is a multi-valued function mapping an activity to its set of (immediate) successors, and δ_i: $\mathbf{A} \longrightarrow \wp(\mathbf{A})$ is a multi-valued function mapping an activity to its set of (immediate) predecessors.
  - $\rho = \rho_i \cup \rho_o$,
    where ρ_o: $\mathbf{A} \longrightarrow \wp(\mathbf{D})$ is a multi-valued function mapping an activity to its set of output repositories, and ρ_i: $\mathbf{A} \longrightarrow \wp(\mathbf{D})$ is a multi-valued function mapping an activity to its set of input repositories.
- Associative Properties
  - $\lambda = \lambda_a \cup \lambda_g$,
    where λ_g: $\mathbf{A} \longrightarrow \mathbf{G}$ is a single-valued function mapping a task-type activity to its invoked application program, and λ_a: $\mathbf{G} \longrightarrow \wp(\mathbf{A})$ is a multi-valued function mapping an invoked application program to its set of associated task-type activities.
  - $\varepsilon = \varepsilon_a \cup \varepsilon_r$,
    where ε_r: $\mathbf{A} \longrightarrow \mathbf{R}$ is a single-valued function mapping a task-type activity to a role, and ε_a: $\mathbf{R} \longrightarrow \wp(\mathbf{A})$ is a multi-valued function mapping a role to its set of associated task-type activities.
  - $\pi = \pi_r \cup \pi_p$,
    where π_p: $\mathbf{R} \longrightarrow \wp(\mathbf{P})$ is a multi-valued function mapping a role to its set of associated performers, and π_r: $\mathbf{P} \longrightarrow \wp(\mathbf{R})$ is a multi-valued function mapping a performer to its set of associated roles.

[2] It is assumed that activities are classified into three activity types: events, gateways, and tasks.
[3] $\wp(\mathbf{A})$ is the powerset of **A**.

- $\kappa = \kappa_i \cup \kappa_o$,
  where κ_i: $(\mathbf{A} \times \mathbf{A}) \longrightarrow \wp(\mathbf{T})$ is a multi-valued function mapping a set of control-transition conditions, ***T***, on directed edges (ordered pairs), $(\delta_i(\alpha), \alpha \in \mathbf{A})$, from $\delta_i(\alpha)$ to α; and κ_o: $(\mathbf{A} \times \mathbf{A}) \longrightarrow \wp(\mathbf{T})$ is a multi-valued function mapping a set of control-transition conditions, ***T***, on directed edges (ordered pairs), $(\alpha \in \mathbf{A}, \delta_o(\alpha))$, from α to $\delta_o(\alpha)$.

Starting and Terminating Nodes. The execution of a workflow model commences with a single χ transition condition. Therefore, we always assume without the loss of generality that there is a single starting event node (α_I), which has no source node (that is, it has no incoming edges). At the commencement, it is also assumed that all input repositories in the set *I* have been initialized with relevant data from the external workflow model:

$$\{\exists \alpha_I \in A \,|\delta_i(\alpha_I) = \{\emptyset\} \wedge \kappa_o(\alpha_I) = \{\{\chi\}\}\}.$$

The execution is terminated with any single λ outgoing transition condition. In addition, we assume, without loss of generality, that there is a single terminating event node (α_F), which has no destination node (that is, it has no outgoing edges). A set of output repositories *O* is a group of data holders that are transferred to the external workflow model after termination:

$$\{\exists \alpha_F \in A \,|\delta_o(\alpha_F) = \{\emptyset\} \wedge \kappa_i(\alpha_F) = \{\{\lambda\}\}\}.$$

Control Flow: Temporal Ordering of Activities. Given a formal definition, the forward temporal ordering of activities (event-type, gateway-type, and task-type activities) in a workflow model can be formally represented as follows. For any type of activity α, in general

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

which can be interpreted as follows:

- Upon the completion of an activity, α, with a single incoming transition, it simultaneously initiates all of the activities β_{i1} through $\beta_{im(i)}$; in this case, all of the initiated transitions are called parallel (conjunctive) transitions.
- After completing an activity, α, with a single incoming transition, only one value out of n with $[m(i) = 1 \wedge 1 \le i \le n]$ is selected as the result of a decision made; in this case, the selected transition is called a decision (disjunctive or exclusive-OR) transition.
- After completing an activity, α, with a single incoming transition, if $n = 1 \wedge m(n) = 1$, then neither decision nor parallelism is needed, and the transition is called a sequential transition.
- After completing an activity, α, with more than one incoming transition, if $1 \le n \le 2 \wedge m(n) = 1$, then only one value out of $i(1 \le i \le n)$ is selected as the result of a decision made; in this case, the selected transition is called a loop-initiation (repetitive) transition.

The backward temporal ordering of activities in the corresponding workflow model can also be formally represented. In this paper, we will not describe this formal definition in detail, because it can be straightforwardly represented according to the forward temporal ordering of activities.

Graphical Formation. Based on the interpretation, we graphically define several primitive transition types. The conjunctive (or parallel) outgoing transitions are connected by a solid dot (•) with a single incoming transition, and a disjunctive (or decision) outgoing transition is connected by a hollow dot (○) with a single incoming transition. These special types of nodes are called gateway nodes in the workflow literature. The starting and terminating event nodes are a medium-sized circle with a thin line and a medium-sized circle with a thick line, respectively. Naturally, activities and roles are represented by large circles and diamonds, respectively. The gateway nodes need to be formed in matched pairs of split and join types. In addition, multiple pairs of split and join gateway nodes should be kept in a properly nested form, to syntactically support safety and soundness.

The example in Fig. 1 depicts an ICN for the library book acquisition workflow procedure, which was first introduced in [7]. The left-hand side of the figure is the control flow aspect of the ICN, which consists of a starting event node, a terminating event node, 12 task-type activities, a pair of split and join parallel gateway-type nodes, a pair of split and join decision gateway-type nodes, and a pair of loop-initiation and loop-termination gateway-type nodes.

Assigning Roles and Performers. For a specific task-type activity α, $\varepsilon_r(\alpha) = \{\eta\}$ means that the task-type activity α is associated with the role, η. In addition, $\varepsilon_a(\eta) = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$, where m is the number of task-type activities associated with the role, means that a specific role η can be associated with several task-type activities in a workflow model. In terms of the role-to-performer association, for any role η, $\pi_p(\eta) = \{p_1, p_2, \ldots, p_n\}$, where n is the number of performers assigned to the role, means that one or more performers (participants) should be assigned to perform a task-type activity via a role. A role is a named designator for one or more participants, who share work skills, access controls, execution controls, and

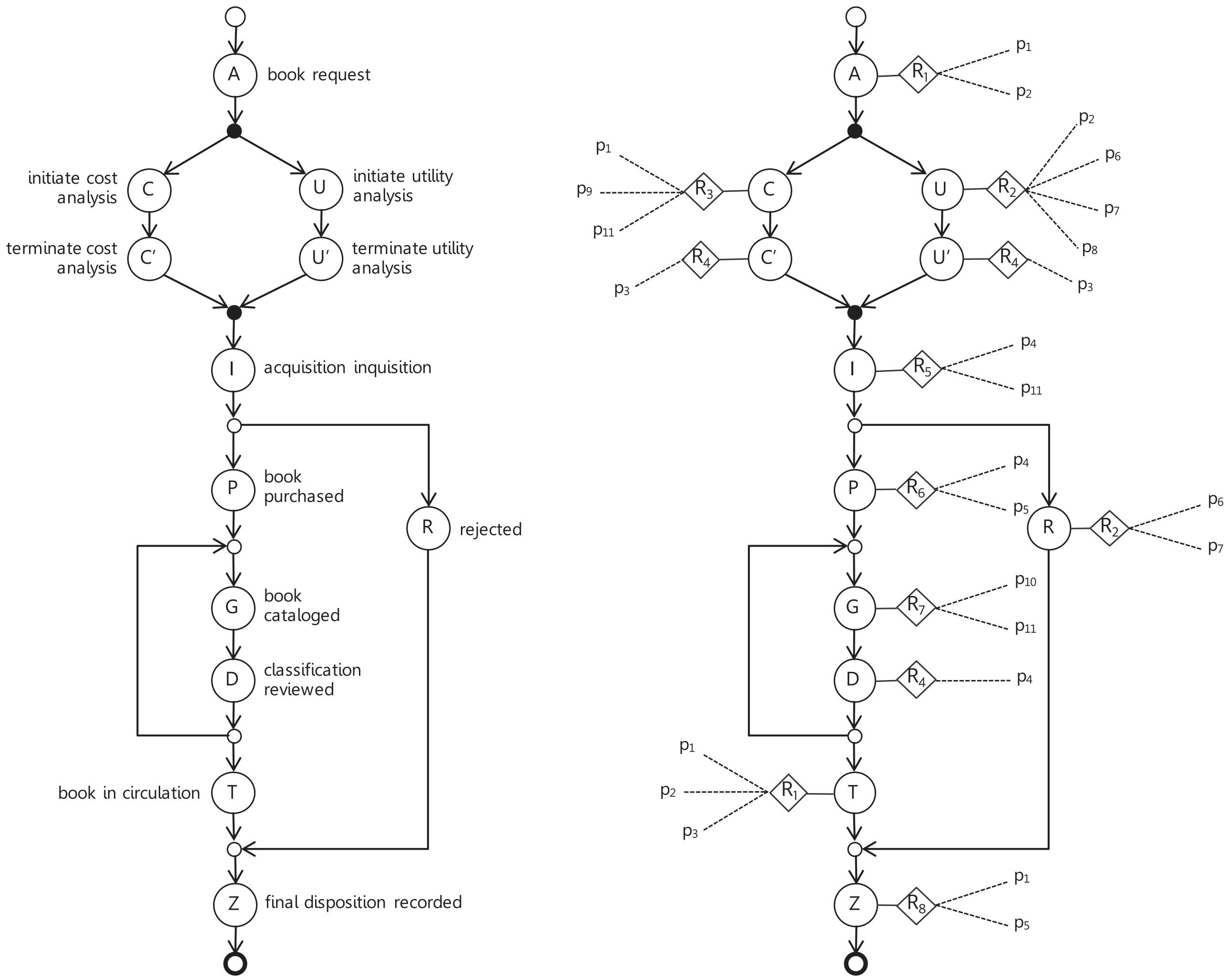

Fig. 1. An example ICN for the library book acquisition workflow model.

authority/responsibility. In addition, for any performer p, $\pi_r(p) = \{\eta_1, \eta_2, \ldots, \eta_m\}$, where m is the number of roles assigned to the performer, means that a specific performer can be assigned to several roles at the same time, and he/she fulfills the roles via their associated task-type activities in a workflow model.

The right-hand side of Fig. 1 shows a graphical formation, representing the role and performer assignment status, in the example workflow model. As the figure shows, the following are some examples in the formal representations for role r_2 and performer p_1 assignments: $\varepsilon_r(\mathrm{U}) = \{r_2\}$, $\varepsilon_a(r_2) = \{\mathrm{U}, \mathrm{R}\}$, $\pi_p(r_2) = \{p_2, p_6, p_7, p_8\}$, and $\pi_r(p_1) = \{r_1, r_3, r_8\}$. In this paper, we particularly focus on these activity-to-role and role-to-performer associative properties, which are directly related to both forming a work transference network in a workflow procedure and discovering the work transference network from the enactment event logs of the corresponding workflow procedure.

3. Work transference networks

In this section, we formally describe the basic concept and definition of work transference networks, which are formed by modeling artifacts of workflow procedures. We know that the procedural activities in a workflow procedure eventually trigger work transferences between the performers who are involved in the workflow procedure. In general, analyzing work transferences between performers is one of the essential planning activities in human resource management and decision-making support management [8]. In particular, the work transference relationships in performing a workflow procedure capture the essential knowledge and criteria for evaluating and validating the human resource performance of the corresponding workflow procedure.

As a formalism to represent such knowledge about work transferences, we formally define a buildtime work transference network model[4], in Definition 2. A work transference network is the formal and graphical structure of a directed graph

[4] The model is discovered from a workflow model defined at build time (the time when the workflow model is constructed).

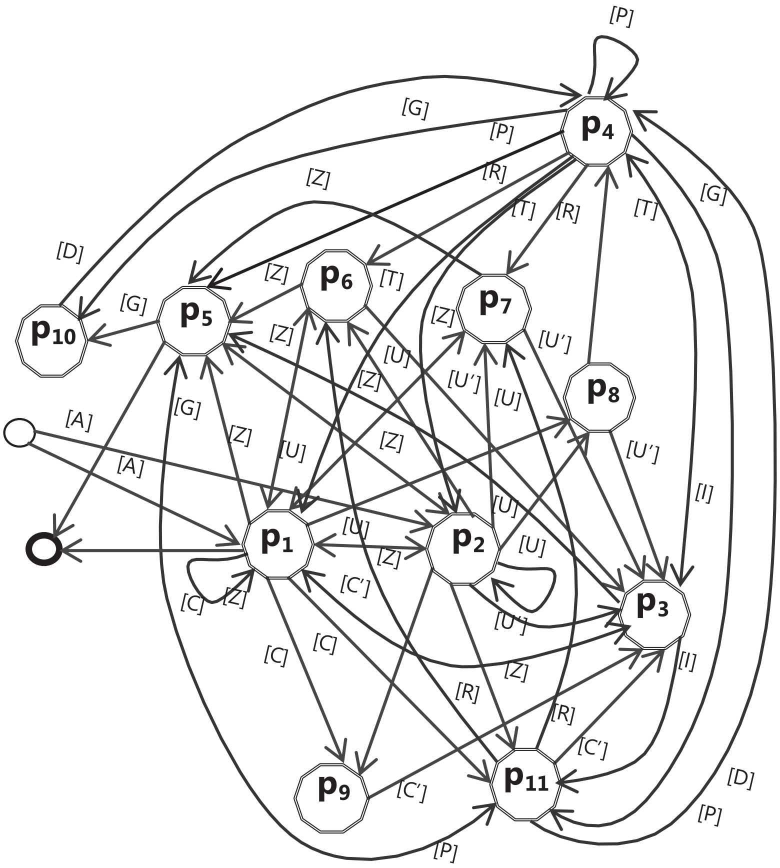

Fig. 2. A buildtime work transference network from the library book acquisition workflow model.

(or digraph) model to represent the relationship of work (activity) transferences, and their associated workitems, between performers who are involved in a corresponding workflow procedure. Each node represents a performer, and each ordered pair of nodes (or directed edge) represents a work transference relationship. Through the formal notation $\sigma\,(=\sigma_i \cup \sigma_o)$, we define the work transference relationships, in which the source and destination nodes of a directed edge represent the transferrer and receiver of work. We also define the formal notation $\psi\,(=\psi_i \cup \psi_o)$ for work association relationships, by labeling each directed edge with the names of workitems that are transferred to the destination node from the source node of the corresponding work transference relationship.

Definition 2. *(Buildtime) Work Transference Network Model.* A buildtime work transference network model is formally defined as $\Lambda^B = (\sigma, \psi, F_r^B, T_o^B)$, over a set $\mathbf{P}$ of performers, and a set $\mathbf{A}$ of activities in a workflow model, where the following hold.

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

Fig. 2 is a buildtime work transference network that graphically illustrates the work transference and work association relationships of the library book acquisition example from the previous section. The network is composed of 11 performers, p_1–p_{11}, and their ordered pairs, labeled with the associated workitems in the activity set, $\mathbf{A} =$