How to Cope with Personnel Unavailability? Process Mining May Help! (DISCUSSION PAPER)

Andrea Chiorrini, Claudia Diamantini, Domenico Potena and Emanuele Storti

DII, Polytechnic University of Marche, Ancona
{a.chiorrini,c.diamantini,d.potena,e.storti}@univpm.it

Abstract. Replacement planning is critical to guarantee continuity of operations in business processes in case of personnel unavailability. In this work, we propose a data-driven approach for supporting resource replacement that makes use of logs of past process executions to model a social network of resources. On this top, a similarity measure among resources is exploited to assign tasks of unavailable resource to the available ones through an Integer Linear Model.

1 Introduction

In organization management, replacement planning consists in addressing short-term lack of workforce that may be due to temporary or permanent unavailability of personnel for several reasons, including for instance vacation, resignation or illness. This issue has a larger impact on organizations when the unavailability cannot be predicted in advance, such as during strikes or emergencies. A recent example of the latter case is the COVID-19 outbreak, which is severely affecting continuity of operations in many public administrations and private companies worldwide.

Traditional approaches in the literature for resource scheduling and replacement planning address the problem typically through business rules modeled by domain experts. On the other hand, nowadays information systems (e.g., ERPs, Workflow Management Systems, CRMs) are capable to track and monitor every event occurring during a business process, such as what activities have been executed, when and by whom. This valuable information can be exploited to support more reliable kinds of data-driven analysis, e.g. through Process Mining techniques.

This paper discusses a data-driven framework for planning of resource replacement. Through the analysis of past executions of business processes the approach is aimed to model a sociogram, i.e. a social network describing how

---

resources collaborate with each other. Several metrics have been devised in the literature to analyse different social relations among resources, including those focusing on how work moves among performers and those considering that people doing similar tasks have stronger relations than people doing completely different things [5]. Here we focus on the both aspects, using a handover of work metric (measuring how frequently work is transferred between two resources) and taking into account capabilities of resources, to evaluate the costs for assigning those activities that should have been performed by the unavailable resource(s), to others. Finally, on the basis of this information, an Integer Linear Programming (ILP) model is defined with the aim of selecting the set of candidate resources to minimize the total cost of replacement.

The work fits into the field of *organizational mining*[4], focusing on the resources (or actors) that performed events and their mutual cooperation in the fulfillment of a process. In [3] an organizational mining framework has been proposed for the discovery, rather than the enforcement, of patterns related to resource assignment using declarative process modelling languages based on rule templates. Our approach shares also the exploitation of an ILP model for task assignment with [2], although the work has a different purpose, namely proposing a recommendation system to support process participants to take a risk-informed decision in choosing the next task to execute out of a set of assigned tasks.

The novel contributions of this work are as follows: (1) we propose a sociogram modeling the handover of work relation that extends traditional models in the literature, by representing the specific type of the transferred activity; (2) on its top, a measure of similarity among resources is defined by considering the degree of collaboration between two resources, the performance and the experience of a resource to perform a given activity; (3) in order to find the best set of resources to replace the unavailable one, an Integer Linear Programming model is proposed to minimize a cost function taking into account similarity among resources.

We propose here an example related to a phone repair process for a company, that will be used throughout the paper. As shown in Figure 1, the process starts with the registration of the repair request (A) and is followed by analysis of defects (B). Then, the user is informed about the outcome (C) and in parallel the repair subprocess is performed. This is achieved by executing either a simple repair (D) or a complex one (E), and is ended by a test to verify whether the repair solved the issues (F). In the negative case, the repair subprocess is reexecuted. Finally, if the test succeeds, the request is archived (G) and the process ends. Let us assume the process involves 6 users, named James, Carrie (in charge of administration activities A, C and G), Mark, Alec, Harrison and Peter (in charge of analysis, repair and test, i.e. activities B, D, E and F). Every record in the log refers to a specific case and is a sequence of events, each of which described in terms of activity, resource and duration in a certain time unit (minutes in the example), as shown in Table 1.

The example is inspired to the ProM 6 tutorial by H.M.W. (Eric) Verbeek at http://www.promtools.org/prom6/downloads/prom-6.0-tutorial.pdf

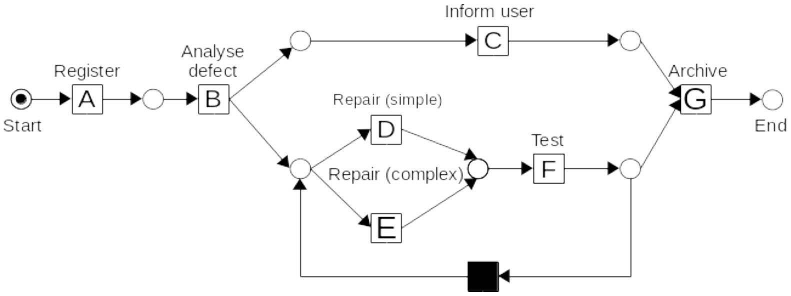

Fig. 1. The example process model.

ID	Trace
1	(A, James, 15), (B, Mark, 60), (C,James, 8), (D, Harrison, 25), (F,Alec, 18), (G, Carrie, 9)
2	(A,Carrie, 12), (B,Harrison, 74), (E,Harrison, 79), (F,Peter, 17), (C,James, 4), (G,James, 13)
3	(A,Carrie, 10), (B,Peter, 43), (D,Peter, 43), (F,Alec, 72), (C,Carrie, 7), (E,Harrison, 115), (F,Alec, 20), (G,James, 14)
4	(A,James, 20), (B,Mark, 96), (C,James, 3), (D,Mark, 31), (F,Alec, 209), (D,Alec, 53), (F,Peter, 23), (G,Carrie, 11)

Table 1. Running example: event log for the repair process. Time unit is minute.

The rest of this work is structured as follows: the next section presents a novel handover of work metric and the sociogram model. Section 3 discusses the proposed methodology for resource replacement. Finally, Section 4 reports preliminary results and draws future research extension.

2 Social network for handover of work

In this section we introduce the terminology and a model of social network for handover of work that is used in the rest of the paper.

Hereby we refer to the term *activity* as a task (or portion thereof) performed to achieve a goal, and to *resource* as any member of personnel that is capable to perform some activity.

Definition 1. *(Event, Trace & Event Log)*
Let $\mathcal{A}$ be a set of activities and $\mathcal{R}$ a set of resources. $V = \mathcal{A} \times \mathcal{R}$ is the set of possible *events*, i.e., combinations of an activity and a resource. Given a resource r, $V(r) \subseteq V$ is the set of events in which r can take part. A *trace* is a possible sequence of events, where $C = V^*$ is the set of all possible traces. An event log $\mathcal{L}$ is a subset of all bags (multi-sets) over C.

In our scenario, we assume the event log includes as attributes at least the case id, the executed activity, the resource name and the timestamp when each event has been performed. For convenience, we introduce the function $\pi_a(\sigma)$ which returns the set of activities related to events of a trace σ.

From the event log, it is possible to characterize a resource $r \in \mathcal{R}$ in terms of his/her capabilities, i.e. the set $\mathcal{A}(r)$ of activities that is able to perform. To make an example, from the log in Table 1 we derive $\mathcal{A}(Mark)$={B, D} and $\mathcal{A}(Harrison)$={B,D,E}.

Among the various relations that can be recognized in an event log between two resources, we focus here on relations of possible causality, and specifically to *handover of work*. Within a case there is a handover of work from a resource r_1 to a resource r_2 if there are two subsequent activities a_1 and a_2 where a_1 is completed by r_1 and a_2 by r_2. The metric can be defined and computed in multiple ways [5], according to the degree of causality (direct or indirect), the existence of multiple succession from a resource to itself (multiple self-transfer), and the kind of succession, i.e. arbitrary succession between two subsequent events in a trace or casual dependency. In this last case, a process model is needed to recognize which is the correct flow of activities in a trace. To make an example, with reference to the case 3, between events (A,Carrie) and (B,Peter) there is a direct and casual succession, while between (F,Alec) and (C,Carrie) there is a direct but arbitrary succession. Indeed, if we compare the log to the process model in Figure 1, there is a causal relation between activities A and B, but this does not hold between activities F and C.

In this work, we refer to the handover of work relation (1) by ignoring self-transfers, (2) by considering indirect succession and (3) casual relation, i.e. we take into consideration succession between activities, with any length, only if aligned to the process model. The reason behind (1) is that the research question of the work is related to resource replacement, for which considering handover from the same resource is not helpful. The reason of (2) and (3) is related to the need to correctly identify handover of work in real processes and avoid spurious relations. Please note that a process model, if not available, can be discovered through Process Discovery algorithms (e.g., Alpha Miner, Heuristic Miner, Inductive Miner). In the following we define the function and the formula for the calculation of the metric.

Definition 2. *(Handover of work)*
Given $r_1, r_2 \in \mathcal{R}$, $a_x \in \mathcal{A}(r_1), a_y \in \mathcal{A}(r_2)$, an event log $\mathcal{L}$, a process model $\mathcal{P}$ and a trace $\sigma \in \mathcal{L}$, the function $r_1 \otimes^{\sigma}_{a_x,a_y} r_2$ returns how many times r_1 and r_2 executed respectively activity $a_x \in \pi_a(\sigma)$ and $a_y \in \pi_a(\sigma)$ such that a causal relation between them in $\mathcal{P}$ exists. For the whole event log $\mathcal{L}$, the function is computed as:

$$q_{xy} = r_1 \otimes^{\mathcal{L}}_{a_x,a_y} r_2 = \sum_{\sigma \in \mathcal{L}} r_1 \otimes^{\sigma}_{a_x,a_y} r_2$$

Finally, the metric *handover of work* between r_1, r_2 for activities a_x, a_y in $\mathcal{L}$ is computed as:

$$h_{xy} = r_1 \odot^{\mathcal{L}}_{a_x,a_y} r_2 = r_1 \otimes^{\mathcal{L}}_{a_x,a_y} r_2 / \Big(\sum_{r_i \in R} \sum_{r_j \in R} r_i \otimes^{\mathcal{L}}_{a_x,a_y} r_j \Big)$$

The metric is computed for a couple of resources r_1, r_2 and with respect to a couple of activities a_x, a_y by dividing the total number of proper casual

	James	Marc	Carrie	Harrison	Alec	Peter
James		(A,B,0.5,2)	(C,G,0.5,2)			
Mark	(B,C,0.5,2)			(B,D,0.3,1)	(D,F,0.25,1)	
Carrie	(C,G,0.25,1)			(A,B,0.25,1)		(A,B, 0.25,1)
Harrison	(B,C,0.25,1)				(D,F,0.25,1) (E,F,0.5,1)	(E,F,0.5,1)
Alec	(F,G,0.25,1)		(F,G,0.25,1)	(F,E,1.0,1)		(D,F,0.25,1)
Peter	(F,G,0.25,1)		(B,C,0.25,1) (F,G,0.25,1)		(D,F,0.25,1)	

Table 2. Running example: handover matrix.

succession (with no self-transfer) by the total number of casual succession of a_x, a_y between any two resources r_i, r_j, with $i \neq j$.

As for the running example, given resources $James$ and $Carrie$, and activities C and G, $James \odot_{C,G}^{\mathcal{L}} Carrie{=}\frac{2}{4} = 0.5$, because the casual succession between them happens 2 times (case 1 and case 4), but the casual succession of C and G between any two resources occurs 4 times.

This information can be represented as a *handover matrix*, that is a matrix $N \times N$, where $N < |\mathcal{R}|$ is the number of active resources (i.e. resources that executed at least 1 activity in the event log). Given two resources $r_i, r_j \in \mathcal{R}$ active in $\mathcal{L}$, the cell (i, j) of the handover matrix includes a tuple $(a_x, a_y, h_{xy}, q_{xy})$, with $a_x \in \mathcal{A}(r_i)$ and $a_y \in \mathcal{A}(r_j)$, if and only if $h_{xy} > 0$. In Table 2 we report the handover matrix for the running example.

From the matrix, a *sociogram* can be defined, i.e. a graph of social relations where nodes are resources and an edge linking two resources is defined if a certain social relation is recognized between them. We refer to the following definition of a sociogram as a labeled multidigraph, i.e. a directed multigraph where two nodes may be linked by multiple labeled edges.

Definition 3. *(Sociogram)*
Given a $N \times N$ handover matrix M, a sociogram G is defined as an 8-tuple $G = (\Sigma_{R'}, \Sigma_E, R', E, s, t, \ell_{R'}, \ell_E)$ where:

- $R' \subseteq \mathcal{R}$ is the finite set of N resources in M and E is a set of arcs representing a handover of work relation;
- $\Sigma_{R'}$ and Σ_E are finite alphabets of the available vertex and arc labels, $s\colon E \to R'$ and $t\colon E \to R'$ are two maps indicating the source and target vertex of an arc;
- $\ell_{R'}\colon R' \to \Sigma_{R'}$ and $\ell_E\colon E \to \Sigma_E$ are two maps describing the labeling of the vertices and arcs.

In the following, for the sake of simplicity, we refer to a sociogram G as a tuple $G = (R', E)$, with E as a multi-set of arcs. Each arc $e \in E$ will be shortly represented as a tuple $e = (r_i, r_j, (a_x, a_y, h_{xy}, q_{xy}))$ linking two nodes $r_i, r_j \in R'$, being r_i the source and r_j the target of e, and $(a_x, a_y, h_{xy}, q_{xy})$ the arc label. For convenience, we introduce the operations $\pi_h(r_i, r_j, a_x) = \{h_{xy} : \exists e = (r_i, r_j, (a_x, a_y, h_{xy}, q_{xy})) \in E\}$ and $\pi_q(r_i, r_j, a_x) = \{q_{xy} : \exists e = (r_i, r_j, (a_x, a_y, h_{xy}, q_{xy})) \in E\}$ which respectively return the values of handover of work between r_i (after a_x has been completed) and r_j, and the numbers of times a_x has been executed by r_i before r_j. Furthermore, the set $G_{r_i}(a_x) = \{r' \in R' : \exists(r_i, r', (a_x, a_y, h_{xy}, q_{xy})) \in E\}$ is introduced to return the subset of resources for which an handover of work relation from r_i after completing a_x exists.

Please note that, differently from the organizational mining literature, we refer to a labeled multidigraph with multiple edges between two nodes. This enables a greater expressiveness as we can encode not only the handover of generic work between two resources, but also take into account which specific activities one resource has handed over the other.

3 Resource replacement methodology

We assume an organization relying on a Business Process Management system with monitoring capabilities of process execution. Execution traces are made available as event logs, which are then analysed and elaborated through *organizational mining* techniques to derive the sociogram as described in Section 2. This information is used to drive the *resource replacement*, that returns the set of specific resources that are selected to replace the missing ones, and their allocation to activities.

Given a sociogram $G = (R', E)$, an unavailable resource $r \in \mathcal{R}$ to replace and a multi-set $T = \{a_1, \ldots, a_h\}$ of activities assigned to r that must be performed by some other resources, the goal is to determine a set of resources $\{r_1, \ldots, r_n\} \subseteq \mathcal{R}$ that is collectively capable to replace r to perform activities in T. We take into account a *similarity* factor such that resources being more compatible to the one to replace are preferred. This takes into account handover of work relations, capabilities of resources, performance and experience.

Similarity measure. Given a resource $r \in \mathcal{R}$ to replace, a candidate resource $r_i \in \mathcal{R}$ and an activity a_j, the similarity between the r and r_j for the specific activity a_j is defined as follows:

$$sim(r, r_i, a_j) = \omega_1 \cdot coll(r, r_i, a_j) + \omega_2 \cdot perf(r, r_i, a_j) + \omega_3 \cdot exp(r, r_i, a_j)$$

The function $sim(r, r_i, a_j)$ returns the degree of affinity between the two resources on the basis of (1) the collaborations that have been established to perform the activity a_j (i.e., $coll(r, r_i, a_j)$), (2) the speed with which a_j has been performed (i.e., $perf(r, r_i, a_j)$) and (3) the experience gained by the two resources in carrying out the activity (i.e., $exp(r, r_i, a_j)$). User parameters $\omega_1, \omega_2, \omega_3 \in [0, 1]$, such that $\omega_1 + \omega_2 + \omega_3 = 1$, are introduced to weigh the three similarity factors.

In details, $G_r(a_j)$ and $G_{r_i}(a_j)$ (see Section 2) are the sets of resources to whom respectively r and r_i handed over work after performing activity a_j. Let $\hat{h}(r_x, r_y, a_j) = \frac{1}{|\pi_h(r_x, r_y, a_j)|} \cdot \sum_{h_i \in \pi_h(r_x, r_y, a_j)} h_i$ be the average handover of work

---

T is a multi-set because an activity may need to be performed multiple times in a process.

between r_x, executing a_j, and r_y. Finally, we introduce the function $\delta(x)$ which returns x if $x \leq 1$, and 1 otherwise. The *coll* function is defined as: $coll(r, r_i, a_j) = \frac{1}{|G_r(a_j)|} \cdot \sum_{r' \in G_r(a_j) \cap G_{r_i}(a_j)} \delta\left(\frac{\hat{h}(r_i, r', a_j)}{\hat{h}(r, r', a_j)}\right)$. It ranges in $[0, 1]$, returning 0 when the two resources do not share any collaborations, 1 if r_i interacts with all collaborators of r with the same handover of work. The *coll* function is introduced under the assumption that replacing a resource with someone acquainted to work in the same team and with similar handover relations is preferable.

The $perf(r, r_i, a_j)$ function compares the time taken by the two resources to carry out the activity a_j. Let t_r and t_{r_i} be the average time respectively taken by r and r_i to perform a_j, the *perf* function is defined as $perf(r, r_i, a_j) = \delta\left(\frac{t_r}{t_{r_i}}\right)$. The *perf* function ranges in $[0, 1]$, values close to 0 mean that r is much faster than r_i. When r_i is on average faster then r, assigning a_j to r_i is also better than having it performed by r, hence the value of *perf* is set to the maximum. The average times to perform an activity, if not available in the event log, can be estimated from the model by considering the difference in time between two causally related events.

Finally, the $exp(r, r_i, a_j)$ function compares the experience of the two resources to perform a_j, computed as the number of times the resource has carried out the activity ($q_r(a_j)$ and $q_{r_i}(a_j)$ respectively). The *exp* function is defined as $exp(r, r_i, a_j) = \delta\left(\frac{q_{r_i}}{q_r}\right)$, where $q_{r'}(a_j) = \sum_{r_x \in G_{r'}(a_j)} \pi_q(r', r_x, a_j)$. This function ranges in $[0, 1]$. Values close to 0 mean that the experience of r_i is much smaller than that of r, while the function takes the maximum value when the experience of r_i is greater than or equal to that of r.

Model specification. An Integer Linear Programming (ILP) model is aimed to select the set of assignments of activities to resources in order to minimize the total cost of resource replacement.

Given a sociogram $G = (R', E)$ and a resource $r \in R'$ to replace, let $C = \{r_i \in R' : \exists a_j \in \mathcal{A}(r_i) \text{ such that } a_j \in T\}$ be the set of candidate resources having at least one capability equal to one activity in T. Finally, Let $|C|$ and $|T|$ be equal to k and h respectively.

We introduce the decision variable $x_{ij}, \forall r_i \in C$ and $\forall a_j \in T$, equal to 1 if the resource r_i is selected for replacement of r to perform activity a_j, 0 otherwise. The objective function $min\left(\sum_{i=1}^{k} \sum_{j=1}^{h} c_{ij} x_{ij}\right)$ represents the total cost of replacing the resource r, taking into account the cost factor $c_{ij} = 1 - sim(r, r_i, a_j)$. An additional constraint is set to impose that each activity in T is assigned to only one resource of C to avoid multiple assignments. Finally, a further constraint defines the binary nature of the decision variables, according to a 0-1 Integer Linear Programming model.

4 Discussion and conclusion

In this work we propose a data-driven approach to support resource replacement in organizations, based on a notion for handover of work that is capable to model collaboration among resources on the basis of causal relations between activities. Although we referred mostly to human-centric processes, the approach is general enough to be applied in cases where a mix of users and machines cooperate together, provided additional constraints on which resources cannot be selected for replacement.

We report some preliminary results of an early experimentation done on the real-world dataset published for the BPI Challenge 2012, referring to an application process for personal loan. From the log we extracted the conform subset, including 7974 traces and 47634 events, with an average number of 6 events per trace (from 3 to 41) and belonging to 8 event types (corresponding to activity names), with 62 resources. We randomly selected a resource and a day in the last 10 days of the log. The procedure is then repeated for all resources. The average number of daily activities that needs to be reassigned per resource is 6.9. On average, in our setting the replace phase took around 6 milliseconds on average per resource to achieve a solution (with an ILP model including up to 1000 variables). The most demanding phase, on the other hand, is the creation of the sociogram, which takes 285 seconds for the whole dataset. Further analysis are not reported here for lack of space and will be deepened in future work. We also plan to extend the model to take into account further social relations among resources and the case of organisations that evolve over time [1]. Furthermore, the average execution time for each activity and the current resource workload will be considered as they can be exploited to add load-balancing constraints in the replacement planning.

References

1. Annalisa Appice. Towards mining the organizational structure of a dynamic event scenario. *Journal of Intelligent Information Systems*, 50(1):165–193, 2018.
2. Raffaele Conforti, Massimiliano de Leoni, Marcello La Rosa, Wil MP van der Aalst, and Arthur HM ter Hofstede. A recommendation system for predicting risks across multiple business process instances. *Decision Support Systems*, 69:1–19, 2015.
3. Stefan Schönig, Cristina Cabanillas, Stefan Jablonski, and Jan Mendling. A framework for efficiently mining the organisational perspective of business processes. *Decision Support Systems*, 89:87 – 97, 2016.
4. Minseok Song and Wil M.P. van der Aalst. Towards comprehensive support for organizational mining. *Decision Support Systems*, 46(1):300 – 317, 2008.
5. Wil MP Van Der Aalst, Hajo A Reijers, and Minseok Song. Discovering social networks from event logs. *Computer Supported Cooperative Work (CSCW)*, 14(6):549–593, 2005.

The experiments run on a 4-core 2.2GHz processor, 16 GB RAM. IBM ILOG CPLEX API (version 12.9) was used for ILP model solving.