Discovering Social Networks from Event Logs

Wil M.P. van der Aalst[1], Hajo A. Reijers[1], Minseok Song[2,1]

[1] Department of Technology Management, Eindhoven University of Technology, P.O.Box 513, NL-5600 MB, Eindhoven, The Netherlands.
w.m.p.v.d.aalst@tm.tue.nl, h.a.reijers@tm.tue.nl

[2] Dept. of Industrial Engineering, Pohang University of Science and Technology, San 31 Hyoja-Dong, Nam-gu, Pohang, 790-784, South Korea. mssong@postech.ac.kr

Abstract. Process mining techniques allow for the discovery of knowledge based on so-called "event logs", i.e., a log recording the execution of activities in some business process. Many information systems provide such logs, e.g., most WFM, ERP, CRM, SCM, and B2B systems record transactions in a systematic way. Process mining techniques typically focus on performance and control-flow issues. However, event logs typically also log the *performer*, e.g., the person initiating or completing some activity. This paper focuses on mining social networks using this information. For example, it is possible to build a social network based on the hand-over of work from one performer to the next. By combining concepts from workflow management and social network analysis, it is possible to discover and analyze social networks. This paper defines metrics, presents a tool, and applies these to a real event log within the setting of a large Dutch organization.

Key words: Process mining, social network analysis, business process management, workflow management, data mining, Petri nets.

1 Introduction

This paper builds on concepts from *business process management* (workflow management in particular) and *sociometry* (social network analysis in particular).

Business process management is concerned with *process-aware information systems*, i.e., systems supporting the design, analysis, and enactment of operational business processes. Typical examples of such process-aware systems are workflow management systems where the process is driven by an explicit process model. However, in many other process-aware information systems the process model is less explicit and users can deviate from the "normal flow", i.e., these systems allow for more flexibility.

Sociometry, also referred to as sociography, refers to methods presenting data on interpersonal relationships in graph or matrix form [12, 43, 46]. The term sociometry was coined by Jacob Levy Moreno who conducted the first long-range sociometric study from 1932-1938 at the New York State Training School for

Girls in Hudson, New York [34]. As part of this study, Moreno used sociometric techniques to assign residents to various residential cottages. He found that assignments on the basis of sociometry substantially reduced the number of runaways from the facility. Many more sociometric studies have been conducted since then by Moreno and others. In most applications of sociometry, the assessment is based on surveys (also referred to as sociometric tests). With the availability of more electronic data, new ways of gathering data are enabled [18]. By analyzing the history of a user's e-mail interactions, personal networks can be extracted. One of the first social-networked tools developed for this purpose is ContactMap [36]. BuddyGraph (www.buddygraph.com) and MetaSight (www.metasight.co.uk) are other examples. By using logs on e-mail traffic as a starting point, meaningful organizational patterns can be distinguished (see e.g., [9, 16, 17, 21, 36, 38]). Similarly, information on the Web can be used for the analysis of social networks. For example, Usenet data has been used to characterize the "authority" of individuals based on posting patterns [44].

For the analysis of social networks around *business processes* such approaches are less useful, since they are based on unstructured information. For example, when analyzing e-mail it is difficult, but also crucial, to distinguish between e-mails corresponding to particular activities within a business process (e.g., the decision with respect to a loan request) and e-mails representing less relevant operational details (e.g., scheduling a meeting). Fortunately, many enterprise information systems store relevant events in a more structured form. For example, workflow management systems typically register the enabling, start and completion of activities [2, 20, 30, 31]. ERP systems like SAP log all transactions, e.g., users filling out forms, changing documents, etc. Business-to-business (B2B) systems log the exchange of messages with other parties. Call center packages but also general-purpose CRM systems log interactions with customers. These examples show that many systems have some kind of *event log* often referred to as "history", "audit trail", "transaction file", etc. [4, 7, 25, 41].

When people are involved in events, logs will typically contain information on the person executing or initiating the *event*. We only consider events both referring to an *activity* and a *case* [4]. The case (also named process instance) is the "thing" which is being handled, e.g., a customer order, a job application, an insurance claim, a building permit, etc. The activity (also named task, operation, action, or work-item) is some operation on the case, e.g., "Contact customer". An event may be denoted by (c, a, p) where c is the case, a is the activity, and p is the person. Events are ordered in time allowing the inference of causal relations between activities and the corresponding social interaction. For example, if (c, a_1, p_1) is directly followed by (c, a_2, p_2), there is some handover of work from p_1 to p_2 (note that both events refer to the same case). If this pattern (i.e., there is some handover of work from p_1 to p_2) occurs frequently but there is never a handover of work from p_1 to p_3 although p_2 and p_3 have identical roles in the organization, then this may indicate that the relation between p_1 and p_2 is stronger than the relation between p_1 and p_3. Using such information it is

possible to build a *social network* expressed in terms of a graph ("sociogram") or matrix.

Social Network Analysis (SNA) refers to the collection of methods, techniques and tools in sociometry aiming at the analysis of social networks [12, 43, 46]. There is an abundance of tools allowing for the visualization of such networks and their analysis. A social network may be dense or not, the "social distances" between individuals may be short or long, etc. An individual may be a so-called "star" (directly linked to many other individuals) or an "isolate" (not linked to others). However, also more subtle notions are possible, e.g., an individual who is only linked to people having many relationships is considered to be a more powerful node in the network than an individual having many connections to less connected individuals.

The work presented in this paper applies the results from sociometry, and SNA in particular, to events logs in today's enterprise information systems. The main challenge is to derive social networks from this type of data. This paper presents the approach, the various metrics that can be used to build a social network, our tool *MiSoN* (Mining Social Networks), and a case study. The paper extends the results presented in [3] by providing concrete metrics and demonstrating these using a case study.

The paper is organized as follows. Section 2 introduces the concept of process mining. Section 3 focuses on the mining organizational relations, introducing concepts from SNA but also showing which relations can be derived from event logs. Section 4 defines the metrics we propose for mining organizational relations. We propose metrics based on (possible) causality, metrics based on joint cases, metrics based on joint activities, and metrics based on special event types (e.g., delegation). Then we present our tool MiSoN. Section 6 presents a case study conducted within a Dutch national public works department employing about 1,000 civil servants. Section 7 presents related work. Finally, Section 8 concludes the paper.

2 Process mining: An overview

The goal of process mining is to extract information about processes from transaction logs [4]. We assume that it is possible to record events such that (i) each event refers to an *activity* (i.e., a well-defined step in the process), (ii) each event refers to a *case* (i.e., a process instance), (iii) each event refers to a *performer* (the person executing or initiating the activity), and (iv) events are totally ordered. Any information system using transactional systems such as ERP, CRM, or workflow management systems will offer this information in some form [2, 20, 30, 31]. An example of an event log is shown in Table 1.

Note that we do not assume the presence of a workflow management system. The only assumption we make, is that it is possible to collect logs with event data. These event logs are used to construct models that explain some aspect of the behavior registered. The term *process mining* refers to methods for distilling a structured process description from a set of real executions [4, 7, 25, 41].

The term "structured process description" may be interpreted in various ways, ranging from a control-flow model expressed in terms of a classical Petri net to a model incorporating organizational, temporal, informational, and social aspects. In Section 7, references to the state-of-the-art using these interpretations are given. In this paper we focus on the social aspect of mining event logs.

case identifier	activity identifier	performer
case 1	activity A	John
case 2	activity A	John
case 3	activity A	Sue
case 3	activity B	Carol
case 1	activity B	Mike
case 1	activity C	John
case 2	activity C	Mike
case 4	activity A	Sue
case 2	activity B	John
case 2	activity D	Pete
case 5	activity A	Sue
case 4	activity C	Carol
case 1	activity D	Pete
case 3	activity C	Sue
case 3	activity D	Pete
case 4	activity B	Sue
case 5	activity E	Clare
case 5	activity D	Clare
case 4	activity D	Pete

Table 1. An event log.

2.1 Discovering social networks

When distilling a process model from an event log, the focus is on the various process activities and their dependencies. When deriving roles and other organizational entities, the focus is on the relation between people or groups of people and the process. Another perspective is to focus on the relations among individuals (or groups of individuals) acting in the process, in other words: the social network. Consider for example the event log of Table 1. Although Carol and Mike can execute the same activities (B and C), Mike is always working with John (cases 1 and 2) and Carol is always working with Sue (cases 3 and 4). Probably Carol and Mike have the same role but based on the small sample shown in Table 1 it seems that John is not working with Carol and Sue is not working with Carol.[1] These examples show that an event log can be used to

[1] Clearly the number of events in Table 1 is too small to establish these assumptions accurately. However, for the sake of argument we assume that the things that did not happen will never happen, cf. Section 2.3.

derive relations between performers of activities, thus resulting in a sociogram. For example, it is possible to generate a sociogram based on the transfers of work from one individual to another as is shown in Figure 1. Each node represents one of the six performers and each arc represents that there has been a transfer of work from one individual to another. The definition of "transfer of work from A to B" is based on whether for the same case an activity executed by A is directly followed by an activity executed by B. For example, both in case 1 and 2 there is a transfer from John to Mike. Figure 1 does not show frequencies. However, for analysis proposes these frequencies can added. The arc from John to Mike would then have weight 2. Typically, we do not use absolute frequencies but weighted frequencies to get relative values between 0 and 1. Figure 1 shows that work is transferred to Pete but not vice versa. Mike only interacts with John and Carol only interacts with Sue. Clare is the only person transferring work to herself.

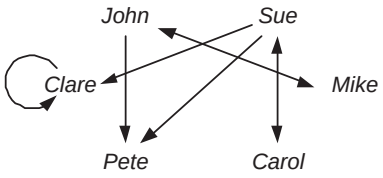


Fig. 1. The sociogram based on the event log shown in Table 1.

For a simple network with just a few cases and performers the results may seem trivial. However, for larger organizations with many cases it may be possible to discover interesting structures. Sociograms as shown in Figure 1 can be used as input for SNA tools that can visualize the network in various ways, compute metrics like the density of the network, analyze the role of an individual in the network (for example the "centrality" or "power" of a performer), and identify cliques (groups of connected individuals). Section 3 will discuss this aspect in more detail and Section 4 will provide concrete metrics to derive sociograms from event logs.

2.2 Other types of mining

Table 1 contains the *minimal information* we assume to be present. Using the information one can also discover other models (i.e., not just sociograms). For example, we have developed techniques and tools to discover the process model. Figure 2 shows the resulting Petri net model after applying our α-algorithm [6] to Table 1. The model shows that the process always starts with A and ends with D. In between these two tasks either E is executed or B and C. B and C are concurrent, i.e., they can be executed in any order. Given the focus of this paper, we will not elaborate further on process discovery. See Section 7 for pointers to related work.

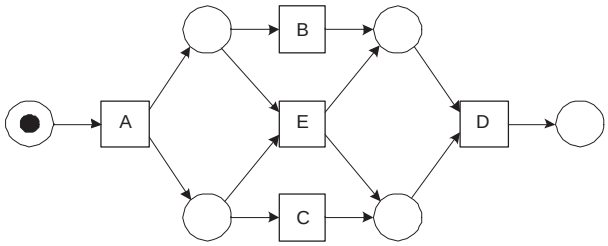


Fig. 2. A process model based on the event log shown in Table 1 discovered by the α-algorithm [6].

In many applications, the event log contains a *time stamp* for each event and this information can be used to extract additional causality information. In addition, a typical log also contains information about the *type of event*, e.g., a start event (a person selecting an activity from a worklist), a complete event (the completion of a activity), a withdraw event (a scheduled activity is removed), etc. Moreover, we are also interested in the relation between attributes of the case and the actual route taken by a particular case or allocation of work to workers. For example, when handling traffic violations: Is the make of a car relevant for the routing of the corresponding traffic violation? (E.g., People driving a Ferrari always pay their fines in time.) Another example directly related to SNA would be to see whether the sociograms for different types of cases (e.g., private and corporate customers) differ.

The presence of timing information and information on cases/activities allows for more advanced forms of process mining, e.g., methods trying to explain the performance indicators like flow times in term of the attributes/performers of cases. Another interesting application of process mining is fraud detection, i.e., detecting suspicions patterns that may indicate security violations (cf. four eyes principle [2]).

2.3 Completeness and noise

For this simple example (i.e., Table 1), it is quite easy to generate the process model shown in Figure 2 or the sociogram shown in Figure 1. For more realistic situations there are however a number of complicating factors:

– *Completeness*
  For larger workflow models mining and models exhibiting alternative and parallel routing, the workflow log will typically not contain all possible routes. Consider 10 activities which can be executed in parallel. The total number of interleavings is 10! = 3628800. It is not realistic that each interleaving is present in the log. Moreover, certain paths through the process model may have a low probability and therefore remain undetected. Similar remarks hold for the organizational model and social network. For example, a person has a role but just by coincidence did not execute some

or all activities corresponding to that role. Another example is that two individuals work together frequently but during the data collection period one of them was on a sabbatical leave. As a result the log is not complete in the sense that it captures possible and/or typical behavior.

– *Noise*
Parts of the log may be incorrect, incomplete, or refer to exceptions. Events can be logged incorrectly because of human or technical errors. Events can be missing in the log if some of the activities are manual or handled by another system/organizational unit. Events can also refer to rare or undesired events. Consider for example the workflow in a hospital. If due to time pressure the order of two events (e.g., make X-ray and remove drain) is reversed, this does not imply that this would be part of the regular medical protocol and should be supported by the hospital's workflow system. Also two causally unrelated events (e.g., take blood sample and death of patient) may happen next to each other without implying a causal relation (i.e., taking a sample did not result in the death of the patient; it was sheer coincidence). Clearly, exceptions which are recorded only once should not automatically become part of the regular workflow.

2.4 Legal issues

To conclude this section, we point out legal issues relevant when mining event logs. Clearly, event logs can be used to systematically measure the performance of employees. The legislation with respect to issues such as privacy and protection of personal data differs from country to country. For example, Dutch companies are bound by the Personal Data Protection Act (Wet Bescherming Persoonsgegeven) which is based on a directive from the European Union. The practical implications of this for the Dutch situation are described in [14, 28, 40]. Event logs are not restricted by these laws as long as the information in the log cannot be traced back to individuals. If information in the log can be traced back to a specific employee, it is important that the employee is aware of the fact that her/his activities are logged and the fact that this logging is used to monitor her/his performance. Note that in a log we can deliberately abstract from information about the workers executing activities and still mine the process, organizational, and social structures (simply hide identities). Therefore, it is possible to avoid collecting information on the productivity of individual workers and legislation such as the Personal Data Protection Act does not apply. Nevertheless, the logs of most workflow systems contain information about individual workers, and therefore, this issue should be considered carefully. Moreover, to use social network analysis as an operational tool to improve work processes, employees should approve and it is vital not to misuse the information gathered.

3 Mining organizational relations

In the previous section, we provided an overview of process mining. In this section, we focus on the main topic of this paper: mining organizational relations

as described in Section 2.1. The goal is to generate a sociogram that can be used as input for standard software in the SNA (Social Network Analysis) domain. In this section we first introduce the fundamentals of SNA and then focus on the question how to derive sociograms from event logs.

3.1 Social network analysis

Applications of SNA range from the analysis of small social networks to large networks. For example, the tool InFlow (www.orgnet.com) has been used to analyze terrorist network surrounding the September 11th 2001 events. However, such tools could also be used to analyze the social network in a classroom. In literature, researchers distinguish between *sociocentric* (whole) and *egocentric* (personal) approaches. Sociocentric approaches consider interactions within a defined group and consider the group as a whole. Egocentric approaches consider the network of an individual, e.g., relations among the friends of a given person. From a mathematical point of view both approaches are quite similar. In both cases the starting point for analysis is graph where nodes represent people and the arcs/edges represent relations. Although this information can also be represented as a matrix, we use the graph notation. The graph can be undirected or directed, e.g., A may like B but not vice versa. Moreover, the relations may be binary (they are there or not) or weighted (e.g., "+" or "-", or a real number). The weight is used to qualify the relation. The resulting graph is named a *sociogram*.

In a mathematical sense such a sociogram is a graph (P, R) where P is the set of individuals (in the context of process mining referred to as performers) and $R \subseteq P \times P$. If the graph is undirected, R is symmetric. If the graph is weighted, there is an additional function W assigning a value to all elements of R. When looking at the graph as a *whole* there are notions like *density*, i.e., the number of elements in R divided by the maximal number of elements, e.g., in a directed graph there are n^2 possible connections (including self loops) where n is the number of nodes. For example the density of the graph shown in Figure 1 is $8/(6 * 6) = 0.22$. Other metrics based on weighted graphs are the maximal geodesic distance in a graph. The geodesic distance of two nodes is the distance of the shortest path in the graph based on R and W.

When looking at one specific individual (i.e., a node in the graph), many notions can be defined. If all other individuals are in short distance to a given node and all geodesic paths (i.e., shorted path in the graph) visit this node, clearly the node is very central (like a spider in the web). There are different metrics for this intuitive notion of *centrality*. The Bavelas-Leavitt index of centrality is a well-known example that is based on the geodesic paths in the graph [8]. Let i be an individual (i.e., $i \in P$) and $D_{j,k}$ the geodesic distance from an individual j to an individual k. The Bavelas-Leavitt index of centrality is defined as $BL(i) = (\sum_{j,k} D_{j,k})/(\sum_{j,k} D_{j,i} + D_{i,k})$. Note that the index divides the sum of all geodesic distances by the sum of all geodesic distances from and to a given resource. Other related metrics are *closeness* (1 divided by the sum of all geodesic distances to a given resource) and *betweenness* (a ratio based on the

number of geodesic paths visiting a given node) [12, 22, 23, 43, 46]. Other notions include the *emission* of a resource (i.e., $\sum_j W_{i,j}$), the *reception* of a resource (i.e., $\sum_j W_{j,i}$), and the *determination degree* (i.e., $\sum_j W_{j,i} - W_{i,j}$) [12, 43, 46]. Another interesting metric is the *sociometric status* which is determined by the sum of input and output relations, i.e., $\sum_j D_{j,i} + D_{i,j}$. All metrics can be normalized by taking the size of the social network into account (e.g., divide by the number of resources). Using these metrics and a visual representation of the network one can analyze various aspects of the social structure of an organization. For example, one can search for densely connected clusters of resources and structural holes (i.e., areas with few connections), cf. [12, 43, 46].

Let us apply some of these notions to the sociogram shown in Figure 1 where the arcs indicate (unweighted) frequencies. The sociometric status of Clare is 2 (if we include self-links), the sociometric status of Pete is 4, the emission of John is 5, the emission of Pete is 0, the reception of Pete is 4, the reception of Sue is 2, the determination degree of Mike is 0, etc. The Bavelas-Leavitt index of centrality of John is 4.33 while the same index for Sue is 3.25. The numbers are unweighted and in most cases these are made relative to allow for easy comparison. Tools like AGNA, Egonet, InFlow, KliqueFinder, MetaSight, NetForm, NetMiner, NetVis, StOCNET, UCINET, and Visone are just some of the many SNA tools available. For more information on SNA we refer to [10, 12, 43, 46].

3.2 Deriving relations from event logs

After showing the potential of SNA and the availability of techniques and tools, the main question is: *How to derive meaningful sociograms from event logs?* To address this question we identify four types of metrics that can be used to establish relationships between individuals: (1) metrics based on (possible) causality, (2) metrics based on joint cases, (3) metrics based on joint activities, and (4) metrics based on special event types.

Metrics based on (possible) causality monitor for individual cases how work moves among performers. One of the examples of such a metric is *handover of work*. Within a case (i.e., process instance) there is a handover of work from individual i to individual j if there are two subsequent activities where the first is completed by i and the second by j. This notion can be refined in various ways. For example, knowledge of the process structure can be used to detect whether there is really a causal dependency between both activities. It is also possible to not only consider direct succession but also indirect succession using a "causality fall factor" β, i.e., if there are 3 activities in-between an activity completed by i and an activity completed by j, the causality fall factor is β^3. A related metric is *subcontracting* where the main idea is to count the number of times individual j executed an activity in-between two activities executed by individual i. This may indicate that work was subcontracted from i to j. Again all kinds of refinements are possible.

Metrics based on joint cases ignore causal dependencies but simply count how frequently two individuals are performing activities for the same case. If individ-

uals work together on cases, they will have a stronger relation than individuals rarely working together.

Metrics based on joint activities do not consider how individuals work together on shared cases but focus on the activities they do. The assumption here is that people doing similar things have stronger relations than people doing completely different things. Each individual has a "profile" based on how frequent they conduct specific activities. There are many ways to measure the "distance" between two profiles thus enabling many metrics.

Metrics based on special event types consider the type of event. Thus far we assumed that events correspond to the execution of activities. However, there are also events like reassigning an activity from one individual to another. For example, if i frequently delegates work to j but not vice versa it is likely that i is in a hierarchical relation with j. From a SNA point of view these observations are particularly interesting since they represent explicit power relations.

The sociogram shown in Figure 1 is based on the causality metric handover of work. In the next section, we will define the metrics in more detail.

4 Metrics

In this section, we define the metrics we have developed to establish relationships between individuals from event logs. We address all types introduced in Section 3.2. Before we define these metrics in detail, we introduce a convenient notation for event logs.

Definition 4.1. (Event log) Let A be a set of activities (i.e., atomic workflow/process objects, also referred to as tasks) and P a set of performers (i.e., resources, individuals, or workers). $E = A \times P$ is the set of (possible) events, i.e., combinations of an activity and a performer (e.g. (a, p) denotes the execution of activity a by performer p). $C = E^*$ is the set of possible event sequences (traces describing a case). $L \in \mathcal{B}(C)$ is an *event log*. Note that $\mathcal{B}(C)$ is the set of all bags (multi-sets) over C.

Note that this definition of an event slightly differs from the informal notions used before. First of all, we abstract from additional information such as time stamps, data, etc. Secondly, we do not consider the ordering of events corresponding to different cases. For convenience, we define two operations on events: $\pi_a(e) = a$ and $\pi_p(e) = p$ for some event $e = (a, p)$.

4.1 Metrics based on (possible) causality

Metrics based on causality take into account both handover of work and subcontracting. The basic idea is that performers are related if there is a causal relation through the passing of a case from one performer to another. For both situations, three kinds of refinements are applied. First of all, one can differentiate with respect to the degree of causality, e.g., the length of handover. It

means that we can consider not only direct succession but also indirect succession. Second, we can ignore multiple transfers within one instance or not. Third, we can consider arbitrary transfers of work or only consider those where there is a casual dependency (for the latter we need to know or be able to derive the process model). Based on these refinements, we derive $2^3 = 8$ variants for both the handover of work and subcontracting metrics. These variants are all based on the same event log. Before defining the metrics, some of the basic notions that can be applied to a single case $c = (c_0, c_1, \ldots)$ are specified.

Definition 4.2. ($\triangleright$,$\trianglerighteq$) Let L be a log. Assume that $\rightarrow$ denotes some causality relation derived from the process model. For $a_1, a_2 \in A$, $p_1, p_2 \in P$, $c = (c_0, c_1, \ldots) \in L$, and $n \in \mathbb{N}$:

- $p_1 \triangleright_c^n p_2 = \exists_{0 \leq i < |c|-n}\ \pi_p(c_i) = p_1\ \wedge\ \pi_p(c_{i+n}) = p_2$
- $|p_1 \triangleright_c^n p_2| = \sum_{0 \leq i < |c|-n} \begin{cases} 1 \text{ if } \pi_p(c_i) = p_1\ \wedge\ \pi_p(c_{i+n}) = p_2 \\ 0 \text{ otherwise} \end{cases}$
- $p_1 \trianglerighteq_c^n p_2 = \exists_{0 \leq i < |c|-n}\ \pi_p(c_i) = p_1\ \wedge\ \pi_p(c_{i+n}) = p_2\ \wedge\ \pi_a(c_i) \rightarrow \pi_a(c_{i+n})$
- $|p_1 \trianglerighteq_c^n p_2| = \sum_{0 \leq i < |c|-n} \begin{cases} 1 \text{ if } \pi_p(c_i) = p_1\ \wedge\ \pi_p(c_{i+n}) = p_2\ \wedge \\ \quad \pi_a(c_i) \rightarrow \pi_a(c_{i+n}) \\ 0 \text{ otherwise} \end{cases}$

$p_1 \triangleright_c^n p_2$ denotes the function which returns *true* if within the context of case c performers p_1 and p_2 both executed some activity such that the distance between these two activities is n. For example, for case 1 shown in Table 1, $John \triangleright_c^1 Mike$ equals 1 (i.e., true) and $John \triangleright_c^3 Pete$ equals 1 (i.e., true). In this definition, if the value of n equals 1, it refers to direct succession. If n is greater than 1, it refers to indirect succession. However, it ignores both multiple transfers within one instance and casual dependencies. $|p_1 \triangleright_c^n p_2|$ denotes the function which returns the number of times $p_1 \triangleright_c^n p_2$ in the case c. In other words, it considers multiple transfers within one instance. $p_1 \trianglerighteq_c^n p_2$ and $|p_1 \trianglerighteq_c^n p_2|$ are similar to $p_1 \triangleright_c^n p_2$ and $|p_1 \triangleright_c^n p_2|$ but in addition they take into account whether there is a real casual dependency. For example, consider case 1 shown in Table 1. The order of events is: A (John), B (Mike), C (John), and D (Pete). If we calculate the relationships among activity B, C, and D, $Mike \triangleright_c^1 John$ equals 1 and $Mike \triangleright_c^1 Pete$ equals 0. However, $Mike \trianglerighteq_c^1 John$ equals 0, i.e., although an activity conducted by Mike is followed an activity conducted by John there is not a causal dependency between B and C because both activities are in parallel. However, there is casual dependency between activity B and D (see Figure 2) and, therefore, $Mike \trianglerighteq_c^2 Pete$ equals 1. The information on causality can be added if the process model is known. If necessary, this information can also be derived from the log by using for example the α-algorithm [6].

Using such relations, we define handover of work metrics. Based on three kinds of refinements mentioned before, eight variants are derived as follows.

Definition 4.3. (Handover of work metrics) Let L be a log. For $p_1, p_2 \in P$ and some β $(0 < \beta < 1)$:

- $p_1 \triangleright_L p_2 = (\sum_{c \in L} |p_1 \triangleright_c^1 p_2|)/(\sum_{c \in L} |c| - 1)$

- $p_1 \dot{\triangleright}_L p_2 = (\sum_{c \in L \ \wedge \ p_1 \triangleright_c^1 p_2} 1)/|L|$
- $p_1 \triangleright_L^\beta p_2 = (\sum_{c \in L} \sum_{1 \leq n < |c|} \beta^{n-1} |p_1 \triangleright_c^n p_2|)/(\sum_{c \in L} \sum_{1 \leq n < |c|} \beta^{n-1}(|c| - n))$
- $p_1 \dot{\triangleright}_L^\beta p_2 = (\sum_{c \in L} \sum_{1 \leq n < |c| \ \wedge \ p_1 \triangleright_c^n p_2} \beta^{n-1})/(\sum_{c \in L} \sum_{1 \leq n < |c|} \beta^{n-1})$
- $p_1 \trianglerighteq_L p_2 = (\sum_{c \in L} |p_1 \trianglerighteq_c^1 p_2|)/(\sum_{c \in L} |c| - 1)$
- $p_1 \dot{\trianglerighteq}_L p_2 = (\sum_{c \in L \ \wedge \ p_1 \trianglerighteq_c^1 p_2} 1)/|L|$
- $p_1 \trianglerighteq_L^\beta p_2 = (\sum_{c \in L} \sum_{1 \leq n < |c|} \beta^{n-1} |p_1 \trianglerighteq_c^n p_2|)/(\sum_{c \in L} \sum_{1 \leq n < |c|} \beta^{n-1}(|c| - n))$
- $p_1 \dot{\trianglerighteq}_L^\beta p_2 = (\sum_{c \in L} \sum_{1 \leq n < |c| \ \wedge \ p_1 \trianglerighteq_c^n p_2} \beta^{n-1})/(\sum_{c \in L} \sum_{1 \leq n < |c|} \beta^{n-1})$

$p_1 \triangleright_L p_2$ means dividing the total number of direct successions from p_1 to p_2 in a process log by the maximum number of possible direct successions in the log. $p_1 \ \dot{\triangleright}_L \ p_2$ ignores multiple transfers within one instance (i.e., case). For example, in Table 1, *John* $\triangleright_L$ *Mike* equals 2/14 and *John* $\dot{\triangleright}_L$*Mike* euqals 2/5. $p_1 \triangleright_L^\beta p_2$ and $p_1 \ \dot{\triangleright}_L^\beta \ p_2$ deal with indirect succession by introducing a "causality fall factor" β in this notation. If within the context of a case there are n events in-between two performers, the causality fall factor is β^n. $p_1 \triangleright_L^\beta p_2$ considers all possible successions, while $p_1 \ \dot{\triangleright}_L^\beta \ p_2$ ignores multiple transfers within one case. For example, in Table 2, if β equals 0.5, then *John* $\triangleright_L$ *Pete* equals 2.5/19.5 and *John* $\dot{\triangleright}_L$ *Pete* equals 2.5/8.5. If we use a β close to 1, the effect of the distance between performers decreased. For example, suppose that only case 1 exists in Table 1, we calculate the handover of metrics from John in Activity A to Mike, John in Activity B, and Pete, according to various values of β. Table 2 shows the results. If the value β increases in value, the variance of resulting values decreases.

beta	$John \triangleright_L^\beta Mike$	$John \triangleright_L^\beta John$	$John \triangleright_L^\beta Pete$
0.1	0.3116 (1/3.21)	0.0312 (0.1/3.21)	0.0031 (0.01/3.21)
0.5	0.2352 (1/4.25)	0.1176 (0.5/4.25)	0.0588 (0.25/4.25)
0.9	0.1783 (1/5.61)	0.1604 (0.9/5.61)	0.1444 (0.81/5.61)

Table 2. Handover of work metrics according to the causality fall factor β.

The remaining four metrics $p_1 \trianglerighteq_L p_2$, $p_1 \dot{\trianglerighteq}_L p_2$, $p_1 \trianglerighteq_L^\beta p_2$, and $p_1 \dot{\trianglerighteq}_L^\beta p_2$ are similar to the previous four kinds of metrics, but take into account real casual dependencies. For example, $p_1 \trianglerighteq_L p_2$ means that the total number of direct successions from p_1 to p_2 in a log is divided by the maximum number of possible direct successions in the log when p_1 and p_2 are casually related.

From above definitions, we derive general formulations of the metrics. The eight metrics mentioned can be merged into the following four metrics.

Definition 4.4. (General forms of handover of work metrics) Let L be a log. For $p_1, p_2 \in P$, some β $(0 < \beta \leq 1)$ and $k \in \mathbb{N}$.

- $p_1 \rhd_L^{\beta,k} p_2 = \frac{(\sum_{c\in L}\sum_{1\le n\le \min(|c|-1,k)} \beta^{n-1}|p_1 \rhd_c^n p_2|/}{(\sum_{c\in L}\sum_{1\le n\le \min(|c|-1,k)} \beta^{n-1}(|c|-n))}$
- $p_1 \dot{\rhd}_L^{\beta,k} p_2 = \frac{(\sum_{c\in L}\sum_{1\le n\le \min(|c|-1,k)\ \wedge\ p_1 \rhd_c^n p_2} \beta^{n-1}/}{(\sum_{c\in L}\sum_{1\le n\le \min(|c|-1,k)} \beta^{n-1})}$
- $p_1 \trianglerighteq_L^{\beta,k} p_2 = \frac{(\sum_{c\in L}\sum_{1\le n\le \min(|c|-1,k)} \beta^{n-1}|p_1 \trianglerighteq_c^n p_2|/}{(\sum_{c\in L}\sum_{1\le n\le \min(|c|-1,k)}\ \beta^{n-1}(|c|-n))}$
- $p_1 \dot{\trianglerighteq}_L^{\beta,k} p_2 = \frac{(\sum_{c\in L}\sum_{1\le n\le \min(|c|-1,k)\ \wedge\ p_1 \trianglerighteq_c^n p_2} \beta^{n-1}/}{(\sum_{c\in L}\sum_{1\le n\le \min(|c|-1,k)} \beta^{n-1})}$

In these alternative formulations, we introduce a "calculation depth factor" k. When we calculate metrics, k specifies maximum degree of casuality. For example, if k equals 3, it considers the case of direct succession, one event in between two performers, and two events in-between two performers. Note that if $\beta = 1, k = 1$, then $p_1 \rhd_L^{1,1} p_2 = p_1 \rhd_L p_2$, and if $k > \max(|c|)$, then $p_1 \rhd_L^{\beta,k} p_2 = p_1 \rhd_L^{\beta} p_2$. This rule is also applied to the other three metrics. Further, when we calculate the metrics, a suitable value for k is important for the efficiency of calculation. Logs are typically very large. Therefore considering all possible successions may be inefficient.

After defining metrics for handover of work we now consider another class of metrics based on (possible) causality: *subcontracting metrics*. In the case of subcontracting, the three refinements mentioned before can also be applied. However the concept of direct and indirect succession is changed. Direct succession means there is only one activity in-between two activities executed by one performer. While indirect succession means, there are multiple activities in-between two activities executed by one performer. We also introduce causality fall factor β for indirect succession. For example, assume that there are four activities. Both the first and the fourth activity are executed by a performer i, while the second and third activity are executed by performer j and k respectively. In this situation, we can derive two relations which are from a performer i to a performer j and from a performer i to a performer k. Again we use a causality fall factor β. The second and third refinements are the same as for handover of work. Before defining metrics, the basic notions applied to a single case $c = (c_0, c_1, \ldots)$ are specified.

Definition 4.5. ($\diamond$,$\underline{\diamond}$) Let L be a log. Assume that $\rightarrow$ denotes some causality relation. In the context of L and $\rightarrow$, we define a number of relations. For $a_1, a_2 \in A$, $p_1, p_2 \in P$, $c = (c_0, c_1, \ldots) \in L$, $|c| > 2$, $n \in \mathbb{N}$, and $n > 1$:

- $p_1 \diamond_c^n p_2 = \exists_{0\le i<j<i+n<|c|} \pi_p(c_i) = p_1 \ \wedge\ \pi_p(c_j) = p_2 \ \wedge\ \pi_p(c_{i+n}) = p_1$
- $|p_1 \diamond_c^n p_2| = \sum_{0\le i<|c|-n} \sum_{i<j<i+n} \begin{cases} 1 \text{ if } \pi_p(c_i) = p_1 \ \wedge\ \pi_p(c_j) = p_2 \ \wedge \\ \quad \pi_p(c_{i+n}) = p_1 \\ 0 \text{ otherwise} \end{cases}$
- $p_1 \underline{\diamond}_c^n p_2 = \begin{array}{l} \exists_{0\le i<j<i+n<|c|} \pi_p(c_i) = p_1 \ \wedge\ \pi_p(c_j) = p_2 \ \wedge\ \pi_p(c_{i+n}) = p_1 \ \wedge \\ \pi_a(c_i) \rightarrow \pi_a(c_j) \rightarrow \pi_a(c_{i+n}) \end{array}$

– $|p_1 \underline{\diamond}_c^n p_2| = \sum_{0 \leq i < |c|-n} \sum_{i<j<i+n} \begin{cases} 1 & \text{if } \pi_p(c_i) = p_1 \ \wedge \ \pi_p(c_j) = p_2 \ \wedge \\ & \pi_p(c_{i+n}) = p_1 \ \wedge \\ & \pi_a(c_i) \rightarrow \pi_a(c_j) \rightarrow \pi_a(c_{i+n}) \\ 0 & \text{otherwise} \end{cases}$

$p_1 \diamond_c^n p_2$ denotes the function which returns *true* if performer p_2 executed an activity in-between two activities executed by performer p_1 and distance between these two activities executed by performer p_1 is n. For example, for case 1 shown in Table 1, *John* $\diamond_c^2$ *Mike* equals 1. However, it ignores both multiple transfers within one instance and casual dependencies. $|p_1 \diamond_c^n p_2|$ denotes the function which returns the number of times $p_1 \diamond_c^n p_2$ in the case c. In other words, it considers multiple transfers within one instance. $p_1 \underline{\diamond}_c^n p_2$ and $|p_1 \underline{\diamond}_c^n p_2|$ are similar to $p_1 \diamond_c^n p_2$ and $|p_1 \diamond_c^n p_2|$ but in addition they take into account whether there is a real casual dependency. For example, consider case 1 shown in Table 1. *John* $\underline{\diamond}_c^2$ *Mike* equals 0, because activity B and C do not have a casual dependency.

Using such relations, we define subcontracting metrics. Again eight variants are identified.

Definition 4.6. (In-between metrics) Let L be a log. For $p_1, p_2 \in P$, $c = (c_0, c_1, \ldots) \in L$, $|c| > 2$, and some β $(0 < \beta < 1)$:

– $p_1 \diamond_L p_2 = (\sum_{c \in L} |p_1 \diamond_c^2 p_2|)/(\sum_{c \in L} (|c| - 2))$
– $p_1 \dot{\diamond}_L p_2 = (\sum_{c \in L \ \wedge \ p_1 \diamond_c^2 p_2} 1)/|L|$
– $p_1 \diamond_L^\beta p_2 = \begin{array}{l} (\sum_{c \in L} \sum_{2 \leq n < |c|} \beta^{n-2} |p_1 \diamond_c^n p_2|)/ \\ (\sum_{c \in L} \sum_{2 \leq n < |c|} \beta^{n-2} (|c| - n)(n-1)) \end{array}$
– $p_1 \dot{\diamond}_L^\beta p_2 = (\sum_{c \in L} \sum_{2 \leq n < |c| \ \wedge \ p_1 \diamond_c^n p_2} \beta^{n-2})/(\sum_{c \in L} \sum_{2 \leq n < |c|} \beta^{n-2})$
– $p_1 \underline{\diamond}_L p_2 = (\sum_{c \in L} |p_1 \underline{\diamond}_c^2 p_2|)/(\sum_{c \in L} (|c| - 2))$
– $p_1 \dot{\underline{\diamond}}_L p_2 = (\sum_{c \in L \ \wedge \ r_1 \underline{\diamond}_c^2 p_2} 1)/|L|$
– $p_1 \underline{\diamond}_L^\beta p_2 = \begin{array}{l} (\sum_{c \in L} \sum_{2 \leq n < |c|} \beta^{n-2} |p_1 \underline{\diamond}_c^n p_2|)/ \\ (\sum_{c \in L} \sum_{2 \leq n < |c|} \beta^{n-2} (|c| - n)(n-1)) \end{array}$
– $p_1 \dot{\underline{\diamond}}_L^\beta p_2 = (\sum_{c \in L} \sum_{2 \leq n < |c| \ \wedge p_1 \ \underline{\diamond}_c^n p_2} \beta^{n-2})/(\sum_{c \in L} \sum_{2 \leq n < |c|} \beta^{n-2})$

$p_1 \diamond_L p_2$ means dividing the total number of direct subcontracting occurrences between p_1 and p_2 in a process log by the maximum number of possible direct subcontracting occurrences in the log. $p_1 \dot{\diamond}_L p_2$ ignores multiple subcontracting occurrences within one instance (i.e., case). For example, in Table 1, *John* $\diamond_L$ *Mike* equals 2/9 and *John* $\dot{\diamond}_L$ *Mike* equals 2/5. $p_1 \diamond_L^\beta p_2$ and $p_1 \dot{\diamond}_L^\beta p_2$ deal with the situation where the distance between these two activities executed by performer p_1 is greater than 2. Again we introduce a "causality fall factor" β in a fashion similar to the handover of work metrics. If within the context of a case there are n events in-between two activities executed by the same performer, the causality fall factor is β^n. $p_1 \diamond_L^\beta p_2$ considers all possible subcontracting occurrences, while $p_1 \dot{\diamond}_L^\beta p_2$ ignores multiple subcontracting within one case. For example, in Table 2, if β equals 0.5, then *John* $\diamond_L$ *Mike* equals 2/13 and *John* $\dot{\triangleright}_L$ *Mike* equals 2/7. Again $p_1 \underline{\diamond}_L p_2$, $p_1 \dot{\underline{\diamond}}_L p_2$, $p_1 \underline{\diamond}_L^\beta p_2$, and $p_1 \dot{\underline{\diamond}}_L^\beta p_2$ are similar but take into account real

casual dependencies. For example, $p_1 \underline{\diamond}_L p_2$ means that the total number of direct subcontracting from p_1 to p_2 in a process log is divided by the maximum number of possible direct subcontracting in the log when p_1 and p_2 are casually related.

As before we can derive more general formulations for the metrics. The eight metrics mentioned above can be merged into four metrics as shown in the following definition.

Definition 4.7. (General forms of in-between metrics) Let L be a log. For $p_1, p_2 \in P$, some β $(0 < \beta \leq 1)$ and $k \in \mathbb{N}$ $(k > 1)$

- $p_1 \diamond_L^{\beta,k} p_2 = \frac{(\sum_{c \in L} \sum_{2 \leq n \leq \min(|c|-1,k)} \beta^{n-2} |p_1 \diamond_c^n p_2|)/}{(\sum_{c \in L} \sum_{2 \leq n \leq \min(|c|-1,k)} \beta^{n-2}(|c|-n)(n-1))}$
- $p_1 \dot{\diamond}_L^{\beta,k} p_2 = \frac{(\sum_{c \in L} \sum_{2 \leq n \leq \min(|c|-1,k) \ \wedge \ p_1 \diamond_c^n p_2} \beta^{n-2})/}{(\sum_{c \in L} \sum_{2 \leq n \leq \min(|c|-1,k)} \beta^{n-2})}$
- $p_1 \underline{\diamond}_L^{\beta,k} p_2 = \frac{(\sum_{c \in L} \sum_{2 \leq n \leq \min(|c|-1,k)} \beta^{n-2} |p_1 \underline{\diamond}_c^n p_2|)/}{(\sum_{c \in L} \sum_{2 \leq n \leq \min(|c|-1,k)} \ \beta^{n-2}(|c|-n)(n-1))}$
- $p_1 \dot{\underline{\diamond}}_L^{\beta,k} p_2 = \frac{(\sum_{c \in L} \sum_{2 \leq n \leq \min(|c|-1,k) \ \wedge \ p_1 \underline{\diamond}_c^n p_2} \beta^{n-2})/}{(\sum_{c \in L} \sum_{2 \leq n \leq \min(|c|-1,k)} \beta^{n-2})}$

Again we also introduce a "calculation depth factor" k. When calculating the metrics, k specifies maximum distance between two activities executed by one performer. For example, if k equals 3, it considers the case of one activity in between two activities executed by one performer and two activities in between two activities executed by one performer. Note that if $\beta = 1, k = 2$, then $p_1 \diamond_L^{1,2} p_2 = p_1 \diamond_L p_2$, and if $k > \max(|c|)$, then $p_1 \diamond_L^{\beta,k} p_2 = p_1 \diamond_L^{\beta} p_2$.

4.2 Metrics based on joint cases

For this type of metrics we ignore causal dependencies and simply count how often two individuals are performing activities for the same case.

Definition 4.8. (Working together metrics) Let L be a log. For $p_1, p_2 \in P$: $p_1 \bowtie_L p_2 = \sum_{c \in L} p_1 \bowtie_c p_2 / \sum_{c \in L} g(c, p_1)$ if $\sum_{c \in L} g(c, p_1) \neq 0$, otherwise $p_1 \bowtie_L p_2 = 0$, where for $c = (c_0, c_1, \ldots) \in L$: $p_1 \bowtie_c p_2 = 1$ if $\exists_{0 \leq i,j < |c| \wedge i \neq j}$ $\pi_p(c_i) = p_1 \ \wedge \ \pi_p(c_j) = p_2$, otherwise $p_1 \bowtie_c p_2 = 0 : g(c, p_1) = 1$ if $\exists_{0 \leq i < |c|} \pi_p(c_i) = p_1$, otherwise $g(c, p_1) = 0$

Note that, in this definition we divide the number of joint cases by the number of cases in which p_1 appeared. It is important to use a relative notation. For example, suppose that p_1 participates in three cases, p_2 participates in six cases, and they work together three times. In this situation, p_1 always work together with p_2, but p_2 does not. Thus, the value for $p_1 \bowtie_L p_2$ has to be larger than the value for $p_2 \bowtie_L p_1$. Let us apply this metric to analyze the relationship between John and Pete based in the log shown in Table 1. In the log, John appeared in two cases, Pete in four cases, and they work together on two cases. Thus, $John \bowtie_L Pete = 2/2$ and $Pete \bowtie_L John = 2/4$.

Moreover, alternative metrics can be composed by taking the distance between activities into account, e.g., use variants like $(p_1 \rhd_L^\beta p_2 + p_2 \rhd_L^\beta p_1)/2$ or $(p_1 \dot{\rhd}_L^\beta p_2 + p_2 \dot{\rhd}_L^\beta p_1)/2$.

4.3 Metrics based on joint activities

To calculate the metrics based on joint activities, first we make a "profile" based on how frequent individuals conduct specific activities. In this paper, we use a *performer by activity matrix* to represent these profiles. This matrix simply records how frequent each performer executes specific activities.

Definition 4.9. ($\triangle$) Let L be a log. For $p_1 \in P$, $a_1 \in A$, and $c = (c_0, c_1, \ldots) \in L$:

- $p_1 \triangle_c a_1 = \sum_{0 \leq i < |c|} \begin{cases} 1 \text{ if } \pi_a(c_i) = a_1 \ \wedge \ \pi_p(c_i) = p_1 \\ 0 \text{ otherwise} \end{cases}$
- $p_1 \triangle_L a_1 = \sum_{c \in L} p_1 \triangle_c a_1$

Note that $\triangle$ defines a matrix with rows P and columns A. Table 3 shows the performer by activity matrix derived from Table 1.

performer	activity A	activity B	activity C	activity D	activity E
John	2	1	1	0	0
Sue	3	1	1	0	0
Mike	0	1	1	0	0
Carol	0	1	1	0	0
Pete	0	0	0	4	0
Clare	0	0	0	1	1

Table 3. The performer by activity matrix.

After creating the matrix, we measure the distance between two performers by comparing the corresponding row vectors. A simple distance measure is *Minkowski distance* which can be seen as a generalization of the Euclidean distance. But the Minkowski distance only gives good results if performers execute comparable volumes of work. Therefore, we also use the *Hamming distance* which does not consider the absolute frequency but only whether it is 0 or not. Another metric is *Pearson's correlation coefficient* which is frequently used to find the relationship among cases.

Definition 4.10. ($\triangle_L^{MD,n}$,$\triangle_L^{HD}$,$\triangle_L^{PC}$) Let L be a log and $\triangle_L$ be a performer by activity matrix. For $p_1, p_2 \in P$, $n \in \{1, 2, 3, \ldots\}$:

- $p_1 \triangle_L^{MD,n} p_2 = (\sum_{a \in A} |(p_1 \triangle_L a) - (p_2 \triangle_L a)|^n)^{1/n}$
- $p_1 \triangle_L^{HD} p_2 = (\sum_{a \in A} \delta(p_1 \triangle_L a, p_2 \triangle_L a))/|A|$
  where $\delta(x, y) = \begin{cases} 0 \text{ if } (x > 0 \ \wedge \ y > 0) \vee (x = y = 0) \\ 1 \text{ otherwise} \end{cases}$

$$- \; p_1 \triangle_L^{PC} p_2 = \frac{\sum_{a \in A}(((p_1 \triangle_L a) - \bar{X})((p_2 \triangle_L a) - \bar{Y}))/}{\sqrt{\sum_{a \in A}((p_1 \triangle_L a) - \bar{X}) \sum_{a \in A}((p_2 \triangle_L a) - \bar{Y})}}$$
$$\text{where } \bar{X} = \sum_{a \in A}(p_1 \triangle_L a)/|A|, \bar{Y} = \sum_{a \in A}(p_2 \triangle_L a)/|A|$$

The Minkowski distance $\triangle_L^{MD,n}$ has a parameter n: $n = 1$ is the Rectilinear distance also referred to as Manhattan distance, $n = 2$ is the Euclidean distance, and for large values of n the metric approximates the Chebyshev distance. The Hamming distance $\triangle_L^{HD}$ does not have a parameter but could be extended with some threshold value. In the case of Pearson's correlation coefficient, the result ranges from +1 to -1. A correlation of +1 means that there is a perfect positive linear relationship between variables. A correlation of -1 means that there is a maximal negative linear relationship between variables. In other words, if the distance between performers is small, the correlation is closer to 1, if it is large, the correlation is closer to -1.

To illustrate the limitations of simple metrics like the Minkowski distance we consider Table 3. Clearly, from an intuitive point of view the distance between Sue and Carol should be smaller than the distance between Carol and Clare because Carol and Clare have no activities in common. The Minkowski distance ($n = 1$) between Sue and Carol equals 3 and the distance between Carol and Clare equals 4. However, if Sue would have executed activity B and activity C also three times, the distance between Sue and Carol would be 7 and thus incorrectly suggest that Carol is closer to Clare than Sue. The Hamming distance is more robust and would indicate in both cases that Carol is closer to Sue: *Sue* $\triangle_L^{HD}$ *Carol* equals 1/5 and *Carol* $\triangle_L^{HD}$ *Clare* equals 4/5. If we calculate the Pearson's correlation coefficient, *Sue* $\triangle_L^{PC}$ *Carol* equals 0.2182 and *Carol* $\triangle_L^{PC}$ *Clare* equals -0.6667. Hence, the result of Pearson's correlation leads to the same conclusion as the Hamming distance.

Note that if the volume of work varies significantly, the metrics are not suitable. For example, it is difficult to compare the profile of a part-time worker with a full-time worker. Thus, in some cases we first apply the $log_k(X + 1)$ function on the values of the performer by activity matrix, i.e., use a logarithmic scale for $\triangle_L$. Note that we need to add "+1" to avoid negative values.

4.4 Metrics based on special event types

The types of metrics mentioned in previous subsections do not consider event types. They more or less assume that all events correspond to the completion of an activity. But events can contain various event types such as *schedule*, *assign*, *withdraw*, *reassign*, *start*, *suspend*, *resume*, *pi_abort*, *ate_abort*, *complete*, *autoskip*, *manualskip*, and *unknown*. For example, *schedule* refers to the enabling of a task for a specific case, *assign* refers to the allocation of such an enabled task to a user, *start* refers to the actual start of a task, and *complete* refers to the completion of a task. Event types such as *withdraw*, *reassign*, *suspend*, *resume*, *pi_abort*, and *ate_abort* may refer to exceptions which are interesting from the viewpoint of SNA.

In this subsection, we take into account metrics based on special event types. In particular, we concentrate on the *reassign* event type. To define metrics based on special event types, we suppose that log lines have an event type. For convenience, we define an operation on events: $\pi_{et}(e) = \textit{event type}$ for some event $e = (a, p)$. Note that Definition 4.1 could be extended to capture event types such as used by commercial systems. In the next section we define an XML format to capture this information.

Before defining metrics, the basic notations used for a single case $c = (c_0, c_1, \ldots)$ are specified as follows.

Definition 4.11. (*follow*,$\triangledown$) Let L be a log. For $p_1, p_2 \in P$, $c = (c_0, c_1, \ldots) \in L$, and some event type *event type*:

- $\textit{follow}(c, i, j) = \pi_a(c_i) = \pi_a(c_j) \wedge \forall_{i<k<j} \pi_a(c_k) \neq \pi_a(c_i)$, for $0 \leq i < j < |c|$
- $p_1 \triangledown_c^{\textit{event type}} p_2 = \begin{array}{l} \exists_{o \leq i < j < |c|} \textit{follow}(c, i, j) \ \wedge \ \pi_p(c_i) = p_1 \ \wedge \\ \pi_{et}(c_i) = \textit{event type} \ \ \wedge \pi_p(c_j) = p_2 \end{array}$
- $|p_1 \triangledown_c^{\textit{event type}} p_2| = \sum_{0 \leq i < |c|} \begin{cases} 1 \text{ if } \exists_{i<j<|c|} \textit{follow}(c, i, j) \ \wedge \ \pi_p(c_i) = p_1 \\ \quad \wedge \ \pi_{et}(c_i) = \textit{event type} \ \ \wedge \pi_p(c_j) = p_2 \\ 0 \text{ otherwise} \end{cases}$

In a log, there may be several events that correspond to the same activity. If the activity a is reassigned from a performer p_1 to a performer p_2, we can find two events c_i and c_j such that $c_i = (a, p_1)$, $\pi_{et}(c_i) = 'reassign'$, $c_j = (a, p_2)$, and $\pi_{et}(c_j)$ is some event type. Thus, we need *follow* to find next event which is related to c_i. $p_1 \triangledown_c^{\textit{event type}} p_2$ denotes the function which returns true if within the context of the case c performers p_1 and p_2 both executed the same activity and p_1 was responsible for a specific type of event and p_2 is the first performer of some event for the same activity. $|p_1 \triangledown_c^{\textit{event type}} p_2|$ denotes the function which returns the number of times $p_1 \triangledown_c^{\textit{event type}} p_2$ in the case c. Using such relations, we define reassignment metrics. Recall that *reassign* is a special event type corresponding to the delegation from one performer to another.

Definition 4.12. (Reassignment metrics) Let L be a log. For $p_1, p_2 \in P$:

- $p_1 \triangledown_L^{'reassign'} p_2 = (\sum_{c \in L} |p_1 \triangledown_c^{`reassign'} p_2|) / (\sum_{c \in L} (|c| - 1))$
- $p_1 \dot{\triangledown}_L^{'reassign'} p_2 = (\sum_{c \in L \ \wedge \ p_1 \triangledown_c^{`reassign'} p_2} 1) / |L|$

$p_1 \triangledown_L^{'reassign'} p_2$ is obtained by dividing the total number of reassignments from p_1 to p_2 in the event log by the maximum number of reassignments in the log. For example, if there are 10 events in a log and *John* has reassigned an activity to *Mike* once, $John \triangledown_L^{'reassign'} Mike$ equals 1/9. $p_1 \dot{\triangledown}_L^{'reassign'} p_2$ ignores multiple reassignment within one instance.

In this section we formalized the metrics introduced in Section 3.2. It is important to note that each of the metrics is derived from some log L and the result can be represented in terms of a weighted graph (P, R, W), where P is the set of performers, R is the set of relations, and W is a function indicating the weight

of each relation (see Section 3.1). For example, the basic handover of work metric $\rhd_L$ defines $R = \{(p_1, p_2) \in P \times P \mid p_1 \rhd_L p_2 \neq 0\}$ and $W(p_1, p_2) = p_1 \rhd_L p_2$. For the Hamming distance $R = \{(p_1, p_2) \in P \times P \mid p_1 \triangle_L^{HD} p_2 \neq 1\}$ and $W(p_1, p_2) = 1 - (p_1 \triangle_L^{HD} p_2)$. For the Pearson's correlation coefficient $R = \{(p_1, p_2) \in P \times P \mid p_1 \triangle_L^{PC} p_2 \geq \alpha\}$ (where α is some threshold value between -1 and 1) and $W(p_1, p_2) = (1 + (p_1 \triangle_L^{PC} p_2))/2$. In other words, given an event log L each metric results in a sociogram that can be analyzed using existing SNA tools.

5 MiSoN

This section introduces our tool *MiSoN (Mining Social Networks)*. MiSoN has been developed to discover relationships between individuals from a range of enterprise information systems including workflow management systems such as Staffware, InConcert, and MQSeries, ERP systems, and CRM systems. Based on the event logs extracted from these systems MiSoN constructs sociograms that can be used as a starting point for SNA. The derived relationships can be exported in a matrix format and used by most SNA tools. With such tools, we can apply several techniques to analyze social networks, e.g., find interaction patterns, evaluate the role of an individual in an organization, etc.

MiSoN has been developed using Java including XML-based libraries such as JAXB and JDOM, and provides an easy-to-use graphical user interface. Figure 3 shows the architecture of MiSoN. The mining starts from a tool-independent XML format which includes information about processes, cases, activities, event times, and performers. From enterprise information systems recording event logs, we can export to this XML format.

Figure 4 shows the XML schema describing this format. It is an extension of the DTD suggested in [4]. The schema has the *WorkflowLog* element as a root element. It has *Data*, *Source*, and *Process* elements. The *Source* element contains the information about software or system that was used to record the log (e.g. Staffware). The *Process* element represents the process where the process log belongs. Note that there may be multiple *Process* elements in a log. Each *Process* element may hold multiple *ProcessInstance* elements that correspond to cases. The *AuditTrailEntry* element represents a log line, i.e., a single event. It contains *WorkflowModelElement*, *EventType*, *Timestamp*, and *Originator* elements. For SNA, the *WorkflowModelElement*, *EventType*, and *Originator* elements are most important. The *WorkflowModelElement* refers to the activity (or subprocess) the event corresponds to. The *EventType* specifies the type of the event, e.g., *schedule* (i.e., a task becomes enabled for a specific instance), *assign* (i.e., a task instance is assigned to a user), *start* (the beginning of a task instance), *complete* (the completion of a task instance), and *reassign* (as discussed in Section 4.4). In total, we identify 12 events. Last but not least the *Originator* element refers to the performer. To make the format more expressive, we define the *Data* element and other elements have it as a sub tags. If users want to specify more information than the basic elements, they can record the additional information using the

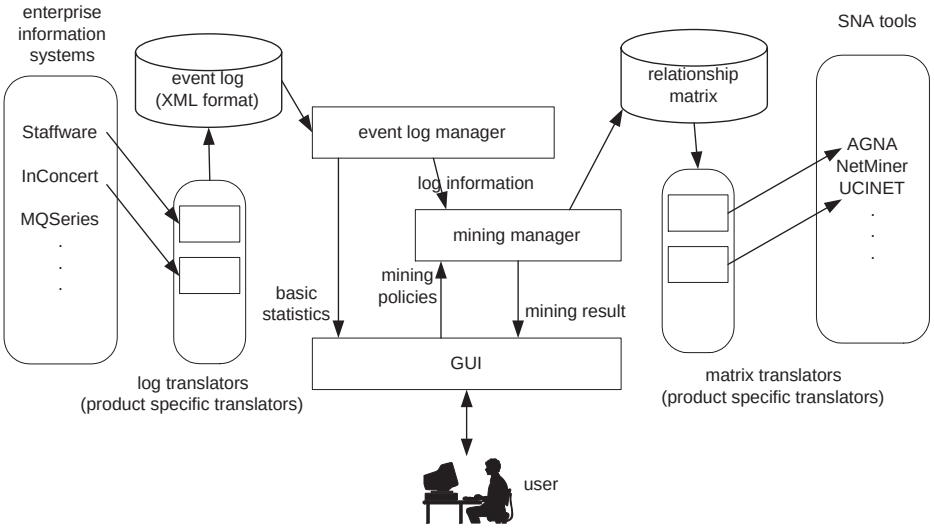


Fig. 3. The architecture of MiSoN.

Data element. Such information can be used for other types of process mining such as performance analysis, process knowledge extraction, etc. The complete XML schema is described in the Appendix.

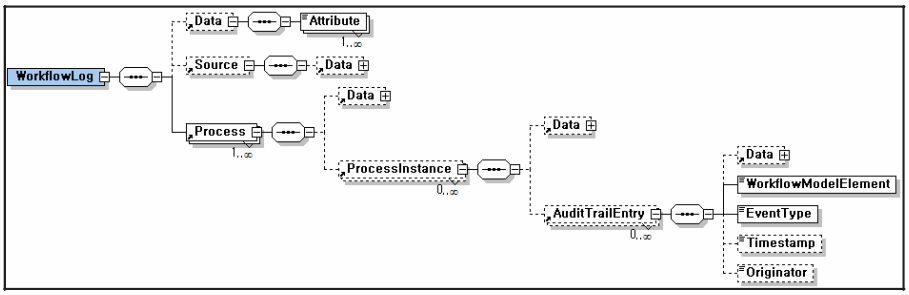


Fig. 4. MiSoN Workflow Mining Format (XML Schema).

After reading an event log that conforms to the XML schema, MiSoN provides functionalities for displaying user statistics and event log statistics. Using the metrics defined in Section 4, MiSoN constructs relationships between individuals. When calculating the relationships, the user can select suitable metrics and set relevant options. The result can be displayed using a matrix representation and a graph representation, but it can also be exported to SNA tools. Exported data

contains the number of performers, names of performers, and a relationship matrix.

To illustrate the MiSoN we have used an event log as generated with Staffware, which was converted to the XML format. For this log, we only consider the "released by" event type to make sociograms. This event corresponds to the *complete* event type in our XML format. We have tested MiSoN with several metrics mentioned in previous section. Figure 5 shows a screenshot of MiSoN when displaying the mining result of handover of work metrics.

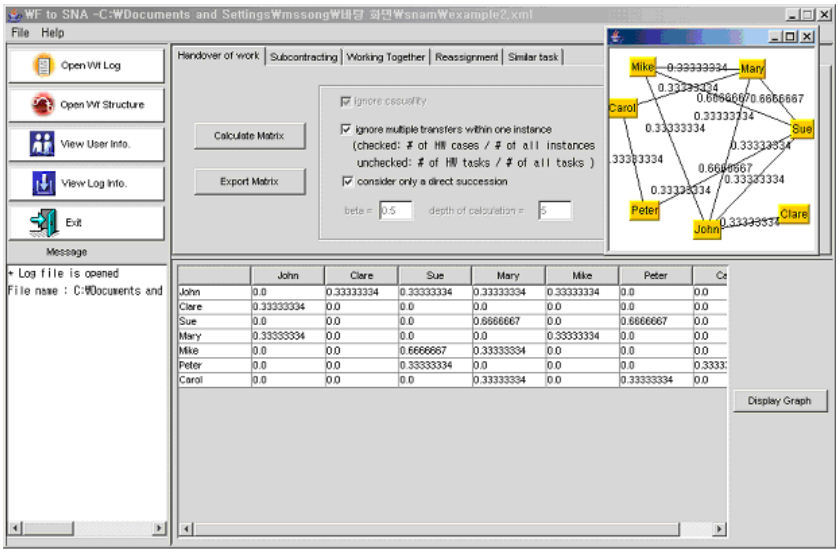


Fig. 5. MiSoN screenshot showing a sociogram based on a Staffware log.

MiSoN can export the mining result using the AGNA-translator (but also other tools like UCINET and NetMiner). AGNA (cf. www.geocities.com/imbenta/-agna/) is an SNA tool that allows for a wide variety of sociometric analysis techniques. For example, AGNA supports various notions of centrality including the Bavelas-Leavitt index described in Section 3.1. John and Sue have the highest Bavelas-Leavitt index (the value is 4.2), while Clare has the smallest value (2.8). Figure 6 shows the analysis using the tool AGNA. It also shows the network structure of result.

MiSoN can also export the mining result to other SNA tools like UCINET (cf. www.analytictech.com) and NetMiner (cf. www.netminer.com). In fact, in the case study described in the next section we will mainly use NetMiner to analyze the social network.

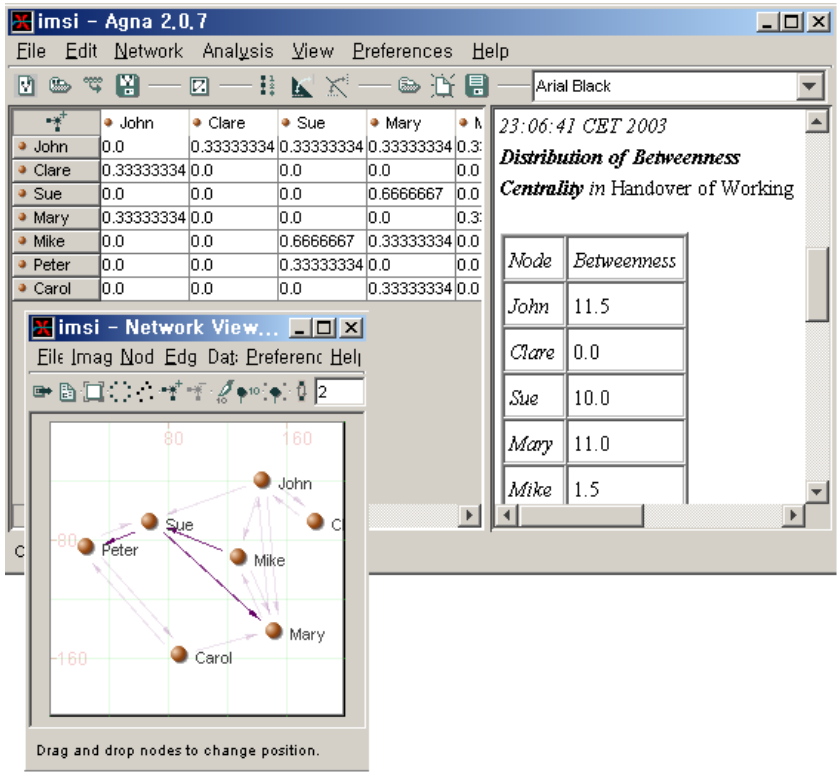


Fig. 6. Screenshot of AGNA when analyzing the input from MiSoN.

6 Case study

6.1 Context

To demonstrate how our metrics can be applied to real workflow logs and what kinds of analysis can be performed, we employed real workflow log data and carried out a case study. The case study we describe here involved one of the twelve provincial offices of the Dutch national public works department, employing about 1,000 civil servants. For reasons of confidentiality, we cannot disclose the name of this specific office.

The office's primary responsibility is the construction and maintenance of the road and water infrastructure within its provincial borders. For this purpose, it subcontracts various parties such as road construction companies, cleaning companies, and environmental agencies. Also, the provincial office purchases services and products to support its construction and maintenance activities on the one hand (e.g. mechanical tools, fuel, and rasters) and its administrative activities on the other (e.g. office supplies).

The process we dealt with concerns the handling of invoices, as received by the provincial office in question. In general, the handling of an invoice involves several validation steps and, if the invoice is approved, it is completed by payment. On a yearly basis, the provincial office processes some 20,000 invoices from its various subcontractors and suppliers.

The provincial office has implemented its own workflow management system to support the processing of invoices. This system records transaction information between activities. We extracted a process log and analyzed it. Since the extracted data are also stored in a relational database, we first developed a translator which converts the process log in the database to an XML file using the format described in the previous section.

The process consists of 17 real activities, aside from logistic steps and splits. The log data contains 4,988 cases. The number of total log lines (i.e. events) is 33,603 and 43 employees participated in the process execution. The log holds no information about reassignments. Hence, we cannot apply the reassignment metrics presented in Section 4.4. However, all other metrics we discussed in Section 4 have been applied in this case study.

6.2 Metrics application

We applied our metrics to the log data and derived several social networks. Moreover, by applying several SNA techniques, we tried to find the characteristics of the social network.

Figure 7 shows a social network which was derived by applying the handover of work metrics. The network represents how cases are transferred among performers. As indicated in Section 4, there are three refinements possible for the handover of work metrics. To generate this network, we take into account direct succession and multiple transfers in a case, but we ignore the real process structure, i.e., we use the metric $\rhd_L$ introduced in Definition 4.3. The network has

43 nodes and 406 links. The density of network is 0.225 and it has no isolated nodes.

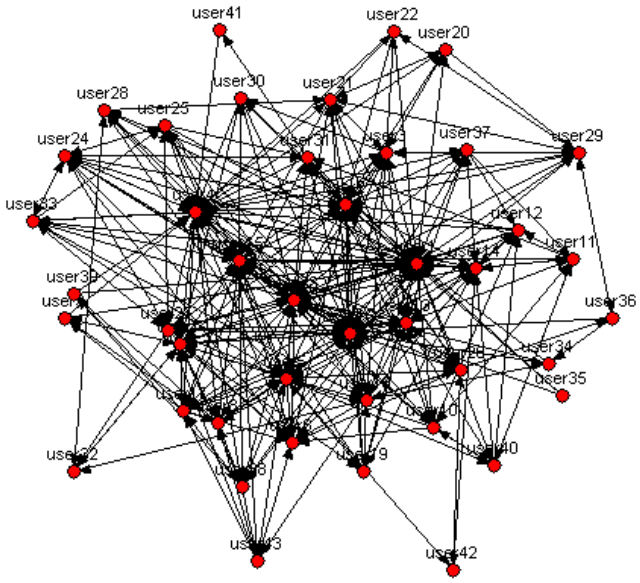


Fig. 7. Social network based on the handover of work metric $\rhd_L$.

In order to find people who are located in the center of the network, we calculate several centrality values such as betweenness, in and out closeness, and power [11] of each node. Normally, the nodes which are the most central have a powerful position in the network. Table 4 shows the top 10 ranked performers among the people involved based on (1) *betweenness* (i.e., the extent to which a node lies between all other pair of nodes on their geodesic paths), (2) *in-closeness* (i.e., the inverse of the sum of distances from all the other nodes to a given node, which is then normalized by multiplying it by the number of nodes minus 1), (3) *out-closeness* (i.e., the normalized inverse of the sum of distances from a node to all the other nodes), and (4) *power* (i.e., Bonacich's metric based on the principle that nodes connected to powerful nodes are also powerful [11]). In this table, we find that *user1* and *user4* have larger values than others in most measurements.[2]

[2] Note that the real user names are changed into anonymous identifiers like *user1*. Although during our analysis and interaction with the organization real user names were used, we abstract from the real user names in this paper to ensure privacy and confidentiality.

ranking	name	betweenness	name	in-closeness	name	out-closeness	name	power
1	user1	0.152	user1	0.792	user23	0.678	user4	4.102
2	user4	0.141	user4	0.792	user1	0.667	user1	2.424
3	user23	0.085	user16	0.75	user4	0.656	user30	1.964
4	user5	0.079	user23	0.689	user5	0.635	user17	1.957
5	user16	0.065	user2	0.667	user13	0.625	user7	1.774
6	user13	0.057	user15	0.618	user18	0.616	user8	1.394
7	user18	0.052	user5	0.609	user2	0.606	user2	1.347
8	user2	0.049	user7	0.592	user16	0.58	user23	1.098
9	user7	0.04	user13	0.568	user7	0.572	user16	1.058
10	user31	0.029	user18	0.568	user17	0.556	user18	0.581

Table 4. Performers having high values for (1) betweenness, (2) in-closeness, (3) out-closeness, and (4) power when analyzing the social network shown in Figure 7.

When generating a social network related to the handover of metrics, we can also consider indirect succession using a "causality fall factor" β. By applying various value of β, we generate several social networks. Despite of value of β, the derived networks have the same structure except the weight of arcs. Table 5 shows the sum, average, standard deviation, minimum value, and maximum value of the arc weights based on different values of β. If we use a small β, the value of arcs between performers who have the relationship of direct succession is larger than between others. However, if we use a large value of β, these differences decrease.

beta	sum	average	standard deviation	Min. value	Max. value
0.1	1.000025	0.000541	0.003269	0	0.086734
0.3	1.000091	0.000541	0.002895	0	0.074274
0.5	1.000001	0.000541	0.002631	0	0.065751
0.7	1.000011	0.000541	0.002522	0	0.063232
0.9	0.999979	0.000541	0.002586	0	0.067214

Table 5. Summary of arc weights for various values of β.

To find subcontracting relationships between people, we apply in-between metrics. Figure 8 shows the resulting social network. The network has 43 nodes and 146 links. The density of network is 0.081 and 8 nodes are isolated from the network. In this network, the direction of arcs is important. The start node of an arc represents a contractor, while the end node of an arc represents a subcontractor. Table 6 shows the ten people of highest in-degree and out-degree of centrality (based on the in-closeness and out-closeness calculated by Netminer).

Figure 9 shows the social network derived by applying the working together metrics and the ego network [33] corresponding to *user41*. In the ego network,

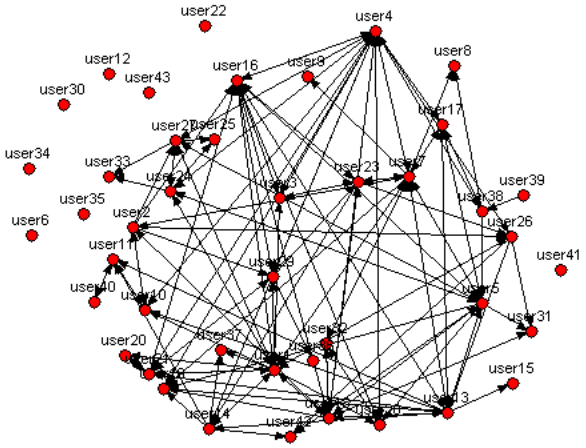


Fig. 8. Social network based on subcontracting metric.

ranking	name	in-closeness	name	out-closeness
1	user4	0.262	user4	0.262
2	user1	0.214	user1	0.214
3	user16	0.214	user7	0.167
4	user18	0.19	user13	0.143
5	user5	0.167	user5	0.167
6	user7	0.167	user16	0.214
7	user13	0.143	user18	0.19
8	user19	0.143	user14	0.095
9	user10	0.119	user23	0.119
10	user17	0.119	user27	0.119

Table 6. A list of people having a high degree of in-/out-closeness based on the subcontracting network shown in Figure 8.

the nodes represent the people working together with *user4* according to this metric. Note that *user41* works together with *user1*, *user4*, *user23*, *user26*, and *user31*. The average size of ego network of the generated network is 24.698 and the standard deviation of this value is 9.709. This means that the social network suggests that an employee on average works with 24 people.

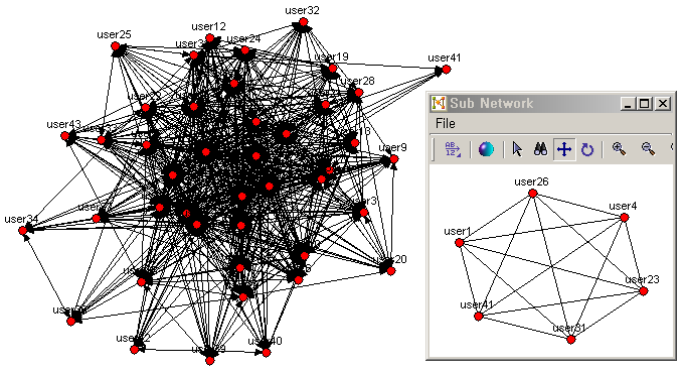


Fig. 9. Social network based on the working together metric (left) and the ego network of *user41* (right).

Applying the metrics based on joint activities, we calculate the distance between people. Figure 10 shows the social network which is derived by applying Pearson's correlation coefficient. From the performer by task matrix, we first apply $log_{10}(x+1)$, then calculate the distances between people. We get 5 clusters and two isolated nodes. The nodes in the same cluster play the same role. In this case, the bridge node can be interpreted as a person who has multiple roles. In the network, *user8*, *user28*, *user37*, and *user43* have multiple roles.

Finally, we explore how cases are transferred among groups. To calculate case transfers among groups, we combine the handover of work metrics with a role model. In this case study, we use the results of correspondence analysis [13] as a role model of performers. (Of course, we can also use the results of the metrics based on joint activities.) Correspondence analysis is frequently used in biological science to analyze ecological systems based on species scores for specific locations [24]. In this paper, we apply correspondence analysis to find relationships between activities and performers. We first make a performer by activity matrix from the workflow logs. Then, by applying correspondence analysis to the matrix, we derive the relationship between activities, between performers, and between activities and performers. Figure 11 shows the graphical result of applying correspondence analysis. In the figure, boxes represent activities and circles represents performers. Closely positioned nodes indicate a strong correspondence from a work handover perspective between the respective users and/or tasks. (Although the distance between user nodes and task nodes

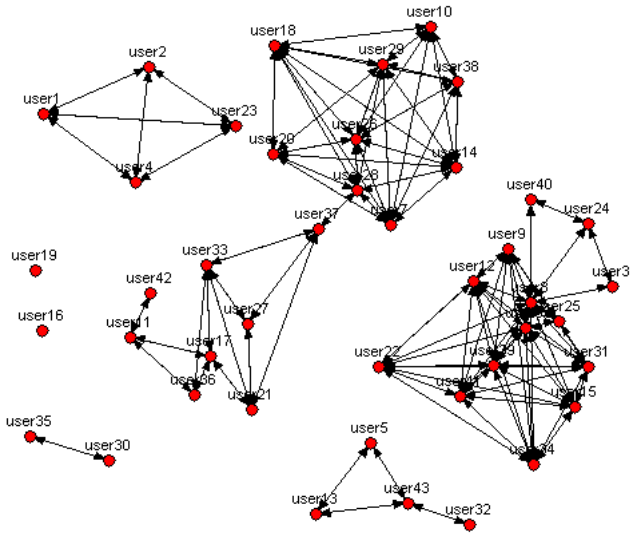


Fig. 10. Social network derived from Pearson's correlation coefficient (threshold value 0.75).

should not be interpreted as an absolute measure.) From this figure, performers and activities are classified into five groups. Table 7 shows the results. In the remainder we will use these five groups as a role model.

Figure 12(a) shows the social network of handover of work metrics considering the role model given in Table 7. By putting the nodes in the same group closely, we have reconstructed the original network. And by summing up the weight of arcs between groups we derive the aggregated network shown in Figure 12(b).

Table 8 shows the information flow of the network according the role model. It is also derived by summing up the weight of arcs between groups. For example, the value form *group1* to *group2* is calculated by adding up the weights of the arcs from nodes in *group1* to nodes in *group2*. Based on Table 8 we can make some observations. First, the highest value (1.330) is in the cell from *group1* to *group1*. It means that the handover of work within *group1* happened most frequently. Second the values from *group1* to *group5* (0.895), from *group4* to *group1* (0.620), and from *group5* to *group4* (0.529) have the high values. It represents that more handover of work happened between these groups.

The goal of this subsection is not to provide a comprehensive overview of all the diagrams we developed or to provide very specific information about the studied process or organization in question. In the next section, we will reflect on the relevance of the various analyses for the organization in question.

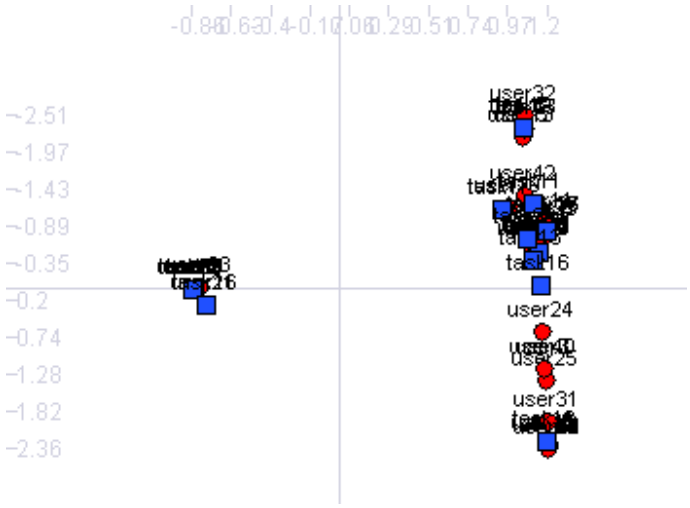


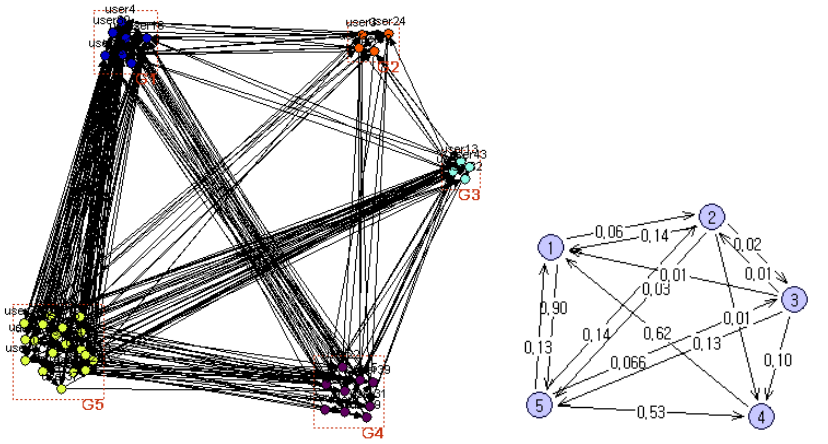
Fig. 11. Graphical result of correspondence analysis.

group	performers	activities
group1	user1, user2, user4, user16, user23, user30, user35	task2, task3, task15, task21, task22
group2	user3, user24, user25, user40	
group3	user5, user13, user32, user43	task8, task19
group4	user6, user8, user9,user12, user15, user22, user31, user39, user41	task18
group5	user7, user10, user11, user14, user17, user18, user19, user20, user21, user26, user27, user28, user29, user33, user34, user36, user37, user38, user42	task5, task7, task11, task13 task16, task17, task20

Table 7. The result of correspondence analysis: users are clustered into five groups.

from \ to	group1	group2	group3	group4	group5	sum
group1	1.330	0.058	0.002	0.002	0.895	2.287
group2	0.143	0.014	0.020	0.005	0.028	0.211
group3	0.014	0.005	0.002	0.104	0.030	0.154
group4	0.620	0.000	0.002	0.005	0.004	0.630
group5	0.132	0.135	0.134	0.526	0.617	1.545
sum	2.239	0.212	0.160	0.642	1.574	4.827

Table 8. Information flow between groups.



(a) Social network of handover of work metrics considering roles

(b) Information flow among roles

Fig. 12. Social network based on handover of work metric using the five groups shown in Table 7.

6.3 Organizational relevance

Prior to our analysis, the involved management did not express any specific needs or questions about the invoice handling process. And yet, they indicated that the handling of invoices is in the center of their attention. There are two main reasons for this. First of all, it is the single most distributed processes under the responsibility of the public works department. For example, if invoices are related to some particular public works project, its project leader must personally certify that delivery has taken place before payment may happen. Project leaders, however, may reside at any location within the provincial borders (in contrast to the performers working at the administrative head office). The distributed nature of the process increases the probability of hand-over errors and work getting lost.

The second reason for the attention for the invoice process is the recent Dutch law about penalty interests. Parties that send invoices to public organizations and receive their rightful payment after more than 30 days are entitled to a compensation proportional to the invoice amount. Due to the current interest rate, this compensation exceeds commercial rates. Sluggish settlement of invoices directly affects the public works department's financial position.

Both issues have contributed to the decision to introduce workflow technology to support the invoice handling process, as this is expected to increase quality and decrease process lead time. Management expressed a general interest in results from SNA to learn about process execution, behavior of the involved parties, and potential opportunities for improvement.

After we applied our metrics to the log data and derived the social networks as shown, we presented the managers of the three departments mostly involved our analysis results in a joint session. Roughly speaking, the three departments are respectively responsible for the administrative, contractual, and financial aspects of the invoice handling. The goal of this meeting from a research perspective was threefold:

1. To validate our understanding of the process.
2. To generate feedback on our analyses.
3. To identify further analysis opportunities.

To determine whether we properly understood the process, we discussed the process model of the invoice handling process and the involved parties for each of the various steps. This led to no surprising new insights. We will reflect on the other two aims of the meeting in more detail.

Feedback on analyses. After we explained the various SNA notions we presented the results from our analyses as presented for a large part in Section 6.2. We started with discussing the top 5 and bottom 5 of the lists of performers as ordered on their scores on betweenness, in and out closeness, and power in respectively the social networks of handover, subcontracting, and working together metrics. Note that we used the lists that included the real names of the actors to facilitate meaningful feedback.

From the responses, we learned that, typically, performers with high scores (e.g. *user1* and *user4* in Table 4) work for the administrative department in supportive functions. This confirms a general insight that highly connected people often are assistants. Because the administrative department is responsible for both the preparation and completion of the handling of each invoice, its staff is involved in the handling of each case, giving them strong ties with other performers. The managers indicated, however, that not all of the people in these positions were present in the top of the lists, indicating that having a supportive function is not sufficient in itself to become highly connected.

Performers with low scores could be categorized as follows. First of all, project leaders were highly represented in the bottom of the lists (e.g. *user9*). As stated before, they play an isolated role in the handling of invoices, being solely responsible for certifying that goods have been delivered (and only if an invoice is related to any project at all). Other performers with limited formal verification responsibilities were identified as well (e.g. *user22*). The second category of relatively unconnected performers could be traced back to auxiliary logins (e.g. *user30*), used by system administrators and management to deal with exceptional circumstances. An example of an exceptional situation is an invoice that is being withdrawn while its processing has already started. The isolated "participation" of this category of users is therefore not very surprising. It did, however, make the managers conscious of the visibility of this type of irregular interference. One manager remarked: "So, auditors can derive this type of information too." The third category turned out to be more surprising, as it involved

senior positions in the contractual and financial departments (e.g. *user41*). At least nominally, they are expected to be actively involved in the process. Their low position could indicate that a large amount of work being executed with workflow technology is delegated to their juniors. Also, one of these performers would retire in a couple of weeks.

After the discussion of the lists of performers, we presented the social network indicating the distance between people (see Figure 10). The relations between users were readily recognized by the involved managers. For example, the subgraph of *user1*, *user2*, *user4* and *user23* concerned the group of highly-connected assistants at the administrative office we encountered earlier. Then, we took a closer look at the two isolated nodes. One of them - rather characteristically - turned out to be the system administrator (*user19*). The isolated position of the other node, *user16*, led to some excitement. At first, the isolation of this performer was not understood, as she was considered to perform an explicit role in the contractual handling of invoices. Then it occurred to one of the managers that the involved person was included in another cluster under a *different user name* as well (*user32*). The existence of such a situation was a complete surprise to the managers and considered highly undesirable for compliancy reasons.

The final result we obtained feedback on, involved the correspondence analysis, such as presented in Table 7 and Figure 11. Managers readily recognized *group1*, *group3*, and *group5*. At the same time, they indicated that they did not differentiate themselves between most of the performers in *group2* and *group4*. This is in line with the observation that some performers from these groups are closely positioned to each other in Figure 10. For example, the positions of *user25* from *group2* and *user31* from *group4* nearly coincide. And yet, the strong correspondence between *group4* and *task18* indicates that a degree of performer specialization has taken place with respect to this specific invoice check that had gone unnoticed with the concerned managers.

Further analyses. Aside from the various surprising aspects of the invoice handling process, the managers of the provincial office were most intrigued by the subcontracting analysis. After some discussion, they expressed their suspicion about three parts of the process where a similar yet undesirable "back-and-forth" behavior may take place. Specifically, they meant that a performer (the contractor) routes a work package to another performer (the subcontractor), who subsequently routes it back to the contractor or one of the contractor's close colleagues, because the subcontractor feels the invoice is received in error. This, for example, takes place when an invoice related to a project is sent for verification to the wrong project leader. Each occurrence of this pattern is highly undesirable, as it slows down the processing of the invoice without making any progress. From an organizational perspective, it is just as unwelcome when the work package is routed back to the original contractor as to a colleague with a similar organizational role.

Our initial analysis did not cover this more general kind of subcontracting pattern, because it focused on the identity of the original contractor only (see

Definition 4.6) . To investigate the expressed suspicions we analyzed the mining log in various ways, using other than SNA techniques as well. Therefore, in the context of this paper, we will be brief about this additional analysis. It turned out that in the handling of over 17 % of all invoices, at least once an undesired subcontracting takes place at either of the three identified places in the process. The exact distribution is shown in Figure 13. As can be seen, there are cases where 10 or more erroneous routings take place.

As a result of the additional analysis we carried out and discussed again, management of the provincial office re-enforced the existing procedure that staff, when in doubt, should contact the intended next performer by phone first. Especially in cases where hand-overs take place between performers at the head and regional offices, management felt that people acted too shy with respect to this procedure.

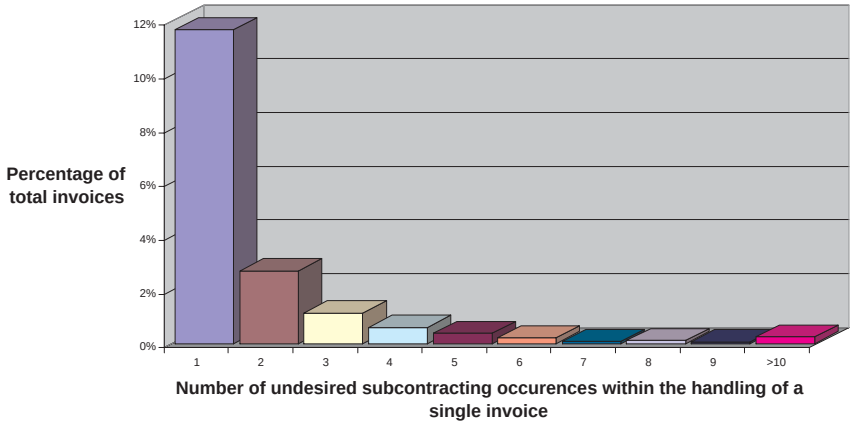


Fig. 13. The distribution of undesired generalized subcontracting within the handling of invoices.

6.4 Discussion

In this section, we demonstrated how our metrics can be applied to a real workflow log of a Dutch organization. Based on the metrics defined in Section 4, we derived various sociograms, some of which have been shown in this paper. Using the sociograms we applied SNA techniques such as betweenness, closeness, power, and ego network, etc. We also showed the possibility of applying other analysis techniques such as correspondence analysis to compare users based on their "profile".

Next we discussed the organizational relevance of our analyses. As we indicated, many of our findings corresponded with existing insights of the involved

management, supporting the correctness of our analyses. At various points, our analyses came as a surprise. These particularly concerned senior performers who did not seem very connected, the clear visibility of the actions of irregular performers, and a degree of unnoticed performer specialization that had taken place. In addition, we found it interesting to observe how our analysis results triggered the management to identify and define additional questions. This, in our eyes, strongly supports the relevance and viability of process mining in an organizational context, even though our additional analyses extended beyond SNA.

As discussed in Section 2.4, ethical and legal issues play an important role in the practical application of process mining in general and SNA analysis in particular. One concern we certainly felt is that the validation and discussion of our analysis results required us to disclose the identity of the involved performers. Note that a discussion of anonymized sociograms with the involved management would not have been less meaningful. We informed the management that it is illegal to perform actions towards individuals based on the presented results. Because of the clear value of this type of analysis, the managers expressed their intent to ask for the consent of their employees for the use of future analyses. Note that the re-enforced policy that resulted from our additional analysis was neither based on information obtained on individual performers, nor did it affect any individual more than others.

7 Related work

Related work can be divided in two categories: process mining and SNA.

7.1 Related work on process mining

The idea of process mining is not new [4, 7, 15] but has been mainly aiming at the control-flow perspective. The idea of applying process mining in the context of workflow management was first introduced in [7]. This work is based on workflow graphs, which are inspired by workflow products such as IBM MQSeries Workflow (formerly known as Flowmark). Cook and Wolf have investigated similar issues in the context of software engineering processes. In [15] they describe three methods for process discovery: one using neural networks, one using a purely algorithmic approach, and one Markovian approach. Schimm [42] has developed a mining tool suitable for discovering hierarchically structured workflow processes. Herbst and Karagiannis also address the issue of process mining in the context of workflow management using an inductive approach [27, 26]. They use stochastic task graphs as an intermediate representation and generate a workflow model described in the ADONIS modeling language. Most of the approaches have problems dealing with parallelism and noise. Our work in [1, 6] is characterized by the focus on workflow processes with concurrent behavior (rather than adding ad-hoc mechanisms to capture parallelism). In [47] a heuristic approach using rather simple metrics is used to construct so-called "dependency/frequency tables" and

"dependency/frequency graphs". These are then used to tackle the problem of noise. The approaches described in [1, 6, 47] are based the α algorithm.

Process mining in a broader sense can be seen as a tool in the context of Business (Process) Intelligence (BPI). In [25, 41] a BPI toolset on top of HP's Process Manager is described. The BPI tools set includes a so-called "BPI Process Mining Engine". However, this engine does not provide any techniques as discussed before. Instead it uses generic mining tools such as SAS Enterprise Miner for the generation of decision trees relating attributes of cases to information about execution paths (e.g., duration). In order to do workflow mining it is convenient to have a so-called "process data warehouse" to store audit trails. Such a data warehouse simplifies and speeds up the queries needed to derive causal relations. In [35] Zur Muehlen describes the PISA tool which can be used to extract performance metrics from workflow logs. Similar diagnostics are provided by the ARIS Process Performance Manager (PPM) [29]. The later tool is commercially available and a customized version of PPM is the Staffware Process Monitor (SPM) [45] which is tailored towards mining Staffware logs. Note that none of the latter tools is extracting models, i.e., the results do not include control-flow, organizational or social network related diagnostics. The focus is exclusively on performance metrics.

For more information on process mining we refer to a special issue of Computers in Industry on process mining [5] and the survey paper [4]. Note that although quite some work has been done on process mining from event logs none of the approaches known to the authors have incorporated the social dimension as discussed in this paper.

7.2 Related work on SNA

Since the early work of Moreno [34], sociometry, and SNA in particular, have been active research domains. There is a vast amount of textbooks, research papers, and tools available in this domain [8, 10, 12, 18, 22, 23, 34, 37, 43, 46]. There have been many studies analyzing organizational activity based on insights from social network analysis. However, some of these studies typically have an ad-hoc character and sociograms are typically constructed based on questionnaires rather than using a structured and automated approach as described in this paper. More structured approaches are often based on the analysis of e-mail interaction and additional electronic sources. Several studies have generated sociograms from email logs in organization [16, 17, 19, 36, 38] to analyze the communication structure. Such studies have resulted in the identification of relevant, recurrent aspects of interaction in organizational contexts [9, 21]. However, these studies are unable to relate the derived social networks to a particular workflow process, as the analyzed data does not reveal to what activity or case it applies.

Most tools in the SNA domain take sociograms as input. MiSoN is one of the few tools that generate sociograms as output. The only comparable tools are tools to analyze e-mail traffic, cf. BuddyGraph (www.buddygraph.com) and MetaSight (www.metasight.co.uk/). However, these tools monitor unstructured

messages and cannot distinguish between different activities (e.g., work-related interaction versus social interaction).

As indicated in the introduction, this paper extends the results presented in [3]. Unlike [3], this paper provides concrete metrics, a more elaborate description of MiSoN, and a case study illustrating the applicability of the approach.

8 Conclusions

This paper presents an approach, concrete metrics, and a tool to extract information from event logs and construct a sociogram which can be used to analyze interpersonal relationships in an organization. Today many information systems are "process aware" and log events in some structured way. As indicated in the introduction, workflow management systems register the start and completion of activities, ERP systems log all transactions (e.g., users filling out forms), call center and CRM systems log interactions with customers, etc. These examples have in common that there is some kind of event log. Unfortunately, the information in these logs is rarely used to derive information about the process, the organization, and the social network. In this paper we focus on the latter aspect and present an approach to discover sociograms. These sociograms are based on the observed behavior and may use events like the transfer of work or delegation from one individual to another. MiSoN can interface with commercial systems such as Staffware and standard SNA tools like AGNA, UCINET and NetMiner, thus allowing for the application of the ideas presented in this paper.

This paper also presents a case study conducted within a Dutch national public works department. The case study shows that the event logs in real organizations allow for social network analysis. Moreover, in this particular case the analysis results provide relevant, surprising organizational information. The established results and resulting discussions have formed the basis for additional process mining to deal with managerial concerns, resulting in the re-enforcement of organizational policies. In the future, we plan to repeat our analysis within the public works department and apply our approach in many other organizations as well. It would be interesting to compare the results we obtain on the basis of event logs to results of the analysis of other communication means usage e.g. e-mail. This would provide an even richer view on organizational interaction and process improvement opportunities.

We also investigate extensions of the approach using filtering techniques and more advanced forms clustering. For example, we now abstract from the results of activities. If activities or cases can be classified as successful or unsuccessful, important or unimportant, standard or special, etc., this information could be used when building sociograms.

Recently, MiSoN has been integrated in the ProM framework[3]. The ProM framework allows for various types of process mining, i.e., given a log it is possible to not only derive sociograms but also process models. The ProM framework also

[3] See www.processmining.org for more information.

provides an LTL checker that can check properties expressed in Linear Temporal Logic (LTL) [32]. This allows for all kinds of questions, e.g., checking the 4-eyes principle (two tasks need to be executed by different people to avoid fraud). This LTL checker can be used to ask more detailed questions based in insights generated from the SNA analysis. In the context of the ProM framework also a prototype of an e-mail analysis tool has been developed. Based on a user's Inbox located on some Exchange server, the prototype can translate the e-mails to the XML format described in this paper. However, since e-mails may refer to different processes and there are no explicit pointers to tasks and cases, and heuristics and/or conventions need to be used. Therefore, we only consider this as means to provide more context to the SNA analysis based on true event logs.

Acknowledgement
Minseok Song is visiting the Department of Technology Management at Eindhoven University of Technology with funding by the BK21 program. He would like to thank the Ministry of Education of Korea for its financial support through the BK21 program. The authors would also like to thank Ton Weijters, Boudewijn van Dongen, Ana Karla Alves de Medeiros, Andriy Anikolov, Laura Maruster, Eric Verbeek, Monique Jansen-Vullers, Michael Rosemann, and Peter van den Brand for their on-going work on process mining techniques and tools at Eindhoven University of Technology.

References

1. W.M.P. van der Aalst and B.F. van Dongen. Discovering Workflow Performance Models from Timed Logs. In Y. Han, S. Tai, and D. Wikarski, editors, *International Conference on Engineering and Deployment of Cooperative Information Systems (EDCIS 2002)*, volume 2480 of *Lecture Notes in Computer Science*, pages 45–63. Springer-Verlag, Berlin, 2002.
2. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT press, Cambridge, MA, 2002.
3. W.M.P. van der Aalst and M. Song. Mining Social Networks: Uncovering interaction patterns in business processes. In M. Weske, B. Pernici, and J. Desel, editors, *International Conference on Business Process Management (BPM 2004)*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2004.
4. W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. *Data and Knowledge Engineering*, 47(2):237–267, 2003.
5. W.M.P. van der Aalst and A.J.M.M. Weijters, editors. *Process Mining*, Special Issue of Computers in Industry, Volume 53, Number 3. Elsevier Science Publishers, Amsterdam, 2004.
6. W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.
7. R. Agrawal, D. Gunopulos, and F. Leymann. Mining Process Models from Workflow Logs. In *Sixth International Conference on Extending Database Technology*, pages 469–483, 1998.

8. A.A. Bavelas. A Mathematical Model for Group Structures. *Human Organization*, 7:16–30, 1948.
9. J. Begole, J. Tang, R. Smith, and N. Yankelovich. Work Rhythms: Analyzing Visualizations of Awareness Histories of Distributed Groups In *ACM conference on Computer supported cooperative work*, pages 334-343, 2002.
10. H.R. Bernard, P.D. Killworth, C. McCarty, G.A. Shelley, and S. Robinson. Comparing Four Different Methods for Measuring Personal Social Networks. *Social Networks*, 12:179–216, 1990.
11. P. Bonacich. Power and Centrality: A family of Measures. *American Journal of Sociology*, 92:1170–1182, 1987.
12. R.S. Burt and M. Minor. *Applied Network Analysis: A Methodological Introduction*. Sage, Newbury Park CA, 1983.
13. S. E. Clausen. *Applied Correspondence Analysis: An Introduction*. Sage Publications, 1998.
14. College Bescherming persoonsgegevens (CBP; Dutch Data Protection Authority). http://www.cbpweb.nl/index.htm.
15. J.E. Cook and A.L. Wolf. Discovering Models of Software Processes from Event-Based Data. *ACM Transactions on Software Engineering and Methodology*, 7(3):215–249, 1998.
16. S. Farnham, S.U. Kelly, W. Portnoy, and J.L.K. Schwartz. Wallop: Designing Social Software for Co-Located Social Networks. *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, Big Island, Hawaii, January 05 - 08, 2004.
17. S. Farnham, W. Portnoy, and A. Turski. Using Email Mailing Lists to Approximate and Explore Corporate Social Networks. In D.W. McDonald, S. Farnham, and D. Fisher, editors, *CSCW'04 Workshop on Social Networks*, Chicago, IL, November 06 - 10, 2004.
18. M. Feldman. Electronic Mail and Weak Ties in Organizations. *Office: Technology and People*, 3:83–101, 1987.
19. D. Fisher and P. Dourish. Social and Temporal Structures in Everyday Collaboration. In E. Dykstra-Erickson and M. Tscheligi, editors, *Proceedings of the 2004 Conference on Human Factors in Computing Systems (CHI2004)*, pages 551-558. Vienna, Austria, April 24 - 29, 2004.
20. L. Fischer, editor. *Workflow Handbook 2001, Workflow Management Coalition*. Future Strategies, Lighthouse Point, Florida, 2001.
21. D. Fischer and P. Dourish. Social and Temporal Structures in Everyday Collboration. In E. Dykstra-Erickson and M. Tscheligi, editors, *Conference on Human Factors in Computing Systems (CHI 200)*, pages 551–558, 2004.
22. L.C. Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40:35–41, 1977.
23. L.C. Freeman. Centrality in Social Networks: Conceptual Clarification. *Social Networks*, 1:215–239, 1979.
24. H. G. Gauch. *Data Analysis in Community and Landscape Ecology*. Cambridge University Press, 1982.
25. D. Grigori, F. Casati, U. Dayal, and M.C. Shan. Improving Business Process Quality through Exception Understanding, Prediction, and Prevention. In P. Apers, P. Atzeni, S. Ceri, S. Paraboschi, K. Ramamohanarao, and R. Snodgrass, editors, *Proceedings of 27th International Conference on Very Large Data Bases (VLDB'01)*, pages 159–168. Morgan Kaufmann, 2001.

26. J. Herbst. A Machine Learning Approach to Workflow Management. In *Proceedings 11th European Conference on Machine Learning*, volume 1810 of *Lecture Notes in Computer Science*, pages 183–194. Springer-Verlag, Berlin, 2000.
27. J. Herbst. *Ein induktiver Ansatz zur Akquisition und Adaption von Workflow-Modellen*. PhD thesis, Universität Ulm, November 2001.
28. B.J.P. Hulsman and P.C. Ippel. Personeelsinformatiesystemen: De Wet Persoonsregistraties toegepast. Registratiekamer, The Hague, 1994.
29. IDS Scheer. ARIS Process Performance Manager (ARIS PPM). http://www.ids-scheer.com, 2002.
30. S. Jablonski and C. Bussler. *Workflow Management: Modeling Concepts, Architecture, and Implementation*. International Thomson Computer Press, London, UK, 1996.
31. F. Leymann and D. Roller. *Production Workflow: Concepts and Techniques*. Prentice-Hall PTR, Upper Saddle River, New Jersey, USA, 1999.
32. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag, New York, 1991.
33. J.C. Mitchell. The Concept and Use of Social Networks. In J.C. Mitchell, editor, *Social Networks in Urban Situations*, pages 1–50. Manchester University Press, Manchester, 1969.
34. J.L. Moreno. *Who Shall Survive?* Nervous and Mental Disease Publishing Company, Washington, DC, 1934.
35. M. zur Mühlen and M. Rosemann. Workflow-based Process Monitoring and Controlling - Technical and Organizational Issues. In R. Sprague, editor, *Proceedings of the 33rd Hawaii International Conference on System Science (HICSS-33)*, pages 1–10. IEEE Computer Society Press, Los Alamitos, California, 2000.
36. B.A. Nardi, S. Whittaker, E. Isaacs, M. Creech, J. Johnson, and J. Hainsworth. *Integrating communication and information through ContactMap. Communications of the ACM*, 45(4):89–95, 2002.
37. H. Nemati and C.D. Barko. *Organizational Data Mining: Leveraging Enterprise Data Resources for Optimal Performance*. Idea Group Publishing, Hershey, PA, USA, 2003.
38. H. Ogata, Y. Yano, N. Furugori, and Q. Jin. Computer Supported Social Networking For Augmenting Cooperation. *Computer Supported Cooperative Work*, 10(2):189–209, 2001.
39. W. Reisig and G. Rozenberg, editors. *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1998.
40. L.B. Sauerwein and J.J. Linnemann. Guidelines for Personal Data Processors: Personal Data Protection Act. Ministry of Justice, The Hague, 2001.
41. M. Sayal, F. Casati, and M.C. Shan U. Dayal. Business Process Cockpit. In *Proceedings of 28th International Conference on Very Large Data Bases (VLDB'02)*, pages 880–883. Morgan Kaufmann, 2002.
42. G. Schimm. Generic Linear Business Process Modeling. In S.W. Liddle, H.C. Mayr, and B. Thalheim, editors, *Proceedings of the ER 2000 Workshop on Conceptual Approaches for E-Business and The World Wide Web and Conceptual Modeling*, volume 1921 of *Lecture Notes in Computer Science*, pages 31–39. Springer-Verlag, Berlin, 2000.
43. J. Scott. *Social Network Analysis*. Sage, Newbury Park CA, 1992.
44. M. Smith. *Invisible crowds in cyberspace: Measuring and mapping the social structure of Usenet*. In M. Smith and P. Kollock, editors, Communities in Cyberspace. Routledge Press, 1999.

45. Staffware. Staffware Process Monitor (SPM). http://www.staffware.com, 2002.
46. S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, 1994.
47. A.J.M.M. Weijters and W.M.P. van der Aalst. Rediscovering Workflow Models from Event-Based Data using Little Thumb. *Integrated Computer-Aided Engineering*, 10(2):151–162, 2003.

Appendix

This appendix provides the XML schema described in Figure 4.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
 <xs:element name="WorkflowLog">
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="Data" minOccurs="0"/>
    <xs:element ref="Source" minOccurs="0"/>
    <xs:element ref="Process" maxOccurs="unbounded"/>
   </xs:sequence>
  <xs:attribute name="description" type="xs:string"
  use="optional"/>
  </xs:complexType>
 </xs:element>
 <xs:element name="Source">
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="Data" minOccurs="0"/>
   </xs:sequence>
   <xs:attribute name="program" type="xs:string"
   use="required"/>
  </xs:complexType>
 </xs:element>
 <xs:element name="Process">
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="Data" minOccurs="0"/>
    <xs:element ref="ProcessInstance" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="id" type="xs:string" use="required"/>
   <xs:attribute name="description" type="xs:string" use="optional"/>
  </xs:complexType>
 </xs:element>
 <xs:element name="AuditTrailEntry">
  <xs:complexType>
```

```
  <xs:sequence>
   <xs:element ref="Data" minOccurs="0"/>
   <xs:element name="WorkflowModelElement" type="xs:string"/>
   <xs:element name="EventType">
    <xs:complexType>
     <xs:simpleContent>
      <xs:restriction base="xs:string">
       <xs:enumeration value="schedule"/>
       <xs:enumeration value="assign"/>
       <xs:enumeration value="withdraw"/>
       <xs:enumeration value="reassign"/>
       <xs:enumeration value="start"/>
       <xs:enumeration value="suspend"/>
       <xs:enumeration value="resume"/>
       <xs:enumeration value="pi_abort"/>
       <xs:enumeration value="ate_abort"/>
       <xs:enumeration value="complete"/>
       <xs:enumeration value="autoskip"/>
       <xs:enumeration value="manualskip"/>
       <xs:enumeration value="unknown"/>
       <xs:attribute name="unknowntype" type="xs:string" use="optional"/>
      </xs:restriction>
     </xs:simpleContent>
    </xs:complexType>
   </xs:element>
   <xs:element name="Timestamp" type="xs:dateTime" minOccurs="0"/>
   <xs:element name="Originator" type="xs:string" minOccurs="0"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="Data">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Attribute" maxOccurs="unbounded">
    <xs:complexType>
     <xs:simpleContent>
      <xs:extension base="xs:string">
       <xs:attribute name="name" type="xs:string" use="required"/>
      </xs:extension>
     </xs:simpleContent>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

```
 <xs:element name="ProcessInstance">
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="Data" minOccurs="0"/>
    <xs:element ref="AuditTrailEntry" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="id" type="xs:string" use="required"/>
   <xs:attribute name="description" type="xs:string" use="optional"/>
  </xs:complexType>
 </xs:element>
</xs:schema>
```