Mining Social Networks: Uncovering Interaction Patterns in Business Processes

Wil M.P. van der Aalst[1] and 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
[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. Increasingly information systems log historic information in a systematic way. Workflow management systems, but also ERP, CRM, SCM, and B2B systems often provide a so-called "event log", i.e., a log recording the execution of activities. Unfortunately, the information in these event logs is rarely used to analyze the underlying processes. Process mining aims at improving this by providing techniques and tools for discovering process, control, data, organizational, and social structures from event logs. This paper focuses on the mining social networks. This is possible because event logs typically record information about the users executing the activities recorded in the log. To do this we combine concepts from workflow management and social network analysis. This paper introduces the approach, defines metrics, and presents a tool to mine social networks from event logs.

1 Introduction

Sociometry, also referred to as sociography, refers to methods presenting data on interpersonal relationships in graph or matrix form [9, 22, 23]. 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 [17]. 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 [11]. For example, BuddyGraph (http://www.buddygraph.com/) and MetaSight (http://www.metasight.co.uk/) are tools that use logs on e-mail traffic as a starting point for sociometric analysis. Similarly, information on the Web can be used for such an analysis. For the analysis of social networks in organizations 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 important decisions (e.g., allocation of resources) 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 like Staffware register the start and completion of activities [2]. 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. [3, 6, 14, 21].

When people are involved, event logs will typically contain information on the person executing or initiating the *event*. We only consider events referring to an *activity* and a *case* [3]. 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 [9, 22, 23]. 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, and our tool *MiSoN* (Mining Social Networks).

The paper is organized as follows. Section 2 introduces the concept of process mining. Section 3 focuses on the mining of 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, a small case study, and 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 [3]. 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]. 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 [3, 6, 14, 21]. The term "structured process description" may be interpreted in various ways, ranging from a control-flow model expressed in terms of classical Petri net to a model incorporating organizational, temporal, informational, and social aspects. In this paper we focus on the social aspect. However, we first provide an example illustrating the broader concept of process mining.

2.1 An Example of a Staffware Log

Table 1 shows a fragment of a workflow log generated by the Staffware system. In Staffware events are grouped on a case-by-case basis. The first column refers to the activity (description), the second to the type of event, the third to the user generating the event (if any), and the last column shows a time stamp. The corresponding Staffware model is shown in Figure 1. Case 10 shown in Table 1 follows the scenario where first activity *Register* is executed followed by *Send questionnaire*, *Receive questionnaire*, and *Evaluate*. Based on the evaluation, the decision is made to directly archive (activity *Archive*) the case without further processing. For Case 9 further processing is needed, while Case 8 involves a timeout and the repeated execution of some activities. Someone familiar with Staffware will be able to decide that the three cases indeed follow a scenario possible in the Staffware model shown in Figure 1. However, three cases are not sufficient to automatically derive the model of Figure 1. Note that there are many Staffware models enabling the three scenarios shown in Table 1. The challenge of process mining is to derive "good" process, organizational, and social models with as little information as possible.

2.2 Discovering Control-flow Structures

To illustrate the principle of process mining in more detail, we consider the event log shown in Table 2 and focus on the *control flow* (cf. [1, 3, 5, 6, 10]). This log abstracts from the time, date, and event type, and limits the information to the order in which activities are being executed. The log shown in Table 2 contains information about five cases (i.e., process instances). The log shows that for four cases (1, 2, 3, and 4) the activities A, B, C, and D have been executed. For the fifth case only three activities are executed: activities A, E, and D. Each case starts

Case 10			
Directive Description	Event	User	yyyy/mm/dd hh:mm
	Start	John	2003/11/26 09:02
Register	Processed To	John	2003/11/26 09:02
Register	Released By	John	2003/11/26 09:09
Send questionnaire	Processed To	Clare	2003/11/26 09:23
Evaluate	Processed To	Sue	2003/11/26 09:58
Send questionnaire	Released By	Clare	2003/11/26 10:11
Receive questionnaire	Processed To	John	2003/11/26 13:05
Receive questionnaire	Released By	John	2003/11/26 13:06
Evaluate	Released By	Sue	2003/11/26 15:23
Archive	Processed To	Mary	2003/11/26 16:20
Archive	Released By	Mary	2003/11/26 16:21
	Terminated		2003/11/26 16:21
Case 9			
Directive Description	Event	User	yyyy/mm/dd hh:mm
	Start	Mike	2003/11/25 11:25
Register	Processed To	Mike	2003/11/25 11:25
Register	Released By	Mike	2003/11/25 11:37
Send questionnaire	Processed To	Mary	2003/11/25 11:51
Evaluate	Processed To	Sue	2003/11/25 11:52
Send questionnaire	Released By	Mary	2003/11/25 13:10
Receive questionnaire	Processed To	Mike	2003/11/25 15:02
Receive questionnaire	Released By	Mike	2003/11/25 15:20
Evaluate	Released By	Sue	2003/11/25 15:31
Process complaint	Processed To	Peter	2003/11/25 16:37
Process complaint	Released By	Peter	2003/11/25 16:51
Check processing	Processed To	Sue	2003/11/25 17:03
Check processing	Released By	Sue	2003/11/25 17:12
Archive	Processed To	Mary	2003/11/25 17:38
Archive	Released By	Mary	2003/11/25 17:41
	Terminated		2003/11/25 17:41
Case 8			
Directive Description	Event	User	yyyy/mm/dd hh:mm
	Start	John	2003/11/25 10:36
Register	Processed To	John	2003/11/25 10:36
Register	Released By	John	2003/11/25 10:40
Send questionnaire	Processed To	Mary	2003/11/25 10:50
Evaluate	Processed To	Sue	2003/11/25 11:25
Send questionnaire	Released By	Mary	2003/11/25 11:51
Receive questionnaire	Processed To	John	2003/11/26 09:36
Receive questionnaire	Expired	John	2003/11/26 09:52
Receive questionnaire	Withdrawn	John	2003/11/26 09:53
...			

Table 1. A Staffware log.

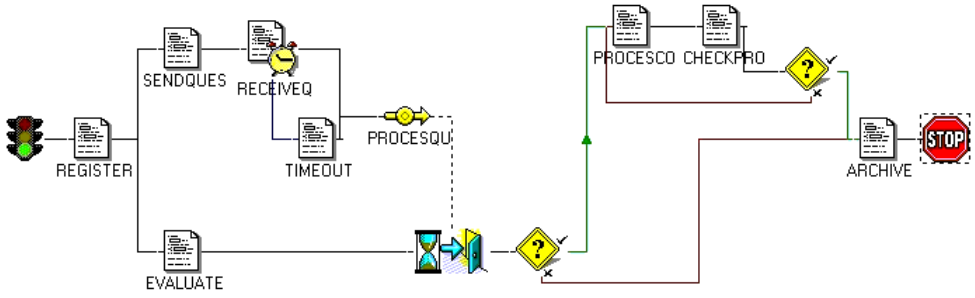

Fig. 1. The staffware model

with the execution of A and ends with the execution of D. If activity B is executed, then also activity C is executed. However, for some cases activity C is executed before activity B. Based on the information shown in Table 2 and by making some assumptions about the completeness of the log (i.e., assuming that the cases are representative and a sufficient large subset of possible behaviors is observed), we can deduce the Petri net shown in Figure 2(a) (cf. [20]).

2.3 Discovering Organizational Structures

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 2. An event log.

Figure 2(a) does not show any information about the performers, i.e., the people executing activities. However, Table 2 shows information about the performers. For example, we can deduce that activity A is executed by either John or Sue, activity B is executed by John, Sue, Mike or Carol, C is executed by John, Sue, Mike or Carol, D is executed by Pete or Clare, and E is executed by Clare. We could indicate this information in Figure 2(a). The information could also be used to "guess" or "discover" organizational structures. For example, a guess could be that there are three roles: X, Y, and Z. For the execution of A role X is required and John and Sue have this role. For the execution of B and C role Y is required and John, Sue, Mike and Carol have this role. For the execution of D and E role Z is required and Pete and Clare have this role. For five cases these choices may seem arbitrary but for larger data sets such inferences capture the dominant roles in an organization. The resulting "activity-role-performer diagram" is shown in Figure 2(b). The three "discovered" roles link activities to performers.

2.4 Discovering Social Networks

When deriving roles and other organizational entities from the event log the focus is on the relation between people or groups of people and the process. Another perspective is not to focus on the relation between the process and individuals but on relations among individuals (or groups of individuals). Consider for example Table 2. 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 2 it seems that John is not working with Carol and Sue is not working with Carol.[1] These examples show that the event log can be used

[1] Clearly the number of events in Table 2 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.

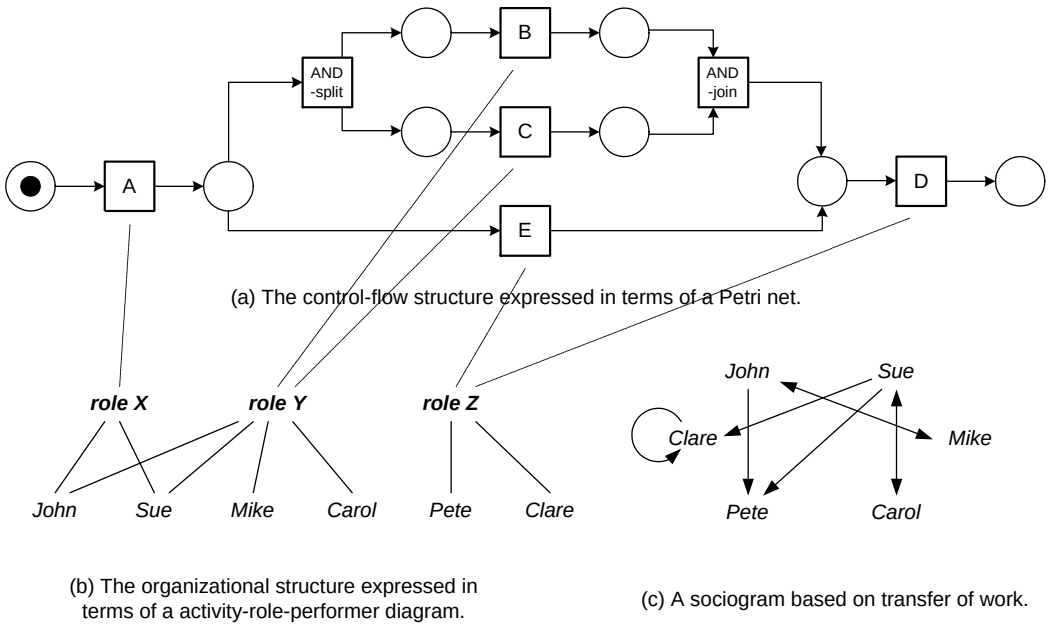

Fig. 2. Three models (control-flow, organizational, and social network structures) based on the event log shown in Table 2.

to 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 2(c). 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 there 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 2(c) 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 2(c) 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.

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 2(c) 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.

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 2.4. 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 (http://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 element 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 2(c) 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 [7]. 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) [9, 12, 13, 22, 23]. 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., $\sum_j W_{j,i} - W_{i,j}$) [9, 22, 23]. 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. [9, 22, 23].

Let us apply some of these notions to the sociogram shown Figure 2(c) 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, NetMiner, Egonet, InFlow, KliqueFinder, MetaSight, NetForm, NetVis, StOCNET, UCINET, and visone are just some of the many SNA tools available. For more information on SNA we refer to [8, 9, 22, 23].

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 individuals 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 Figure 2(c) 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 some of the metrics we have developed to establish relationships between individuals from event logs. We address only examples of the first three types introduced in Section 3.2. Before we define these examples 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 a case is passed 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 the process model). Based on these refinements, we derive $2^3 = 8$ variants for both the handover of work and subcontracting metrics. These variant metrics are all

based on the same event log. Before defining metrics, the basic notions 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 $n \in \mathbb{N}$:

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

$p_1 \triangleright_c^n p_2$ 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 2, *John* $\triangleright_c^1$ *Mike* equals 1 and *John* $\triangleright_c^3$ *Pete* equals 1. 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 2. 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 and *Mike* $\trianglerighteq_c^2$ *Pete* equals 1, because activity B and C do not have a casual dependency but activity B and D do (see Figure 2(a); B and C are in parallel but are both causally followed by D).

Using such relations, we define handover of work metrics. The following metrics only deal with first and second refinements. If we replace $\triangleright$ with $\trianglerighteq$, we can calculate the relationships considering only real casual dependencies and thus deal with the third refinement.

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

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

$p_1 \triangleright_L p_2$ 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. For example, in Table 2, *John* $\triangleright_L$ *Mike* equals 2/14. $p_1 \dot{\triangleright}_L p_2$ ignores multiple transfers within one instance (i.e., case). $p_1 \triangleright_L^\beta p_2$ 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.

$p_1 \triangleright_L^\beta p_2$ consider all possible successions, while $p_1 \dot{\triangleright}_L^\beta p_2$ ignores multiple transfers within one case.

In the case of subcontracting, we only describe a basic relation and a basic metrics, i.e., again there are 8 variants but we only consider the basic one.

Definition 4.4. (In-between metrics) 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 $n > 1$:

- $p_1 \diamond_c^n p_2 = \exists_{0 \leq i < j < i+n < |c|} \pi_p(c_i) = p_1 \ \wedge \ \pi_p(c_j) = p_2 \ \wedge \ \pi_p(c_{i+n}) = p_1$
- $p_1 \diamond_L p_2 = (\sum_{c \in L} |p_1 \diamond_c^2 p_2|)/(\sum_{c \in L} (|c| - 2))$

In subcontracting, the three refinements mentioned 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 first and 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.

4.2 Metrics Based on Joint Cases

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

Definition 4.5. (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 which p_1 appeared, since the appearance is relative to the performers. Let us apply this metric to analyze the relationship between John and Pete based in the log shown in Table 2. *John* $\bowtie_L$ *Pete* equals 2/2 and *Pete* $\bowtie_L$ *John* equals 2/4.

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

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 4.6. ($\triangle$) 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 a part of the performer by activity matrix derived from Table 2.

performer	activity A	activity B	activity C	activity D	activity E
Sue	3	1	1	0	0
Carol	0	1	1	0	0
Clare	0	0	0	1	1

Table 3. A part of the performer by activity matrix.

Based on this matrix, we defined several metrics to measure the distance between two performers. These metrics are all based on a comparison of the corresponding row vectors.

In this section we introduced only some of the metrics we have developed. 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 $W(p_1, p_2) = p_1 \rhd_L p_2$. 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. 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.

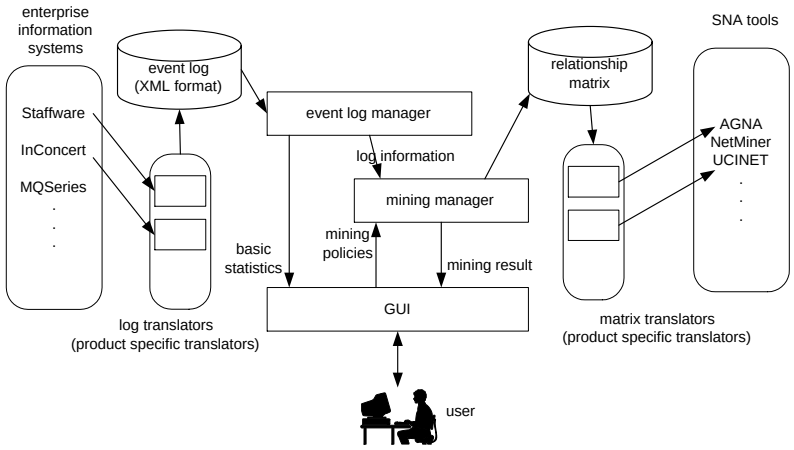

Fig. 3. The architecture of MiSoN

6 Example: Applying MiSoN to a Staffware log

Although MiSoN and the underlying analysis routines are tool-independent, we focus on a concrete system to illustrate the applicability of the results presented in this paper. The Staffware audit trail referred to by Table 1 is converted by MiSoN to the XML format described in the previous section. In this sample data, we only consider the "released by" event type to make sociograms. We have tested MiSoN with several metrics mentioned in previous section. Figure 4 shows a screenshot of MiSoN when displaying the mining result of handover of work metrics. MiSoN

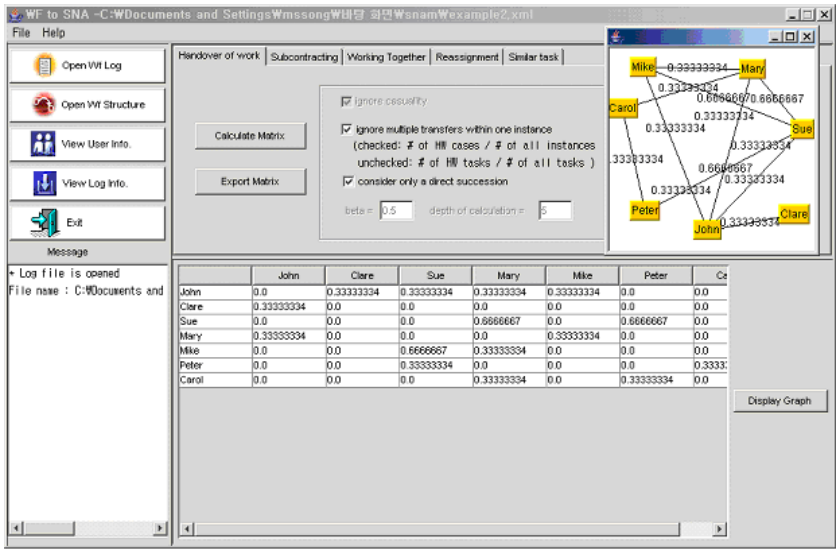

Fig. 4. MiSoN screenshot showing a sociogram based on the Staffware log

can export the mining result using the AGNA-translator (but also other tools like UCINET and NetMiner). AGNA (cf. http://www.geocities.com/imbenta/agna/)

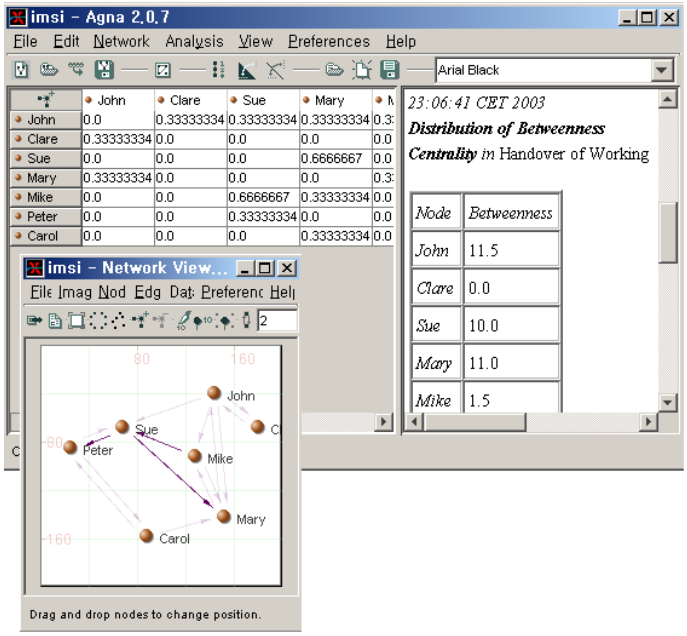

Fig. 5. Screenshot of AGNA when analyzing the input from MiSocN

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 5 shows the analysis using the tool AGNA. It also shows the network structure of result.

7 Related Work

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

The idea of process mining is not new [1, 3, 5, 6, 10, 15, 16, 18, 21, 24] but has been mainly aiming at the control-flow perspective. In this paper, it is impossible to do justice to the work done in this area. Therefore, for more information on process mining we refer to a special issue of Computers in Industry on process mining [4] and the survey paper [3]. 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.

Since the early work of Moreno [17], 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 [7–9, 11–13, 17, 19, 22, 23]. There have been many studies analyzing workflow processes based on insights from social network analysis. However, 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. 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

(http://www.buddygraph.com/) and MetaSight (http://www.metasight.co.uk/). However, these tools monitor unstructured messages and cannot distinguish between different activities (e.g., work-related interaction versus social interaction).

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.

At this point in time we are applying MiSoN to a real data set, and we plan to report on this in a future paper. 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.

Acknowledgement
Minseok Song is visiting Department of Technology Management at Eindhoven University of Technology with fund by BK21 program. He would like to thank the Ministry of Education of Korea for its financial support through the BK21 program.

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, 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.
4. 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.

5. W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. QUT Technical report, FIT-TR-2003-03, Queensland University of Technology, Brisbane, 2003. (Accepted for publication in IEEE Transactions on Knowledge and Data Engineering.).
6. 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.
7. A.A. Bavelas. A Mathematical Model for Group Structures. *Human Organization*, 7:16–30, 1948.
8. 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.
9. R.S. Burt and M Minor. *Applied Network Analysis: A Methodological Introduction*. Sage, Newbury Park CA, 1983.
10. 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.
11. M. Feldman. Electronic mail and weak ties in organizations. *Office: Technology and People*, 3:83–101, 1987.
12. L.C. Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40:35–41, 1977.
13. L.C. Freeman. Centrality in Social Networks: Conceptual Clarification. *Social Networks*, 1:215–239, 1979.
14. 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.
15. 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.
16. IDS Scheer. ARIS Process Performance Manager (ARIS PPM). http://www.ids-scheer.com, 2002.
17. J.L. Moreno. *Who Shall Survive?* Nervous and Mental Disease Publishing Company, Washington, DC, 1934.
18. 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.
19. H. Nemati and C.D. Barko. *Organizational Data Mining: Leveraging Enterprise Data Resources for Optimal Performance*. Idea Group Publishing, Hershey, PA, USA, 2003.
20. 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.
21. 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.
22. J. Scott. *Social Network Analysis*. Sage, Newbury Park CA, 1992.
23. S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, 1994.
24. 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.