Towards comprehensive support for organizational mining
Song, M.S.; van der Aalst, W.M.P.

Published: 01/01/2007

Document Version
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):
Towards Comprehensive Support for Organizational Mining

Minseok Song and Wil M.P. van der Aalst

Eindhoven University of Technology
P.O.Box 513, NL-5600 MB, Eindhoven, The Netherlands.
{m.s.song, w.m.p.v.d.aalst}@tue.nl

Abstract. Process mining has emerged as a way to analyze processes based on the event logs of the systems that support them. Today’s information systems (e.g., ERP systems) log all kinds of events. Moreover, also embedded systems (e.g., medical equipment, copiers, and other high-tech systems) start producing detailed event logs. The omnipresence of event logs is an important enabler for process mining. The primary goal of process mining is to extract knowledge from these logs and use it for a detailed analysis of reality. Lion’s share of the efforts in this domain has been devoted to control-flow discovery. Many algorithms have been proposed to construct a process model based on an analysis of the event sequences observed in the log. As a result, other aspects have been neglected, e.g., the organizational setting and interactions among coworkers. Therefore, we focus on organizational mining. We will present techniques to discover organizational models and social networks and show how these models can assist in improving the underlying processes. To do this, we present new process mining techniques but also use existing techniques in an innovative manner. The approach has been implemented in the context of the ProM framework.

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

1 Introduction

Business Process Management (BPM) systems provide a broad range of facilities to enact and manage operational business processes. Ideally, these systems should provide support for the complete BPM life-cycle: (re)design, configuration, execution, control, and diagnosis of processes. However, existing BPM tools are unable to support the full life-cycle [38]. There are clearly gaps between the various phases (i.e., users need to transfer or interpret information without any support) and some of the phases (e.g., the redesign and diagnosis phases) are not supported satisfactorily.

Process mining techniques can be used to support the redesign and diagnosis phases by analyzing the processes as they are being executed. Process mining can be seen in the broader context of Business (Process) Intelligence (BI) and Business Activity Monitoring (BAM). Commercial BI and BAM tools are not
doing any process mining. They typically look at aggregate data seen from an external perspective (frequencies, averages, utilization, service levels, etc.). Unlike BI and BAM tools, process mining looks “inside the process” (What are the causal dependencies?, Where is the bottleneck?, etc.) and at a very refined level. In the context of a hospital, BI tools focus on performance indicators such as the number of knee operations, the length of waiting lists, and the success rate of surgery. Process mining is more concerned with the paths followed by individual patients and whether certain procedures are followed or not.

Process mining requires the availability of an event log. Luckily, event logs are widely available today and the total volume of events being recorded is still growing at a spectacular rate. Events logs may originate from all kinds of systems ranging from enterprise information systems to embedded systems. Process mining is a very broad area both in terms of applications (from hospitals and banks to embedded systems in cars, copiers, and sensor networks). Most of the process mining research has been focusing on control-flow discovery, i.e., constructing a process model based on an event log while other aspects have been neglected, e.g., the organizational setting and interactions among coworkers.

The focus of this paper is on organizational mining. The observation that human behavior is highly relevant for the performance of processes, suggests that comprehensive support for this is needed. Process mining is most interesting in situations where processes are not completely controlled by systems. This is of course the case in any environment where humans play a dominant role. For example, in a hospital and many other professional organizations, processes “emerge” because of human decision making. The discovery of organizational knowledge, such as organizational structures and social networks, enables managers to understand organizational structures and improve business processes. Therefore, organizational mining assists in understanding and improving organizational and social structures. For example, social networks show the communication structures in enterprises. This can be used to design communication infrastructures or office layouts.

In this paper, we focus on organizational mining. We describe the challenges related to organizational mining and try address them in a comprehensive manner. Our process mining tool (ProM) supports the methods proposed in this paper.

The remainder of this paper is organized as follows. We provide an overview of process mining and organizational mining in Section 2. Section 3 presents a simple example process that is used throughout this paper. Then, Section 4 introduces important notions such as process log and organizational model in much more detail. Section 5 explains the organizational mining methods along with an example. Section 6 describes the implementation of our methods in ProM. Section 7 reviews related work. Finally, Section 8 concludes the paper.
2 Process Mining

Process mining is applicable to a wide range of systems. These systems may be pure information systems (e.g., ERP systems) or systems where the hardware plays a more prominent role (e.g., embedded systems). The only requirement is that the system produces event logs thus recording (parts of) the actual behavior.

An interesting class of information systems that produce event logs are the so-called Process-Aware Information Systems (PAISs) [22]. Examples are classical workflow management systems (e.g. Staffware), ERP systems (e.g. SAP), case handling systems (e.g. FLOWer), PDM systems (e.g. Windchill), CRM systems (e.g. Microsoft Dynamics CRM), middleware (e.g., IBM’s WebSphere), hospital information systems (e.g., Chipsoft), etc. These systems provide very detailed information about the activities that have been executed.

This section first provides an overview of process mining and the focuses on organizational mining.

2.1 Overview of process mining

The goal of process mining is to extract information (e.g., process or organizational models) from these logs, i.e., process mining describes a family of a-posteriori analysis techniques exploiting the information recorded in the event logs. Typically, these approaches assume that it is possible to sequentially record events such that each event refers to an activity (i.e., a well-defined step in the process) and is related to a particular case (i.e., a process instance). Furthermore, some mining techniques use additional information such as the performer or originator of the event (i.e., the person / resource executing or initiating the activity), the timestamp of the event, or data elements recorded with the event (e.g., the size of an order).

Process mining addresses the problem that most “process/system owners” have limited information about what is actually happening. In practice, there is often a significant gap between what is prescribed or supposed to happen, and what actually happens. Only a concise assessment of reality, which process mining strives to deliver, can help in verifying process models, and ultimately be used in system or process redesign efforts.

The idea of process mining is to discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs. As shown in Figure 1, we consider three basic types of process mining: (1) discovery, (2) conformance, and (3) extension.

Traditionally, process mining has been focusing on discovery, i.e., deriving information about the original process model, the organizational context, and execution properties from enactment logs. There is no a-priori model, i.e., based on an event log some model is constructed. An example of a technique addressing the control flow perspective is the $\alpha$-algorithm, which constructs a Petri net model [19, 41] describing the behavior observed in the event log. However, process mining is not limited to process models (i.e., control flow) and recent process mining techniques are more and more focusing on other perspectives,
e.g., the organizational perspective or the case perspective. For example, there are approaches to extract social networks from event logs and analyze them using social network analysis [5]. This allows organizations to monitor how people, groups, or software/system components are working together.

Conformance checking compares an a-priori model with the observed behavior as recorded in the log. In this case, there is an a-priori model. This model is used to check if reality conforms to the model. For example, there may be a process model indicating that purchase orders of more than one million Euro require two checks. Another example is the checking of the four-eyes principle. Conformance checking may be used to detect deviations, to locate and explain these deviations, and to measure the severity of these deviations. In [43] it is shown how a process model (e.g., a Petri net) can be evaluated in the context of a log using metrics such as “fitness” (Is the observed behavior possible according to the model?) and “appropriateness” (Is the model “typical” for the observed behavior?). However, it is also possible to check conformance based on organizational models, predefined business rules, temporal formulas, Quality of Service (QoS) definitions, etc.

There are different ways to extend a given process model with additional perspectives based on event logs, e.g., decision mining, performance analysis, and user profiling. There is an a-priori model. This model is extended with a new aspect or perspective, i.e., the goal is not to check conformance but to enrich the model with the data in the event log. Decision mining, also referred to as decision point analysis, aims at the detection of data dependencies that affect the routing of a case [44]. Starting from a process model, one can analyze how data attributes influence the choices made in the process based on past process executions. Classical data mining techniques such as decision trees can be leveraged for this purpose. Similarly, the process model can be extended with timing information (e.g., bottleneck analysis).
Orthogonal to the three types of process mining depicted in Figure 1 (i.e., discovery, conformance, and extension), we distinguish three different perspectives: (1) the process perspective (“How?”), (2) the organizational perspective (“Who?”) and (3) the case perspective (“What?”). The process perspective focuses on the control-flow, i.e., the ordering of activities. The goal of mining this perspective is to find a good characterization of all possible paths, e.g., expressed in terms of a Petri net [41] or Event-driven Process Chain (EPC) [30]. The organizational perspective focuses on the originator field, i.e., which performers are involved and how are they related. The goal is to either structure the organization by classifying people in terms of roles and organizational units or to show relations between individual performers. The case perspective focuses on properties of cases. Cases can be characterized by their path in the process or by the originators working on a case. However, cases can also be characterized by the values of the corresponding data elements. For example, if a case represents a replenishment order, it may be interesting to know the supplier or the number of products ordered.

Figure 2 relates the two dimensions. As shown, the traditional focus of process mining research has been on process discovery, i.e., constructing control-flow models from event logs. Data mining (e.g., decision trees) and Business Intelligence (BI) tools mainly focus on the case perspective, i.e., cases with attribute values are analyzed without constructing some kind of process model. This paper will focus on the organizational perspective. Therefore, the following subsection elaborates on this perspective.

2.2 Organizational mining

After providing an overview of process mining, we now focus on organizational mining. Therefore, we first discuss issues related to organizational mining ac-
According to three types of mining mentioned before (i.e. *discovery*, *conformance*, and *extension*).

*Discovery* aims at constructing a model that reflects current situations. For organizational mining, two kinds of models are relevant. These are (1) the organizational model that represents the current organizational structure and (2) the social network that shows the communication structure in an organization. An organizational model usually consists of organizational units (e.g. functional units), roles (e.g. duty), originators, and their relationships (i.e. who belongs to which functional unit, who plays what roles, hierarchy among organizational units). When we analyze the process logs, it is difficult to find an explicit hierarchy of organizational units. However, it is possible to derive originator groups in which the people are allowed to execute similar tasks. Only a specific originator group and not all originators are allowed to carry out similar tasks. Thus, from a “profile” describing how frequently individuals conduct specific tasks, we can derive groups. A originator group could be a organizational unit or a grouping of people who perform the same roles in real life. A social network is a network in which nodes represent individuals or organizational units, and arcs between the nodes denote the relationships between them. It is possible to derive social networks from the logs as shown in [5]. The generated social networks allow organizations to monitor how people and groups work together. The social networks can be analyzed using a wide variety of SNA (Social Network Analysis) techniques that compute metrics such as centrality, position, density, etc [14, 48, 50]. SNA can also be augmented by other techniques from social sciences as shown in [18, 10, 17].

Furthermore, we can take into account discovery of rules, such as staff assignment rules and originator allocation rules. Staff assignment rules contain the guidelines on how a task is assigned to roles or organizational units. One example of rule is the requirement that the task of repairing a mobile phone should be assigned to an engineer who belongs to the mobile phone team. While staff assignment rules define who is allowed to do which tasks, originator allocation rules define to whom the specific task is assigned in run-time. We can assign work based on the priority of the work, capacity of originators, or FIFO (First In, First Out)/LIFO (Last In, First Out) policies. For example, the schedule events of these three tasks appear in a particular sequence (i.e. task A, task B, task C) in both cases in Table 1. In the first case, these tasks started in the same order as scheduled (i.e. task A, task B, task C). If this is recurring pattern in the log, then one could conclude that tasks are assigned to the originators based on FIFO policy. For the second case in Table 1, the tasks start in a different order. The actual start events take place in reversed order (task C, task B, and task A). Thus, the originator allocation rule might be the LIFO policy if this is recurring pattern in the log.

*Conformance checking* examines whether the modeled behavior matches the observed behavior. As indicated before, there are two dimension of conformance measures in the control flow perspective: *fitness* and *appropriateness* [43]. *Fitness* is the degree of the association between the log traces and the execution paths.
specified by the process model. Appropriateness is the degree of accuracy with which the process model describes observed behavior. These concepts can also be applied to the organizational mining. For example, in staff assignment rule mining, we can redefine fitness as the extent to which the actual originators in the logs can be associated with task roles specified by staff assignment rules. We can also redefine appropriateness as the degree of accuracy with which the staff assignment rules describe observed behavior. For example, ten originators can be assigned to a task according to the staff assignment rule, while only three of them are actually involved in the execution of some instance of this task. We might say that they have a low appropriateness.

Extension aims at enriching an existing model by extending the model through the projection of information extracted from the logs onto the initial model. An example of this is the extension of a social network with performance data, i.e., bottlenecks can be projected onto an a-priori social network in this way. This extended model can then be used to identify communication problems in the organizational perspective.

In the remainder of this paper, we will show a comprehensive approach to organizational mining. We will present new analysis techniques and show how existing techniques (e.g., for discovering control-flow) can be adapted for organizational mining.

### 3 Running Example

The example model used throughout the paper is the “repair” process of products within an electronic company that makes mobile phones and GPS systems. In Figure 3, the process model is expressed in terms of a WorkFlow net, i.e. a Petri net describing the lifecycle of a case. The process starts with the “Receive an item and a repair request” task (A). The customer sends his broken item to the company and requests repair. After receiving the request, a preliminary check (B) is carried out to find its faults. In parallel, the warranty is checked (C). Then, based on the status of the item and the warranty of the customer, repair costs are calculated and passed back to the customer. If the customer decides to repair the product, the product is repaired (E) and subsequently a bill for payment is issued (F). Otherwise, a cancellation letter (F) is sent. After that, the item is returned (H) and the case is closed.

Figure 4 shows the organizational model of the company. The model has three organizational units, three roles, and nine originators. The organization units consist of “Customer Service Team”, “Mobile Phone Team”, and “GPS
Team”, while the roles are clerk, engineer, and financial administrator. “Customer Service Team” has only one originator whose role is that of clerk. She is in charge of both the “Receive an item and a repair request” and the “Check the warranty” task. The “Mobile Phone Team” and “GPS Team” have four originators each. Since the company deals with two kinds of products, the item can be either a mobile phone or a GPS product. The case is forwarded to the appropriate team according to the product type. Each team consists of a clerk, two employees, and a financial administrator. Clerks are involved in administrative work, i.e. “Notify the customer” (D), “Send a cancellation letter” (G), and “Return the item” (H). Engineers perform preliminary checks (B) and repair the broken item (C). Financial administrators handle the “Issue payment” task (F).

Table 2 shows an event log in a schematic way. The log is consistent with the process mentioned above. Each row refers to a single case and is represented as a sequence of events. Events are represented by the case identifier (denoted by the row), activity identifier (first element), and originator (second element). In the remainder of the paper, we use the process model, the organizational model, and the example log to show how organizational information is derived.

4 Process Logs and Organizational Model

Before explaining the organizational mining in more detail, this section discusses the MXML process log and the meta model used for representing organizations. As indicated before, a process log consists of several instances or cases, each of
which may be made up of several audit trail entries. An audit trail entry corresponds to an atomic event, e.g., the scheduling, start, or completion of a task. Each audit trail entry records task name, event type, originator and time stamp. This information is defined by the MXML schema, a standard XML format used in ProM. We can use the ProMimport to convert logs from existing (commercial) process-aware systems (e.g. Staffware and FLOWer) to the MXML format [26, 20]. Figure 5 shows a screenshot of the event log. The process log starts with the WorkflowLog element that contains Source, and Process elements. The Source element refers to the information about the software or the system that was used to record the log, while the Process element represents the process to which the process log belongs. The 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. The WorkflowModelElement refers to the activity 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), and complete (the completion of a task instance), etc. The Timestamp refers to the time when the event occurred and the Originator corresponds to the originator, i.e., the resource initiating the event.

To describe organizational concepts, we introduce OMML (Organizational Model Markup Language) in this paper. Figure 6 illustrates the XML schema describing this format. The schema has the OrgModel element as its root element. This root element contains OrgEntity, Resource, and Task elements. The OrgEntity element refers to an organizational entity. It has EntityID, EntityName, and EntityType elements as attributes. An OrgEntity can be an organizational unit, a role, or a user defined type. This type information is specified in the EntityType element. The Resource element represents an originator. It contains ResourceID, ResourceName, and HasEntity elements. The former two elements are used to describe the originator’s ID and name. The HasEntity element refers to an OrgEntity element. It refers to the functional unit of the originator, his role, or etc. The Task element refers to a task. It has TaskID, TaskName, EventType, and HasEntity elements. The former two elements are used to describe the task’s ID and name. EventType element refers to the event type of the task. Based on

<table>
<thead>
<tr>
<th>Case ID</th>
<th>log events</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(A,John),(B,Mike),(C,John),(D,Sue),(E,Pete),(F,Jane),(H,Sue)</td>
</tr>
<tr>
<td>2</td>
<td>(A,John),(B,Fred),(C,John),(D,Clare),(E,Robert),(F,Mona),(H,Clare)</td>
</tr>
<tr>
<td>3</td>
<td>(A,John),(C,John),(B,Pete),(D,Sue),(E,Mike),(F,Jane),(H,Sue)</td>
</tr>
<tr>
<td>4</td>
<td>(A,John),(C,John),(B,Fred),(D,Clare),(G,Clare),(H,Clare)</td>
</tr>
<tr>
<td>5</td>
<td>(A,John),(C,John),(B,Robert),(D,Clare),(E,Fred),(F,Mona),(H,Clare)</td>
</tr>
<tr>
<td>6</td>
<td>(A,John),(B,Mike),(C,John),(D,Sue),(G,Sue),(H,Sue)</td>
</tr>
</tbody>
</table>

Table 2. Example process logs (A: Receive a item and repair request, B: Check the item, C: Check the warranty, D: Notify the customer, E: Repair the item, F: Issue payment, G: Send the cancellation letter, H: Return the item)
the event type, the task can be assigned to a different organizational entity. For example, schedule events are activated by a system, and start events are invoked by an originator who can execute the task. The HasEntity element describes an organizational unit or a role that corresponds to the task.

Figure 7 shows the example organizational model which reflects the organizational structure in Figure 4. Three organizational entities and three roles are defined using the OrgEntity elements. Nine originators are specified with the Resource elements. Each Resource element shows his/her organizational unit and role using the HasEntity elements. For example, Sue has two HasEntity elements: (1) “Mobile Phone Team” and (2) “Clerk”.

5 Organizational Mining

This section describes a comprehensive approach towards organizational mining. We distinguish three types of organizational mining (1) Organizational model mining, (2) Social network analysis, and (3) Information flows between organizational entities. In the remainder we elaborate on each of the three types.

5.1 Organizational model mining

Organizational model mining aims at deriving the organizational model from process logs. As mentioned before, we do not derive an explicit organizational hierarchy but a group of originators that has similar characteristics in process execution. A group could be either an organizational unit or role.

In this paper, we explain three kinds of mining methods. The first one is “default mining” that is a simple way to derive a role for each task. Before we formally define the default mining method, we introduce a convenient notation...
for event logs. This can be seen as an abstraction of the MXML format defined in Section 4.

**Definition 5.1. (Event log)** Let $T$ be a set of tasks (i.e., atomic workflow/process objects, also referred to as activities) and $P$ a set of originators (i.e., persons, resources, or agents). $E = T \times P$ is the set of (possible) events, i.e., combinations of an activity and an originator (e.g. $(t, p)$ denotes the execution of task $t$ by originator $p$). $C = E^*$ 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 $C$. Each element of $L$ denotes a case.

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: $\pi_t(e) = t$ and $\pi_p(e) = p$ for some event $e = (t, p)$. Now the default mining method is defined as follows.

**Definition 5.2. (Default Mining)** Let $L$ be a log, $T$ be a set of tasks, and $c = (e_0, e_1, \ldots) \in L$. $O_t$ and $A_S$ are defined as follows:

(i) For each $t \in T$, $O_t = \{\pi_p(e) | e \in c \land \pi_t(e) = t\}$.

(ii) $A_S = \{(t, O_t) | t \in T\}$. 
Fig. 7. The example organizational model in MXML format viewed using XML Spy

$O_t$ stands for the organizational entity (i.e. role or organizational unit) for a task $t$ and has originators who performed the task $t$. For example, in the log shown in Table 2, $O_A = \{\text{John}\}$, $O_B = \{\text{Robert, Fred, Mike, Pete}\}$, etc. We can obtain seven entities from the log, since $O_t$ is derived for each task. $A_S$ is the entity assignment that shows the relationship between organizational entity and tasks. From the example log, we attain $A_S = \{(A, O_A), (B, O_B), \ldots, (H, O_H)\}$.

Other methods are inspired by the metrics based on joint activities and the metrics based on joint cases proposed in [5]. Metrics based on joint activities focus on the activities that individuals perform. We assume that originators doing similar things are more closely linked than originators doing completely different tasks. Each originator has a “profile” (i.e. originator by activity matrix) based on how frequently they conduct specific activities. Table 3 shows the originator by activity matrix derived from Table 2.

<table>
<thead>
<tr>
<th>originator</th>
<th>act A</th>
<th>act B</th>
<th>act C</th>
<th>act D</th>
<th>act E</th>
<th>act F</th>
<th>act G</th>
<th>act H</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sue</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Mike</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pete</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Jane</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Clare</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fred</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Robert</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mona</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3. The originator by activity matrix
From the profile, we can measure the “distance” between the profiles of different originators by comparing the corresponding row vectors. We can calculate Minkowski distance, Hamming distance, Pearson’s correlation coefficient to quantify this “distance”. Figure 8(a) shows the network derived by applying Pearson’s correlation coefficient to Table 3. Note that Pearson’s correlation coefficient uses values ranging from -1 to +1. Since the positive values imply positive linear relationships between variables, we applied the threshold value of 0.0 and removed negative arcs from the network. Four clusters (i.e. \{John\}, \{Jane, Mona\}, \{Mike, Pete, Fred, Robert\}, \{Sue, Clare\}) are derived. They coincide with the roles shown in Figure 4. This is because each task is assigned to the proper originator based on the associated roles. For example, the “Issue payment” task (F) is assigned to the role financial administrator (i.e., Jane and Mona). Thus they have the same profile and belong to the same cluster.

Metrics based on joint cases count how frequent two originators are performing activities on the same case. If originators work together on cases, they will have a stronger bond than originators who rarely work together. For example, in the log shown in Table 2, the value from Mike to Jane is 2/3, since Mike appears in three cases and they work together twice. Figure 8(b) shows the network derived by applying the metrics based on joint cases to the example log. The network has two sub parts. The upper part is associated with “GPS team”, while the lower part refers to “Mobile Phone team”. They are connected through John. If we disconnect the node John from the rest of the network, three clusters (i.e. \{John\}, \{Sue, Mike, Pete, Jane\}, \{Clare, Fred, Robert, Mona\}) are obtained, and these clusters are relevant to functional units shown in Figure 4. This is because the case is assigned to the proper team based on the product type and handled within the functional unit.

After applying the metrics based on joint activities and the metrics based on joint cases, we obtain clusters that correspond to possible organizational entities. We use the following entity assignment method to derive the relationship between organizational entities and tasks.

**Definition 5.3. (Entity assignment \(A_S\))** Let \(L\) be a log, \(T\) be a set of tasks, and \(P\) be a set of originators. Moreover, \(\hat{O} \subseteq \mathcal{P}(P)\) is the set of organizational entities. Based on this we define the entity assignment \(A_S\) as follows: \(A_S = \{(t, O) \in T \times \hat{O} | \exists c \in L, \exists e \in c, \pi_t(e) = t \land \pi_p(e) \in O\}\).

Any \(O \in \hat{O}\) represents an organizational entity such as an organizational unit or a role. \(\hat{O}\) is a set of all organizational entities. If an originator executed a task, the task is assigned to the organizational entity to which the originator belongs. For example, in the example log, since Sue executed task D in the first case, task D is assigned to all organizational entities she belongs to.

---

\(^1\) \(\mathcal{P}(X)\) is the power set of \(X\), i.e., \(\hat{O}\) is a set of organizational entities (set of sets) and \(O \in \hat{O}\) is a set of originators representing e.g. a role.
5.2 Social network analysis

To derive social networks from process logs, different kinds of metrics have been developed in [5]. For a better understanding of our approach, we should briefly examine the basic concept. The idea is monitoring how individual cases are routed between originators. A typical example is the handover of work metric. If there are two subsequent (causally related) activities within a case (i.e., process instance) where the first is completed by originator \( i \) and the second by originator \( j \), it is likely that there is a handover of work from originator \( i \) to originator \( j \). Hence, we can add an arc from the node \( i \) to the node \( j \). 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” \( \beta \), i.e., if there are \( n \) activities in-between an activity completed by originator \( i \) and an activity completed by originator \( j \), the causality fall factor is \( \beta^n \). Another example is the subcontracting metric where the main idea is to count the number of times originator \( j \) executed an activity in-between two activities executed by originator \( i \). This may indicate that work was subcontracted by originator \( i \) to originator \( j \). Using these metrics, we can generate social networks. Figure 9 shows a social network derived from the log in Table 2 by applying the handover of work metric. It shows a relationship among originators in terms of process flow. For example, John is connected six originators such as Fred, Robert, Clare, Mike, Pete, and Sue. It means that after John finishes his task, the case is transferred to one of the six originators.
The weights on arcs represent the ratios of transfers. The fact that the weight on the arc from John to Fred, Mike, and Sue is higher than the others shows that cases are more frequently transferred from John to Fred, Mike, and Sue. Since the case is assigned to either “GPS” or “Mobile Phone” team based on the product type and handled within the team, there are no transfers between different team members. (i.e. between \{Robert, Fred, Mona, Clare\} and \{Pete, Mike, Jane, Sue\})

![Fig. 9. The social network](image)

After generating a social network, various SNA techniques such as density, centrality, cohesion, equivalence, etc. can be applied. For example, betweenness (a ratio based on the number of geodesic paths visiting a given node) [50] can be used to find possible bottlenecks. In social networks generated by applying the handover of work metric, nodes with no incoming arcs are originators who only initiate processes, while nodes with no outgoing arcs are originators who perform only final activities. In social networks generated by applying the subcontracting metric, the start node of an arc represents a contractor and the end node means a subcontractor. Thus, nodes with a high out-degree of centrality are originators that usually play the role of contractors and nodes with a high in-degree of centrality are originators that usually act as subcontractors. In social networks generated by applying metrics based on joint cases, high density means that more originators work together and an ego network (a focal node and the nodes to whom ego is directly connected to) shows the originators that work together. The
average size of ego networks of a social network is an indicator of the degree of cooperation between originators. For example, if the average size of ego networks is five, then it means that a originator usually works with four other originators. Practical experience [5] shows that social network analysis based on event logs in a powerful tool for analyzing cooperation and coordination patterns. Unlike approaches based on the mining of e-mail messages, our approach is based on actual work-related events and is not “polluted” by non-work related events (e.g. betting on soccer games).

5.3 Information flows between organizational entities

Besides generating social networks where the nodes are originators, we can also construct social networks where the nodes correspond to organizational entities (i.e., groups of originators). Social networks based on organizational entities such as organizational units or roles, provide additional insights at a higher aggregation level.

So far we did not formalize social networks, but as the diagrams clearly show a social network is simply a weighted graph. Such a graph can be represented as $G = (P, R, W)$, where $P$ is the set of originators, $R \subseteq P \times P$ is the set of relations, and $W : R \rightarrow \mathbb{R}$ is a function indicating the weight of each relation, i.e., $W(p_1, p_2)$ is the Real valued weight of the relation from $p_1$ to $p_2$. From the social network, we can derive a graph $G_O$ where the nodes are organizational entities using the following method. This methods aggregates “originator nodes” into “organizational entity nodes”.

**Definition 5.4. (Deriving $G_\hat{O}$ from $G$)** Let $G = (P, R, W)$ be a social network for originators and $\hat{O} \subseteq P(P)$ be a set of organizational entities. $G_O = (\hat{O}, R_O, W_O)$ is defined as follows:

(i) $R_O = \{(O_1, O_2) \in \hat{O} \times \hat{O} \mid \exists (p_1, p_2) \in \{O_1 \times O_2\} (p_1, p_2) \in R\}$,
(ii) $W_O(O_1, O_2) = \sum_{(p_1, p_2) \in R \cap \{O_1 \times O_2\}} W(p_1, p_2)$, for $(O_1, O_2) \in R_O$.

$G_O = (\hat{O}, R_O, W_O)$ is a social network for organizational entities, where $\hat{O}$ is the set of organizational entities, $R_O$ is the set of relations, and $W_O$ is a function indicating the weight of each relation. By applying the method to the network in Figure 9, we derive the social network for organizational units (a) and the social network for roles (b) in Figure 10. For example, for Figure 10(a), $\hat{O}$ for organizational units is defined as $\hat{O} = \{O_{CS}, O_{MP}, O_{GPS}\}$, where $O_{CS} = \{\text{John}\}$, $O_{MP} = \{\text{Sue, Mike, Pete, Jane}\}$, $O_{GPS} = \{\text{Clare, Fred, Robert, Mona}\}$. $R_O$ is derived as a set of $\{(O_{CS}, O_{CS}), (O_{MP}, O_{MP}), (O_{GPS}, O_{GPS}), (O_{CS}, O_{MP}), (O_{CS}, O_{GPS}), (O_{MP}, O_{CS}), (O_{GPS}, O_{CS})\}$. For example, since there is an arc from John to Robert in Figure 10(a) and John and Robert belong to $O_{CS}$ and $O_{GPS}$ respectively, $(O_{CS}, O_{GPS})$ is included in $R_O$. The weight value ($W_O$) is calculated by summing up the values on arcs between originators in different organizational units. For example, to calculate the weight value ($W_O(O_{CS}, O_{GPS})$) on the arc from $O_{CS}$ to $O_{GPS}$, the values on the arcs from John to Clare, from
John to Fred, and from John to Robert are considered. Thus \( W_O(O_{CS}, O_{GPS}) = W(John, Clare) + W(John, Fred) + W(John, Robert) = 0.118. \)

![Diagram of social networks for organizational entities](image)

**Fig. 10.** Social networks for organizational entities

6 System Implementation

In this section, we present the implementation of the approach proposed in this paper. ProM\(^2\) has been developed to support various process mining algorithms. It enables rapid development of new algorithms and techniques by means of plug-ins [21]. A plug-in is basically the implementation of an algorithm that is of use in the process mining area. Such plug-ins can be added to the framework relatively easily. To support the methods described in Section 5, we have implemented five new plug-ins in ProM. These are briefly described below.

Figure 11 shows the overview of the implementation supporting organizational mining. There are five components that support the organizational perspective in ProM. The organizational model miner and social network miner read process logs and generate an organizational model and social networks respectively. From the social network and the organizational model, we can execute the grouping plug-in to derive a social network for organizational entities. The organizational model from the organizational model miner can be provided as a reference organizational model. The social network analysis plug-in provides

---

\(^2\) See [www.processmining.org](http://www.processmining.org) for more information and to download ProM and the five plug-ins developed in the context of this paper.
several social network analysis measures. The *replacement plug-in* is a filter that replaces an attribute of an event in the log by another attribute of the event. It enables users to reuse the existing mining plug-ins for the organizational perspective. The remainder of this section explains each plug-in in detail. Note that we will use the process log introduced in Section 3 as an example.

![Diagram of the five plug-ins and their relationships](image)

**Fig. 11.** Overview of the five plug-ins and their relationships

### 6.1 Organizational model mining

The organizational model miner supports deriving organizational models. Figure 12 shows a screenshot of the plug-in. The left-hand side of the figure shows the options. As mentioned before, it supports default mining, metrics based on doing similar tasks, and metrics based on working together. For the last two metrics, users can select a relevant option. The right-hand side shows the result window. The upper part enables users to change the organizational model by adding or deleting originators, organizational entities, tasks, and their relationships. The part below shows a graphical representation of the organizational model. The ovals, pentagons and boxes represent originators, organizational entities, and tasks respectively.

### 6.2 Social network analysis

To support social network analysis, three plug-ins have been implemented. Figure 13 shows the social network miner. The left-hand side of the figure shows
the main panel where users can select suitable metrics and set relevant options. Users can also filter originators according to their frequencies. By selecting the suitable metrics and options, users can start mining and create a social network. The right-hand side of the figure depicts the mining result. The social network is displayed both as matrix and as network. In the resulting network, users can remove arcs by applying threshold values and eliminate any isolated nodes.

After generating the social network, users can activate the social network analysis plug-in shown in Figure 14. We use the JUNG (Java Universal Network/Graph) library [39] to implement the plug-in. It provides several APIs in order to easily implement social network analysis features. The plug-in supports
several centrality measures such as degree [50], betweenness [13], hubs-and-authorities [32]. The centrality values are displayed in the left-hand side of the panel. Display options and the graph are shown in the right-hand side. It supports several layout options (e.g., spring layout, circle layout, etc.); enabling users to select an appropriate layout that suits the analysis objective. The vertex size can be changed according to the number of internal flows or to the appearance frequency of originators. If the social network contains the organizational model corresponding to the network, it can be shown. Users can also remove arcs from the network based on betweenness or weight of the arcs. As a result, the network is disconnected and becomes several clusters.

Fig. 14. Analyzing social networks

6.3 Information flow between organizational entities
From the result of the social network miner, the grouping plug-in can be executed to generate social networks for organizational entities. Figure 15 shows the screenshot of the plug-in. The upper part of the figure is the option panel. Users can add organizational information by manually providing the information or selecting an organizational model. If an originator has more than one organizational entity, a suitable one can be selected. The lower part shows the resulting network.

6.4 Replacement filter
The filtering plug-ins support “massaging” the log before applying process mining techniques. For example, there are filters to remove infrequent activities,
clean the log by deleting incomplete cases, etc. The replacement filter is a filter that systematically replaces a specific attribute of an event with another attribute. Figure 16 shows a screenshot of the filter. As shown, the task ID or originator ID in a log line can be replaced by other elements. For example, an originator ID can be replaced by the task ID of the same event, the originator’s role or the originator’s organizational entity.
In the ProM, there are several plug-ins that deal with mining the control flow perspective from logs. The replacement filter enables users to reuse these plug-ins to analyze logs in the organizational perspective (i.e. after replacing task IDs with originator IDs, we can execute control flow mining plug-ins.). For example, Figure 17 shows one of the example applications. After replacing task IDs with organizational units, a so-called “heuristics net” for originators is derived by the heuristics miner. The net is converted to a Petri net by one of ProM’s conversion plug-ins. Then, performance analysis with Petri net plug-in is executed to view the performance information such as sojourn time in each place, time in between two organizational units, bottleneck points, etc.

![Fig. 17. Organizational performance analysis](image)

The other way round is also possible. To reuse organizational mining plug-ins, we can swap originator IDs with other elements. Figure 18 shows an example. In the figure, the originator IDs are substituted by task IDs and the social network for tasks is generated and analyzed by social network miner and social network analysis plug-in. Thus we can analyze the relationship between tasks with social network analysis measure. For example, we can calculate centrality measure to find out which activity is located in the center of the network.

7 Related work

Related work can be divided in two categories: process mining and organizational issues in workflow area. There is a growing interest in process mining.
Process mining allows for the discovery of knowledge based on so-called “event logs”, i.e., a log recording the execution of activities in some business processes [7]. The concept of process mining was introduced by Cook et al. [16]. They started to mine process models from event logs in the context of software engineering [8]. Agrawal et al. first applied process mining in the context of workflow management. Recently many techniques and tools for process mining have been developed [3, 5–7, 28, 37, 47, 27]. The mainstream of process mining is to discover process models from process logs [3, 51, 4, 23]. It aims at creating a process model that best describes the set of process instances. To check whether the modeled behavior matches the observed behavior, the research on conformance checking has been carried out. The concept of conformance and several measures are proposed in [1, 4, 43, 2, 25].

Even though process mining deals with the organizational context of business processes, relatively little research has been carried out on analyzing business processes from the organizational perspective. Only a few research results in this area have been reported [5, 34]. In our work in [5], we developed methods for mining social networks from process logs to analyze relationships between originators involved in processes. We also implemented the social network miner in ProM. In this paper, we provide a much more comprehensive approach towards organizational mining. We focus not only on social networks for originators, but also on mining organizational models and analyzing relationships between organizational entities. Li et al. focused on mining staff assignment rules from process logs and an explicitly given organizational model [34]. They applied a decision tree learning method to enable rule discovery. Their approach required a-priori
knowledge (i.e. an organizational models) and focused on mining rules. But in this paper, we only used process logs and concentrated on mining organizational models and social networks.

Organizational aspects have been considered by many authors in workflow literature. However, in comparison with the research on the control-flow aspect of business process management, the research on mining organizational aspects has been largely neglected [45, 33]. A more prominent line of research in the workflow domain is organizational meta models. Several researchers have developed organizational meta models. Bussler proposed a generic organizational meta model [15]. Bertino et al. developed a logic based model that supports not only static authorization constraints, but also dynamic authorization constraints that refer to the history of the workflow instance [9]. Zur Mühlen pointed out the lack of attention for the link between the organizational elements and process activities. He developed several organizational meta models and guidelines for the design of a workflow-enabled organization [35, 36].

RBAC (Role based access control) [46, 24] is one of the more popular techniques to manage resources in workflow area [11, 12, 29, 49]. It uses roles as intermediates between tasks and originators. Roles are allocated to tasks in processes, and originators are made members of roles. The RBAC model is a useful mechanism for managing resources and the results of this paper could easily be translated to this model.

The handling of resources at runtime is also discussed in [33, 45]. Kumar et al. present dynamic work distribution in workflow management systems [33]. They have developed a mechanism that allows on-the-fly balancing of quality and performance considerations. Russell et al. define 43 resource patterns and evaluate several commercial workflow systems using these patterns [45]. Pesic and Van der Aalst model work distribution mechanisms of several commercial systems using colored Petri nets [40]. As a result, a core dynamic model emerges that allows for the comparison of workflow systems. In the adaptive workflow area, researchers focus on the change of organizational models. Klarmann proposes eight categories for structural changes in organizational model [31]. Rinderle et al. suggest a method to support organizational model changes considering access rules defined in organizational entities [42].

This section shows that related work on the one hand has been concentrating on the definition and implementation of work distribution mechanisms and on the other hand on the control-flow discovery. Few papers have been focusing on organizational mining.

8 Conclusion

The paper focuses on organizational mining. As shown, lion’s share of attention in the process mining area has been devoted to the process perspective (control-flow discovery) while classical data mining approaches have been devoted to the analysis of case attributes. Given the importance of people and organizational
entities in business process management, organizational mining deserves more attention, thus motivating our work.

In this paper, we distinguished three types of organizational mining (1) Organizational model mining, (2) Social network analysis, and (3) Information flows between organizational entities. We have shown how each of these types can be supported. Moreover, we showed how organizational mining can benefit from creatively using approaches developed for the process perspective. All of this is supported by the open-source process mining framework ProM. For this paper, we have implemented five new plug-ins that together constitute a comprehensive approach towards organizational mining.

Acknowledgements
This research is supported by EIT, NWO-EW, the Technology Foundation STW, and the SUPER project (FP6). Moreover, we would like to thank the many people involved in the development of ProM.

References


