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

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 and has been applied in various case studies. In this paper, we demonstrate the applicability of our techniques by analyzing the logs of a municipality in the Netherlands.

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

Email addresses: m.s.song@tue.nl (Minseok Song), w.m.p.v.d.aalst@tue.nl (Wil M.P. van der Aalst).

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 [23]. 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 aggregated 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 describe the challenges related to organizational mining

and try to address them in a comprehensive manner. We elaborate issues in organizational mining and distinguish three types of organizational mining (1) *Organizational model mining*, (2) *Social network analysis*, and (3) *Information flows between organizational entities*. For organizational model mining, we explain four kinds of methods and their characteristics. For the social network analysis, we summarize our previous approach [1] and explain the applicability of the social network analysis. A method to derive organizational entities from social networks is also proposed. 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 case study which demonstrates the applicability of our approach. 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) [13]. 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 α-algorithm, which constructs a Petri net model [24] 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 [1]. 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 [26] 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 [27]. 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 [24] or Event-driven Process Chain (EPC) [16]. 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 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 [1]. 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 [31,33]. SNA can also be augmented by other techniques from social sciences as shown in [6,10].

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 at runtime. 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, consider the two cases discussed in Table 1. For both cases, the schedule events of the three tasks appear in a particular sequence (i.e. task A, task B, task C). 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* [26]. *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 (G) 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 teams, 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. 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 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.

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. Since the process log has only limited information that is relevant to process execution (e.g. performed tasks, originators, etc.), we cannot derive the actual organizational model in an organization. However we can derive a group of originators that has similar characteristics in process execution and the relationship between the mined groups and the tasks. There are two kinds of organizational entities that we can extract from process logs. They are *task-based team* and *case-based team*. A task-based team consists of people who are allowed to execute similar tasks and a case-based team contains people who are involved in the same case. The task-based team can be relevant to the functional departmentalization in which employees possess similar skills and knowledge to perform the tasks. For example, departments in a company such as financial, accounting, marketing, manufacturing departments, etc. belong to this category. And the case-based team is related to the project team in which employees usually possess different skills and work together at the same case. Even though many organizations have functional structures, some organizations such as hospitals, consultancy firms, etc. have a tendency to build teams with individuals having different specialties to achieve certain goals (e.g. surgical operation, consulting project, etc.). In those cases, identifying case-based

teams in an organization is useful for understanding the organization.

In this paper, we explain three kinds of mining methods for task-based team discovery and one method for case-based team discovery. 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, event types, data, etc. Note that we do not take into account this information in this paper. 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 $e = (t, p)$.

For the organizational model, an organizational entity and an entity assignment are defined as follows.

Definition 5.2. (Organizational Entity) Let P be a set of originators. $O \subseteq P$ is an organizational entity (i.e. organizational unit, role, etc.).

Definition 5.3. (Entity Assignment) Let T be a set of tasks and $\breve{O} \subseteq \mathcal{P}(P)$ be a set of organizational entities.[1] $A \in T \times \breve{O}$ is an entity assignment.

An organizational entity defines as a set of originators and represents organizational unit, role, etc. And in this paper an organizational entity refers to either a task-based team or a case-based team. An entity assignment is a pair of a task and an organizational entity and shows the assigned organizational entity for the task. Now the default mining method is defined as follows.

Definition 5.4. (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 $t \in T$, $O_t = \{\pi_p(e) | \exists_{c \in L}\ e \in c \wedge \pi_t(e) = t\}$.

(ii) $A_S = \{(t, O_t) | t \in T\}$.

[1] $\mathcal{P}(X)$ is the power set of X, i.e., $\breve{O}$ is a set of organizational entities (set of sets) and $O \in \breve{O}$ is a set of originators representing an organizational entity, e.g., a role.

O_t stands for the organizational entity for a task t and has originators who performed the task t. For example, in the log shown in Table 2, O_A ={John}, O_B ={Robert, Fred, Mike, Pete}, etc. We can obtain seven entities from the log, since O_t is derived for each task. Note that, the organizational entities do not disjoint, since an originator can perform more than one task. A_S is the set of entity assignments (A) that shows the relationship between organizational entity and tasks. From the example log, we attain $A_S = \{(A, O_A), (B, O_B), .., (H, O_H)\}$.

Default mining is simple and straightforward. It clearly shows the relationship between tasks and originators. However, since the number of organizational entities depends on the number of tasks in a log, we will also have many organizational entities, if we have many tasks in the log. To avoid this problem, we can use *metrics based on joint activities* proposed in [1].

Metrics based on joint activities also 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.

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". After that, we can apply a threshold value to remove less important arcs from the network. Then, each sub network can be mapped onto an organizational entity. Figure 7(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.

The third method is *hierarchical organizational mining*. The two methods mentioned above can derive a flat model where organization hierarchy is excluded from the derived model. However organizational models are usually hierarchical. To derive a hierarchical organizational model, we apply AHC (Agglomerative Hierarchical Clustering) technique [12]. The major steps in the AHC algorithm are as follows.

Definition 5.5. (Agglomerative Hierarchical Clustering) Let P be a set of originators, k be the desired number of final clusters.

(i) begin initialize k, $\hat{k} \leftarrow |P|, D_i \leftarrow \{x_i | x_i \in P\}, i = 1, .., |P|$

(ii) do $\hat{k} \leftarrow \hat{k} - 1$

(iii) find nearest clusters, say, D_i and D_j

(iv) merge D_i and D_j

(v) until $k = \hat{k}$

(vi) return k clusters

(vii) end

The first step is a partition into $|P|$ clusters, each cluster contains one originator. The next is a partition into $|P|-1$ clusters by combining the most nearest two clusters. To calculate the distance between clusters, we use the "profile" (i.e. originator by activity matrix) and the "distance" measures explained in *metrics based on joint activities*. The next is a partition into $|P|-2$ clusters, $|P|-3$ clusters, and so on, until obtaining the k number of clusters. For example, Figure 8 shows an AHC result represented as a dendrogram. This result is based on the running example. Since {Clare, Sue}, {Jane, Mona}, {Pete, Robert}, and {Fred, Mike} have the same profile in Table 2, they are merged into the same group respectively. After that, {Pete, Robert} and {Fred, Mike} are merged into the same group. Then {Pete, Robert, Fred, Mike} and {Jane, Mona} are combined. This way, the dendrogram is constructed. From the dendrogram, we can derive the organizational model in Figure 9. In the figure, the ovals and the pentagons represent originators and organizational entities respectively. Note that from the dendrogram we can also derive flat and disjointed organizational entities by cutting it with a certain value. For example, in Figure 8, by cutting the dendrogram using a cut-off value of 0.5, we obtain two groups such as {John} and {Clare, Sue, Jane, Mona, Pete, Robert, Fred, Mike}. If we use 0.3 as a cut-off value, then {John}, {Clare, Sue}, and {Jane, Mona, Pete, Robert, Fred, Mike} are obtained.

The fourth method is *metrics based on joint cases* proposed in [1]. On the contrary to the previous methods, it focuses on cases and derives case-based team structures. The metrics count how frequent two originators are performing activities on the same case. 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. If originators work together on cases, they will have a stronger bond than originators who rarely work together. There are two ways to derive an organizational model from the network. The first method is to apply a threshold value. A threshold value can be used to erase less important arcs from the network. Then, each sub network can be mapped onto an organizational entity that represents a cased-based team in the organization. The second method is to remove nodes which have large centrality values (e.g. degree, betweenness, etc.) and to map each sub network onto an organizational entity. If an originator works with several teams, the teams are connected to the originator and the first method is not enough to identify different teams. In this case, the originator who works with several teams has a large centrality value and is located as a hub between the teams in the network. Thus, if we

disconnect nodes which have large centrality values from the network, we can obtain several sub networks which refer to the teams. Figure 7(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*, since *John* works with both teams. In the network, the betweenness of *John* is higher than those of others. Thus, we disconnect the node *John* from the rest of the network. Then, three clusters (i.e. {John}, {Sue, Mike, Pete, Jane}, {Clare, Fred, Robert, Mona}) are obtained, and these clusters are relevant to teams shown in Figure 4. This is because the case is assigned to the proper team based on the product type and handled within the team.

Since the metrics focus not on tasks but on cases, the generated model may deviate from the functional structure of an organization. However, it shows cooperation in an organization and the entities in the model are relevant to identify case-based teams in which employees work together at the same case in the organization.

After applying the *metrics based on joint activities*, the *hierarchical organizational mining*, 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 5.6. (Entity assignments) Let L be a log, T be a set of tasks, and P be a set of originators. Moreover, $\breve{O} \subseteq \mathcal{P}(P)$ is the set of organizational entities. Based on this we define the entity assignments A_S as follows: $A_S = \{(t, O) \in T \times \breve{O} \mid \exists_{c \in L} \exists_{e \in c}\ \pi_t(e) = t \wedge \pi_p(e) \in O\}$.

A_S is the set of entity assignments and show the relationships between organizational entities and tasks. 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.

5.2 Social network analysis

To derive social networks from process logs, different kinds of metrics have been developed in [1]. For a better understanding of our approach, we should briefly examine the basic concept. The idea is to monitor 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 origi-

nator 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" β, 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 β^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 10 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 to 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})

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) [33] 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 [1] 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 $W \in R \to \mathbb{R}$ 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 5.7. (Deriving $G_{\breve{O}}$ from G) Let $G = (P, R, W)$ be a social network for originators and $\breve{O} \subseteq \mathcal{P}(P)$ be a set of organizational entities. $G_O = (\breve{O}, R_O, W_O)$ is defined as follows:

(i) $R_O = \{(O_1, O_2) \in \breve{O} \times \breve{O} \mid \exists_{(p_1,p_2)\in(O_1\times O_2)}\ (p_1, p_2) \in R\}$,

(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 $(O_1, O_2) \in R_O$.

$G_O = (\breve{O}, R_O, W_O)$ is a social network for organizational entities, where $\breve{O}$ 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 10, we derive the social network for organizational units (a) and the social network for roles (b) in Figure 11. For example, for Figure 11(a), $\breve{O}$ for organizational units is defined as $\breve{O} = \{O_{CS}, O_{MP}, O_{GPS}\}$, where $O_{CS} = \{John\}, O_{MP} = \{Sue, Mike, Pete, Jane\}, O_{GPS} = \{Clare, Fred, Robert, Mona\}$. 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 11(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 $W_O(O_{CS}, O_{GPS}) = W(John, Clare) + W(John, Fred) + W(John, Robert) = 0.118$. Note that the above definition does not consider the number of originators in an organizational entity. Thus the entity which has more originators

seems to have arcs with larger values. To see the relative information flows, we can use $W_O(O_1, O_2) = \sum_{(p_1,p_2)\in R\cap(O_1\times O_2)} W(p_1, p_2)/|O_1|$, for $(O_1, O_2) \in R_O$.

6 Case study

To validate the approach discussed in this paper, we have performed a case study. It is based on a process log from a municipality in the Netherlands. In the case study, we focus on how the methods proposed in this paper can be applied in a real case and what kinds of organizational information can be derived. This section consists of three parts. First we explain the ProM tool that we used to analyze the process log. Then we explain the context of the case study. After that the case study results are discussed.

6.1 ProM framework

To perform the case study, we used ProM framework. Figure 12 shows a screenshot of ProM. 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 [11]. 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.*

Figure 13 shows the overview of the implementation supporting organizational mining. 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 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.

6.2 Context

Starting point for this case study is the process log of an invoice handling process, which we obtained from the Urban Management Service of a munici-

[2] See http://www.processmining.org for more information and to download ProM and the five plug-ins developed in the context of this paper.

pality of 90,000 citizens, situated in the northern part of the Netherlands. They have implemented their own custom-made workflow system. From the workflow system, we extracted process logs and converted them into the MXML format. We use the log of the handling of invoices in 2005. From the log, we have extracted 570 cases. The number of total events is 3,023. The process consists of 9 activities. The general procedure is that an invoice is scanned and subsequently sent by the workflow management system to the central financial department. A clerk registers the invoice, after that it is sent to the proper local financial office. Depending on the kind of invoice, there are various checks that need to take place: the person responsible for the budget that is used for the purchase must approve (the budget keeper); the fit between the purchase with the supplier's contract (if any) must be established; various managers may be required to authorize the invoice depending on the amount of money involved etc. Eventually, a purchase may be paid by the central financial office. There are 109 employees participating in the process execution. They are distributed over 9 locations in the city, such as council office, town hall, theater, fire station, ice rink, museum, sports park, cleansing service, and swimming pool. They have a hierarchical organizational model that consists of three layers. The first layer has 13 departments. The second and the third layers have 44 and 63 subgroups respectively.

6.3 Mining result

This section describes organizational mining result. First, we focus on organizational models. Several organizational models are derived from the process log with the organizational mining plug-in. For example, Figure 14 shows the screenshot of the AHC mining result. In the figure, the ovals, pentagons and boxes represent originators, organizational entities, and tasks respectively. In the case study, we derived eight organizational models that include a default mining result, two models by metrics based on joint activities (with threshold of 0.7, and 0.9), two models by metrics based on joint cases (with threshold of 0.7, and 0.9), and two models obtained through AHC mining.

To compare organizational models, we introduce the concept of organizational congruence which loosely states that an organization that is matched structurally to the overall mission performs better than others [19]. We assume that each activity has its own distinct purpose and the default mining result can be a good reference model for organizational congruence because an organizational entity in the mined model corresponds to each activity in the log. We calculate the similarity between the default mining result and not only original organizational models but other minded models. In this paper, we used *entropy measure* that is normally used to evaluate the performance of a classification model [32].

Suppose that there are an organizational model (O) and the default mining

result (D). For an organizational entity i from the model O and an organizational entity j from the model D, we can compute the probability that an originator of the entity i belongs to the entity j as $p_{ij} = \frac{m_{ij}}{m_i}$, where m_i is the number of originator in the entity i and m_{ij} is the number of shared originators of the entity i and the entity j. For example, when we compare the location based model and the default mining result, there are five originators in "cleansing service" department in the locational model. Four of them belong to the organizational entity related to "checking invoice" activity. Thus the probability p_{ij} from "cleansing service" department to the organizational entity of the "checking invoice" activity is 4/5. Then the entropy of each organizational entity (i) can be calculated as $e_i = -\sum_{j=1}^{L} p_{ij} log_2 p_{ij}$, where L is the number of organizational entities in the model D. The total entropy for an organizational model O is $e = \sum_{i=1}^{K} \frac{m_i}{m} * e_i$, where K is the number of organizational entities in the model O. If two models are equal, the entropy value is 0. The higher value it has, the bigger difference two models have.

Table 4 shows the analysis result. Note that, if the AHC mining is used, we cut the dendrogram with certain values and use two models in which the organizational entities are disjoint. In the table, among the original organizational models, the third layer model has the lowest value. It is obvious because the third layer model more specifically defines the role of originators. Among the minded models, the models derived by *metrics based on joint activities* have lower values. Since the metrics based on joint cases focus not on organizational congruence, but on cooperation between originators, the mined models have higher values.

To analyze the relationship between performers and departments, we have performed social network analysis. Figure 15 shows the social network generated from the log. We used the handover of work metric to derive it. This shows transfer of works among originators. We applied the threshold value of 0.004 and highlighted the major flows in the network. 34 performers play major roles in the flow: twenty of them belong to council offices, while four performers are affiliated to town hall, The others are fairly distributed across the other locations.

To investigate the information flow between physical locations, the social network for organizational entities is calculated using Definition 5.7. Figure 16 shows the screenshot of the result. The upper part of the figure is the grouping plug-in which calculates the aggregated network from the result of the social network miner. The lower part of the figure shows the resulting network. The council office and the town hall are located in the center of the network. The other eight nodes are connected to those two nodes in a *hub* and *spoke* fashion. The link from "ice rink" to "swimming pool" is not a desired link in the context of process execution.

We can also use existing process mining techniques which focus on process perspective. In the case study, we analyzed the performance of organization.

To do this, we need to massage the log before applying existing process mining techniques. We used the replacement filter what systematically replaces a specific attribute of an event with another attribute. For example, the task ID or originator ID in a log line can be replaced by other elements. Figure 17 shows the analysis result. Note that, based on the case duration time, we have used 362 cases out of 570 cases to focus on the long duration cases. 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. In the analysis, we found that there are some delays from "town hall" to "ice rink" and from "town hall" to "cleansing service"

In the case study, we have derived several organizational mining results from a real-life process log with the methods proposed in this paper. We have investigated differences between the mined organizational models using the concept of organizational congruence, and explained the social network analysis results and the way of using existing mining techniques. Since the aim of this case study is showing the applicability of the proposed methods, we have not focused on the detailed analysis of a particular aspect but presented several mining results.

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 [3]. The concept of process mining was introduced by Cook et al. [9]. They started to mine process models from event logs in the context of software engineering [4]. Agrawal et al. first applied process mining in the context of workflow management. Recently many techniques and tools for process mining have been developed [1–3,22,30,15]. The mainstream of process mining is to discover process models from process logs [2]. 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 [26,14]

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 [1,20]. In our work in [1], 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 [20]. 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 [28,18]. 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 [8]. 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 [5]. 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 [21].

RBAC (Role based access control) [29] is one of the more popular techniques to manage resources in workflow area [7]. 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 [18,28]. Kumar et al. present dynamic work distribution in workflow management systems [18]. 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 [28]. In the adaptive workflow area, researchers focus on the change of organizational models. Klarmann proposes eight categories for structural changes in organizational model [17]. Rinderle and Reichert suggest a method to support organizational model changes considering access rules defined in organizational entities [25].

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 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 explained organizational mining issues in the context of discovery, conformance, and extension. And we addressed three issues (1) *Organizational model mining*, (2) *Social network analysis*, and (3) *Information flows between organizational entities*. With a case study, we have shown how each of these issues 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.

In this paper, we addressed some issues in discovery (i.e. Organizational model mining) and extension (i.e. social network analysis and information flows between organizational entities). As future work, conformance issues should be addressed. To evaluate the organizational model mining results, conformance test methods should be developed. The development of new mining methods is also essential. For example, we can apply non-disjoint clustering methods to reflect an organization in which originators play multiple roles.

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

[1] W.M.P. van der Aalst, H.A. Reijers, and M. Song. Discovering Social Networks from Event Logs. *Computer Supported Cooperative work*, 14(6):549–593, 2005.

[2] 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.

[3] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.

[4] 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.

[5] E. Bertino, E. Ferrari, and V. Alturi. The specification and enforcement of authorization constraints in WFMS. *ACM Transactions on Information and System Security*, 2(1):65–104, 1999.

[6] S. Borgatti and R. Cross. A Social Network View of Organizational Learning: Relational and Structural Dimensions of 'Know Who'. *Management Science*, 49:432–445, 2003.

[7] R.A. Botha and J.H.P. Eloff. A framework for access control in workflow systems. *Information Management and Computer Security*, 9(3):126–133, 2001.

[8] C. Bussler and S. Jablonski. Policy resolution for workflow management systems. *System Sciences, 1995. Vol. IV. Proceedings of the Twenty-Eighth Hawaii International Conference on*, 4, 1995.

[9] 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.

[10] R. Cross, J. Liedtka, and L. Weiss. A practical guide to social networks. *Harvard Business Review*, 83(3):124–132, 2005.

[11] B.F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, and W.M.P. van der Aalst. The ProM framework: A new era in process mining tool support. In G. Ciardo and P. Darondeau, editors, *26th International Conference on Applications and Theory of Petri Nets (ICATPN 2005)*. Springer, 2005.

[12] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[13] M. Dumas, W.M.P. van der Aalst, and A.H.M. ter Hofstede. *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley & Sons, 2005.

[14] G. Greco, A. Guzzo, and L. Pontieri. Discovering Expressive Process Models by Clustering Log Traces. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1010–1027, 2006.

[15] S.-Y. Hwang and W.-S. Yang. On the discovery of process models from their instances. *Decision Support Systems*, 34(1):41–57, 2002.

[16] G. Keller and T. Teufel. *SAP R/3 Process Oriented Implementation*. Addison-Wesley, Reading MA, 1998.

[17] J. Klarmann. A Comprehensive Support for Changes in Organizational Models of Workflow Management Systems. In *Proceedings of the 4th International Conference on Information Systems Modelling (ISM?1)*, pages 165–172, 2001.

[18] A. Kumar, W.M.P. van der Aalst, and H.M.W. Verbeek. Dynamic Work Distribution in Workflow Management Systems: How to Balance Quality and Performance? *Journal of Management Information Systems*, 18(3):157–193, 2002.

[19] G.M. Levchuk, D.L. Kleinman, S. Ruan, and K.R. Pattipati. Congruence of Human Organizations and Missions: Theory versus Data. *Proceedings of the 8th International Command and Control Research Symposium*, Washington, DC, June 2003. University of Muenster.

[20] L.T. Ly, S. Rinderle, P. Dadam, and M. Reichert. Mining Staff Assignment Rules from Event-Based Data. In C. Bussler and A. Haller, editors, *Business Process Management 2005 Workshops*, volume 3812 of *Lecture Notes in Computer Science*, pages 177–190. Springer-Verlag, Berlin, 2006.

[21] M. zur Mühlen. Organizational Management in Workflow Applications–Issues and Perspectives. *Information Technology and Management*, 5(3):271–291, 2004.

[22] 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.

[23] M. Netjes, H.A. Reijers, and W.M.P. van der Aalst. Supporting the BPM Lifecycle with FileNet. In T. Latour and M. Petit, editors, *Proceedings of the EMMSAD Workshop at the 18th International Conference on Advanced Information Systems Engineering (CAiSE'06)*, pages 497–508. Namur University Press, 2006.

[24] 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.

[25] S. Rinderle and M. Reichert. On the Controlled Evolution of Access Rules in Cooperative Information Systems. In R. Meersman and Z. Tari et al., editors, *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, OTM Confederated International Conferences*, volume 3812 of *LNCS*, pages 238–255. Springer-Verlag, Berlin, 2005.

[26] A. Rozinat and W.M.P. van der Aalst. Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models. In C. Bussler et al., editor, *BPM 2005 Workshops (Workshop on Business Process Intelligence)*, volume 3812 of *Lecture Notes in Computer Science*, pages 163–176. Springer-Verlag, Berlin, 2006.

[27] A. Rozinat and W.M.P. van der Aalst. Decision Mining in ProM. In S. Dustdar, J.L. Faideiro, and A. Sheth, editors, *International Conference on Business Process Management (BPM 2006)*, volume 4102 of *Lecture Notes in Computer Science*, pages 420–425. Springer-Verlag, Berlin, 2006.

[28] N. Russell, W.M.P.van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Workflow Resource Patterns: Identification, Representation and Tool Support. In O. Pastor and J. Falcao e Cunha, editors, *Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05)*, volume 3520 of *Lecture Notes in Computer Science*, pages 216?-232. Springer-Verlag, Berlin, 2005.

[29] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-Based Access Control Models. *IEEE Computer*, 29(2):38–47, 1996.

[30] M. Sayal, F. Casati, U. Dayal, and M.C. Shan. Business Process Cockpit. In *Proceedings of 28th International Conference on Very Large Data Bases (VLDB'02)*, pages 880–883. Morgan Kaufmann, 2002.

[31] J. Scott. *Social Network Analysis*. Sage, Newbury Park CA, 1992.

[32] P-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, Boston, MA, USA, 2005.

[33] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, 1994.

Case ID	log events
1	..(A,Jane,‘schedule’)(B,Jane,‘schedule’)(C,Jane,‘schedule’) ..(A,Jane,‘start’)(B,Jane,‘start’)(C,Jane,‘start’)
2	..(A,Mike,‘schedule’)(B,Mike,‘schedule’)(C,Mike,‘schedule’) ..(C,Mike,‘start’)(B,Mike,‘start’)(A,Mike,‘start’)

Table 1
Fragment of a process log containing only two cases

Case ID	log events
1	(A,John),(B,Mike),(C,John),(D,Sue), (E,Pete),(F,Jane),(H,Sue)
2	(A,John),(B,Fred),(C,John),(D,Clare), (E,Robert),(F,Mona),(H,Clare)
3	(A,John),(C,John),(B,Pete),(D,Sue), (E,Mike),(F,Jane),(H,Sue)
4	(A,John),(C,John),(B,Fred),(D,Clare),(G,Clare),(H,Clare)
5	(A,John),(C,John),(B,Robert),(D,Clare), (E,Fred),(F,Mona),(H,Clare)
6	(A,John),(B,Mike),(C,John),(D,Sue),(G,Sue),(H,Sue)

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)

originator	act A	act B	act C	act D	act E	act F	act G	act H
John	6	0	6	0	0	0	0	0
Sue	0	0	0	2	0	0	1	3
Mike	0	2	0	0	1	0	0	0
Pete	0	1	0	0	1	0	0	0
Jane	0	0	0	0	0	2	0	0
Clare	0	0	0	2	0	0	1	3
Fred	0	2	0	0	1	0	0	0
Robert	0	1	0	0	1	0	0	0
Mona	0	0	0	0	0	2	0	0

Table 3
The originator by activity matrix

models	# of entities	# of originators	Entropy value
Location	9	109	1.917
1st layer	13	109	1.691
2nd layer	44	109	0.800
3rd layer	63	109	0.520
Default mining	9	109	0.0
MJA(0.7)	6	109	0.435
MJA(0.9)	10	109	0.297
AHC(6)	6	109	2.072
AHC(8)	8	109	2.024
MJC(0.7)	6	109	2.371
MJC(0.9)	12	109	2.314

Table 4
Entropy values of the organizational models

List of Figures

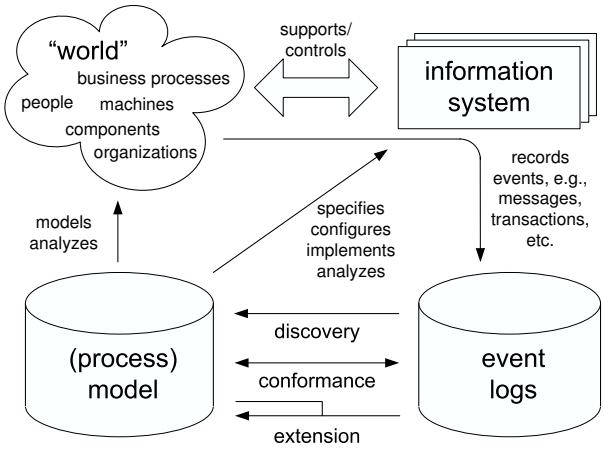


Fig. 1. The process mining overview

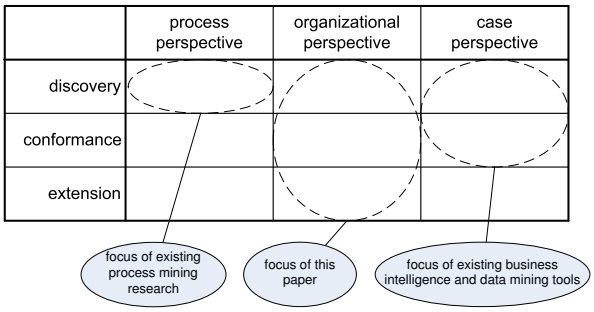


Fig. 2. Diagram showing the focus of this paper using two dimensions: (1) type of mining (discovery, conformance, and extension) and (2) perspective (process, organization, and case)

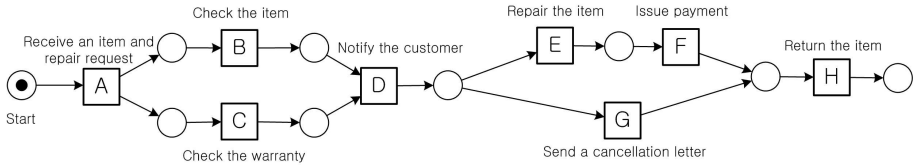


Fig. 3. The example process model

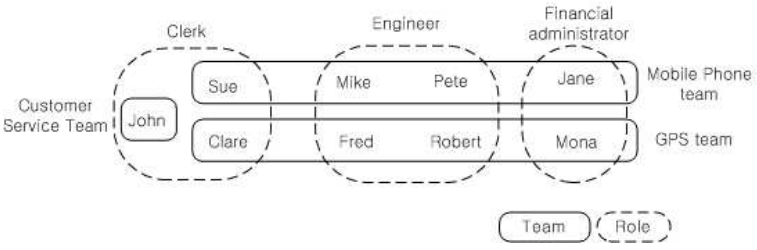


Fig. 4. The example organizational model

```
<?xml version="1.0" encoding="UTF-8"?>
<WorkflowLog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="http://www.is.tm.tue.nl/research/processmining/
                               WorkflowLog.xsd">
    <Source program="Eistream"/>
      <Process id="process" description="none">
            <ProcessInstance id="01" description="none">
            <AuditTrailEntry>
            <WorkflowModelElement>Receive_repair_request</WorkflowModelElement>
                <EventType>complete</EventType>
                <Originator>John</Originator>
                <Timestamp>2004-09-22T15:13:00+01:00</Timestamp>
            </AuditTrailEntry>
            <AuditTrailEntry>
            <WorkflowModelElement>Preliminary_check</WorkflowModelElement>
                <EventType>complete</EventType>
                <Originator>Mike</Originator>
                <Timestamp>2004-09-23T12:08:01+01:00</Timestamp>
            </AuditTrailEntry>
            ...
```

Fig. 5. Fragment of the example log in MXML format

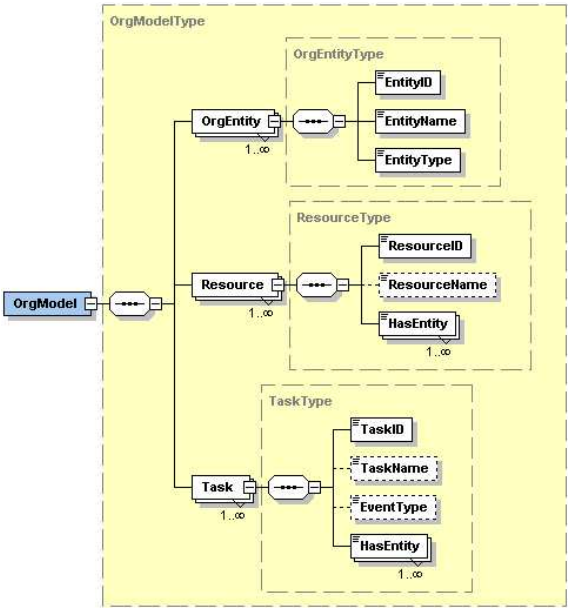
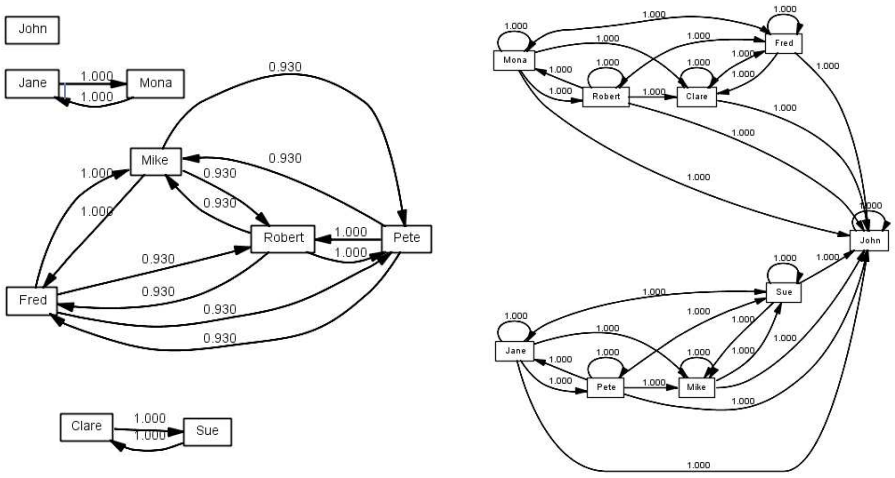


Fig. 6. Organizational Model Markup Language



(a) doing similar task (b) working together

Fig. 7. The organizational model mining result

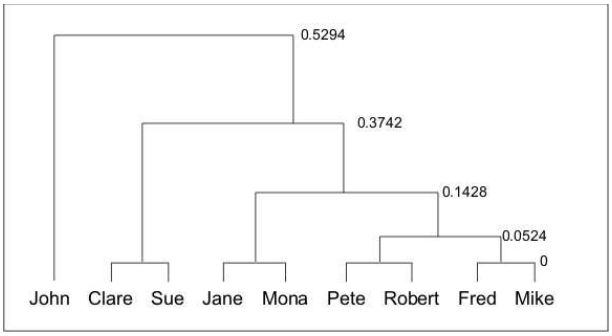


Fig. 8. A dendrogram of the example

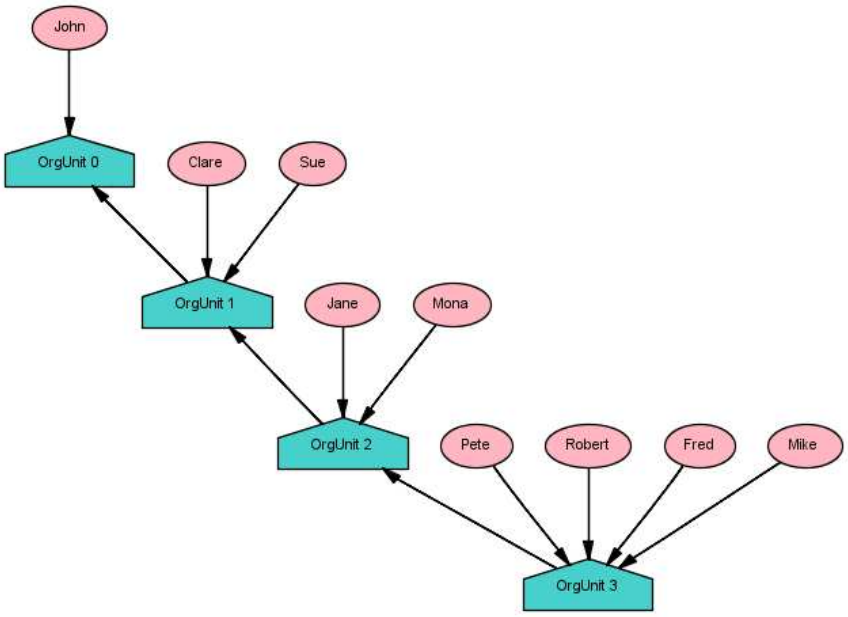


Fig. 9. A hierarchical organizational model

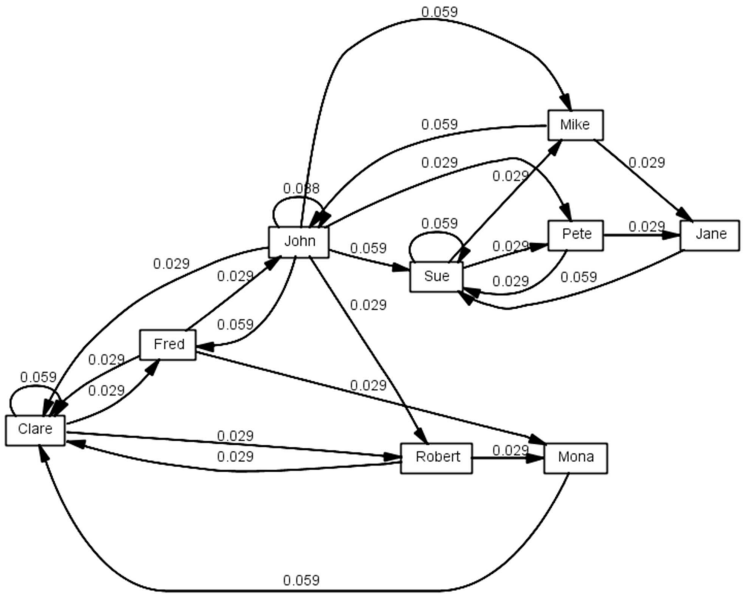


Fig. 10. The social network

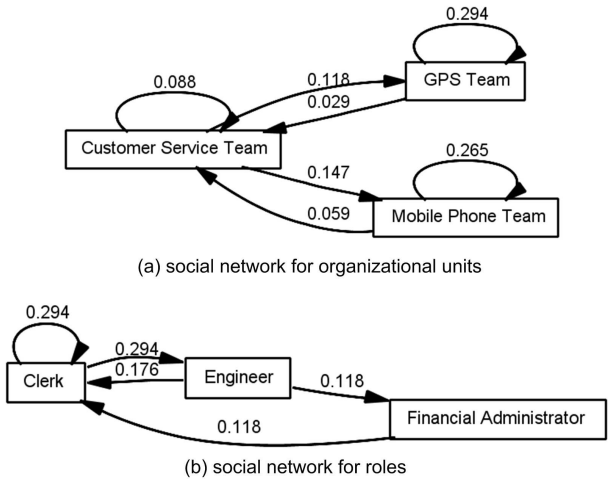


Fig. 11. Social networks for organizational entities

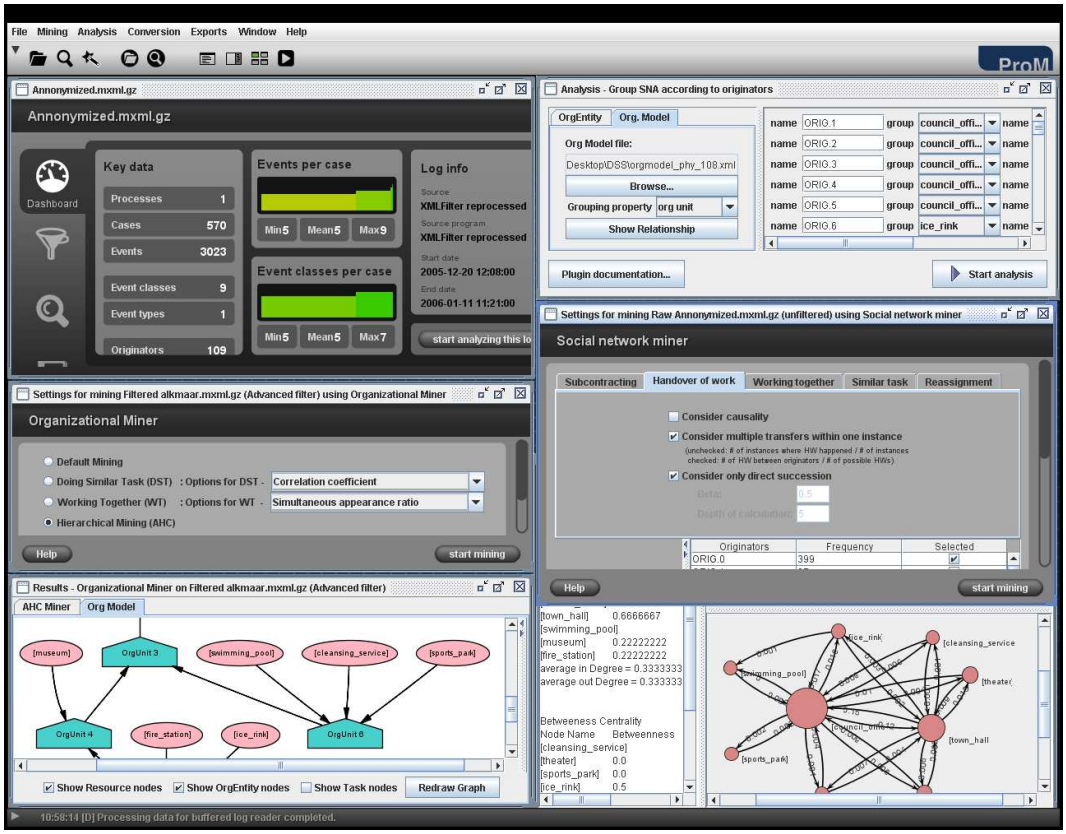


Fig. 12. ProM screenshot showing organizational mining plug-ins

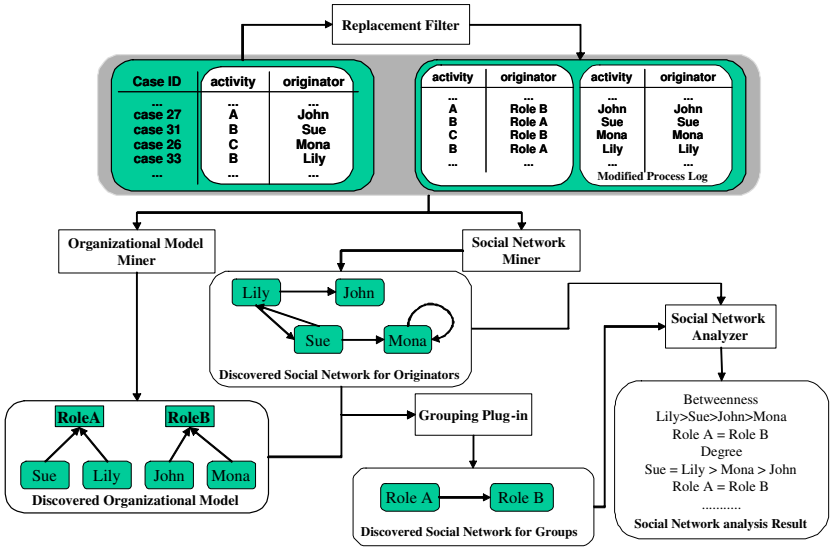


Fig. 13. Overview of the five plug-ins and their relationships

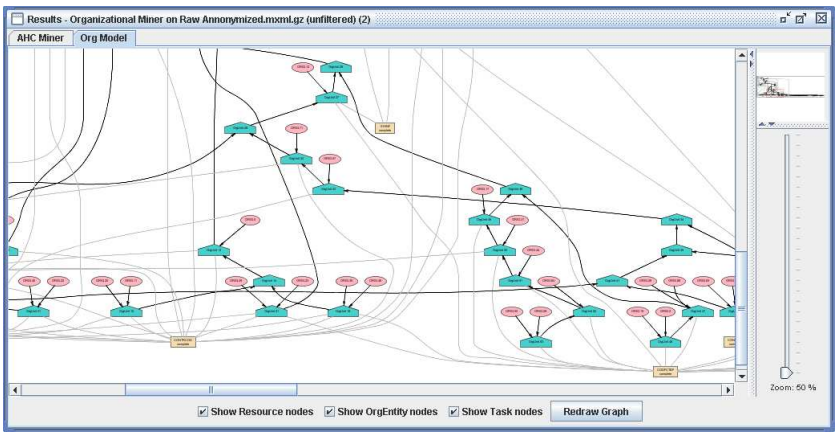


Fig. 14. Organizational mining result

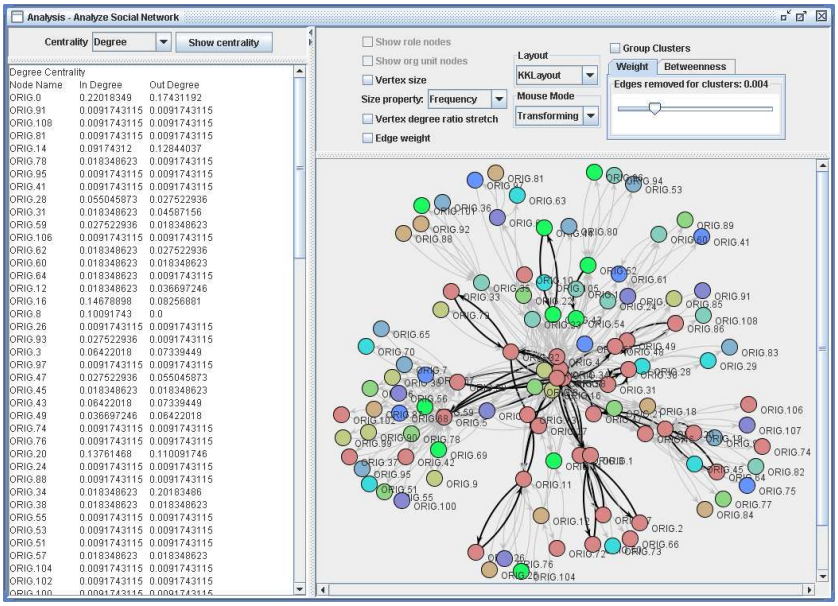


Fig. 15. Social network (handover of work metric)

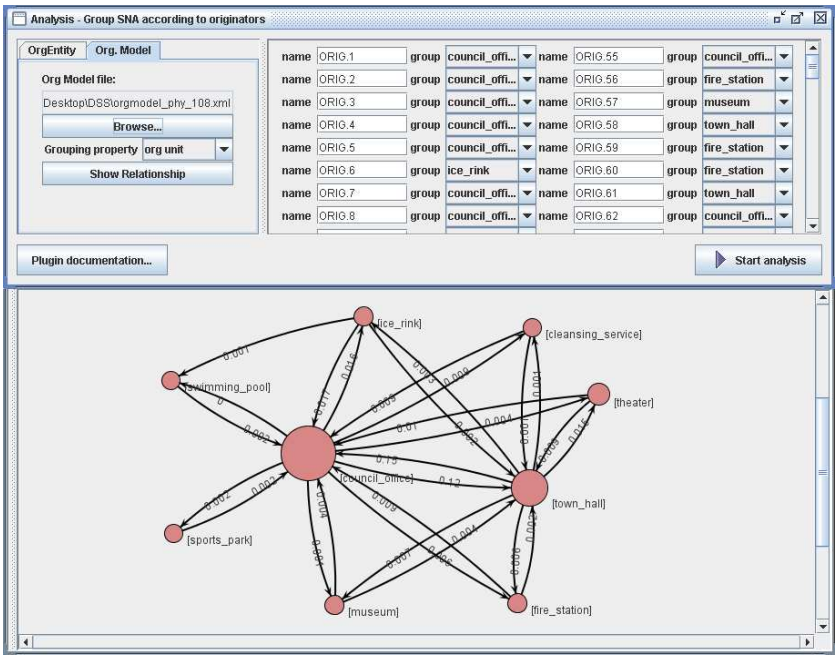


Fig. 16. Information flow between groups

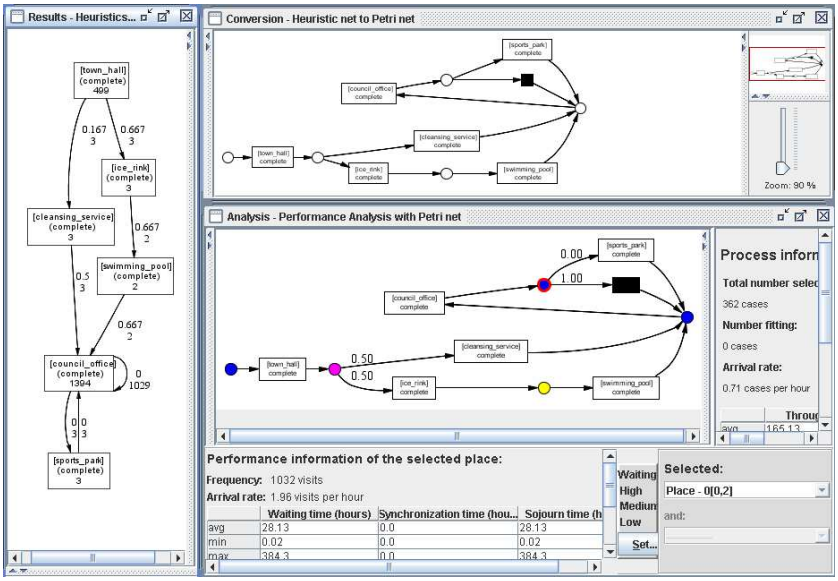


Fig. 17. Organizational performance analysis