J Intell Inf Syst (2018) 50:165–193
DOI 10.1007/s10844-017-0451-x

Towards mining the organizational structure of a dynamic event scenario

Annalisa Appice[1,2,3]

Received: 7 July 2016 / Revised: 12 January 2017 / Accepted: 15 February 2017 /
Published online: 23 February 2017

Abstract The increasing volume and value of data is an important enabler for data science. In this study, we consider the event data, i.e. information on things that happen in organizations, machines, systems and people's lives. Each event refers to a well-defined activity in a certain business process execution, the resource (i.e. person or device) executing or initiating the activity, the timestamp of the event, as well as to various data elements recorded with the event (e.g. the geo-location of an activity). Process mining aims to analyze event data, in order to mine knowledge that can contribute to improving a business process behavior. In particular, the focus of this study is on organizational mining, that is a sub-field of process mining that aims at understanding the life cycle of a dynamic organizational structure (i.e. a configuration of organization units) and the interactions among co-workers (resources) arising from the analysis of real-world event logs. The innovative contribution of this study is that the organizational mining goal is here achieved by combining concepts from process mining, stream mining and social network analysis. This combination is an original contribution of this study, not still explored in organizational mining field. In an assessment, benchmark event data are explored, in order to understand how the presented solution allows us to identify the life cycle a dynamic organizational structure.

Keywords Process mining · Organizational mining · Internet of events · Social network analysis

✉ Annalisa Appice
annalisa.appice@uniba.it

[1] Dipartimento di Informatica, Università degli Studi di Bari “Aldo Moro”, via Orabona 4, 70125 Bari, Italy

[2] CINI - Consorzio Interuniversitario Nazionale per l'Informatica, Bari, Italy

[3] CILA - Centro Interdipartimentale di Logica e Applicazioni, Bari, Italy

1 Introduction

The amount of data in our world is exploding, due to the spectacular growth of the digital universe in only a few years (Hilbert and Lopez 2011). Data are collected about any type of event, at any time and any place. Event data are the most important source of information (van der Aalst 2016). They may concern "life events", "machine events", or both. They are associated with the execution of a certain business process and registered in a wide variety of data sources (i.e. databases, flat files, message logs, transaction logs, ERP systems and document management systems). Each event refers to a specific activity (i.e. a well-defined step in a business process). It may be recorded with additional information such as the resource (i.e. person or device) executing or initiating the activity, the timestamp of the event, or the data elements recorded with the event (e.g. the place where the activity is performed).

However, the ultimate goal is not to store more events, but to turn event information into real values (van der Aalst 2014). Process mining is an integral part of data science fueled by the omnipresence of event data. In the last decade, it has inspired the development of a large plethora of techniques, in order to explore event information in a meaningful way and learn knowledge related to the behavior of people, organizations, machines and systems (van der Aalst 2016). In these techniques, the wealth of information recorded with events is related to several perspectives (Song et al. 2009; van der Aalst 2011; Appice and Malerba 2015), such as the control-flow perspective (ordering of activities), the execution perspective (time performances and frequency of activities) and the organizational perspective (organization of resources). However, the majority of current process mining developments (see van der Aalst (2011, 2016) for surveys) focuses mainly on control-flow discovery, while no comparable effort has been made to facilitate the understanding of complex organizations arising from the analysis of real-world event logs.

Going against the trend of process mining research, the focus of this paper is on organizational mining. It is a sub-field of process mining that analyzes event data along their organization perspective (van der Aalst 2011). Its primary goals are: (1) identifying meaningful relationships between resources and (2) deriving an interpretable model of the organizational structure of a business process. In particular, addressing organizational structure discovery problems corresponds to identifying the organization units of a business process and classifying resources according to the units they participate in.

Recently, the main stream of research in organizational mining has contributed to the formulation of various event-based metrics (e.g. handover of work, subcontracting, working together and joint activities), in order to derive meaningful relationships between resources and represent such resources as a sociogram (social network) (van der Aalst and Song 2004). In addition, it has adopted a clustering formulation for the organizational structure discovery problems and applied traditional clustering techniques, in order to group "similar" resources into clusters (organization units) (van der Aalst et al. 2005; Song and van der Aalst 2008; Ferreira and Alves 2012). This means that, although a part of research in organization mining has begun to highlight that social network analysis may have many chances to succeed as a comprehensive approach towards organizational mining (Song and van der Aalst 2008), the organizational structure discovery is still performed without considering the specific discovery algorithms (e.g. community detection algorithms), which may account for the network property directly.

The network property, as is exhibited by several social networks, is the property of containing community structures (Palla et al. 2009). In particular, social networks naturally divide into groups of nodes with denser connections inside each group and fewer connections crossing between groups. Consequently, community detection in a social network is the gathering of its nodes into groups in such a way that the nodes in each group are densely connected inside and sparser outside.

We observe that clustering and community detection algorithms are closely related to each other due to their nature. Both share the same objective of partitioning nodes into groups. However, clustering algorithms group nodes into clusters according to their similarity, while community detection algorithms are often modularity-based algorithms that analyze the structure of the similarity matrix. Modularity (Clauset et al. 2004) is a quality function that allows us to compare different partitions of a given network, rewarding those partitions that are more cohesive internally than externally. Therefore, community discovery allows us to display the whole network organization at a compact and more understandable level, where each community can represent a functional group or an entity in the system. At this level, community structure provides meaningful insights into the network organizational principles (Nguyen et al. 2014). These considerations have mainly inspired this research study, that intends to assess the viability of community discovery as a more powerful tool than clustering in the field of organizational mining.

As an additional contribution in this study, we counterpose the discovery of overlapping organization units to the traditional discovery of disjoint units. While all existing studies have focused on the discovery of disjoint organization units, we have decided to determine organizational structures that may fit resources, which may cover a different role in various organization units simultaneously. This is the case of a software engineer who may participate in various organization units involved in the different phases of a software life cycle, e.g. design the program and build the system. Based upon these considerations, we point out that a flexible model with overlapping organization units can provide interesting insights, otherwise missed, of complex organizations, where roles may not be so strictly separated.

As a final contribution, we outdo the state-of-the-art in organizational mining, which only considers event data provided as a static, finite dataset and is able to discover a single organizational structure configuration from this dataset. We consider that this static view of event data is restrictive, as it neglects the actual scenario, where new events can be continuously generated from (new) running executions of a business process. On the other hand, adopting an event stream (van Zelst et al. 2015) perspective leads to expecting resources organized in dynamic social networks, which may change over time (new resources are active, old resources are inactive and resources change role). This means that a new scope is emerging in organizational mining. It moves the attention from the discovery of an organizational structure to the discovery and understanding of the life cycle of the dynamic organization of a business process. This scope is also consistent with a recent research trend (Greene et al. 2010; Spiliopoulou 2011; Oliveira et al. 2014; Dhouioui and Akaichi 2014) in social network analysis, which has started to consider the problem of tracking the progress of clusters/communities over time in a dynamic social network.

In short, this study actually advances organizational mining research, as it seminally applies a (overlapping) community detection approach, in order to discover and understand the dynamic organizational structure of a certain business process along the time line. This organizational analysis of a business process promises a wide range of applications such as resource assignment and schedule, skill management and organizational design. Procedurally, the presented approach operates in two phases. The online phase decomposes an event stream into consecutive time-based windows. It analyzes events, window-by-window, as they arrive from the stream and constructs the social network of the windowed resources. The organizational structure of each social network is derived by resorting to a modularity-based algorithm that discovers overlapping communities in the network. The offline phase is repeatable. It allows us to analyze the series of organizational structures (sets of (overlapping) communities) discovered at consecutive time windows, in order to identify the organizational changes in any dynamic business organization . In particular, it derives the life cycle of every organization unit (resource community) expressed as a series of significant events (e.g. birth, death, merging, splitting, expansion and contraction) along the time line.

The paper is organized as follows. Section 2 describes the background of this study in the field of organizational mining. Section 3 introduces basic concepts, while Section 4 illustrates the organization discovery approach. Section 5 describes two business process applications, the processed events, the experimental setup and discusses the relevant results. Finally, Section 6 draws some conclusions and outlines some future work.

2 Background

The seminal study in organizational mining is authored by van der Aalst and Song (2004). This study has introduced various metrics to identify relationships between resources recorded in an event log. The proposed metrics are based on (possible) causality (e.g. handover of work and subcontracting), joint executions (e.g. working together), joint activities (e.g. how frequently two resources perform the same activity) and special event types (e.g. reassignment metrics). Following this research direction, van der Aalst et al. (2005) have promoted the use of sociography, in order to represent the interpersonal relationships of any organizational structure. They have suggested that a resource sociogram (i.e. a social network) can be constructed from event data. The nodes of this social network correspond to the organizational entities. More specifically, van der Aalst (2011) have noted that often, although not always, there is a one-to-one correspondence between the resources found in a log of event data and the organization entities (i.e. nodes which correspond to people). The arcs correspond to the relationships between such organization entities (or equivalently organization resources). They may be associated with weights, which indicate the importance of the relationship between the connected entities. The higher the weight, the higher the importance of the relationships. Once the social network is constructed, social network metrics like centrality, closeness or betweenness can be computed, in order to characterize the role of individual nodes/resources in a sociogram (van der Aalst and Song 2004; Song and van der Aalst 2008).

The idea of applying sociography to organizational mining has attracted few followers in recent years. For example, Saravanan and Rama Sree (2011) derived the social network representation of the worker work load process in an emerald dyeing unit. They constructed various social networks by using metrics based on handover of work, working together and joint activities, respectively. Similarly, Sunindyo et al. (2010) constructed social networks in the context of production automation system engineering, by looking for relationships between machines that do similar tasks, as well as the relationships between machines that work together to create some products. In their study, the relationships of the constructed social networks are based on metrics like joint activities/doing similar tasks and joint traces/working together.

While paving the way to sociography in organizational mining, van der Aalst and Song (2004) and van der Aalst et al. (2005), as well as Song and van der Aalst (2008) also applied traditional clustering techniques (e.g. k-means clustering and agglomerative hierarchical clustering), in order to group similar resources in the same cluster. Proceeding in this research direction in the clustering field, Song et al. (2009) investigated the use of resource information, in order to derive the organization profile of the execution of a certain business process. They mapped each business process execution into a vector of resource-based features, one feature for every resource. Each resource-feature denotes the number of times the resource participates in an event of the execution. Finally, they applied either a clustering technique (e.g. k-means clustering, quality threshold clustering or agglomerative hierarchical clustering) or a Self Organization Map, in order to identify groups of business process executions sharing a similar organization. Procedurally, they measured the distance between the organization profile of two traces by computing traditional distance measures (e.g. Euclidean distance, Jaccard distance, Hamming distance). We note that, by remaining under the umbrella of clustering research, it is still possible to derive a social network representation of the business process organization with a resource connected to every other resource. In this formulation, the importance (weight) of a relationship can be, for example, expressed as the inverse of the distance computed between the connected resources, so that the lower (higher) the distance, the higher (lower) the importance.

More recently, Ferreira and Alves (2012) have highlighted the importance of constructing clusters of highly “interconnected” resources, namely communities, as an effective means to identify the organization units of a business process. They have also observed that, depending on the metric adopted to construct the social network, the discovery of a group of densely interconnected resources, which exhibits few connections to resources outside the group, may take on a different meaning. For example, a joint activity-based metric leads to the discovery of communities that group resources playing a similar role in the business process, while a joint trace-based metric can promote the discovery of communities that represent resources working together in a team. On the other hand, despite the intent of arousing organizational mining interest in community detection research, Ferreira and Alves (2012) still used a traditional clustering procedure, namely agglomerative hierarchical clustering, to identify organization units of an event log. They detected similar nodes, while relegating the use of the social network theory to a post-processing phase. In particular, from the social network research, they adopted the idea of performing the inspection of a resource network structure by resorting to the analysis of the so-called network modularity.

Modularity (Clauset et al. 2004) is one measure of the structure of a network. It was designed to measure the strength of division of a network into modules (also called groups, clusters or communities). Networks with high modularity have dense connections between the nodes within modules, but sparse connections between nodes in different modules. Modularity has often been used in optimization methods for detecting community structure in several kind of networks (Clauset et al. 2004). However, Ferreira and Alves (2012) considered modularity only as a way to estimate the goodness of a clustering configuration previously discovered. They computed the modularity of the network according to the clustering configuration detected at each level of the agglomerative hierarchical clustering. In this way, they were able to output the cluster configuration that achieves the maximum modularity.

Therefore, the current state of research is that community discovery has been indicated as a potential organizational mining tool, but the mainstream of research is still focused on the investigation of traditional clustering techniques in this field. In fact, organization units are mainly discovered based on the analysis of a distance or similarity matrix overlooking that the network data structure may lead to the application of specific community discovery algorithms, which use the graph property directly (Lei and Huan 2010). To the best of our knowledge, our preliminary work (Appice et al. 2016) is the seminal study applying a specific community detection algorithm to the discovery of organization units, also paying attention to aspects like discovering overlapping organization units, as well as tracking evolutions of organization units in dynamic scenarios. The actual intent of this research is to foster the use of community detection as a comprehensive tool to perform organizational mining in dynamic resource scenarios. In fact, this study does not advance research in community detection by designing novel algorithms, but defines the viability of community detection in the context of organizational mining.

3 Basics

In this section, we introduce the basic concepts of this study, namely the event streams and social networks.

3.1 Event stream

An *event* $\epsilon(\mathcal{P}id, a, r, t)$ describes something happened during the execution of a certain business process $\mathcal{P}$ (van der Aalst 2011). *It* is characterized by a set of mandatory characteristics, that is, the event corresponds to an activity a, is triggered by a resource r and has a timestamp t, that represents date and time of occurrence. An event can be linked to a particular execution $\mathcal{P}id$ of the logged business process and is globally unique.

Focusing on the temporal dimension (van Zelst et al. 2015; Appice et al. 2016) of the event data, events linked to (various) running executions of a specific business process can be ordered by the timestamp independently of the execution producing them. In this way, a continuous, unbounded *event stream* (see the example reported in Table 1) is populated

Table 1 A fragment of an event stream

BusinessProcessID	Activity	Timestamp	Resource
1	Register request (R)	2010-12-30:11:02	Pete
2	Register request (R)	2010-12-30:11:32	Mike
2	Check ticket (CT)	2010-12-30:12:12	Mike
2	Check ticket (CT)	2010-12-30:12:14	Sue
2	Examine causally (EC)	2010-12-30:14:16	Pete
3	Register request (R)	2010-12-30:14:32	Pete
1	Examine thoroughly (ET)	2010-12-31:10:06	Sue
2	Check ticket (CT)	2010-12-31:15:31	Pete
2	Decide (D)	2011-01-05:11:22	Sara
1	Check ticket (CT)	2011-01-05:15:12	Mike
1	Decide (D)	2011-01-06:11:18	Sara
1	Reject request (RR)	2011-01-07:14:24	Pete
2	Pay compensation (PC)	2011-01-08:12:05	Ellen
...	...	...	...

Each event is linked to a specific execution of a certain business process. It corresponds to an activity, has a timestamp and is triggered by a resource. It is organized as an unbounded series of time-ordered events. It is virtually divided into consecutive windows, which are 24 hours long, according to the time-based window model with $\Delta(t) = 24h$

(Appice et al. 2016). Formally, an event stream $\mathcal{S}$ is an ordered, unbounded sequence of timestamped events:

$$\mathcal{S} = \epsilon(\mathcal{P}id_1, a_1, r_1, t_1), \epsilon(\mathcal{P}id_2, a_2, r_2, t_2), \ldots, \epsilon(\mathcal{P}id_i, a_i, r_i, t_i), \ldots, \quad (1)$$

where events of the stream arrive sequentially, at consecutive time points (i.e. $t_i \leq t_{i+1}$), from the running executions of business process $\mathcal{P}$. We observe that the running executions feeding an event stream may change over time, since old executions can be completed, while new executions can be started at a certain time point.

An event stream, like any data stream, is unbounded in length. It is impractical to query all the data of a stream. A window model approach is a stream approach commonly used to query open-ended data. Instead of computing an answer over the whole data stream, the query (or operator) is computed, maybe several times, over a finite window of events. Coherently with the considerations formulated by Gaber et al. (2005), computing a new answer as a new event window is collected can allow us to represent the recent organization of a business process by adapting naturally this representation to a possible change in occurring events over time.

Several window models (e.g. time-based and count-based windows, sliding windows, landmark windows and titled windows) are defined in the literature. We consider the *time-based window model* (Gaber et al. 2005), which decomposes an event stream into consecutive (non overlapping) windows of a fixed temporal size (see the example reported in Table 1). When a window is completed, it is queried. The answer is stored in a synopsis structure for the mining phase, while the windowed events are discarded. The synopsis is constructed in a way which is friendly to the needs of the particular problem being solved by the mining phase (in this case the organizational structure discovery). In this study the synopsis used to store the queried data window is a graph structure representing resources as a social network (see details in Section 3.2).

Formally, let $\Delta(T)$ be the window temporal size of the model, then a time-based window model decomposes a stream $\mathcal{S}$ into non overlapping windows:

$$\mathcal{S}(\Delta(T)) = \begin{cases} t & \rightarrow t + \Delta(T), \\ t + \Delta(T) & \rightarrow t + 2\Delta(T), \\ \ldots, & \\ t + (j-1)\Delta(T) & \rightarrow t + j\Delta(T), \\ \ldots & \end{cases} \quad (2)$$

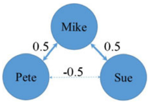
so that every window $t + (j-1)\Delta(T) \rightarrow t + j\Delta(T)$ selects stream events $\epsilon(\mathcal{P}id_i, a_i, t_i, r_i) \in \mathcal{S}$, acquired at each time point $t_i \in [t + (j-1)\Delta(T), t + j\Delta(T)[$.

3.2 Social network

The query operator is a resource *social network* constructor transforming an event window into a resource social network synopsis. The resource social network is a graph $(\mathcal{N}, \mathcal{A})$. Node set $\mathcal{N}$ is a set of nodes, where each node represents a resource triggering one or more events in the event window. Arc set $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N} \times \Re^+$ is a set of arcs, where each arc expresses a relationship between connected resources. Every relationship is associated with a real-valued positive weight that expresses the importance of the relationship: the higher the weight, the more important the relationship (arc).

Resource	Register Request	Check ticket	Examine Causally
Mike	1	1	0
Pete	2	1	1
Sue	0	1	0

(a) Activity profile



(b) Activity-based social network

Fig. 1 The activity profile of Pete, Mike and Sue (Fig. 1a) and their social network (Fig. 1b). These resources are active in the data window [2010-12-30:0:00, 2010-12-30:24:00] of the stream reported in Table 1. The social network extracted from this data window contains three nodes associated with Pete, Mike and Sue, as well as weighted arcs connecting these resources. The weight associated with every arc is the Pearson correlation coefficient computed between the activity profiles of the edged nodes

Procedurally, we compute a metric based on the joint activities, in order to determine interesting relationships between resources.[1]The selected metric focuses on the activities that every resource performs. As described by Song and van der Aalst (2008), we assume that resources doing similar things are more closely connected (and then have the same role) than resources doing completely different tasks (i.e. covering different roles in the business process). In particular, each resource is associated with an *activity profile*, that is, a vector of attributes, where every attribute represents a distinct activity. This profile describes how frequently a resource performs every specific activity (Song et al. 2009) during the window time. An example of activity profiles is reported in Fig. 1a.

For each pair of resources, the joint activity degree is measured over this pair of resources. If this degree is significant, an arc connects these resources in the social network, while the joint activity degree is the weight assigned to the arc. We apply Pearson's correlation coefficient to quantify this degree/weight.[2] It is computed as the covariance of the two resources divided by the product of their standard deviations. For each pair of resources r_i and r_j, we compute Pearson's correlation coefficient between the activity profiles associated with r_i and r_j, respectively. Formally,

$$w(r_i, r_j) = \frac{\sum\limits_{A} r_i(A) r_j(A) - n\overline{r_i}\,\overline{r_j}}{\sqrt{\sum\limits_{A} r_i(A)^2 - n\overline{r_i}^2}\sqrt{\sum\limits_{A} r_j(A)^2 - n\overline{r_j}^2}}, \tag{3}$$

where A denotes an activity associated with the compared resource profiles, $r_i(A)$ $(r_j(A))$ is the number of times A is performed by r_i (r_j) in the data window, $\overline{r}_i$ $(\overline{r}_j)$ is the average $\overline{r}_i = \frac{1}{n}\sum\limits_{A} r_i(A)$ $\left(\overline{r}_j = \frac{1}{n}\sum\limits_{A} r_j(A)\right)$ and n is the number of activities in the resource profile.

The computation of Pearson's correlation coefficient is motivated here by various considerations. This coefficient is invariant to both scale and location change. It ranges from -1 to 1. A value of 1 implies that a linear equation describes the relationship between compared variables perfectly, with all data points lying on a line for which one variable increases as

[1] Alternative metrics, e.g. handover of work or subcontracting (Song and van der Aalst 2008), can be equally considered to determine relationships between resources, without changing the general contribution of the theory described in this study.

[2] Song and van der Aalst (2008) describe the use of several distance/similarity measures (e.g. Minkowski distance, Hamming distance, Pearson's correlation coefficient), in order to quantify the "weight" associated with the arcs of a resource social network.

the other increases. A value of -1 implies that all data points lie on a line for which one variable decreases as the other increases. A value of 0 implies that there is no linear correlation between the variables. Therefore, only positive values of Pearson's correlation coefficient indicate that activity profiles of compared resources tend to be simultaneously greater than, or simultaneously less than their respective means. Accounting for this interpretation, Song and van der Aalst (2008) observed that if Pearson's correlation coefficient is used to assign a weight to the relationship between the resources performing the joint activities, it is natural to remove negative arcs from the social network they are associated with. We highlight that, in their study, Song and van der Aalst (2008) also illustrated business process applications where Pearson's correlation coefficient worked well in combination with the joint activity metric, when both are used to identify clusters of resources doing similar tasks. Based upon the considerations formulated in Song and van der Aalst (2008), we decide to rank potential resource relationships according to Pearson's correlation coefficient of the joint activities associated with arcs. Only arcs associated with the top p ranked positive Pearson's correlation coefficient values are finally added to the social network synopsis.

An example of a social network synopsis whose relationships are defined with the joint activity metric and assigned to weights computed by Pearson's correlation coefficient is reported in Fig. 1b.

4 Organizational mining: a community detection approach

We describe an organizational mining process which operates in two phases.

The online phase (see Fig. 2) consumes events as they arrive from the event stream. It analyzes buffered events, window-by-window, in order to construct the social network of the recorded resources. The social network is loaded in a graph data synopsis (see the description in Section 3). For each data window, (overlapping) communities are detected in the social network as a model of the current organizational structure of the business process. A community detection algorithm is applied, in order to group cohesive resources into communities; each community describes a specific organization unit that has more connections inside the unit than outside. In this context, overlapping communities are discovered, in order to represent resources that may also simultaneously play various roles in the organizational structure of a business process. The organizational model (set of (overlapping) communities) discovered from the data window is permanently stored and available for the off-line analysis; windowed events are definitely discarded.

The off-line phase (see Fig. 3) is repeatable. It resorts to the model proposed by Greene et al. (2010), in order to describe the evolution of a dynamic organizational structure as it is discovered across the time line. In particular, it follows the life cycle of every organization unit (resource community); every life cycle is characterized by a series of significant events. These events include birth, death, merging, splitting, expansion and contraction of dynamic units.

The on-line organization unit discovery, as well as the off-line organization unit lifecycle analysis are described in Sections 4.1 and 4.2, respectively.

4.1 On-line organization unit discovery

In the online phase, we resort to the Louvain algorithm (Blondel et al. 2008), in order to perform the community detection (i.e. organization unit discovery) in the constructed social network. This algorithm is a greedy optimization that attempts to optimize the modularity

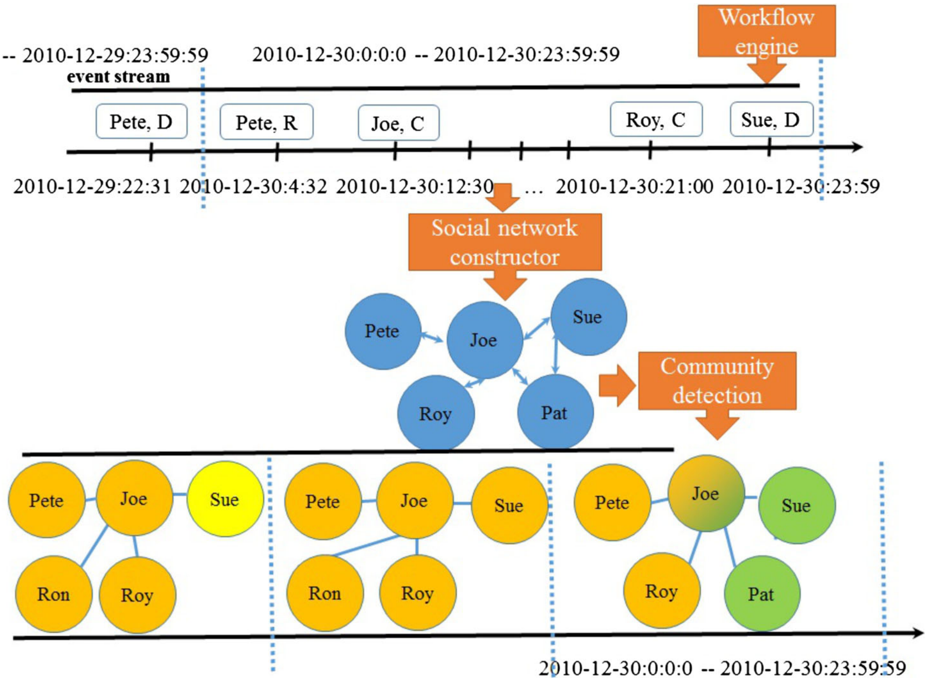


Fig. 2 Online organizational mining: resources grouped in the same community are in the same color

of a partition of the network. Blondel et al. (2008) have proved that the Louvain algorithm achieves a modularity comparable to pre-existing algorithms, typically in less time, so that it is suitable for studying large networks. This founds the common opinion that considers the Louvain algorithm an efficient and easy-to-implement algorithm to identify communities in large networks. In addition, this algorithm reveals a hierarchy of communities at different

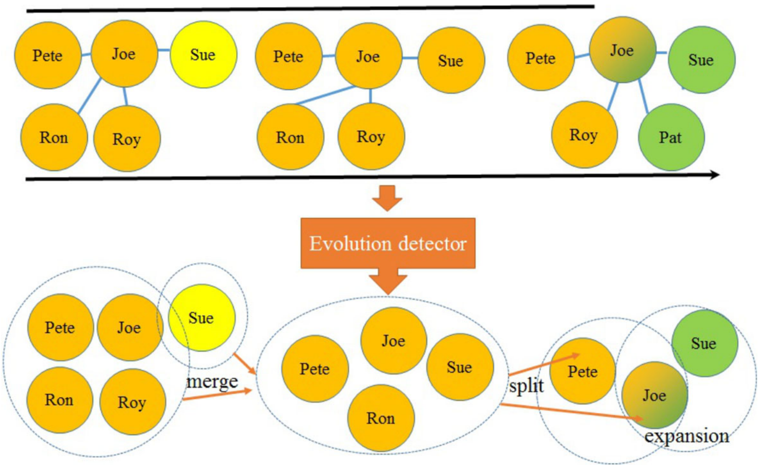


Fig. 3 Offline organizational mining: resources grouped in the same community are in the same color

scales, without requiring us to pre-specify the number of communities. We observe that, as for hierarchical agglomeration clustering (Ward 1963), this hierarchical perspective, where communities with different sizes are organized in a tree, can provide valuable information. In particular, it allows us to control the level of detail at which we intend to understand the working principle of an organization: the higher levels of the hierarchy describe global organizations, while the local levels describe local organizations. However, the Louvain algorithm, in its original formulation, cannot discover overlapping communities. To avoid this pitfall, we apply the Louvain algorithm to the linear network that can be constructed from the social network. The linear network is constructed by following the theory formulated by Evans and Lambiotte (2010) and representing the arcs of the social network as nodes of the linear network. Disjoint communities discovered in the linear network are finally mapped into overlapping communities of the original social network.[3] The three phases of this discovery process, i.e. the linear network construction, the Louvain discovery of communities in the linear network and the transformation of every linear community into a resource community, are described in the following.

4.1.1 Linear network

Let $\mathcal{R} = (\mathcal{N}, \mathcal{A})$ be a resource social network, so that $\mathcal{N}$ is the node set (i.e. set of resources) and $\mathcal{A}$ is the arc set (i.e. set of weighted relationships between resources). A linear network $\mathcal{L} = (\mathcal{N}_{\mathcal{L}}, \mathcal{A}_{\mathcal{L}})$ can be constructed from $\mathcal{R} = (\mathcal{N}, \mathcal{A})$, so that $\mathcal{N}_{\mathcal{L}}$ is the linear node set (i.e. set of linear nodes) and $\mathcal{A}_{\mathcal{L}}$ is the linear arc set (i.e. set of weighted linear relationships between linear nodes). Procedurally, for every arc $(u, v, w) \in \mathcal{A}$ (where u and v are resources connected by a relationship, while w is the weight associated with this relationship), a linear node $\langle uv \rangle \in \mathcal{N}_{\mathcal{L}}$ is constructed in correspondence with (u, v, w). Similarly, for every pair of arcs $(u_i, v_i, w_i), (u_j, v_j, w_j) \in \mathcal{A}$, which share one vertex denoted as x (i.e. $x = u_i = u_j$ or $x = u_i = v_j$ or $x = v_i = u_j$ or $x = v_i = v_j$), a linear arc $(\langle u_i v_i \rangle, \langle u_j v_j \rangle, w) \in \mathcal{A}_{\mathcal{L}}$ is constructed to express the relationship between linear nodes $\langle u_i v_i \rangle$ and $\langle u_j v_j \rangle \in \mathcal{N}_{\mathcal{L}}$. Linear weight w is defined as follows:

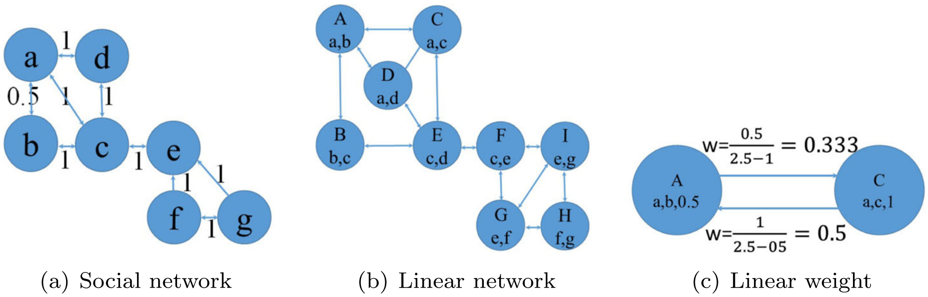
$$w = \frac{w_i}{deg(x) - w_j}, \tag{4}$$

where deg(x) is the sum of weights associated with arcs of $\mathcal{A}$ incoming to /outcoming from shared node x. An example of a linear network associated with a social network is reported in Fig. 4a–c.

4.1.2 Louvain discovery in a linear network

The algorithm inputs the linear network $\mathcal{L} = (\mathcal{N}_L, \mathcal{A}_L)$ and finds node partitioning that tries to maximize the modularity of nodes $\mathcal{N}_L$ based on arcs $\mathcal{A}_L$. A definition of the modularity measure calculated to evaluate the quality of a node partitioning is reported below.

[3] The idea of discovering overlapping communities by processing the linear network associated with a social network is mainly based on the considerations reported in Evans and Lambiotte (2010), which can be easily applied to the setting in this study. In fact, although resources may also belong to various organization units simultaneously, the arcs between them represent, in this formulation, a single type of interaction. This makes it reasonable to discover disjoint communities when the focus is on networks representing these interactions in the nodes. On the other hand, transforming detected linear communities into resource communities would naturally represent overlaps.



(a) Social network (b) Linear network (c) Linear weight

Fig. 4 An example of how a resource social network (see Fig. 4a) is transformed into the linear network (see Fig. 4b) and an example of how the linear weights are computed in the linear network Fig. 4c)

The discovery is divided into two phases repeated iteratively (Blondel et al. 2008). In the first phase, a different community is initially assigned to each node. Then, for each node i, the algorithm considers every neighbor j of i, in order to evaluate the gain of modularity (details are reported below) that would take place by transferring i from its community into the community of j. The node i is transferred into the community for which this maximum positive gain can be achieved. If no positive gain is possible, i stays in its original community. This evaluation is applied by processing, repeatedly and sequentially, all linear nodes until no further modularity improvement can be achieved. The first phase stops when a local maximum of the modularity is attained and no individual transfer can further increase the modularity. In the second phase, a new meta-network is built. Its meta-nodes represent the communities discovered during the first phase. Meta-arcs between these new meta-nodes are given by the sum of the weight of the arcs between linear nodes in the corresponding two communities. The arcs between the nodes of the same community lead to self-loops for this community in the new network. Once the second phase is completed, the first phase of the algorithm is reapplied to the resulting meta-network, in order to iterate the discovery process, until there are no more changes in discovered communities and/or maximum modularity is attained.

Computing modularity The modularity of a community partitioning is measured by computing the Reichardt-Bornholdt measure (Reichardt and Bornholdt 2006). This measure is considered since it allows us to tune the number and size of communities by resorting to a resolution parameter γ. Higher values of γ foster the discovery of a larger number of communities of smaller size and vice-versa (Reichardt and Bornholdt 2006). Let $\mathcal{C}_{\mathcal{L}} = \left\{C^1_{\mathcal{L}}, C^2_{\mathcal{L}}, \ldots, C^l_{\mathcal{L}}\right\}$ be a community partitioning of the node set $\mathcal{N}_L$, so that: (1) $\bigcup_{C^k_{\mathcal{L}} \in \mathcal{C}_L} C^k_{\mathcal{L}} = \mathcal{C}_{\mathcal{L}}$ and (2) for all $C^h_{\mathcal{L}}, C^k_{\mathcal{L}} \in \mathcal{C}_{\mathcal{L}}$ $(h \neq k)$, $C^h_{\mathcal{L}} \cap C^k_{\mathcal{L}} = \oslash$. The Reichardt-Bornholdt measure $\mathcal{RB}$ is formulated as follows:

$$\mathcal{RB}(\mathcal{C}_{\mathcal{L}}) = \frac{1}{2m} \sum_{i,j \in \mathcal{N}_{\mathcal{L}}} \left[\left(A_{ij} - \gamma \frac{deg^{in}(i) deg^{out}(j)}{2m} \right) \delta\left(C^i_{\mathcal{L}}, C^j_{\mathcal{L}}\right) \right], \tag{5}$$

where $C^i_{\mathcal{L}}$ and $C^j_{\mathcal{L}}$ denote communities, which group nodes i and j, respectively; $A_{ij} = w_{ij}$ if arc $(i, j, w_{ij}) \in \mathcal{A}_{\mathcal{L}}$ exists, 0 otherwise; m is the sum of defined weights

$\left(\text{i.e. } m = \sum_{(i,j,w_{ij}) \in \mathcal{A}_\mathcal{L}} w_{ij}\right)$; γ is the resolution parameter; $deg^{in}(i)$ and $deg^{out}(j)$ are the sum of weights incoming to/outcoming from i and j , respectively; $\delta\left(C^i_\mathcal{L}, C^j_\mathcal{L}\right)$ is the Kronecker function, so that $\delta\left(C^i_\mathcal{L}, C^j_\mathcal{L}\right) = 1$, if i and j are grouped in the same community $\left(\text{i.e. } C^i_\mathcal{L} = C^j_\mathcal{L}\right)$, 0 otherwise.

Moving the focus from arcs to communities, Formula 5 can be rewritten as follows (see details in Appendix A):

$$\mathcal{RB}(\mathcal{C}_L) = \sum_{C^h_\mathcal{L} \in \mathcal{C}_L} \left(e_{hh} - \gamma a^{in}_h a^{out}_h\right), \tag{6}$$

where $e_{hh} = \frac{1}{2m} \sum_{i,j \in \mathcal{N}_\mathcal{L}} A_{ij} \delta\left(C^i_\mathcal{L}, C^h_\mathcal{L}\right) \delta\left(C^j_\mathcal{L}, C^h_\mathcal{L}\right)$ describes the arcs falling in a community $C^h_\mathcal{L}$ with respect to the entire set of arcs, while $a^{in}_h = \frac{1}{2m} \sum_{i \in \mathcal{N}_\mathcal{L}} deg(i)^{in} \delta\left(C^i_\mathcal{L}, C^h_\mathcal{L}\right)$ and $a^{out}_h = \frac{1}{2m} \sum_{j \in \mathcal{N}_L} deg(j)^{out} \delta\left(C^j_\mathcal{L}, C^h_\mathcal{L}\right)$ describe the arcs incoming to/outcoming from nodes falling in $C^h_\mathcal{L}$ with respect to the entire set of arcs.

Based upon this arc-centered formulation of $\mathcal{RB}$, Aynaud et al. (2013) showed that the gain of modularity when a node x is moved from the old community $C^{old}_\mathcal{L}$ to the new community $C^{new}_\mathcal{L}$ (i.e. the difference between the network modularity with the community set produced after the re-assignment of x and the modularity before this re-assignment) can be easily computed as follows:

$$\begin{aligned} \Delta(\mathcal{RB})\left(x, C^{old}_\mathcal{L}\ ,\ C^{new}_\mathcal{L}\right) = \\ &= \frac{1}{2m} \sum_{n \in C^{new}_\mathcal{L}} [A_{nx} + A_{xn}] - \frac{1}{2m} \sum_{n \in C^{old}_\mathcal{L} - \{x\}} [A_{nx} + A_{xn}] \\ &\quad - \frac{1}{2m} \sum_{n \in C^{new}_\mathcal{L}} \left[\frac{deg^{in}(n) deg^{out}(x) + deg^{in}(x) deg^{out}(n)}{2m}\right] \\ &\quad + \frac{1}{2m} \sum_{n \in C^{old}_\mathcal{L} - \{x\}} \left[\frac{deg^{in}(n) deg^{out}(x) + deg^{in}(x) deg^{out}(n)}{2m}\right]. \end{aligned} \tag{7}$$

An intuitive interpretation Formula 7 is that the modularity gain can be measured by:

1. adding up the weight of every arc that connects x and a node $n \in C^{new}_\mathcal{L}$;
2. subtracting the weight of every arc that connects x and a node $n \in C^{old}_\mathcal{L} - \{x\}$;
3. subtracting the product between the degree incoming to /outcoming from node x and the degree outcoming from / incoming to any node $n \in C^{new}_\mathcal{L}$;
4. adding up the product between the degree incoming to /outcoming from node x and the degree outcoming from / incoming to any node $n \in C^{old}_\mathcal{L} - \{x\}$.

We observe that this optimization has been based upon the consideration that all the arcs, which are outside $C_{\mathcal{L}}^{old}$ and $C_{\mathcal{L}}^{new}$, do not contribute to changing the modularity of the community-based network partitioning, hence they can be omitted from the calculation. As proved by Aynaud et al. (2013), determining the modularity gain in this way allows us to diminish the computation burden when evaluating community re-assignments.

4.1.3 From linear communities to resource communities

Finally, every linear community $C_{\mathcal{L}}^{i} \in \mathcal{C}_{\mathcal{L}}$ (see Fig. 5a) is one-to-one associated with a resource community $C^{i} \in \mathcal{C}$ (see Fig. 5b). This set of resource communities represents the instantaneous model of the organizational structure of the business process described at the window time. Formally, C^i groups every resource node that contributes to the definition of a linear node belonging to $C_{\mathcal{L}}^{i}$, that is:

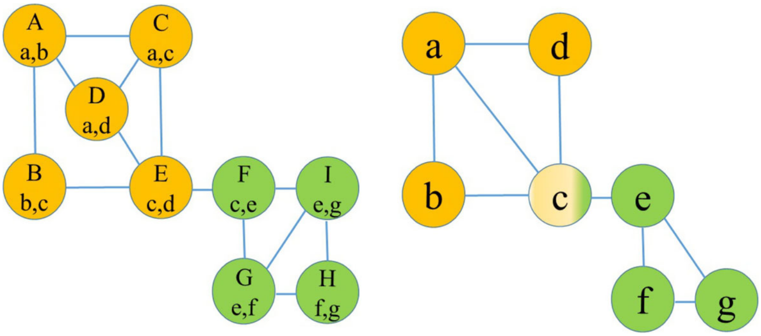
$$C^i = \left(\bigcup_{\langle uv \rangle \in C_{\mathcal{L}}^{i}} u \right) \cup \left(\bigcup_{\langle uv \rangle \in C_{\mathcal{L}}^{i}} v \right). \tag{8}$$

The degree function $degree\colon \mathcal{N} \times \mathcal{C} \mapsto [0, 1]$ is computed, in order to quantify the membership degree according to which a resource node is part of a resource community (organization unit). Let $\mathcal{N}_{\mathcal{L}}(u)$ be the set of linear nodes, which are associated with resource node $u \in \mathcal{N}$, that is,

$$\mathcal{N}_{\mathcal{L}}(u) \subseteq \mathcal{N}_{\mathcal{L}} = \bigcup_{i \in \mathcal{N}_{\mathcal{L}} \wedge (i = \langle u,v \rangle \vee i = \langle v,u \rangle)} i, \tag{9}$$

$\mathcal{C}_{\mathcal{L}}(u)$ be the sequence of linear communities, which are associated with every linear node of $\mathcal{N}_{\mathcal{L}}(u)$, that is:

$$\mathcal{C}_{\mathcal{L}}(u) = \langle C_{\mathcal{L}}(i_1), C_{\mathcal{L}}(i_2), \ldots, C_{\mathcal{L}}(i_h) \rangle \; with \; i_j \in \mathcal{N}_{\mathcal{L}}(u), \tag{10}$$



(a) Linear community set (b) Resource community set

Fig. 5 Overlapping resource community discovery: disjoint linear communities discovered by the Louvain algorithm in the linear network (see Fig. 5a) are mapped into overlapping resource communities (see Fig. 5b). The resource node c of the social network is part of the yellow community with degree 0.75, while it is part of the green community with degree 0.25

where $C_{\mathcal{L}}(i)$ denotes the linear community that groups linear node i. For each resource community C^i, the degree according to which u is part of C^i is measured as follows:

$$degree(u, C^i) = \begin{cases} \frac{|\{C^j_{\mathcal{L}} \in \mathcal{C}_{\mathcal{L}}(u) | C^j_{\mathcal{L}} = C^i_{\mathcal{L}}\}|}{|\{C_{\mathcal{L}}(u)\}|} & if\ C^i_{\mathcal{L}} \in \mathcal{C}_{\mathcal{L}}(u) \\ 0 & otherwise \end{cases} , \tag{11}$$

where $C^i_{\mathcal{L}}$ denotes the linear community one-to-one associated with C^i.

4.2 Off-line organization life cycle discovery

In the off-line phase, we track the life cycle of a dynamic organizational structure after its series of instantaneous organization units has been discovered from the business events, buffered into consecutive time windows. This life cycle is represented as a time line, whose consecutive time points are associated with the on-line processed consecutive time windows. Considering that the organization units are here modeled as overlapping resource communities, the time line of a life cycle is formally expressed as a sequence of instantaneous communities, ordered by time, with one instantaneous community for each window-stamped time point. In particular, the evolution of a dynamic resource community along the time line is encoded in a sequence of predefined evolution events (Greene et al. 2010) (see Fig. 6), described as follows:

- A *birth* describes a new resource community observed at time t with no corresponding community in the set of instantaneous communities detected in the past time $t-1$. A new life cycle is created with the time line starting at this birth time.
- A *death* describes the dissolution of a resource community that does not appear for several consecutive time points. The time line of this community ends when it disappears.
- A *merge* occurs when two or more distinct resource communities observed at time $t-1$ can be similar to a single community at time t. A branch is added to connect the single communities at time $t-1$ to the merged community created at time t. The single communities subsequently share the same time line.
- A *split* occurs when a single resource community present at time $t-1$ can be similar to two or more distinct resource communities at time t. A branching occurs from the starting community to the split ones, with the creation of an additional resource community that shares the time line up to time $t-1$, but has a distinct time line from time t onwards.

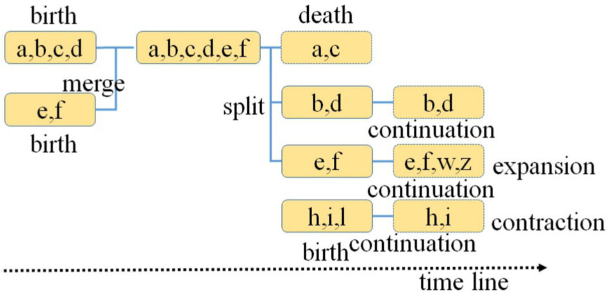


Fig. 6 The life cycles of dynamic organization units (*resource communities*) of a business process as they are detected along the time line. Frontier communities are drawn with a dotted line

- A *continuation* of a resource community occurs when its composition is (near-)stable at time t.
- An *expansion* of a resource community occurs when its cardinality grows at time t.
- A *contraction* of a resource community occurs when its cardinality decreases at time t.

We resort to the algorithm proposed by Greene et al. (2010),[4] in order to discover the evolution events described above and use them to construct the life cycle of the dynamic organzational structure. The discovery is based on the analysis of the degree of similarity between the pairs of resource communities discovered at consecutive time points. The algorithm inputs a time series $\mathcal{T} = \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_t, \ldots, \mathcal{C}_n$, where each instantaneous organzational structure $\mathcal{C}_t$ is a set of overlapping resource communities extracted at a specific time window and timestamped with t in $\mathcal{T}$. The output is the life cycle of this organization time series. This is modeled as a time-step organization network (Greene et al. 2010). The nodes of this network, organized by their timestamp, correspond to the instantaneous communities discovered at consecutive time points. In particular, step t represents a snapshot of the nodes timestamped at time t. The inter-step arcs of this network link nodes/communities at successive steps by representing evolution events (see Fig. 6). Every inter-step link path (time line path) describes the life cycle of a dynamic organization unit starting from its birth. The most recent community in a time line path denotes its frontier community. The algorithm operates in two phases (see Algorithm 1):

Algorithm 1 Organization life cycle discovery

```
Require: $\mathcal{T} = \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_t, \ldots, \mathcal{C}_n$: organzational structure time series
Ensure: $\mathcal{D}$: time-step organization network (life cycle) of $\mathcal{T}$
 1: {initialization phase $(t = 1)$}
 2: $\mathcal{F} \leftarrow \mathcal{C}_1$
 3: Add each community of $\mathcal{C}_1$ at the step 1 of $\mathcal{L}$
 4: {Mining phase $(t \geq 2)$}
 5: for $t = 2$ to $n$ do
 6:     for $(C, C_{\mathcal{F}}) \in \mathcal{C}_t \times \mathcal{F}$ do
 7:         Compute jaccard$(C, C_f)$
 8:     end for
 9:     Determine the event set $\mathcal{E}$ based on the analysis of Jaccard coefficients
10:     Adding a time step $t$ to $\mathcal{D}$ by constructing links to previous steps according to $\mathcal{E}$
11:     Update $\mathcal{F}$ according to $\mathcal{E}$
12: end for
```

1. The initialization phase considers $t = 1$. It processes the organizational structure C_1 and adds every resource community $C \in \mathcal{C}_1$ both to step 1 of the node set of the time-step organization network $\mathcal{D}$ (line 2, Algorithm 1) and to the frontier set $\mathcal{F}$ (line 3, Algorithm 1).
2. The mining phase, that is iterated for each time point t ($t = 2, 3, \ldots, n$), discovers the evolution events at time t (lines 6-9, Algorithm 1), adds a new step t in $\mathcal{D}$ (line 10, Algorithm 1), updates $\mathcal{F}$ (line 11, Algorithm 1). Let $\mathcal{E}_t$ be the set of evolution events, which describe how the community set $\mathcal{C}_t$ evolves with respect to the frontier set $\mathcal{F}$, then step t of $\mathcal{D}$ is grown with nodes associated with communities of $\mathcal{C}_t$ and inter-step arcs associated with events of $\mathcal{E}_t$. $\mathcal{F}$ is updated, in order to store the most recent communities of the time line paths in current $\mathcal{D}$.

[4] The life cycle discovery algorithm is independent of the algorithm used on-line, in order to discover instantaneous organizational structures of the business process under analysis.

In particular, the detection of evolution events and their transformation into inter-step arcs of $\mathcal{D}$ is based on the analysis of similarities between each instantaneous community $C \in \mathcal{C}_t$ and each frontier community $C_{\mathcal{F}} \in \mathcal{F}$ (community matching analysis). The Jaccard coefficient is computed as a similarity measure so that $jaccard(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}$. The compared communities are similar if and only if their Jaccard coefficient exceeds a user-defined threshold σ. In the described algorithm, the computed similarities are explored in three steps.

1. First, we consider every $C \in \mathcal{C}_t$. If there is no frontier $C_{\mathcal{F}} \in \mathcal{F}$, so that $jaccard(C, C_{\mathcal{F}}) \geq \sigma$, then a birth event has happened. C becomes a new node at step t of $\mathcal{D}$. A new time line path would start in this node. If there is one and only one frontier $C_{\mathcal{F}} \in \mathcal{F}$ so that $jaccard(C, C_{\mathcal{F}}) \geq \sigma$, this indicates the continuation from $C_{\mathcal{F}}$ to C. If there are two or more frontiers $C_{\mathcal{F}_1}, \ldots, C_{\mathcal{F}_k} \in \mathcal{F}$ so that $jaccard(C, C_{\mathcal{F}_i}) \geq \sigma$ $(i = 1, 2, \ldots, k)$, then a merge event has happened. The new merged community is linked to the time lines of the frontier communities contributing to the merge.
2. Then, we consider every $C_{\mathcal{F}} \in \mathcal{F}$. If there are two or more resource community $C_1, \ldots, C_k \in \mathcal{C}_t$ so that $jaccard(C_i, C_{\mathcal{F}}) \geq \sigma$ $(i = 1, 2, \ldots, k)$, then a split event has happened. Every new split community is connected to the time line of the frontier community that has initiated the split.
3. Finally, we consider every pair $(C, C_{\mathcal{F}}) \in \mathcal{C}_t \times \mathcal{F}$, so that $jaccard(C, C_{\mathcal{F}}) \geq \sigma$, if $\frac{cardinality(C - C_{\mathcal{F}})}{cardinality(C)} > \theta$, then an expansion event has happened; if $\frac{cardinality(C_{\mathcal{F}} - C)}{cardinality(C_{\mathcal{F}})} > \theta$, then a contraction event is manifested. The expansion/contraction threshold θ is a user-defined parameter.

An example of evolution events tracked according to the procedure described above is reported in Example 1.

Example 1 (Evolution event discovery) Let us consider frontier set $\mathcal{F}$ composed of frontier communities $C_{\mathcal{F}_1} = \{a, b, c\}$, $C_{\mathcal{F}_2} = \{d, e, f\}$ and $C_{\mathcal{F}_3} = \{h, i, l\}$ and organizational structure $\mathcal{C}$ composed of communities $C_1 = \{a, b, c, d, e, f\}$, $C_2 = \{h, i\}$, $C_3 = \{i, l, m\}$ and $C_4 = \{m, o\}$. Jaccard similarities between $\mathcal{F}$ and $\mathcal{C}$ are reported in the following:

$$\left(\begin{array}{lllll} & C_1 = \{a, b, c, d, e, f\} & C_2 = \{h, i\} & C_3 = \{i, l, m\} & C_4 = \{m, o\} \\ C_{\mathcal{F}_1} = \{a, b, c\} & \frac{3}{6} = 0.5 & 0 & 0 & 0 \\ C_{\mathcal{F}_2} = \{d, e, f\} & \frac{3}{6} = 0.5 & 0 & 0 & 0 \\ C_{\mathcal{F}_3} = \{h, i, l\} & 0 & \frac{2}{3} = 0.66 & \frac{2}{4} = 0.5 & 0 \end{array}\right).$$

Let us explore Jaccard similarities with $\sigma = 0.5$ and $\theta = 0.5$. We discover the following events:

1. A birth event on $C_4 = \{m, o\}$, as there is no frontier community $C_{\mathcal{F}} \in \mathcal{F}$ so that $jaccard(C_4, C_{\mathcal{F}}) \geq \sigma$.
2. A continuation event on $C_2 = \{h, i\}$, as there is only frontier community $C_{\mathcal{F}_3} = \{h, i, l\}$ so that $jaccard(C_2, C_{\mathcal{F}_3}) \geq \sigma$.
3. A continuation event on $C_3 = \{i, l, m\}$, as there is only frontier community $C_{\mathcal{F}_3} = \{h, i, l\}$ so that $jaccard(C_3, C_{\mathcal{F}_3}) \geq \sigma$.
4. A merge event on $C_1 = \{a, b, c, d, e, f\}$, as there are frontier communities $C_{\mathcal{F}_1} = \{a, b, c\}$ and $C_{\mathcal{F}_2} = \{d, e, f\}$ so that $jaccard(C_1, C_{\mathcal{F}_1}) \geq \sigma$ and $jaccard(C_1, C_{\mathcal{F}_2}) \geq \sigma$.
5. a split event on $C_2 = \{h, i\}$ and $C_3 = \{i, l, m\}$, as there is frontier community $C_{\mathcal{F}_3} = \{h, i, l\}$ so that $jaccard(C_2, C_{\mathcal{F}_3}) \geq \sigma$ and $jaccard(C_3, C_{\mathcal{F}_3}) \geq \sigma$.

6. An expansion event on $C_3 = \{i, l, m\}$, as there is a continuation from $C_{\mathcal{F}_3}$ to C_3 and $\frac{cardinality(C_3 - C_{\mathcal{F}_3})}{cardinality(C)} = \frac{2}{3} > \theta$.

5 Applications

We consider event streams produced in two benchmark process mining applicative scenarios. Our goal is to assess the viability of performing the organizational inspection of the involved business processes by applying the described comprehensive approach, comprising social network analysis and stream data mining. In particular, we evaluate the accuracy of the instantaneous organizational structure discovery in the on-line processing phase, as well as assess the significance of the organizational life cycle discovery performed in the off-line inspection phase.

5.1 Business process scenario

The first business process scenario is taken from a Dutch Academic Hospital and provided in the Business Processing Intelligence Challenge 2011 (BPI 2011).[5] The log contains 150.291 events from January 1, 2005 to April 15, 2008 in 1143 traces. It tracks the organizational behavior of 42 resources (e.g. Emergency room, Cardiovascular clinics, Nursing ward and Recovery room / high care). Each trace is a patient of a Gynaecology department. It indicates how the patient goes through different (maybe overlapping) phases, where a phase consists of the combination Diagnosis and Treatment. The second business process scenario is taken from a Dutch Financial Institute and provided in the Business Processing Intelligence Challenge 2012 (BPI 2012).[6] The log contains 262.200 events from October 01, 2011 to March 3, 2012 and 69 resources, in 13.087 traces. It tracks the organizational behavior of 63 resources identified by a 5-digit code (e.g. 10821, 10629, 10609 and 11289). The business process is an application process for a personal loan or overdraft within a global financing organization.

In both scenarios, event streams are processed with the time-based window model having $\Delta(T) = 120$ days for BPI 2011 and $\Delta(T) = 15$ days for BPI 2012, respectively. These window sizes have been decided based upon the common duration of a business process execution in both scenarios. We drop out time windows, where no event has been produced. Resource social networks are constructed with the top $p = 75$ % ranked positive Pearson correlation values (see details in Section 3.2). The structure (number of nodes, number of arcs and average degree) of the series of social networks, constructed in both scenarios, is described in Table 2.

5.2 Organizational structure discovery

We start the analysis of the organizational structures discovered through the organizational structure discovery phase (see Section 4.1) by comparing the overlapping community discovery to the baseline disjoint community discovery. This baseline is defined by applying the Louvain algorithm to the resource social network directly (i.e. without resorting to the

[5] http://www.win.tue.nl/bpi/2011/challenge

[6] http://www.win.tue.nl/bpi/2012/challenge

Table 2 Resource social networks (BPI 2011 and BPI 2012): number of nodes (N), number of arcs (A) and average node degree (deg)

	BPI 2011				BPI 2012		
Time horizon	N	A	deg	Time horizon	N	A	deg
2005/01/01-2005/05/01	22	13	1.18	2011/10/01-2011/10/16	46	455	19.78
2005/05/01-2005/08/29	19	8	0.84	2011/10/16-2011/10/30	44	408	18.55
2005/08/29-2005/12/27	27	24	1.78	2011/10/30-2011/11/14	50	516	20.64
2005/12/27-2006/04/26	29	23	1.59	2011/11/14-2011/11/29	52	556	21.38
2006/04/26-2006/08/24	28	21	1.50	2011/11/29-2011/12/14	50	531	21.24
2006/08/24-2006/12/22	29	16	1.10	2011/12/14-2011/12/29	49	550	22.45
2006/12/22-2007/04/21	27	20	1.48	2011/12/29-2012/01/13	47	543	23.11
2007/04/21-2007/08/19	27	24	1.78	2012/01/13-2012/01/28	47	567	24.13
2007/08/19-2007/12/17	27	24	1.78	2012/01/28-2012/02/12	44	487	22.14
2007/12/17-2008/04/15	19	9	0.95	2012/02/12-2012/02/27	47	579	24.64
				2012/02/27-2012/03/13	46	586	25.48
				2012/03/13-2012/03/29	22	97	8.82

linear network transformation of the network). The compared organizational structures are discovered by varying resolution parameter γ of the Reichardt-Bornholdt measure among 0.5, 1, 1.5 and 2. We proceed in this study by comparing the organizational structures discovered by resorting to the community detection approach presented in this paper, to the structures discovered by resorting to the clustering approach introduced by Song and van der Aalst (2008).

5.2.1 Evaluation metrics

Following the mainstream of research in social network analysis, a modularity metric is used, in order to measure the quality of every organizational structure discovered by the community detection approach. This metric allows us to quantify the ability of putting highly interconnected resources in the same community (organization unit), while putting separate resources in distinct communities. Taking into account that, in this study, the organization structures are primarily discovered in the form of "overlapping" resource community structures, the extended modularity metric (Shen et al. 2009) is considered. This metric is formulated as follows:

$$\mathcal{E}(\mathcal{C}) = \frac{1}{2m} \sum_{i,j \in \mathcal{N}} \frac{1}{O_i O_j} \left[\left(A_{ij} - \frac{deg^{in}(i) deg^{out}(j)}{2m} \right) \delta(\mathcal{C}^i, \mathcal{C}^j) \right], \tag{12}$$

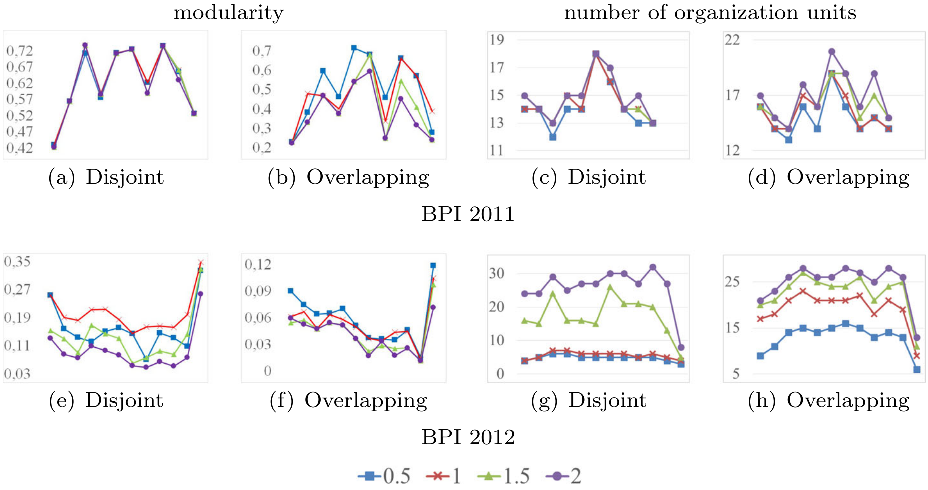
where $\mathcal{C}$ is the organizational structure (i.e. resource community set) ; O_i (O_j) is the number of communities to which i (j) belongs, $A_{ij} = w_{ij}$ if arc (i, j, w_{ij}) exists in the resource social network, 0 otherwise; m is the sum of weights on the entire network, while $deg^{in}(i)$ $(deg^{out}(j))$ is the sum of weights incoming to /outcoming from i (j); $\delta(\mathcal{C}^i, \mathcal{C}^j) = 1$ if i and j are grouped in the same community, 0 otherwise. This modularity metric equals 0 when all the resources belong to the same organization unit (community). The higher the metric, the more significant the community structure of the discovered organization. As already observed by Shen et al. (2009), the extended modularity reduces to Reichardt-Bornholdt measure $\mathcal{RB}$ of modularity (see Formula 5) when $\mathcal{RB}$ is formulated with $\gamma = 1$ and both $\mathcal{RB}$ and $\mathcal{E}$ are computed with each node/resource belonging to only one community (i.e. disjoint communities are discovered). We also observe that extended modularity $\mathcal{E}$ allows us to measure the modularity of both disjoint and overlapping organizational structures in a similar fashion, directly on the resource social network. In this way, we can reasonably compare the disjoint and overlapping organizations, which can be extracted from the same resource network, even though the approach proposed in this study discovers the overlapping organizational structure by computing the Reichardt-Bornholdt measure on the linearization of the resource social networks.

The extended modularity is analyzed in combination with structural metrics such as the number of organization units, in order to evaluate the granularity of the discovered organizational structures, as well as the number of resources grouped in several units simultaneously, in order to evaluate the overlapping degree of these structures. We also consider the learning time spent completing the discovery process.

Finally, as the community detection approach is compared to the clustering approach, we also consider the Silhouetted index (Rousseeuw 1987). This metric is generally adopted in machine learning, in order to evaluate clustering algorithms. It measures the ability of detecting clusters (organization units) with similar (interconnected) resources (high cohesion) compared to other clusters (separation). It ranges between -1 and 1. The higher the Silhouette index, the more appropriate the community/cluster configuration.

5.2.2 *Disjoint community detection vs Overlapping community detection*

The extended modularity of the organization structures, discovered at the consecutive time windows of the streaming time, is reported in Fig. 7a (disjoint organization units) and b (overlapping organization units) for BPI 2011, and in Fig. 7e (disjoint organization units) and f (overlapping organization units) for BPI 2012. The number of organization units discovered per structure is reported in Fig. 7c (disjoint organization units) and d



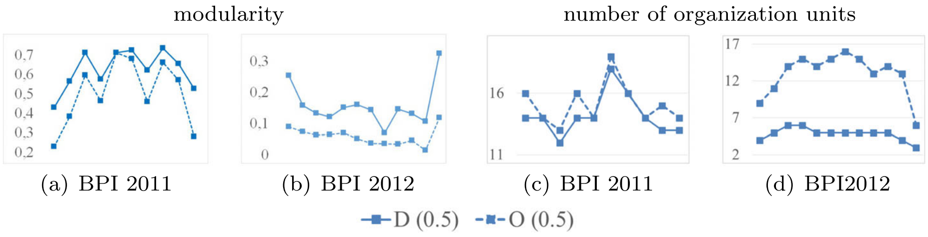
(a) Disjoint (b) Overlapping (c) Disjoint (d) Overlapping

BPI 2011

(e) Disjoint (f) Overlapping (g) Disjoint (h) Overlapping

BPI 2012

Fig. 7 Organizational structure analysis: modularity and number of (disjoint and overlapping) organization units (*Y axis*) with respect to time window (*X axis*). The organizational structure discovery is performed by varying the resolution parameter γ of the Reichardt-Bornholdt measure between 0.5, 1, 1.5 and 2



(a) BPI 2011 (b) BPI 2012 (c) BPI 2011 (d) BPI2012

Fig. 8 Organizational structure analysis: modularity and number of organization units (*Y axis*) with respect to time window (*X axis*). The disjoint organization units (normal line - D) are compared to the overlapping organization units (dotted line - O). Both are discovered with $\gamma = 0.5$

(overlapping organization units) for BPI 2011 and in Fig. 7g (disjoint organization units) and h (overlapping organization units) for BPI 2012.

The inspection of the evolution trend exhibited by these metrics confirms that the presented approach is actually able to control the granularity of the discovered organizational structure by resorting to the resolution parameter γ. In general, the lower the resolution value ($\gamma = 0.5$ and 1.0), the lower the number of discovered units and the higher the modularity of the derived organizational structure. This behavior, that is generally exhibited by both the overlapping and disjoint discovery approach, confirms the ability to control the resolution power of an organizational structure model by computing the Reichardt-Bornholdt measure in combination with the hierarchical Louvain learning during the community detection phase.

Additional considerations regard the pairwise analysis of the disjoint and overlapping organizational structures constructed with $\gamma = 0.5$. Their metrics are compared in Fig. 8a (modularity) and c (number of units) for BPI 2011 and in Fig. 8b (modularity) and d (number of units) for BPI 2012. Normal lines represent disjoint communities, while dotted lines represent overlapping communities. We observe that, in both business process scenarios, the discovery of overlapping communities (dotted line) increases the number of detected organization units, while decreasing the modularity of the final organizational structure. The finer granularity is caused by the higher degree of overlap that produces the lower modularity of the structure as an extra-outcome.[7]

Studying more closely the interpretation of these results, we note that, although the discovery of overlapping organization units can only be obtained at the cost of a reasonable loss of the modularity, it may crucially contribute to improving the comprehension of the structural and functional properties of organizational structures, which do not necessarily involve a strict separation of their organization units. For example, let us analyze the organization units in which "Radiotherapy" (BPI 2011) is involved from January 1, 2005 to May 1, 2005. In the disjoint organizational structure, "Radiotherapy" is part of a single organization unit composed of:

– *"Internal Specialisms clinic ","Anesthesiology clinic", "Obstetrics and Gynaecology clinic" and " Radiotherapy".*

In the overlapping organizational structure, "Radiotherapy" is part of *three* distinct units (which probably have different missions in the Gynecology department), that is:

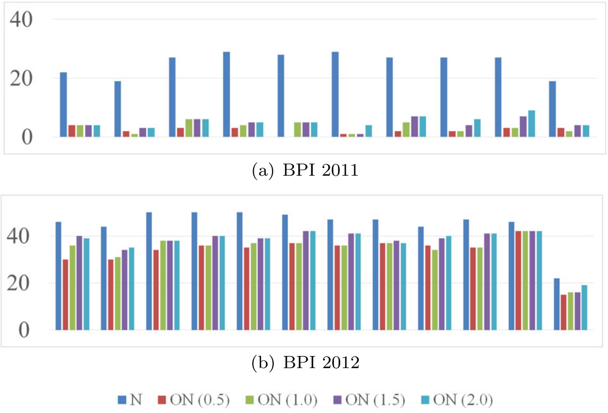
[7] Similar behavior can be observed by performing this pairwise comparison for the overlapping and disjoint organization structures discovered with $\gamma = 0.5$, 1, 1.5 and 2.

- *“Emergency room”, “Internal Specialisms clinic”, “Obstetrics and Gynecology clinic” and “Radiotherapy”* (that involves“Radiotherapy” with degree 0.25);
- *“Emergency room”, “Internal Specialisms clinic”, “Radiotherapy” and “Recovery room / high care”* (that involves“Radiotherapy” with degree 0.5);
- *“Anesthesiology clinic”, “Cardiovascular clinic”, “Radiotherapy” and “Recovery room / high care”* (that involves“Radiotherapy” with degree 0.25).

We observe that the overlapping structure is able to fit the presence of interesting interconnections between “Radiotherapy” and “Emergency room”, “Radiotherapy” and “Cardiovascular clinic”, as well as “Radiotherapy” and “Recovery room / high care”. This knowledge is completely missing in the disjoint organizational structure, since the strength of the interconnection between “Emergency room”, “Cardiovascular clinic” and “Recovery room / high care” is so weak that it prevents their grouping in a single organization unit. On the other hand, the discovery of three different units around “Radiotherapy” outlines three different Radiotherapy missions in the Gynecology department. These missions correspond to different resource structures, which probably have different functions, although all include “Radiotherapy”.

Proceeding in this quality inspection, we analyze the number of resources grouped in various organization units simultaneously by varying the resolution parameter γ. This metric is reported in Fig. 9a for BPI 2011 and Fig. 9b for BPI 2012. We note that the higher number of organization units (i.e. the higher resolution parameter γ) is always associated with the improved capability of detecting small specific units, at a higher level of resolution. This reveals multifaceted resources which are able to cover several roles in various units, simultaneously. On the other hand, the intensity of the overlapping phenomenon is stronger in BPI 2012 (see Fig. 9b) than in BPI 2011 (see Fig. 9a). This reveals that the hospital business process shows a stricter separation of roles than the financial business process.

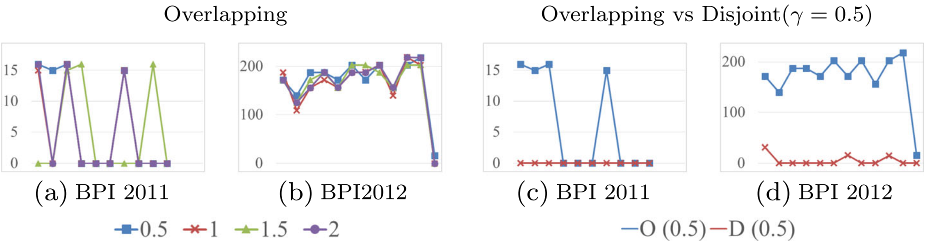
Finally, we investigate the efficiency of the discovery process. The learning time (in milliseconds), spent by the overlapping community detection approach with γ varying between 0.5, 1, 1.5 and 2.0, is reported in Fig. 10a for BPI 2011 and Fig. 10b for BPI 2012. The learning time (in milliseconds), spent discovering disjoint and overlapping communities with $\gamma = 0.5$ is reported in Fig. 10c for BPI 2011 and Fig. 10d for BPI 2012. We note that



(a) BPI 2011

(b) BPI 2012

Fig. 9 Organizational structure analysis: number of nodes (N - blue bar) vs number of overlapping nodes (ON). The overlapping organizational structure discovery is performed by varying the resolution parameter γ of the Reichardt-Bornholdt measure between 0.5 (red bar), 1 (*green bar*), 1.5 (*purple bar*) and 2 (*cyan bar*)



(a) BPI 2011 (b) BPI2012 (c) BPI 2011 (d) BPI 2012

Fig. 10 Organizational structure analysis: learning time (in milliseconds, Y axis) with respect to time window (X axis).The overlapping organizational structure discovery is performed by varying the resolution parameter γ of the Reichardt-Bornholdt measure between 0.5, 1, 1.5 and 2 (see Fig. 10a and b). The overlapping discovery is compared to the disjoint discovery for $\gamma = 0.5$ (see Fig. 10c and d). The learning time is collected on Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz 16.0GB RAM running Windows 8.1

this learning time depends on the number of processed resources, as well as the number of discovered organization units, but it is (near-)stable with respect to the resolution parameter γ. On the other hand, we note that the discovery of overlapping organization units is time-consuming, although the learning time effort is, in general, small.

5.2.3 *Community detection vs clustering*

We complete this investigation by focusing on the social network perspective adopted in this study and comparing the performances of both the community detection-based approach, described in this paper, and the clustering-based approach, illustrated in the seminal research of Song and van der Aalst (2008). Since this clustering-based approach is defined, in order to discover disjoint resource clusters, we compare here disjoint clusters with disjoint communities. For this comparative analysis, we construct the activity profile of the resources by considering the entire event log without any window-based decomposition of the logged events. Coherently with the theory reported in Section 3.2, we compute Pearson's correlation coefficient to measure the joint activities performed by every couple of resources. As reported in Song and van der Aalst (2008), the agglomerative hierarchical clustering algorithm is adopted for the clustering phase.

To evaluate the quality of the discovered organization structures (sets of communities and/or clusters), we consider again the extended modularity metric (see Formula 12), as the modularity is a metric generally adopted to evaluate community detection algorithms. In addition, we consider the Silhouette index (Rousseeuw 1987), which is generally adopted to evaluate clustering algorithms. For each business process scenario, we select the clustering configuration having the number of detected clusters equal to the number of detected communities.

The results are reported in Table 3. These results contribute to validating our intuition that a community detection approach can outperform a clustering approach when both are used to mine the organizational structure that fits the organizational behavior of a business process. In fact, in both scenarios, the discovered community set exhibits higher modularity, as well as higher appropriateness than the discovered cluster set.

5.2.4 *Learned lessons*

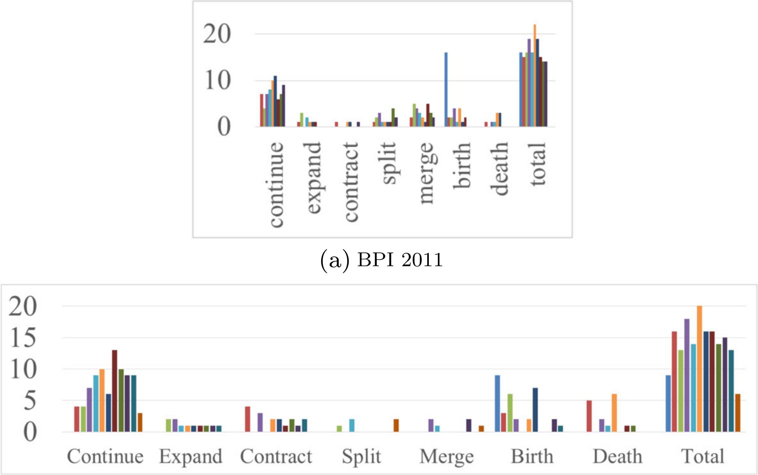
We note that this empirical study contributes to assessing the actual viability of our idea of applying a community detection approach in the context of organizational mining. To

Table 3 Community detection ($\gamma = 0.5$) vs Clustering: Extended modularity and Silhouette index. The best results are in bold

Scenario	n. units	Community model		Clustering model	
		modularity	Silhouette index	modularity	Silhouette index
BPI 2011	18	**0.676**	**0.279**	0.616	0.271
BPI 2012	4	**0.123**	**0.714**	0.122	0.669

highlight the advantages and limits of the proposed solution, we briefly summarize the learned lessons:

1. The study confirms that a community-based organizational structure generally exhibits a higher modularity and better appropriateness than its cluster-based counterpart (see the results in Section 5.2.2). This is an empirical confirmation of how this study takes a step forward from the seminal work of Song and van der Aalst (2008).
2. The study allows us to conclude that a limitation of the presented approach is that the discovery of overlapping organizational units can only be obtained at the cost of a reasonable loss of the modularity. However, the discovery that a resource may also participate in various units simultaneously can be an effective means to model appropriately various complex organizations, which do not admit such a strict distinction of roles. In particular, our qualitative analysis reveals that an organizational structure comprising overlapping units is a more flexible organization model. It is able to reveal which resources pursue "different" missions simultaneously, as well as to derive a structural characterization of resources involved in each mission. This kind of knowledge may be missed with the discovery of disjoint organization units (see the results in Section 5.2.3).



(a) BPI 2011

(b) BPI 2012

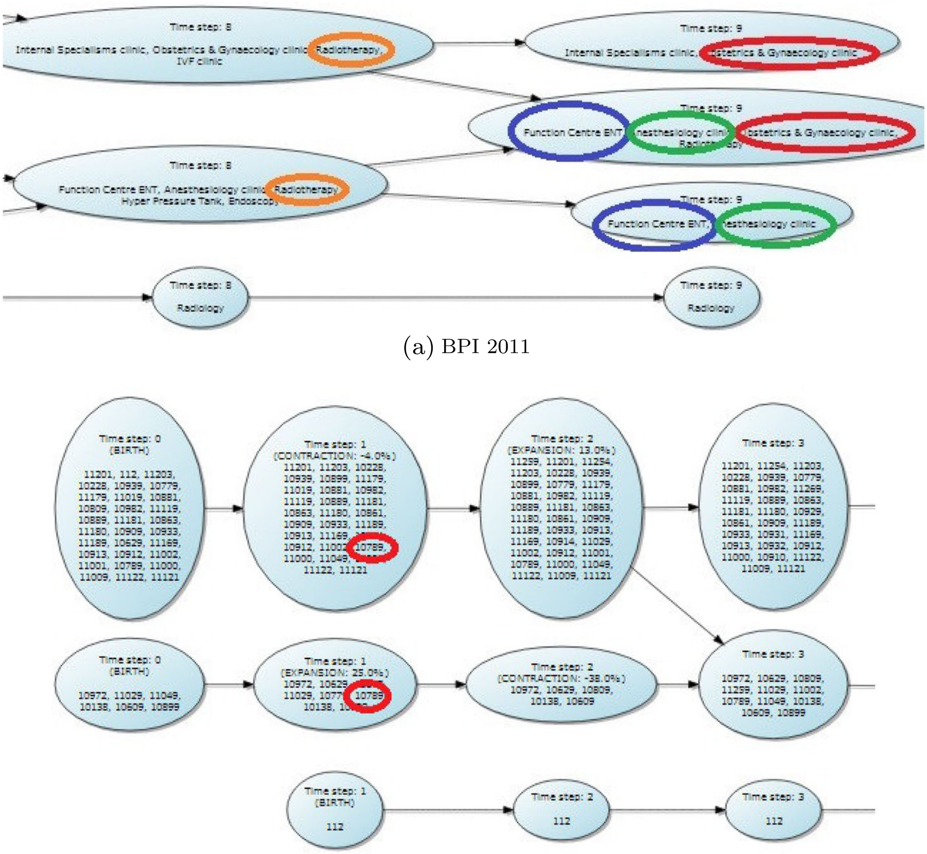
Fig. 11 Organization life cycle analysis: number of evolution events grouped per event type (continuation, contraction, expansion, merge, split, birth and death). A bar group represents each event type; one bar for each time window. The overlapping organizational structure discovery is made with $\gamma = 0.5$. The community matching is performed with a similarity threshold $\sigma = 0.75$ and a contraction/expansion threshold $\theta = 0.35$

3. The analysis of the set-up of the specific algorithm, used to discover communities, shows that its resolution parameter can be decided based on specific application-requirements (e.g. number of units and modularity of the structure). In general, the higher the parameter, the higher the number of units and the lower the modularity (see the results in Section 5.2.2).

5.3 Organizational life cycle discovery

We analyze the performance of the organizational life cycle discovery phase (see Section 4.2) along the type of detected evolution events. We consider the instantaneous overlapping organization structures discovered with $\gamma = 0.5$. The evolution events are discovered with similarity threshold $\sigma = 0.75$ and contraction/expansion threshold $\theta = 0.35$.

The number of continuations, expansions, contractions, splits, merges, births and deaths detected by processing every instantaneous organizational structure along the time line is



(a) BPI 2011

(b) BPI 2012

Fig. 12 A fragment of organizational life cycles discovered for both BPI 2011 (Fig. 12a) and BPI 2012 (Fig. 12b). Resources belonging to overlapping communities are highlighted with a circle

reported in Fig. 11a for BPI 2011 and Fig. 11b for BPI 2012. We note that the majority of detected life cycles starts at the beginning of the stream (a high number of birth events is detected at the first time window). A high number of continuation events is detected at each processed instantaneous organizational structure. This means that the time life of an organization unit typically continues over time in both these business processes. This pattern suggests that a resource often covers a specific role uninterruptedly over time. This consideration is also confirmed by the low number of both contractions and splits detected at consecutive time points. On the other hand, there are cases of merges revealing the presence of organization units whose roles evidently converge at a particular time.

To complete this analysis, we analyze fragments of discovered community life cycles, discovered for both BPI 2011 (see Fig. 12a) and BPI 2012 (see Fig. 12b). They allow us to track the evolutions (birth, split, merge, and so on) of the organizational units discovered by analyzing the events produced by the executions of the two business processes considered in this study. We note that, in both scenarios, several resources participate in different organizational structures simultaneously. This phenomenon is appropriately captured by the discovery of overlapping communities.

6 Conclusion

Process mining aims to exploit event information, in order to learn knowledge related to the behavior of people, organizations, machines and systems. Organizational mining is a sub-field of process mining that focuses on the organization information of event data by learning more about people, machines, organizational roles, work distribution and work patterns.

In this study, we follow the research direction initiated by van der Aalst and Song (2004) and Song and van der Aalst (2008), so we still consider social network analysis as a basic approach to organizational mining. However, we take a step forward in this direction, as we propose combining stream data mining and social network analysis, in order to derive a comprehensive approach to perform organizational mining in a *dynamic* resource scenario. In particular, we consider a stream model to represent events produced by business executions. We use a time-based window model to decompose event streams into consecutive windows, while formalizing a community detection approach to discover and understand the dynamic organizational structure of the logged business process. The discovery phase is defined, in order to mine overlapping, dynamic organization units, to quantify the degree of membership of a resource to several organization units and to understand the evolution (life cycle) of an organizational structure along the time line.

We investigate the viability of the described approach by considering event streams produced in two benchmark process mining applicative scenarios. This investigation contributes to understanding prominent organizational patterns in both business processes. We note that these patterns promise a wide range of applications such as resource assignment and schedule, skill management and organizational design.

As future work, we plan to investigate the performance of the proposed approach in the analysis of resource-intensive business process executions. In addition, we intend to investigate alternative community detection algorithms in this process mining scenario. Finally, we are interested in the study of algorithms, which perform incrementally the discovery of the dynamic organization structures, by integrating the event evolution discovery phase into the community discovery phase.

Acknowledgments This work fulfills the research objectives of the the project MAESTRA "Learning from Massive, Incompletely annotated, and Structured Data" (Grant number ICT-2013-612944) funded by the European Commission, as well as the ATENEO 2014 project "Mining of network data" funded by the University of Bari Aldo Moro. The authors wish to thank Marco Di Pietro and Claudio Greco for their support in developing the software and Lynn Rudd for her help in reading the manuscript.

Appendix A

Let us start from the formulation of Reichardt-Bornholdt measure $\mathcal{RB}(\mathcal{C}_{\mathcal{L}}) = \frac{1}{2m}\sum_{i,j\in\mathcal{N}_{\mathcal{L}}}\left[(A_{ij} - \gamma\frac{deg^{in}(i)deg^{out}(j)}{2m})\delta(C^i_{\mathcal{L}}, C^j_{\mathcal{L}})\right]$ as reported in Formula 5. Let us consider that the Kronecker function $\delta(C^i_{\mathcal{L}}, C^j_{\mathcal{L}})$ can also be written as $\delta\left(C^i_{\mathcal{L}}, C^j_{\mathcal{L}}\right) = \sum_{C^h_{\mathcal{L}}\in\mathcal{C}_{\mathcal{L}}} \delta\left(C^i_{\mathcal{L}}, C^h_{\mathcal{L}}\right)\delta\left(C^j_{\mathcal{L}}, C^h_{\mathcal{L}}\right)$, where $\delta(X, Y) = 1$ 1 iff $X = Y$, 0 otherwise. Therefore, $\mathcal{RB}(\mathcal{C}_{\mathcal{L}})$ can be rewritten as follows:

$$\begin{aligned}
\mathcal{RB} &= \frac{1}{2m}\sum_{i,j\in\mathcal{N}_{\mathcal{L}}}\left[(A_{ij} - \gamma\frac{deg^{in}(i)deg^{out}(j)}{2m})\sum_{C^h_{\mathcal{L}}\in\mathcal{C}_{\mathcal{L}}}\delta\left(C^i_{\mathcal{L}}, C^h_{\mathcal{L}}\right)\delta\left(C^j_{\mathcal{L}}, C^h_{\mathcal{L}}\right)\right]\\
&= \frac{1}{2m}\sum_{i,j\in\mathcal{N}_{\mathcal{L}}}\left[A_{ij}\sum_{C^h_{\mathcal{L}}\in\mathcal{C}_{\mathcal{L}}}\delta\left(C^i_{\mathcal{L}}, C^h_{\mathcal{L}}\right)\delta\left(C^j_{\mathcal{L}}, C^h_{\mathcal{L}}\right) - \gamma\frac{deg^{in}(i)deg^{out}(j)}{2m}\sum_{C^h_{\mathcal{L}}\in\mathcal{C}_{\mathcal{L}}}\delta\left(C^i_{\mathcal{L}}, C^h_{\mathcal{L}}\right)\delta\left(C^j_{\mathcal{L}}, C^h_{\mathcal{L}}\right)\right]\\
&= \sum_{C^h_{\mathcal{L}}\in\mathcal{C}_{\mathcal{L}}}\left[\frac{\sum_{i,j\in\mathcal{N}_{\mathcal{L}}} A_{ij}\delta\left(C^i_{\mathcal{L}}, C^h_{\mathcal{L}}\right)\delta\left(C^j_{\mathcal{L}}, C^h_{\mathcal{L}}\right)}{2m} - \gamma\frac{\sum_{i,j\in\mathcal{N}_{\mathcal{L}}} deg^{in}(i)\delta\left(C^i_{\mathcal{L}}, C^h_{\mathcal{L}}\right)deg^{out}(j)\delta\left(C^j_{\mathcal{L}}, C^h_{\mathcal{L}}\right)}{2m}\right]\\
&= \sum_{C^h_{\mathcal{L}}\in\mathcal{C}_{\mathcal{L}}}\left[\frac{\sum_{i,j\in\mathcal{N}_{\mathcal{L}}} A_{ij}\delta\left(C^i_{\mathcal{L}}, C^h_{\mathcal{L}}\right)\delta\left(C^j_{\mathcal{L}}, C^h_{\mathcal{L}}\right)}{2m} - \gamma\frac{\sum_{i\in\mathcal{N}_{\mathcal{L}}} deg^{in}(i)\delta\left(C^i_{\mathcal{L}}, C^h_{\mathcal{L}}\right)}{2m}\frac{\sum_{j\in\mathcal{N}_{\mathcal{L}}} deg^{out}(j)\delta\left(C^j_{\mathcal{L}}, C^h_{\mathcal{L}}\right)}{2m}\right].
\end{aligned}$$

Introducing the following notation:

$$\begin{aligned}
e_{hh} &= \frac{1}{2m}\sum_{i,j\in\mathcal{N}_{\mathcal{L}}} A_{ij}\delta\left(C^i_{\mathcal{L}}, C^h_{\mathcal{L}}\right)\delta\left(C^j_{\mathcal{L}}, C^h_{\mathcal{L}}\right),\\
a^{in}_h &= \frac{1}{2m}\sum_{i\in\mathcal{N}_{\mathcal{L}}} deg(i)^{in}\delta\left(C^i_{\mathcal{L}}, C^h_{\mathcal{L}}\right),\\
a^{out}_h &= \frac{1}{2m}\sum_{j\in\mathcal{N}_{\mathcal{L}}} deg(j)^{out}\delta\left(C^j_{\mathcal{L}}, C^h_{\mathcal{L}}\right),
\end{aligned}$$

the Reichardt-Bornholdt measure can be written as $\mathcal{RB} = \sum_{C^h_{\mathcal{L}}\in\mathcal{C}_L}\left(e_{hh} - \gamma a^{in}_h a^{out}_h\right)$, as reported in Formula 6.

References

Appice, A., & Malerba, D. (2015). A co-training strategy for multiple view clustering in process mining. *IEEE Transactions on Services Computing* PP(99).

Appice, A., Pietro, M.D., Greco, C., & Malerba, D. (2016). Discovering and tracking organizational structures in event logs. In M. Ceci, C. Loglisci, G. Manco, E. Masciari & Z.W. Ras (Eds.), *New Frontiers in Mining Complex Patterns - 4th International Workshop, NFMCP 2015, Held in Conjunction with ECML-PKDD 2015, Revised Selected Papers, Springer, Lecture Notes in Computer Science* (Vol. 9607, pp. 46–60).

Aynaud, T., Blondel, V.D., Guillaume, J.L., & Lambiotte, R. (2013). *Multilevel Local Optimization of Modularity* (pp. 315–345). John Wiley and Sons, Inc.

Blondel, V., Guillaume, J.L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *10*, P10008.

Clauset, A., Newman, M.EJ., & Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, *70*(6), 1–6.

Dhouioui, Z., & Akaichi, J. (2014). Tracking dynamic community evolution in social networks. In X. Wu, M. Ester & G. Xu (Eds.), *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2014, IEEE Computer Society* (pp. 764–770).

Evans, T., & Lambiotte, R. (2010). Line graphs of weighted networks for overlapping communities. *The European Physical Journal B*, *77*(2), 265–272.

Ferreira, D.R., & Alves, C. (2012). Discovering user communities in large event logs. In F. Daniel, K. Barkaoui & S. Dustdar (Eds.), *Business Process Management Workshops - BPM 2011 International Workshops, Revised Selected Papers, Part I, Springer, Lecture Notes in Business Information Processing* (Vol. 99, pp. 123–134).

Gaber, M.M., Zaslavsky, A., & Krishnaswamy, S. (2005). Mining data streams: a review. *ACM SIGMOD Record*, *34*(2), 18–26.

Greene, D., Doyle, D., & Cunningham, P. (2010). Tracking the evolution of communities in dynamic social networks. In *ASONAM 2010* (pp. 176–183).

Hilbert, M., & Lopez, P. (2011). The world's technological capacity to store, communicate, and compute information. science. *Science*, *332*(6025), 60–65.

Lei, T., & Huan, L. (2010). *Community Detection and Mining in Social Media*. Morgan and Claypool Publishers.

Nguyen, N.P., Dinh, T.N., Shen, Y., & Thai, M.T. (2014). Dynamic social community detection and its applications. *PLOS One*, 9(4):Open Access.

Oliveira, M.DB., Guerreiro, A., & Gama, J. (2014). Dynamic communities in evolving customer networks: an analysis using landmark and sliding windows. *Social Netw Analys Mining*, *4*(1), 208.

Palla, G., Pollner, P., Barabási, A. L., & Vicsek, T. (2009). Social group dynamics in networks. In T. Gross & H. Sayama (Eds.), *Adaptive Networks: Theory, Models and Applications* (pp. 11–38). Springer Berlin Heidelberg.

Reichardt, J., & Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E*, *74*(1), 016,110.

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math*, *20*(1), 53–65.

Saravanan, M., & Rama Sree, R. (2011). Process mining in dyeing unit using control flow perspective: A case study. *Data Mining and Knowledge Engineering*, *3*(6), 351–356.

Shen, H., Cheng, X., Cai, K., & Hu, M. (2009). Detect overlapping and hierarchical community structure in networks. *Physica A*, *388*(2009), 3888:1706–1712.

Song, M., & van der Aalst, W.M.P. (2008). Towards comprehensive support for organizational mining. *Decision Support Systems*, *46*(1), 300–317.

Song, M., Gùnther, C.W., & van der Aalst, W.M.P. (2009). Trace clustering in process mining. In D. Ardagna, M. Mecella & J. Yang (Eds.), *Business Process Management Workshops, BPM 2008 International Workshops, Revised Papers, Springer, Lecture Notes in Business Information Processing* (Vol. 17, pp. 109–120).

Spiliopoulou, M. (2011). Evolution in social networks: A survey. In *Social Network Data Analytics, Springer US* (pp. 149–175).

Sunindyo, W.D., Moser, T., Winkler, D., & Biffl, S. (2010). *Process analysis and organizational mining in production automation systems engineering*. Tech. rep.

van der Aalst, W.M.P. (2011). *Process mining - discovery, conformance and enhancement of business processes*. Springer.

van der Aalst, W.M.P. (2014). No knowledge without processes - process mining as a tool to find out what people and organizations really do. In J. Filipe, J.L.G. Dietz & D. Aveiro (Eds.), *Proceedings of the International Conference on Knowledge Engineering and Ontology Development, KEOD 2014, SciTePress* (pp IS–11).

van der Aalst, W.M.P. (2016). *Process mining - data science in action*, 2nd Edition. Springer.

van der Aalst, W.M.P.., & Song, M. (2004). Mining social networks: Uncovering interaction patterns in business processes. In *BPM 2004* (Vol. 3080, pp. 244–260). Springer: LNCS.

van der Aalst, W.M.P., Reijers, H.A., & Song, M. (2005). Discovering social networks from event logs. *Computer Supported Cooperative Work*, *14*(6), 549–593.

van Zelst, S.J., van Dongen, B.F., & van der Aalst, W.M.P. (2015). Know what you stream: Generating event streams from CPN models in prom 6. In F. Daniel & S. Zugal (Eds.), P*roceedings of the BPM Demo Session 2015 Co-located with the 13th International Conference on Business Process Management (BPM 2015), CEUR-WS.org, CEUR Workshop Proceedings* (Vol. 1418, pp. 85–89.

Ward, J. Jr. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*(301), 236–244.