Multi-Perspective Clustering of Process Execution Traces

Stefan Jablonski[a], Maximilian Röglinger[b], Stefan Schönig[*,a], Katrin M. Wyrtki[c]

[a] Institute for Computer Science, University of Bayreuth, Germany
[b] FIM Research Center, University of Bayreuth, Germany
[c] Project Group Business & Information Systems Engineering of the Fraunhofer FIT

Abstract. *Process mining techniques enable extracting process models from process event logs. Problems can arise if process mining is applied to event logs of flexible processes that are extremely heterogeneous. Here, trace clustering can be used to reduce the complexity of logs. Common techniques use isolated criteria such as activity profiles for clustering. Especially in flexible environments, however, additional data attributes stored in event logs are a source of unused knowledge for trace clustering. In this paper, we present a multi-perspective trace clustering approach that improves the homogeneity of trace subsets. Our approach provides an integrated definition of similarity between traces by defining a distance measure that combines information about executed activities, performing resources, and data values. The evaluation with real-life event logs, one from a hospital and one with traffic fine data, shows that the homogeneity of the resulting clusters can be significantly improved compared to existing techniques.*

Keywords. Process mining • Trace clustering • Multiple perspectives

Communicated by S. Strecker. Received 2018-03-26. Accepted after 2 revisions on 2018-11-28.

1 Introduction

Process mining is a well-established method for extracting models of business processes by analyzing process event logs (Aalst and Weijter 2004; Dumas et al. 2013). Problems arise when applying process mining techniques to event logs of flexible processes (Günther and Aalst 2007; Schönig et al. 2016a,b). Flexible processes are typical in environments such as healthcare where, for example, patient diagnosis and treatment processes require flexibility to cope with unanticipated circumstances (Schönig et al. 2013). Executions of the same process model can therefore differ significantly, resulting in a huge and potentially unplannable number of different process variants (Song et al. 2009). As a process variant, we define a certain way of process execution that differs significantly from other process executions. Especially in flexible environments, processes can have a multitude of execution variants (Lee et al. 2013; Montani and Leonardi 2014; Rebuge and Ferreira 2012) or outliers (Folino et al. 2011). In such environments, event logs have the particular property of *heterogeneity*. Recorded *traces*, i. e., the digital footprints of process executions, differ fundamentally. As a result, mined models tend to be unstructured and very difficult to understand (Günther and Aalst 2007).

One way to cope with the heterogeneity of process logs for flexible processes is to structure logs by *clustering* traces into homogeneous subsets (Song et al. 2009), with each cluster representing a certain process variant with similar characteristics (Ferreira et al. 2007; Song et al. 2009). This allows, examining each trace cluster separately. Hence, mined process models – for clusters – are less complex and easier to understand. These process models can also be used to predict (process) variant-specific attributes, e. g., execution time or costs, during runtime (Aalst et al. 2011).

* Corresponding author.
E-mail. stefan.schoenig@uni-bayreuth.de

Most existing trace clustering approaches mainly focus on a single process perspective. A first group of approaches leverages activity sets and control flow information to identify clusters (Bose and Aalst 2009, 2010; Greco et al. 2006; Lee et al. 2013; Rebuge and Ferreira 2012). Another approach focuses on resource profiles of recorded traces to define clusters (Song et al. 2009). Instead of considering various perspectives for clustering separately, which does not satisfy the idea of processes as multi-perspective entities, we propose to consider multiple perspectives of a process in an integrated manner (M. H. Baumann et al. 2014). Besides the perspectives mentioned above, also the data perspective of processes contains valuable information. Considering several perspectives comprehensively, the importance of the specific perspectives for the clustering is not clear. How high is the impact, how high is the weight of each perspective on the total distance measure so that the resulting clustering leads to homogeneous subsets. Beyond that background, our research addresses the following research question: How does an integrated consideration of multiple process perspectives look like that improves the homogeneity of trace clusters? Are there combinations of perspectives which are transferable to other event logs? To the best of our knowledge, there is no other approach to trace clustering that presents a weighting schema for multi-perspective trace clustering and that has been validated on real-life event logs.

Against this background, we present an approach following a design science research methodology, i. e., the main goal of this paper is to develop a multi-perspective process trace clustering approach with respective tool support that process analysts can use to preprocess available process event logs in order to improve process mining results (Vaishnavi and Kuechler 2004). The multi-perspective trace clustering approach enables reducing the complexity of mined process models in terms of improving the homogeneity of trace subsets. We consider four different perspectives simultaneously: Executed activities, direct following relations between activities, performing resources, and data. Depending on the process log, each perspective has another importance when it comes to structure the log into homogeneous clusters. In order to identify clusters of process traces and maximize the homogeneity of clusters, we define a distance measure that includes the four perspectives, each weighted individually. Therefore, we develop a weighting schema, i. e., the optimal weighting of the perspectives. Especially, in case of the long term evaluation of a process log, it is crucial to know the weighting that leads to the subsets with high homogeneity. Finally, we built clusters by applying a common hierarchical clustering algorithm. We have implemented the approach by means of common hierarchical clustering algorithms using the software environment *R*. The evaluation w. r. t. two real-life event logs, a hospital event log and a traffic event log, shows that the proposed approach improves the homogeneity of resulting clusters significantly compared to classical clustering techniques.[1] Additionally, we identify a weighting that is transferable to other logs.

This paper is structured as follows: Section 2 gives an overview of relevant background and related work. In Section 3, we introduce the proposed integrated trace clustering approach. Section 4 shows the implementation and describes experimental results using two real-life event logs. Section 5 concludes by presenting limitations of our approach and pointing out to future research.

2 Background and Related Work

In this section, we describe the building blocks of trace clustering, provide a short classification framework, and give an overview of related research.

A challenging property of event logs of flexible business processes is their heterogeneity, which results in a huge number of different process execution variants. Each variant is characterized by different sets of executed activities, varying

[1] Prepared event logs and R scripts are available at (Jablonski et al. 2015)

execution orders of activities, different performing resources, and different data used in the processes (Song et al. 2009). Process discovery techniques frequently lack to incorporate other perspectives than activities and control flow. However, the focus of the work is trace clustering and in particular the benefits that can be achieved w. r. t. cluster improvement when considering several process perspectives.

Clustering is used for structuring process logs into clusters of traces which helps to reduce the heterogeneity and improve *understandability* as well. It can be differentiated between reducing the complexity of mined models (Weerdt et al. 2012, 2013) and decomposing resulting models (Ekanayake et al. 2013).

Additionally, clustering techniques can be classified based on how traces are represented. Using a vector space model, an *abstract representation* of a trace is derived. Different properties characterizing a trace, e. g., activity or resource profiles, are transformed into elements of a vector and serve as a mathematical abstract view of the trace (Thaler et al. 2015). This view serves as a basis for calculating a distance measure of traces. The second category uses a *concrete representation* of a trace. Traces are not transformed, but represented based on exact node labels. The similarity of traces is derived by assessing standard string distance metrics, such as Levenshtein edit distance (Ekanayake et al. 2013).

Greco et al. (2006) were the first to propose an approach to cluster execution traces in the domain of process mining. Using a vector space model, traces are represented considering activities and transitions. The works in (Lee et al. 2013; Rebuge and Ferreira 2012) also use information about executed activities to cluster traces. Song et al. (2009) introduced an extended approach and a generic method for trace clustering. They introduce a set of profiles, each addressing a specific perspective of the log that serves as basis for measuring the similarity between traces. Depending on the objective, a suitable profile is chosen to define the similarity between traces. Song et al. define profiles w. r. t. executed activities, organizational aspects as well as case data and case performance (Song et al. 2009). They demonstrate the applicability of their approach using the activity profile, i. e., the multiset of executed activities. The application of other profiles was neither conceived, applied nor validated (Song et al. 2009). Further, all profiles can only be used separated from one another. Latest techniques for context-aware trace clustering (Bose and Aalst 2009, 2010) propose a generic edit distance that takes control flow context information into account. Further, (Bose and Aalst 2010) presents context-aware feature sets based on conserved patterns for the vector space model. Montani and Leonardi (2014) propose a distance measure based on the concrete representation of a trace that is able to take into account temporal information as well.

Evermann et al. (2016) propose a trace clustering method using sequence alignment. They try to bridge the gap between clustering and evaluation by incorporating process mining and model evolution into the clustering method. They use a method for sequence alignment to derive the trace distances and apply multidimensional scaling to construct a feature space that serves the base for clustering.

Leoni et al. (2016) present a process mining framework for correlating, predicting and clustering dynamic behaviour. Regression- and decision-trees are used to cluster event logs into clusters of traces with similar behaviour. They take the correlation of different characteristics into account.

Appice and Malerba (2016) present Co-TraDic, a multiple view aware approach to trace clustering, based on a co-training strategy. All trace profiles defined by Song et al. (2009) are used to generate a clustering pattern. They show that the computed process models have high conformance and comprehensibility compared to process models discovered from the traces clustered with other trace clustering approaches. But, using the co-training strategy, all perspectives have the same weight. Plus it is not clear, which perspective had most impact to the resulting distance measure.

One can easily imagine that the various profiles do not have the same impact on improving the homogeneity of resulting clusters. Therefore, a weighting schema should be used to specify the importance of the trace profiles for the distance measure.

Apart from dedicated clustering approaches, there exist several more general approaches dealing with the analysis and mining of process variants (Ballambettu et al. 2017; Bolt and Aalst 2015; Bolt et al. 2018; Li et al. 2011). In the context of flexible environments, processes can have a multitude of different variants. With these circumstances, the conventional approach of mining, identification and analysis of a limited number of process variants is no longer possible. For the scope of this paper, we especially focus on works that propose a *clustering* approach to identify process variants.

Only a few trace clustering approaches focus exclusively on multiple process perspectives, i. e., either activity or resource profiles, to derive clusters. The approaches mainly take all perspectives equally into account and do not consider a weighting of perspectives. In order to reduce the complexity of trace clusters, identifying similar process traces is necessary. We assume, that depending on the specific process log some perspectives have a higher impact on the similarity than others. For instance, in the context of flexible processes, where activities not necessarily follow a strict process model, the resource and time perspective have a high impact on identifying the homogeneous subset of similar instances. We assume that clusters resulting of a weighted distance measure lead to a higher homogeneity of trace subsets. A weighting schema for the trace clustering approach that focuses on the concurrent integration of several perspectives is missing. We thus propose a multi-perspective trace clustering approach using the abstract representation of traces to reduce the complexity of mined process models in terms of improving the homogeneity of trace subsets.

3 Multi-Perspective Trace Clustering

In this section, we present an integrated multi-perspective trace clustering approach. The approach requires an event log as input that contains specific information about cases that have been executed for a certain process.

3.1 Running Example

The example process contains patient treatment procedures of a hospital. The log contains information about the diagnosis and treatment activities related to the patients in a hospital. Tab. 1 shows an exemplary event log of the process in focus. Each row refers to a *case*, i. e., one patient, and is represented by a *trace*, the sequence of events within a case. Events are represented by the properties: *case identifier* (denoted by the row), *activity identifier* and *resource identifier*, and the specific data attribute *duration time*.

As the log reveals, the cases differ in multiple ways: *(i)* in the sets of executed activities, *(ii)* in the sets of resources that are assigned to a specific activity, and *(iii)* in the execution time of each event. Inspecting the duration time of a case, it is obvious that duration times of the different activities vary. Comparing duration times of equal activities, it is apparent that the same activities can have different execution times. It is obvious that there are other features that influence the duration of activities. These could be features available in the log, e. g., assigned resources, as well as other unknown features as well as high workload or waiting times. In the context of a hospital, cases classified as *urgent* might have lower duration times than cases that are *non-urgent*.

Referring to the example log, one can think of different cluster solutions: regarding the activity profile (multiset of activities) the similarity between trace 3 and 6 is high since the executed activities of both cases are exactly the same. Comparing the overall case duration, a difference of 44 units is observable. The duration of all events of trace 6 are lower than those in trace 3. That could be a hint that case 6 is a case of a patient with an urgent treatment procedure. In order to

Case ID	log events	duration time
1	(1,A,Mike,5),(1,D,Mary,15),(1,B,Pam,32),(1,C,Carol,21), (1,E,Anne,12),(1,F,Wil,3),(1,G,Anne,10),(1,H,Sam,27)	125
2	(2,A,Anne,8),(2,C,John,16),(2,B,Sam,26),(2,D,Mike,20), (2,E,Anne,9),(2,F,Wil,2),(2,G,Mike,4),(2,H,Sam,26), (2,G,Mike,5),(2,H,Pam,17)	133
3	(3,A,Anne,9),(3,B,Pam,36),(3,C,Carol,25),(3,E,Pam,19), (3,F,Wil,5)	94
4	(4,A,Mike,4),(4,B,Mary,12),(4,D,Mary,13),(4,E,Pam,20), (4,F,Wil,3),(4,G,Mike,3), (4,H,Pam,15),(4,G,Anne,9), (4,H,Sam,24)	103
5	(5,A,Anne,11),(5,C,Carol,24),(5,B,Pam,28),(5,E,Pam,21), (5,F,Wil,2),(5,G,Anne,12),(5,H,Sam,24)	122
6	(6,A,Mike,4),(6,B,Sam,20),(6,C,John,13),(6,E,Pam,11), (6,F,Wil,2)	50
7	(7,A,Anne,12),(7,B,Mary,15),(7,C,Sue,36),(7,E,Sam,25), (7,F,Wil,6)	94

Table 1: Example process log, events are characterized by features: case ID, activity, resource, duration time. (A: Consultation, B: Charge administration fee, C: Demanding Second Opinion, D: Medical Examination, E: Surgical Treatment, F: Extraordinary Treatment, G: Ambulatory Treatment, H: Release)

gain insights of the process variants in such a flexible environment, e. g. to recognize patterns and similarities in subsets (as identifying urgent cases or non-urgent cases), only considering the activity perspective is not sufficient. By just using the information of activities to define the similarity of cases, one omits valuable information and those two cases would be erroneously defined as similar. The example highlights that, in some cases, it can be insufficient to define the similarity of traces only w. r. t. the set of executed activities.

There are multiple ways of defining the similarity of two traces using diverse combinations of case properties. The definition of similarity essentially affects the resulting clustering. In the following section, we thus define a similarity measure that combines three perspectives to get an integrated similarity measure and to ensure homogeneous clustering.

3.2 Preliminaries and Definitions

First, we define a process model, a trace, and further requirements that should be fulfilled (Aalst et al. 2011).

Definition 1 (Process model)
Let $\mathfrak{R} = \{r_1, ..., r_{n_\mathfrak{R}}\}$ be a finite population (human resources), and $\mathfrak{D} = \{d_1, ..., d_{n_\mathfrak{D}}\}$ a finite set of data objects (e. g. a timestamp). Then, a process graph G is a tuple (T,E,λ) with

- *T being a set of activities*
- *$E \subseteq T \times T$ a set of edges and*
- *λ: $T \rightarrow \mathcal{P}(\mathfrak{R}) \times \mathcal{P}(\mathfrak{D})$ a function, that maps nodes to entities.*

Note, that $\mathcal{P}(\cdot)$ indicates the power set.

Definition 2 (Activity, Event)
Let $\mathcal{E}$ be the set of all possible event identifiers. Each event e is a tuple (c,t,r,d) characterized by specific properties: An event is part of a trace $c \in C$ corresponds to an activity description $t \in T$, is executed by a particular human resource $r \in \mathfrak{R}$, and has data $d \in \mathfrak{D}$.

Definition 3 (Trace, Event log, Case)
$\mathcal{E}^*$ denotes a set of finite sequences over $\mathcal{E}$. A trace *$c \in \mathcal{E}^*$ is a finite sequence of events. In a trace, each event appears only once and time is non-decreasing. Let C be the set of all possible traces. An* event log *is a set of traces $L \subseteq C$. A trace $c \in L$ in a log represents a process instance, also referred to as "case". Each case is characterized by a defined set of activities, a set of resources, and a set of data objects.*

3.3 Similarity of Cases and Distance Measure

The multi-perspective distance measure is composed of different perspectives. First, we introduce how the overall case distance is derived, before we model specific perspectives. The distance measure represents the distance, or similarity, between two traces. Usually the values of such distance measures start from 0 (the objects are equal) with no upper boundary (completely different). For comparisons and threshold values, a normalized distance measure, for which an upper boundary exists, is better suited (M. H. Baumann et al. 2014; M. Baumann et al. 2014; Thaler et al. 2015). Since the clustering of a set of traces mainly depends on the distance measure, it is crucial to determine how we define the similarity of traces. In line with the literature, we assume the similarity of cases to be defined as follows (M. H. Baumann et al. 2014).

Assumption 1 (Similarity of cases)
The similarity of cases is derived by the aggregation of perspective-specific similarities. Two cases are similar, if the set of executed activities, the resources mapped to the activities, and the data objects are similar.

Using this definition of case similarity, the distance measure can be derived. We define a general distance measure that is applicable using different perspectives, e. g., activity labels and resource identifiers between two cases c_x and c_y executed of one process model G. For each perspective i that is relevant concerning similarity, we calculate the distances and use these results for calculating a measure $dist(c_x, c_y)$ for the distance of c_x and c_y. Different perspectives can be combined depending on the specific application scenario at hand.

Definition 4 (Case distance)
For two cases c_x and c_y with $(x \neq y)$ and weights w_i with $\sum_{i \in I} w_i = 1$ and $w_i \geq 0$, the case distance $dist(c_x, c_y)$ is

$$dist(c_x, c_y) = \sum_{i \in I} w_i \cdot d_i(c_x, c_y)$$

where i is a distinct perspective from the set $I = \{A, T, R, D_e, D_c\}$, A denoting the activity, T the transition, R the resource, D_e the data perspective based on event level, and D_c the data perspective based on case level. $d_i(c_x, c_y)$ is a normalized distance for the distinct perspective between c_x and c_y.

The overall case distance is a linear combination of the weighted perspectives-specific distances that are taken into account. The weights influence the overall case distance and in the same way affect the resulting clustering. The quality of a clustering is measured by the notion of homogeneity, a widely used measure in this context that is defined as follows (Montani and Leonardi 2014; Yip et al. 2003).

Definition 5 (Homogeneity)
Homogeneity is a measure of the quality of clustering results. It measures the similarity of the cases pooled in one cluster. Clusters that contain traces of similar or equal variants have a high homogeneity. The homogeneity of a cluster k is

$$H(k) = \frac{\sum\limits_{c_x, c_y \in k} sim(c_x, c_y)}{\binom{|k|}{2}}$$

The similarity $sim(c_x, c_y)$ refers to the variant of a case. If two cases c_x and c_y are of the same variant, $sim(c_x, c_y) = 1$, otherwise $sim(c_x, c_y) = 0$.

The average homogeneity of a clustering solution with a set of N clusters is defined as the weighted average of the homogeneity of all clusters

$$clustHom(k^{(N)}) = \frac{1}{\sum\limits_{i \in N} |k_i|} \sum_{i \in N} |k_i| H(k_i)$$

By considering the objective of increasing the homogeneity of each resulting cluster, the weights used in the distance measure should be set in a way to reach this objective.

Definition 6 (Optimal weighting)
The optimal weighting of the several perspectives is

$$w_i^* = \arg\max_{w_i}(clustHom(k^{(N)}))$$

where $w_i^* \in [0,1]$ *and* $arg\,max_{w_i}$ *is the arguments of maxima function which refers to inputs* w_i^* *for which* $clustHom(k^{(N)})$ *depending on* $dist(c_x, c_y)$ *and on* w_i *is maximized.* $clustHom(k^{(N)})$ *is the homogeneity of the clustering result.*

Section 4 presents a detailed procedure on how to determine the optimal weights that lead to the maximal homogeneity in each cluster.

As mentioned above, the case distance can be composed of different perspectives. To determine the similarity, or dissimilarity respectively, between two cases a closer look concerning the possible perspectives is necessary.

Activities

The main and essential elements of a case are the executed activities. Therefore, the activity perspective has to take an integral part when defining the similarity between two cases. The similarity or distance of activities depends on their labels such that activities are either the same (label is the same, $distance = 0$) or not (label is different, $distance = 1$) (Dijkman et al. 2011). Using the definition of profiles presented by Song et al. (2009), we use the *activity profile* to characterize the activities related to a specific trace. The items of an activity profile represent all possible activities of a process. The value of each item indicates the number of occurrences of the activity that is related to the item in the case. To calculate the activity-based distance, we use euclidean distance measure. Other distance measures like Hamming, or Jaccard distance are possible, but studies show good results using Euclidean (Bose and Aalst 2009; Francescomarino et al. 2015).

$$d_A(c_x, c_y) = \sqrt{\sum_{a \in A} |i_{xa} - i_{ya}|^2}$$

Each case c_x corresponds to a vector of items $\langle i_{x1}, i_{x2}, ..., i_{xn} \rangle$, where i_{xa} denotes the number of appearances of activity a in case x. Table 2 shows the distance matrix of the cases of the example log from Tab. 1. Cases 3, 6, and 7 are similar regarding the executed activities. The dissimilarity between case 4 related to cases 3, 6, or 7 is the highest.

CaseID	2	3	4	5	6	7
1	$\sqrt{2}$	$\sqrt{3}$	$\sqrt{3}$	1	$\sqrt{3}$	$\sqrt{3}$
2		3	1	$\sqrt{3}$	3	3
3			$\sqrt{10}$	$\sqrt{2}$	0	0
4				2	$\sqrt{10}$	$\sqrt{10}$
5					$\sqrt{2}$	$\sqrt{2}$
6						0

Table 2: Euclidean trace distance of example log based on activities

Transition

Apart from the activities appearing in a trace, the order of the execution is relevant to measure whether traces are similar or not. Therefore, Song et al. defined the transition profile (Song et al. 2009). The items represent the direct following relations between the activities of the trace. For any combination of two activity names, e. g. (A, B), the profile contains the number how many times an event with name A has been directly followed by another event name B (Song et al. 2009).

$$d_T(c_x, c_y) = \sqrt{\sum_{t \in T} |i_{xt} - i_{yt}|^2}$$

Each case c_x corresponds to a vector of items $\langle i_{x1}, i_{x2}, ..., i_{xn} \rangle$, where i_{xt} denotes the number of appearances of transition t in case x. Table 2 shows the distance matrix of the cases of the example log from Tab. 1. Cases 3, 6, and 7 are

similar regarding the order of executed activities. The dissimilarity between case 2 related to cases 3, 6, or 7 is the highest.

Resources

As the running example reveals, two cases can have the same activities executed, but are not really the same. The same activities can be executed by different resources that results in different execution times or different quality. By defining the similarity of cases apart from control flow aspects, the organizational perspective has to be considered as well. The choice of a resource that is assigned to a specific activity has an impact on e. g. the case performance. Therefore, for a comprehensive trace clustering approach, it is crucial to consider resources as well. Song et al. (2009) present an originator profile reflecting how many events were executed by each resource. This profile is, however, too inaccurate to define similarity. Unlike activities executed in a case, the resources must not be considered isolated. To determine the similarity of two cases, it is not only important who was generally involved in a case. It is important to distinguish two cases based on the fact who was assigned to which activity in a case. Thus, it is essential to consider resource items always associated to the activities they have been assigned to. We define a new more comprehensive *resource perspective*. Its items are all possible combinations of resources and activities.

$$d_R(c_x, c_y) = \sqrt{\sum_{r \in R} |i_{xr} - i_{yr}|^2}$$

Each case c_x corresponds to a vector of items $\langle i_{x1}, i_{x2}, ..., i_{xn} \rangle$, where i_{xr} denotes the number of appearances of activity/resource combination r in case x.

Additional Data Attributes

Apart from activities or resources, more information is typically available in event logs. The timestamp of each activity is information which provides valuable insights on cases. For example, the timestamp enables computing the duration of whole cases or of distinct activities in particular as well as the time between activities. One can characterize activities based on their execution duration and one can compare resources based on their performance by execution distinct activities. In the hospital environment of the running example, the waiting and execution time is a hint on the fact that a patient is a urgent case or not. Data attributes can be used to calculate performance measures like time or costs, e. g., case duration time or time between two activities. To increase the precision of trace clustering, it is necessary to define a profile that captures the data perspective. Therefore, it is necessary to differentiate between data attributes on a case and on a event level. In commonly used event log formats such as XES, data on a case level is data that is associated to the whole case and cannot (without further information) be directly assigned to single events. In contrast, data on an activity level is directly associated with the specific event (Verbeek et al. 2010). If the data is available for all events, this finer granular level can be aggregated and used as case information. Therefore, we define two profiles that capture the data perspective.

1) Event-level profile

For attributes that are available on event level, the new profile is introduced.

$$d_{D_e}(c_x, c_y) = \sqrt{\sum_{d \in D_e} |i_{xd} - i_{yd}|^2}$$

Each case c_x corresponds to a vector of items $\langle i_{x1}, i_{x2}, ..., i_{xn} \rangle$. Each item i_{xd} corresponds to one activity-related data attribute d in case x. The value of i_{xd} is computed by the *mean* over the values of the data attribute related to the same activity appearing in case x.

2) Case-level profile

Using a data attribute (aggregated) on case level, each case is represented by one data value. The case level data distance is calculated using the euclidean distance measure.

$$d_{D_c}(c_x, c_y) = \sqrt{|i_x - i_y|^2}$$

Each case c_x corresponds to one item i_x, where i_x denotes the value of the data object of case x.

Multiple data profiles, each referring to another data attribute, can be integrated in the overall distance measure. In this running example, cases, which are similar regarding the other two perspectives, are dissimilar regarding the data perspective. The integrated distance measure is the weighted sum of these perspectives. The weights can be set individually and optimized in order to maximize the homogeneity within the clusters.

3.4 Trace Clustering Approach and Implementation

The basic idea of the clustering algorithms is to group sets of similar process instances. The similarity of instances is defined by the new distance measure presented before. Instances with low distances between one another are clustered. The resulting groups of instances, the clusters, should have a high distance among one another, whereas the distance between the instances in one cluster, should be low (Thaler et al. 2015). Various clustering techniques exist that can be divided into three main categories – hierarchical, partitioning, and density-based techniques (Han 2005). We do not focus on the strength and weaknesses of these techniques, instead, for a more detailed description of the approaches we refer to the specific literature (Duda et al. 2000). We apply an agglomerative hierarchical clustering algorithm using Ward's method to derive distances between two clusters. This method does not require an optimal number of clusters in advance and offers transparency of the complete clustering approach. Besides, it leads to comprehensible cluster solutions, computes the complete hierarchy of clusters and performed well in similar studies and is therefore well suited for our experiments (Montani and Leonardi 2014; Weerdt et al. 2013). We implemented and evaluated the proposed approach using *R*. We have implemented the presented approach, the calculation of the distance measure and the trace clustering approach. The R scripts contain specific sections for *(i)* importing the source event log; *(ii)* calculations of the distance matrices of each perspective; *(iii)* the relative waiting of the perspectives; and *(iv)* the configuration and execution of the clustering approach.

4 Evaluation

Given that the distance measure is a weighted combination of the presented profiles, a high variety of different distance measures is possible. We present an approach to identify the weighting that results in clusters with a high homogeneity and traces that lead to compact process models. The individual use-case determines the appropriate number of clusters, not too many, since they cannot be analysed sensibly, not too little, since heterogeneity is not reduced. We use a process-specific trace characteristic to determine the optimal weighting. Common evaluation metrics (like fitness, generalization, precision, simplicity) determine the quality of process models mainly based on the activity perspective of the corresponding Petri net. We enrich the evaluation by using a process-specific characteristic to show that the resulting clusters have a higher conformance and contain more similar traces.

In this section, we demonstrate the applicability of our approach using two real-life event logs. We present a weighting schema for multi-perspective trace clustering and analyze the importance of each perspectives. Additionally, we evaluate the presented approach against various profile-based distance measures (Appice and Malerba 2016; Song et al. 2009). In order to compare the multi-perspective distance measure with other distance measures, we compute the conformance measures. Therefore, we compare our distance measure to *(i)* CoTraDic the multi-view clustering approach (Appice and Malerba 2016); *(ii)* an equal weighting of the several perspectives; *(iii)* the isolated perspectives. We apply the clustering using the different distance measures on each log and compare the resulting clusters.

4.1 Experiments with a Hospital Log

We first use a log taken from a Dutch Academic Hospital (called *hospital log*) (Dongen 2011). The

log contains data about the treatment procedures of gynecological oncology patients. Each patient corresponds to a single process instance. The log consists of 1,143 instances, 675 different activities executed by 117 different resource units. Some cases in the log are declared as *urgent*. We assume that urgent cases are not only similar regarding the label *urgent*, but also they have to be similar in terms of executed activities or assigned resources. Furthermore, we assume that the time between two executed activities (the waiting time) is smaller in an urgent case. A patient with an urgent diagnosis should not have to wait long during her treatment procedure. Urgent cases might have a similar low waiting time between two activities. The overall waiting time of a case depends on the number of executed activities. The relative waiting time (adjusted by the number executed activities) is used for further calculations. The waiting time is not explicitly given in the log, but can be derived using the timestamp of the activities. Some patients (104) have a number of activities smaller than three, are treated on one day and have a waiting time of 0. Those instances are not taken into account, because they are already similar and represent a first cluster. This results in 1,039 instances, 655 activities executed by 117 different resource units. Here, 772 cases are non-urgent cases, 267 are urgent.

Figure 1 shows the histogram of non-urgent and urgent cases related to their relative waiting time. It confirms the assumption that urgent cases have a similar low relative waiting time, which lies in an interval of [0; 16] days. In contrast, 452 non-urgent cases can have a similar low relative waiting time, but 320 cases have high waiting times up to 183 days. We clear the log of the information whether a case is urgent or not and try to reproduce this information using the new trace clustering approach. We assume that, with the new distance measure, we are able to cluster the urgent instances together in a homogeneous cluster by using the new proposed perspectives and not particularly knowing whether they are urgent or not. We transform each case of the log along the proposed profiles (activity, transition, resource, case data, and event data).

The optimal weights are identified in order to calculate the distance measure that leads to the most homogeneous clusters based on the chosen characteristic. Therefore, we perform the clustering with all weight-combinations for weights at 0.01 intervals and determine the combination that leads to the cluster allocation representing the highest homogeneity.

The homogeneity is derived, by calculating the maximum percentage of cases with the equal characteristic (urgent or non-urgent) of each cluster, weight it with the number of cases in this cluster, sum it up, and divide it by the overall number cases. In an optimal case, the homogeneity would be 100% (all instances of a cluster are either urgent or non-urgent).

4.2 Experiments with a Traffic Fine Log

The second log contains data of a traffic fine management process (Leoni and Mannhardt 2015). The log contains 150,370 cases each referring to one traffic fine. The "traffic offender" has to pay a fine of a certain amount. In some cases a judge is appealed. We assume this specific activity depends on the duration time of the process of the duration time of specific activities as well as on the amount of the fine that has to be paid. Cases that contain the activity *appeal to judge* should be similar regarding those data values. With the integrated distance measure it should be possible to detect the cases that involve a judge, without specificly using this information during clustering. For experiments we subset the log so that the relation between judge-cases and non-judge-cases is equal. For evaluation, we want to avoid any bias due to imbalance in the data. The log contains 1,111 instances. Figures 2 and 3 show the histogram of cases regarding the fine amount as well as duration time. The fine amount of cases where a judge is included ranges from 19.95 up to 2,775. In contrast, the fine amount of cases where no judge is needed ranges from 19.95 up to 205.8. Figure 3 shows similar varieties of the duration time. The duration time of cases where a judge

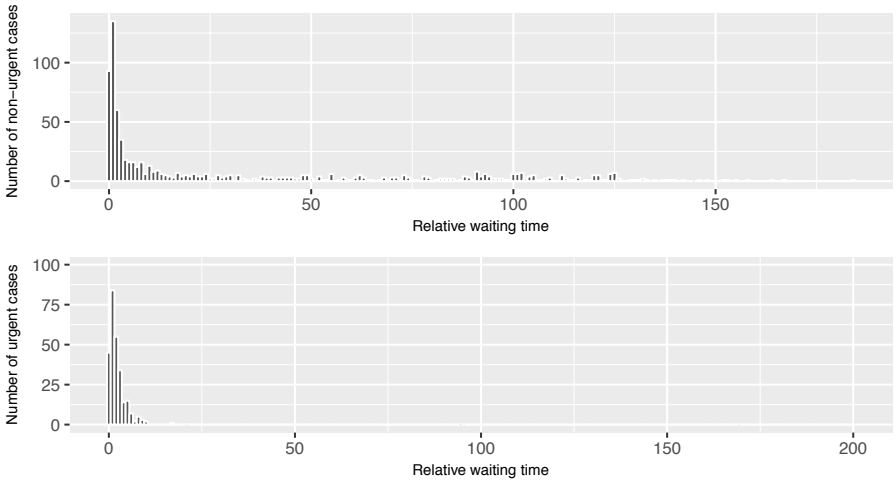

Figure 1: Histogram of relative waiting time in days of urgent/non-urgent cases

is appealed ranges from 47 up to 3,841 days. The duration time of cases where no judge is needed just ranges between 0 and 954 days with peaks around 0 and 500 days.

Apart from activities and resources, the relevant perspectives for the distance measure in this scenario are: *(i)* total duration time of a case; *(ii)* the event based duration time; *(iii)* the total fine value aggregated at a case level; and *(iv)* the amount of a fine on an event level. In total the new distance measure combines seven different values. Again we transform the log along the proposed profiles and calculate the integrated distance measure. Therefore, we optimize the weights of each perspective in order to receive the cluster solution with the highest homogeneity. Our objective is to receive clusters that separate the cases where a judge is appealed from those where no judge is consulted.

4.3 Weighting Schema

The weighting is computationally costly. We provide a weighting schema in order to simplify the choice of the optimal weights. To derive the optimal weighting we use process-specific case information. We take a case attribute to define whether traces are similar or not. Regarding the environment of the hospital log, two cases are considered similar, if they have the attribute urgent and not urgent, respectively. Analogously, cases of the traffic log are considered similar, if they include a judge during the execution. In case of long-term evaluation of an event log the exact computation of the optimal weighting makes sense. Given that the event log might change over time because of additional executed cases, the initial event log can be seen as a training data set. The previous computed optimal weighting is still a valid weighting combination that can be used for the enlarged event log.

General Weighting Guidelines

For both event logs, we computed the cluster solutions for different weighting combinations. We analyze the solutions to clarify the relation between the weighting of the single perspective and the respective range of resulting homogeneity. For both logs, we propose general weighting guidelines that can be used for these specific logs, but can be transferred to other event logs as well.

Hospital log: Figure 4 gives an overview of the relation between the weighting of the activity perspective and the resulting homogeneity of the specific clusters. For a weighting between 0 and 0.5, the main number of clusters have a homogeneity of only 0.743 (the minimal homogeneity for separation of urgent/non urgent cases). For a weighting over 0.6, the mean of resulting homogeneity is increasing. But, the weight between 0.9

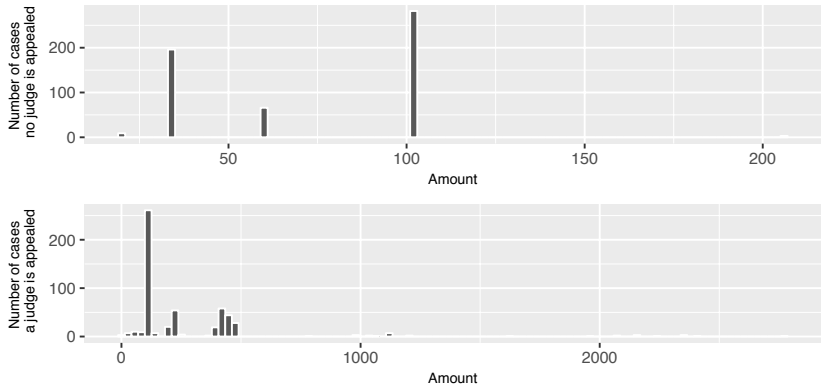

Figure 2: Histogram depicting the fine amount of cases with and without judge

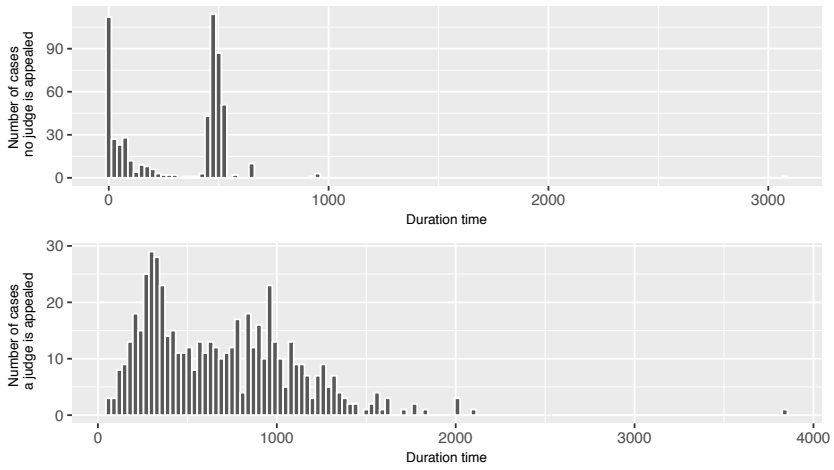

Figure 3: Histogram depicting the duration time of cases with and without judge

and 1 does not lead to the optimal homogeneities. This validates the need of a multi-perspective distance measure instead of the use of isolated measures. We therefore recommend the weighting of the activity perspective to a value between 0.6 and 0.9. The transition profile leads to a homogeneity around 0.76 and 0.78 for weights larger than 0.6. Analysing the resource perspective shows a similar distribution as the activity profile. With growing importance, growing weighting, the homogeneity is increasing as well. For a good weighting, one can determine the weight of one of those both perspectives to at least 0.6.

The histograms of the data perspective per event and per case reveal that high homogeneities only can derived with a weighting less or equal than 0.3.

We set a range for the weight of the activity perspective to larger or equal than 0.6. With this parametrization, we evaluate again the weighting of the other perspectives. The different weighting of the transition profile (Fig. 5) does not lead to clear differences in distributions of homogeneity.

Figure 6 shows the weighting of the resource perspective provided the weighting of the activity perspective. The resource perspective should be weighted in the range between 0.1 and 0.3 to increase to probability to get a result with a high homogeneity. A high weighting of the data perspective "duration time" aggregated for the whole case does not lead to a increase in homogeneity (Fig. 7). We suggest a weighting of that perspective of larger than 0 and smaller than

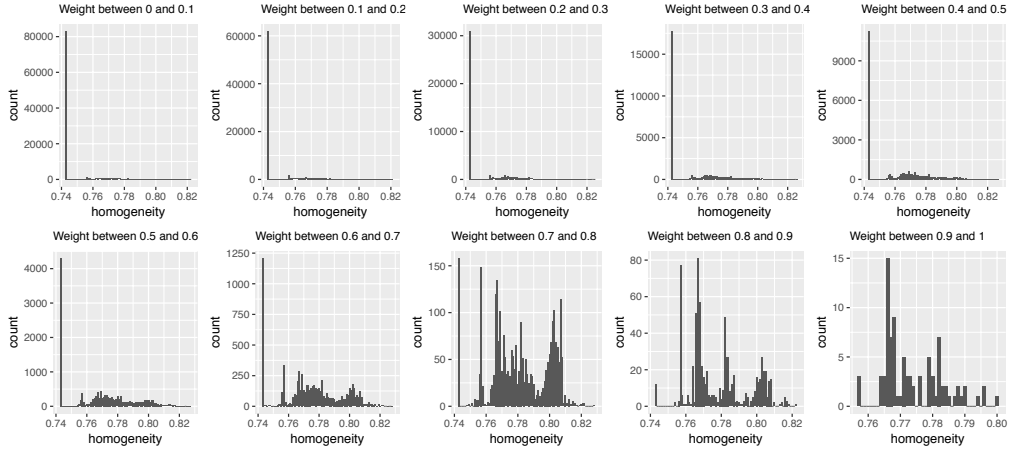

Figure 4: Hospital log: Relation between weighting of activity perspective and homogeneity

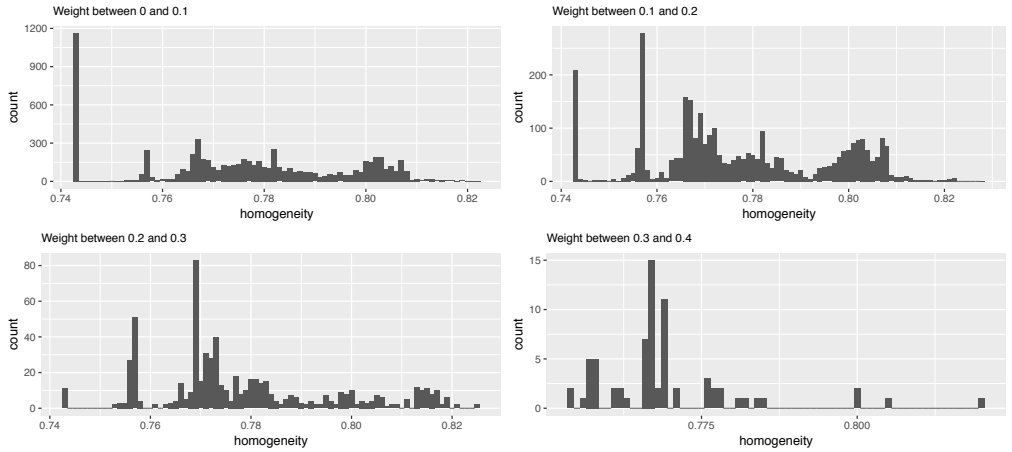

Figure 5: Hospital log: Relation between weighting of transition perspective and homogeneity under constraint activity perspective > 0.6

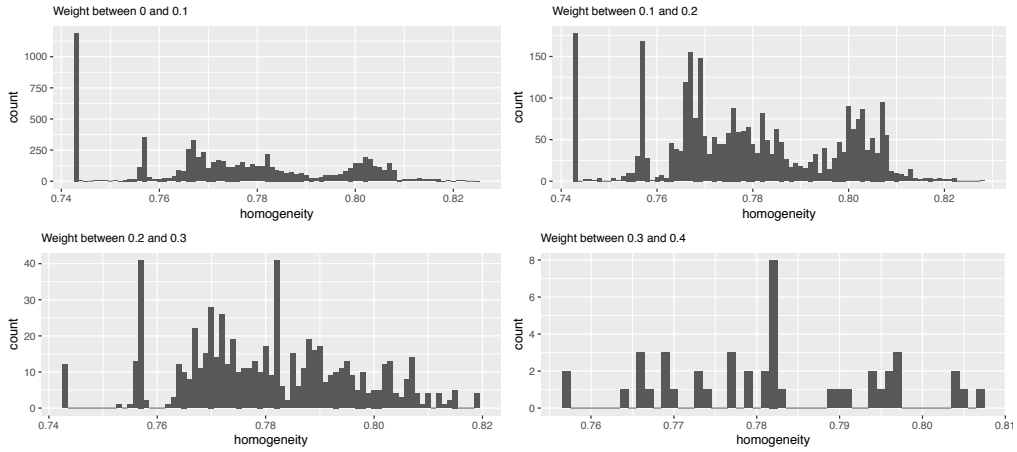

Figure 6: Hospital log: Relation between weighting of resource perspective and homogeneity under constraint activity perspective > 0.6

0.1. The data perspective on an event level has a higher impact on the homogeneity than the data perspective aggregated on case level (Fig. 8). We suggest a weighting between 0.1 and 0.3.

For a complex log like the hospital log (654 different activities, on average 151 events per case), the main perspectives are either Activity or Resource. The influence of the other perspectives is significant as well to obtain better cluster results. We analyze the histograms for the traffic log. The lowest value of homogeneity is 0.50. The clear correlation between importance of the activity or resource perspective and increasing results in homogeneity as seen for the hospital log can not be observed (Fig. 9 and Fig. 10). The data perspective duration time on event level seems to have a high impact on the homogeneity (Fig. 11).

We determine the weighting of the perspective duration based on event-level to larger or equal than 0.6 and analyze the influence of the other perspectives. For a weighting of > 0.1 of the activity perspective (Fig. 12) the majority of clusters have a high homogeneity around 0.65. A similar distribution is detected for the transition profile (Fig. 13). The resource perspective (Fig. 14) should be weighted with a value between 0 and 0.2. All other perspectives should be weighted with a value of the interval $[0; 0.1]$

We apply the activity weighting of the hospital log (activity perspective larger than 0.6) and analyse whether the weighting of the one log can be transferred to the other. By setting the weight for the activity perspective to at least 0.6 the weightings of the other perspectives can be analysed. Again, the important perspectives are resource (Fig. 15) and duration time on event level (Fig. 16). For clusters with high homogeneity the resource perspective and the perspective of the duration time on event level should be weighted between 0.2 and 0.4. The data perspectives on case level are again not as important as the data perspective on event level. The transition profile is not as important as activity or resource perspective.

The characteristics of the traffic log may be an explanation that the weighting of the activity perspective and resource perspective do not have a clear influence on the homogeneity as seen for the hospital log. Traffic log contains 1,111 traces, each trace contains on average 5 events. Overall only 10 different activities are executed. The log is not as complex as the hospital log, that is why the relation between high weighting and high homogeneity can not be detected clearly. But, as shown, the weighting of the hospital log is transferable and results in a set of clusters with high homogeneities as well.

4.4 Evaluation metrics

The objective of clustering traces from a process mining perspective is to facilitate the discovery of process models. We evaluate the clusters by evaluating the process models, which can be mined from the traces of each cluster. In meaningful clusters, all traces belonging to related cases are in the same cluster, unrelated traces are not. Furthermore, clusters are meaningful, when the derived process models are less complex, more comprehensible, and have a high degree of fitness (Bose and Aalst 2009). The process models are generated using the Inductive miner algorithm that is available in the ProM framework (Leemans et al. 2014). The discovered process models are represented in Petri nets. For each real-life event log, a process model from the entire log of traces is computed, as well as one process model from each cluster derived using the different clustering approaches.

Rozinat and Aalst (2008) propose the fitness metric to measure the conformance of the discovered process models to the related trace clusters. With the plugin *Replay a log on Petri Net for Conformance Analysis* in ProM, the fitness metric is derived (Aalst et al. 2012). Fitness measures the proportion process models capture the observed behavior seen with traces in the event log (Rozinat and Aalst 2008). The fitness of a process model is high (good) if the traces of a log can be replayed in the related process model. Fitness takes values in the range $[0; 1]$. Apart from fitness values, we measure the precision and generalization of the

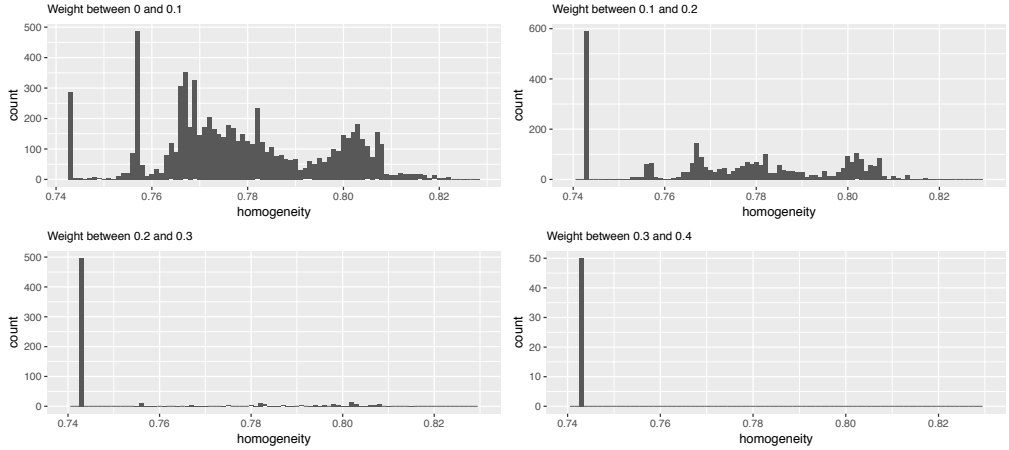

Figure 7: Hospital log: Relation between weighting of perspective "case-level duration time" and homogeneity under constraint activity perspective > 0.6

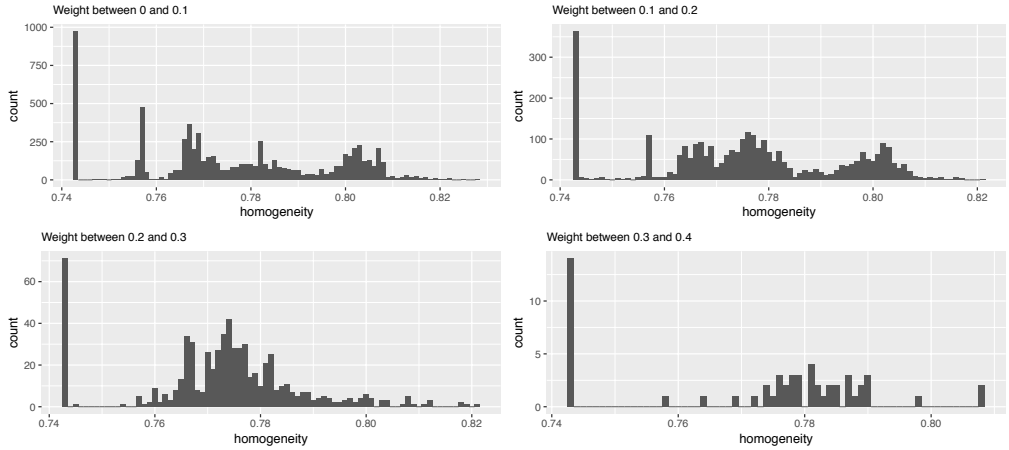

Figure 8: Hospital log: Relation between weighting of perspective "event-level duration time" and homogeneity under constraint activity perspective > 0.6

obtained solutions using the ProM plugin "Measure precision/generalization" (Adriansyah et al. 2013). Precision determines the model's ability to disallow behaviour which is not wanted, whereas generalization indicates whether the model is able to avoid overfitting (Broucke et al. 2014). With the intention of achieving the objective of simplifying the model complexity, we compute a two cluster solution for each approach. For the conformance and quality metrics, we calculate the weighted average of each fitness, precision, and generalization value over the resulting clusters per approach (Bose and Aalst 2009). We show that a comprehensive multi-perspective case distance leads to Petri nets with a higher conformance and quality compared to commonly used clustering approaches. We compare the metrics of the entire log to the multi-perspective distance measure to *(i)* CoTraDic, *(ii)* the equal weighting of the several perspectives, *(iii)* the isolated perspectives of the integrated distance measure. The results are shown in Tab. 3. The results show that clustering using the multi-perspective approach improves conformance and quality of both process models. Comparing the multi-view algorithms, we can observe that the multi-perspective approach

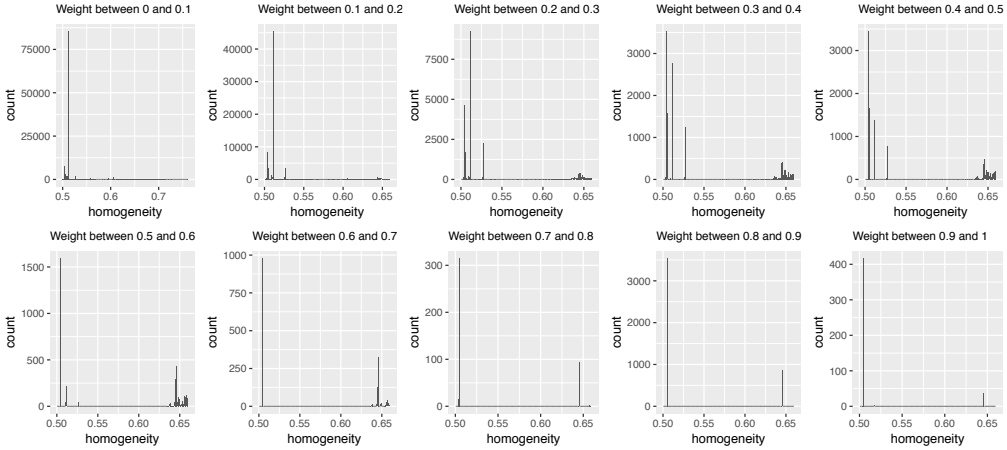

Figure 9: Traffic log: Relation between weighting of activity perspective and homogeneity

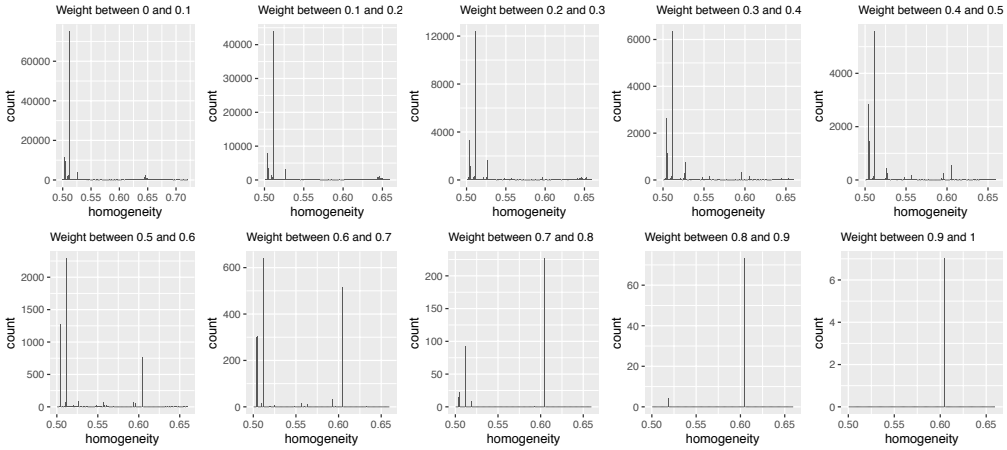

Figure 10: Traffic log: Relation between weighting of resource perspective and homogeneity

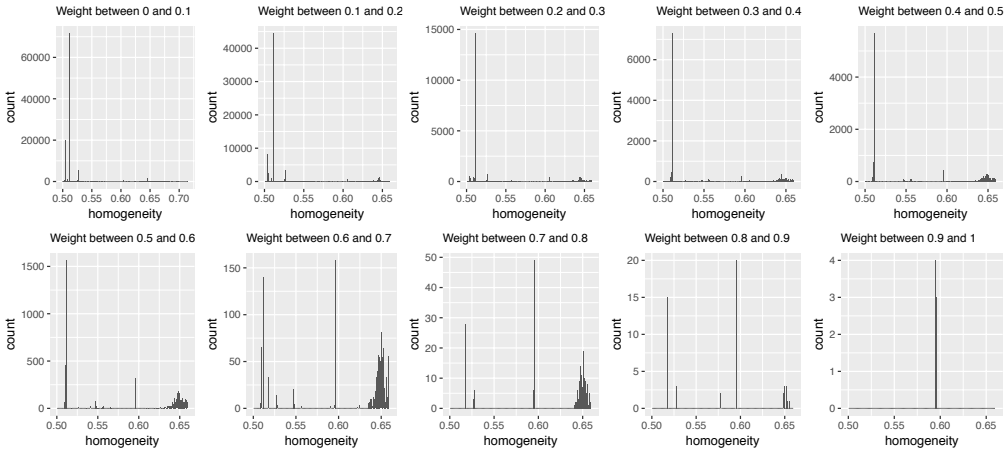

Figure 11: Traffic log: Relation between weighting of perspective "event-level duration time" and homogeneity

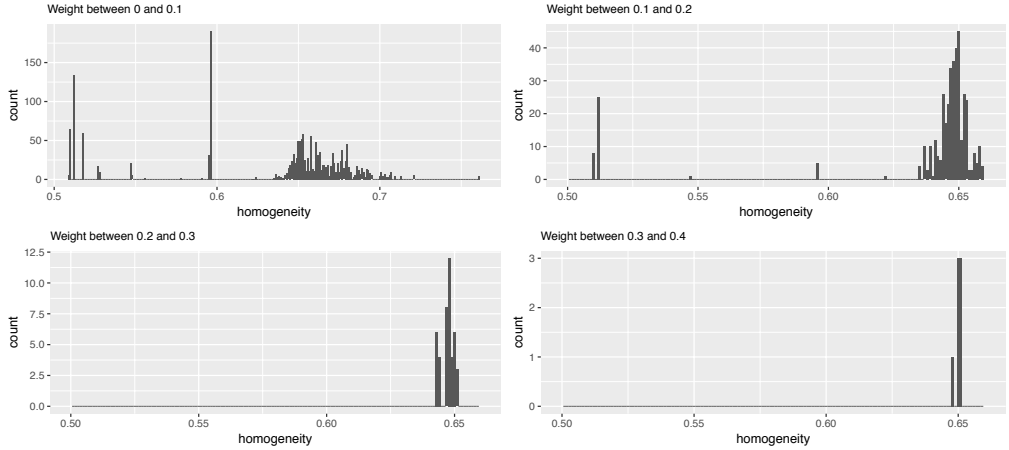

Figure 12: Traffic log: Relation between weighting of activity perspective and homogeneity under constraint "event-level duration time" > 0.6

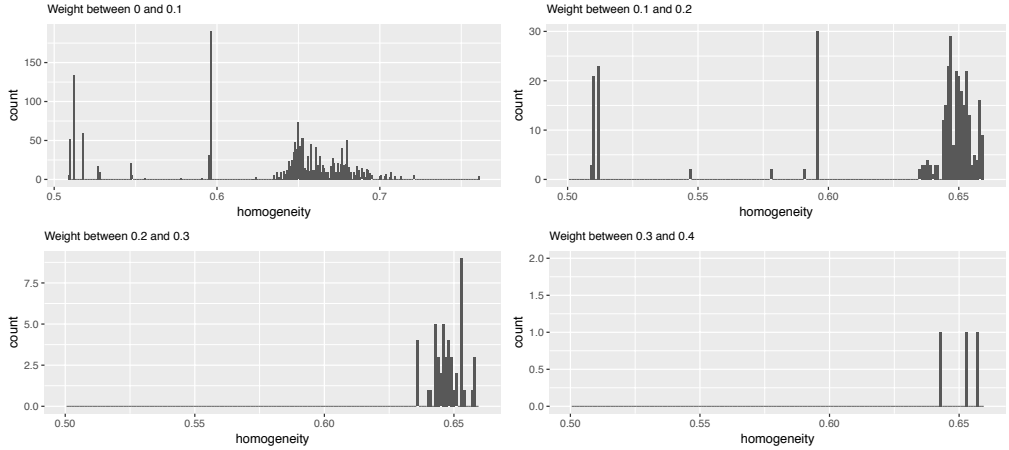

Figure 13: Traffic log: Relation between weighting of transition perspective and homogeneity under constraint "event-level duration time" > 0.6

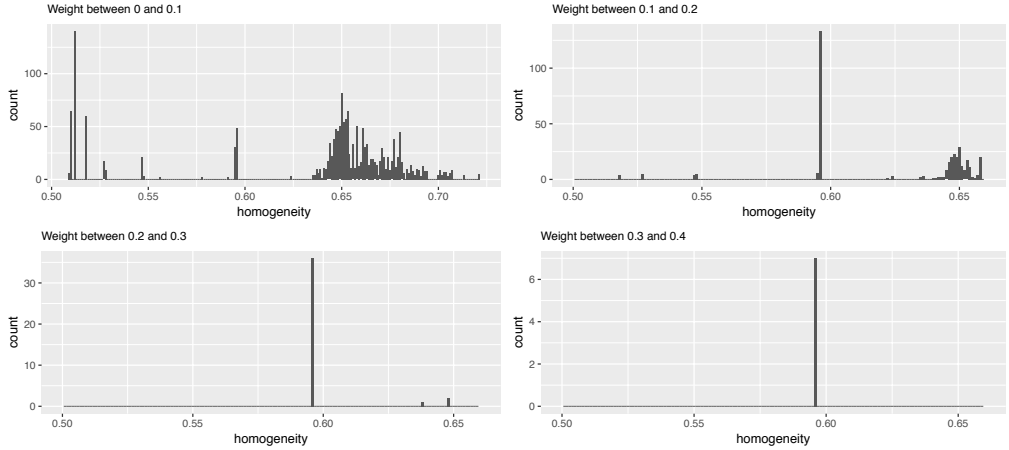

Figure 14: Traffic log: Relation between weighting of resource perspective and homogeneity under constraint "event-level duration time" > 0.6

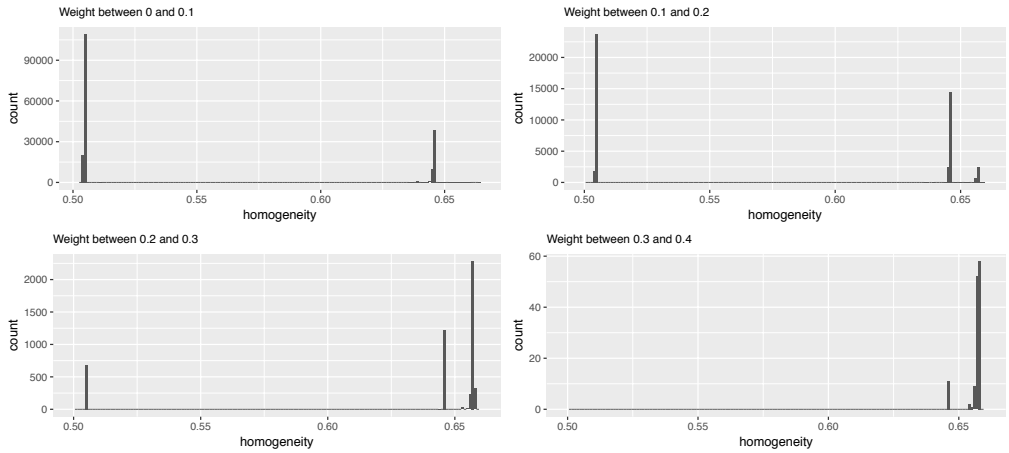

Figure 15: Traffic log: Relation between weighting of resource perspective and homogeneity under constraint activity perspective > 0.6

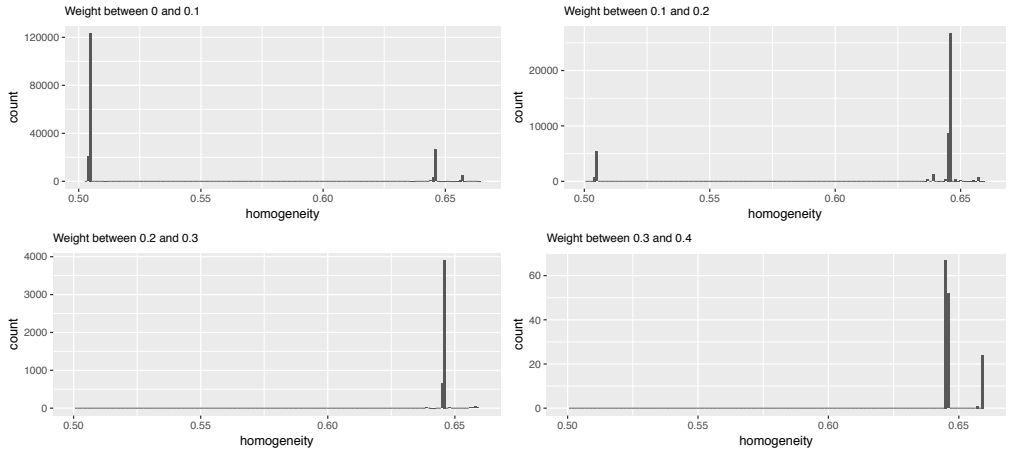

Figure 16: Traffic log: Relation between weighting of perspective "event-level duration time" and homogeneity under constraint activity perspective > 0.6

with optimized weights outperforms the two other approaches. The Co-Training strategy leads to a higher conformance than the entire event log but a lower quality regarding precision and generalization. This shows that the weighting of the several perspective is a crucial choice in order to gain process models with a higher conformance and quality. The equal weighting of the several perspectives only improves the conformance of the process model in case of the traffic log. Regarding the single view algorithms, it is not observable that one view outperforms the others. The evaluation validates the hypothesis that the proposed clustering approach outperforms other multi-view approaches and the single-view algorithms. The new approach enables increasing conformance and quality of generated clusters.

5 Strengths and Limitations of the presented approach

The presented clustering approach has its strengths but its weaknesses as well. Regarding the strengths, our clustering approach provides the possibility to consider multiple different perspectives and weight them individually. The distance measure is not limited to the presented perspectives. Even further perspectives can be included individually. In the evaluation, we present a way

Log	Fitness	Precision	Generalization
Hospital	0.93	-*	-
Multi	0.98	0.71	0.91
CoTraDiC	0.94	-	-
equal	0.93	0.65	0.77
activity	0.96	0.53	0.71
transition	0.97	0.61	0.86
resource	0.96	0.49	0.74
duration per case	0.97	0.56	0.81
duration per event	0.97	0.59	0.87
Traffic	0.90	0.49	0.88
Multi	0.98	0.76	0.94
CoTraDiC	0.95	0.43	0, 70
equal	0.95	0.54	0.67
activity	0.95	0.40	0.88
transition	0.97	0.66	0.92
resource	0.95	0.59	0.90
amount per case	0.95	0.53	0.66
amount per event	0.95	0.46	0.36
duration per case	0.95	0.54	0.67
duration per event	0.95	0.68	0.92

*Table 3: Cluster Approach Analysis: Conformance (fitness, precision, generalization) of the computed Petri nets. *Computations were limited to 24 hours of computational time.*

to discover the optimal weighting of the several perspectives in order to achieve homogeneous clusters. Even more, we propose a set of weights that is transferable to other logs. With the comprehensive multi-perspective clustering approach we are able to discover insights which go beyond a process variant analysis mainly based on an isolated perspective.

Our trace clustering approach has some limitations. For these first analyses, we chose simple metrics to calculate the distance measures of the several perspectives. They may not always capture the real similarity. It is also possible integrating more sophisticated metrics, e. g. considering the similarity of activities based on contextual information instead of labels. The guidelines to determine the optimal weighting result from the analysis of two event logs. Additional experiments on event logs are required to validate the weighting schema. As expected, the computational performance decreases with the size of a process log and the number of cases. Since for good and robust results, a large amount of data is needed, there is trade-off between performance and valid results. In our examples, we used a given characteristic of the process to compute cluster homogeneity. In some scenarios, such a characteristic is not available or has to be detected first. In sum, the multi-perspective clustering approach is an adaptable approach that can be easily applied to other event logs and enables to detect insights on the process log.

6 Conclusions and Future Work

In this paper, we presented a multi-perspective trace clustering approach that leads to more homogeneous clusters of process variants. Our approach is based on a multi-perspective distance measure that integrates activities, resources, and additional data attributes for a comprehensive determination of the similarity between traces. The evaluation with two real-life process logs shows that this approach significantly improves the conformance, quality, and homogeneity of resulting trace clusters. We proposed a weighting procedure and gave general guidelines to determine the optimal weights. We compared our approach to commonly used distance-measures and outperformed them. Our trace clustering approach has some limitations. The computation of the optimal weights is costly. We provide a weighting schema which can be transferred to other logs. And for long-term process log evaluations, the one-time computation of the weights is reasonable. For future work, we plan to combine the approaches of reducing complexity of an event log and simultaneously detect different process variants.

References

van der Aalst W. M. P., Adriansyah A., van Dongen B. (2012) Replaying history on process models for conformance checking and performance analysis. In: Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2(2), pp. 182–192

van der Aalst W. M. P., Schonenberg M. H., Song M. (2011) Time Prediction Based on Process Mining. In: Information Systems 36(2), pp. 450–475

van der Aalst W. M. P., Weijter A. J. M. M. (2004) Process Mining: A Research Agenda. In: Computers and Industry Vol. 70, pp. 231–244

Adriansyah A., Munoz-Gama J., Carmona J., van Dongen B. F., van der Aalst W. M. P. (2013) Alignment Based Precision Checking. In: La Rosa M., Soffer P. (eds.) Business Process Management Workshops. Lecture Notes in Business Information Processing Vol. 132. Springer, Berlin, Heidelberg, pp. 137–149

Appice A., Malerba D. (2016) A Co-Training Strategy for Multiple View Clustering in Process Mining. In: IEEE Transactions on Services Computing 9(6), pp. 832–845

Ballambettu N. P., Suresh M. A., Bose R. P. J. C. (2017) Analyzing Process Variants to Understand Differences in Key Performance Indices. In: Dubois E., Pohl K. (eds.) Advanced Information Systems Engineering. Lecture Notes in Computer Science Vol. 10253. Springer International Publishing, Cham, pp. 298–313

Baumann M. H., Baumann M., Schönig S., Jablonski S. (2014) Towards Multi-perspective Process Model Similarity Matching. In: Barjis J., Pergl R. (eds.) Enterprise and Organizational Modeling and Simulation. Lecture Notes in Business Information Processing Vol. 191. Springer, Berlin, Heidelberg, pp. 21–37

Baumann M., Baumann M. H., Schönig S., Jablonski S. (2014) Resource-Aware Process Model Similarity Matching. In: ICSOC Workshops. Lecture Notes in Computer Science Vol. 8954. Springer International Publishing, Cham, pp. 96–107

Bolt A., van der Aalst W. M. P. (2015) Multidimensional Process Mining Using Process Cubes. In: Gaaloul K., Schmidt R., Nurcan S., Guerreiro S., Ma Q. (eds.) Enterprise, Business-Process and Information Systems Modeling. Lecture Notes in Business Information Processing Vol. 214. Springer International Publishing, Cham, pp. 102–116

Bolt A., de Leoni M., van der Aalst W. M. P. (2018) Process variant comparison: Using event logs to detect differences in behavior and business rules. In: Information Systems 74, pp. 53–66

Bose R. P. J. C., van der Aalst W. M. P. (2009) Context Aware Trace Clustering: Towards Improving Process Mining Results. In: Proceedings of the SIAM International Conference on Data Mining (SDM 2009), pp. 401–412

Bose R. P. J. C., van der Aalst W. M. P. (2010) Trace Clustering Based on Conserved Patterns: Towards Achieving Better Process Models. In: Rinderle-Ma S., Sadiq S., Leymann F. (eds.) Business Process Management Workshops. Lecture Notes in Business Information Processing Vol. 43. Springer, Berlin, Heidelberg, pp. 170–181

vanden Broucke S. K. L. M., Weerdt J. D., Vanthienen J., Baesens B. (2014) Determining Process Model Precision and Generalization with Weighted Artificial Negative Events. In: IEEE Transactions on Knowledge and Data Engineering 26(8), pp. 1877–1889

Dijkman R., Dumas M., van Dongen B., Käärik R., Mendling J. (2011) Similarity of Business Process Models: Metrics and Evaluation. In: Information Systems 36(2), pp. 498–516

van Dongen B. F. (2011) Real-life event logs - Hospital log. https://doi.org/10.4121/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54

Duda R. O., Hart P. E., Stork D. G. (2000) Pattern Classification, 2nd Ed. Wiley Interscience New York, NY, USA

Dumas M., Rosa M. L., Mendling J., Reijers H. A. (2013) Fundamentals of Business Process Management. Springer, Berlin, Heidelberg

Ekanayake C. C., Dumas M., Garcia L., Rosa M. L. (2013) Slice, Mine and Dice: Complexity-Aware Automated Discovery of Business Process Models. In: Business Process Management. Lecture Notes in Computer Science Vol. 8094. Springer, Berlin, Heidelberg, pp. 49–64

Evermann J., Thaler T., Fettke P. (2016) Clustering Traces Using Sequence Alignment. In: Business Process Management Workshops. Lecture Notes in Business Information Processing Vol. 256. Springer International Publishing, Cham, pp. 179–190

Ferreira D., Zacarias M., Malheiros M., Ferreira P. (2007) Approaching Process Mining with Sequence Clustering: Experiments and Findings. In: Alonso G., Dadam P., Rosemann M. (eds.) Business Process Management. Lecture Notes in Computer Science Vol. 4714. Springer, Berlin, Heidelberg, pp. 360–374

Folino F., Greco G., Guzzo A., Pontieri L. (2011) Mining usage scenarios in business processes: Outlier-aware discovery and run-time prediction. In: Data & Knowledge Engineering 70(12), pp. 1005–1029

Francescomarino C. D., Dumas M., Maggi F. M., Teinemaa I. (2015) Clustering-Based Predictive Process Monitoring. In: ArXIv e-prints

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

Günther C. W., van der Aalst W. M. P. (2007) Fuzzy mining–adaptive process simplification based on multi-perspective metrics. In: Business Process Management. Lecture Notes in Computer Science Vol. 4714. Springer, Berlin, Heidelberg, pp. 328–343

Han J. (2005) Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA

Jablonski S., Schönig S., Röglinger M., Wyrtki K. M. (2015) R scripts. http://mp-clustering.kppq.de

Lee S., Kim B., Huh M., Cho S., Park S., Lee D. (2013) Mining transportation logs for understanding the after-assembly block manufacturing process in the shipbuilding industry. In: Expert Systems with Applications 40(1), pp. 83–95

Leemans S. J. J., Fahland D., van der Aalst W. M. P. (2014) Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour. In: Lohmann N., Song M., Wohed P. (eds.) Business Process Management Workshops. Lecture Notes in Computer Science Vol. 7927. Springer, Berlin, Heidelberg, pp. 66–78

de Leoni M., van der Aalst W. M. P., Dees M. (2016) A General Process Mining Framework for Correlating, Predicting and Clustering Dynamic Behavior Based on Event Logs. In: Information Systems 56(C), pp. 235–257

de Leoni M., Mannhardt F. (2015) Road Traffic Fine Management Process. https://doi.org/10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5

Li C., Reichert M., Wombacher A. (2011) Mining business process variants: Challenges, scenarios, algorithms. In: Data and Knowledge Engineering 70(5), pp. 409–434

Montani S., Leonardi G. (2014) Retrieval and Clustering for Supporting Business Process Adjustment and Analysis. In: Information Systems 40, pp. 128–141

Rebuge A., Ferreira D. R. (2012) Business Process Analysis in Healthcare Environments: A Methodology Based on Process Mining. In: Information Systems 37(2), pp. 99–116

Rozinat A., van der Aalst W. M. P. (2008) Conformance Checking of Processes Based on Monitoring Real Behavior. In: Information Systems 33(1), pp. 64–95

Schönig S., Cabanillas C., Jablonski S., Mendling J. (2016a) A framework for efficiently mining the organisational perspective of business processes. In: Decision Support Systems 89, pp. 87–97

Schönig S., Rogge-Solti A., Cabanillas C., Jablonski S., Mendling J. (2016b) Efficient and Customisable Declarative Process Mining with SQL. In: Nurcan S., Soffer P., Bajec M., Eder J. (eds.) Advanced Information Systems Engineering. Lecture Notes in Computer Science Vol. 9694. Springer International Publishing, Cham, pp. 290–305

Schönig S., Zeising M., Jablonski S. (2013) Supporting collaborative work by learning process models and patterns from cases. In: International Conference on Collaborative Computing: Networking, Applications and Worksharing. Piscataway, NJ, USA, pp. 60–69

Song M., Günther C. W., van der Aalst W. M. P. (2009) Trace Clustering in Process Mining. In: Ardagna D., Mecella M., Yang J. (eds.) Business Process Management Workshops. Lecture Notes in Business Information Processing Vol. 17. Springer, Berlin, Heidelberg, pp. 109–120

Thaler T., Ternis S. F., Fettke P., Loos P. (2015) A Comparative Analysis of Process Instance Cluster Techniques. In: Smart Enterprise Engineering: 12. Internationale Tagung Wirtschaftsinformatik, WI 2015, Osnabrück, Germany, pp. 423–437

Vaishnavi V., Kuechler W. (2004) Design Science Research in Information Systems. In: Association for Information Systems www.desrist.org/design-research-in-information-systems/

Verbeek H. M. W., Buijs J. C. A. M., van Dongen B. F., van der Aalst W. M. P. (2010) XES, XESame, and ProM 6. In: Information Systems Evolution - CAiSE Forum 2010, Hammamet, Tunisia, pp. 60–75

Weerdt J. D., vanden Broucke S. K. L. M., Vanthienen J., Baesens B. (2012) Leveraging process discovery with trace clustering and text mining for intelligent analysis of incident management processes. In: 2012 IEEE Congress on Evolutionary Computation, pp. 1–8

Weerdt J. D., vanden Broucke S. K. L. M., Vanthienen J., Baesens B. (2013) Active Trace Clustering for Improved Process Discovery. In: IEEE Transactions on Knowledge and Data Engineering 25(12), pp. 2708–2720

Yip A. M., Chan T. F., Mathew T. P. (2003) A Scale Dependent Model for Clustering by Optimization of Homogeneity and Separation. In: CAM Technical Report 03-37, Department of Mathematics, University of California