3-4-2015

A Comparative Analysis of Process Instance Cluster Techniques

Tom Thaler

Simon Felix Ternis

Peter Fettke

Peter Loos

A Comparative Analysis of Process Instance Cluster Techniques

Tom Thaler, Simon Ternis, Peter Fettke, Peter Loos

Institute for Information Systems at the German Research Center for Artificial Intelligence (DFKI GmbH) and Saarland University, Saarbrücken, Germany
{tom.thaler|simon.ternis|peter.fettke|peter.loos}@iwi.dfki.de

Abstract. The application of process mining and analysis techniques to the process logs of information systems often leads to highly complex results, e.g. in terms of a high number of elements in the mined model. Thus, clustering corresponding log files is mandatory in the context of an expedient analysis. Against that background, many cluster techniques have been developed during the last years but, at the same time, it is unclear how powerful they operate in particular application scenarios. Therefore, the paper at hand aims at analyzing and comparing the capabilities of existing cluster techniques with regard to different objectives. As a result, it is shown that some techniques are more suitable for the handling of particular scenarios than others and there are also general challenges in their application, which should be addressed in future work.

Keywords: Process Clustering, Process Mining, Business Process Management

1 Introduction

The execution of business processes often causes unexpected dynamics, e.g. in terms of their behavior depending on a variety of parameters. At the same time, upcoming legal regulations as well as industry or company standards make it necessary to consequently check the process behavior against different demands. Moreover, the need for an elicitation and analysis of not yet covered business processes as a model, are of major importance for today's companies. Process-supporting business software, like ERP and workflow systems, generally produces process execution logs, which serve as a basis for such inquiries. However, analyzing these logs with e.g. process mining techniques is challenging as they may contain very heterogeneous execution information on many different processes and process variants. Thus, deriving process models based on the raw logs often leads to models of high complexity in terms of the number of contained elements [27], which are hard to understand and to interpret.

Against that background, cluster techniques are used for separating the execution logs into groups containing similar process instances. Thereby, the intentions of clustering are manifold and range from the identification of different processes and process variants to the derivation of understandable process models with a limited number of elements. In fact, there are already isolated works like [28], which evaluate

12th International Conference on Wirtschaftsinformatik,
March 4-6 2015, Osnabrück, Germany

Thaler, T.; Ternis, S. F.; Fettke, P.; Loos, P. (2015): A Comparative Analysis of Process Instance Cluster Techniques, in: Thomas. O.; Teuteberg, F. (Hrsg.): Proceedings der 12. Internationalen Tagung Wirtschaftsinformatik (WI 2015), Osnabrück, S. 423-437

different process mining algorithms with process models from practice or [22] comparing selected process clustering techniques in specific contexts. However, a general overview on existing techniques as well a comprehensive comparison are missing. Moreover, it is still unclear how existing cluster techniques are characterized and how powerfully they operate in different application scenarios.

Hence, the paper at hand aims at filling that research gap in terms of analyzing the capabilities of existing techniques with regard to different objectives, whereby both theoretical analytical and practical empirical aspects are considered. Thus, the goal is not the evaluation of particular techniques in the context of sample experiments (which is partially carried out in the papers describing the algorithms and techniques) but their fundamental characterization, which is rooted in the established literature, as well as a comparative analysis of their capabilities in realistic contexts.

Within the theoretical analytical investigation, existing process instance cluster techniques are characterized by process mining specific and cluster theoretical aspects. In the practical empirical analysis, two areas of major interest are identified in the literature and serve as a basis for the design of two real life application scenarios. The first scenario analyzes the capabilities of existing techniques to separate a process log with regard to different processes, while the second scenario aims at reducing the complexity of the mined models and, thus, at improving its understandability.

In order to get a framework for the theoretical analytical investigation, a morphological box will be developed in section 2. Afterwards, in section 3, the relevant literature providing corresponding techniques is identified and characterized using that morphological box. In section 4, a selection of the cluster techniques is then analyzed in the practical empirical part of the work. The limitations of the analysis are discussed in Section 5, while Section 6 provides some concluding recommendations on the usage of particular techniques and their further development.

2 Morphological Box Describing Process Instance Cluster Techniques

2.1 Preliminary Note

Developing a morphological box describing process instance cluster techniques requires the identification of relevant aspects within the cluster theory in general and in the field of process mining in particular. The development of a cluster technique is generally motivated by a concrete objective. Since the objective essentially affects the design of a particular technique, it is of high interest in both the field of process mining and the comparative analysis at hand. In addition to that, the representation of traces is also important in that context as it affects the choice of a particular cluster method. The distance measurement and the cluster approach with its specific characteristics are two important cluster-theoretical aspects, between which the basic literature (e.g. [14]) generally distinguishes. With regard to the practical empirical analysis, the availability of an implementation of a technique is obligatory, too. Thus, both aspects are considered in the following as well.

2.2 Description of Characteristics

Objective. As mentioned in Section 1, one objective is the differentiation of business processes [6] which are covered by a particular log file. This is also a general challenge of process mining as processes which do not have any relationships to each other might be modeled in an integrated manner. Thus, it might be meaningful to derive not only one but several process models from a log file. If that differentiation is already done or if the log file a priori covers exactly one process, it may be necessary to identify different execution variants [20] or outliers [7]. Variants can be characterized by several aspects, as e.g. the proceeded activities, the involved employees, the affected commodity group, the process duration / costs et cetera [21]. As a refinement of that, outliers are variants with infrequent occurrence (e.g. exceptions). Against that background, the objective of process identification can be understood as a special case of variant identification. However, only that enables a detailed analysis of the as-is processes.

Another objective is to improve the understandability of the mined model(s). Hereby, it is distinguished between the reduction of complexity in terms of the number of elements in general and a decomposition of the resulting models in particular [5]. While the reduction of the number of elements solely leads to a growing number of single models, the decomposition also connects the resulting models in order to clarify their relationships. In addition, one might distinguish different levels of granularity. Thus, there are also cluster approaches focusing the hierarchization of the mined process model(s) [10].

Representation of Traces. Generally, a trace can be represented in an *abstract* and in a *concrete* manner. The abstract representation provides a mathematical abstract view on a trace by using the vector space model. Different properties or characteristics describing a trace from a particular point of view are transformed to numerical values and serve as the elements of a vector. The authors of [21] suggest some corresponding vector profiles including the control flow (activity profile and transition profile), organizational aspects (originator profile), specific case data (case attributes profile and event attributes profile) and performance aspects such as the size of a trace and the execution durations. The most common features are presented in Table 1.

In contrast to that, the *concrete* representation provides a linguistically exact view on the traces based on node labels without any transformations. This view is often used for the description of examples in the traditional process mining literature (e.g. traces ABC, ACD, ACE). Corresponding distance measures solely work on the recorded activity sequences.

Distance Measure. A distance measure is a numerical value, calculating the distance between two objects. The value ranges from 0 (the objects are equal) and has no general upper bound in most cases (completely different). However, there are also normalized distance measures, for which an upper bound exists. Generally, the distance between the objects *i* and *j* is equal to *j* and *i*. The used distance measure depends on

the representation of a trace. In case of an abstract trace representation, the most common distance measures are the Euclidean, the Hamming and the Jaccard distance but the correlation between vectors and the cosine distance are relevant in some approaches as well. In case of a concrete trace representation, different kinds of edit distances are used to measure the distance between two strings. Thus, the number of edit operations (insert, delete, move) needed for the transformation of one string into another are calculated, whereby it is possible to use fixed or dynamic costs [13] for the different operations. Furthermore, the Markov chain is sometimes used for the representation of clusters. In that case, a trace is allocated to the cluster, whose Markov chain has the highest probability to reproduce the trace. This strategy serves as an alternative to the traditional distance measures.

Cluster Approach. In order to divide a multiset of process instances into different groups, the basic idea is to determine the distance between all elements of the multiset and to put elements with low distances between each other in one group. The resulting groups should have a high inner density (low distance between all elements), while the distance between the produced groups should be high. Thereby, a variety of corresponding cluster approaches exists, which are generally divided into three categories – hierarchical, partitioning and density-based approaches [12, 14].

Table 1. Most common features for a vector representation of traces

Property	Description
activity profile [21]	number of occurrences of each function in a trace
originator profile [21]	number of events which have been caused by each originator
transition profile [21]	number of occurrences of each combination of two activities
case attributes profile [21]	individual case data attributes serve as the vector elements
event attributes profile [21]	number of events with particular attributes
performance profile [21]	trace length, case / task durations, minimum, maximum, mean and median time difference between events of a trace
custom profile [21]	individual case / event properties
maximal repeat feature set [1]	number of occurrences of each maximal repeat (a subsequence that occurs in a maximal pair; a maximal pair is a pair of identical subsequences, which cannot be extended without destroying the equality)
super maximal repeat feature set [1]	number of occurrences of each super maximal repeat (a maximal repeat, that never occurs as a substring of another maximal repeat)
near super maximal repeat feature set [1]	number of occurrences of each near super maximal repeat (a super maximal repeat, which is not contained in any other maximal repeat)
maximal repeat alphabet feature set [1]	sum of occurrences of each maximal repeat under the equivalence class of the repeat alphabet
super maximal repeat alphabet feature set [1]	sum of occurrences of each super maximal repeat under the equivalence class of the repeat alphabet
near super maximal repeat alphabet feature set [1]	sum of occurrences of each near super maximal repeat under the equivalence class of the repeat alphabet

Hierarchical approaches are subdivided into agglomerative and divisive algorithms, which differ in the order in which they create clusters. Agglomerative algorithms start with *n* clusters of size *1*, whereas divisive techniques start with *1* clusters of size *n*. An agglomerative cluster algorithm merges *two* clusters in every step until only one all-embracing cluster remains. On the contrary, divisive algorithms split an all-embracing starting cluster to *n* clusters of size *1*. Partitioning approaches construct *k* cluster centroids which are then iteratively altered. Density-based approaches focus on the inner density of each cluster. The input objects are examined to determine regions with a high density.

An important property of a cluster algorithm is the handling of the amount of clusters which should be produced. Three different characteristics can be distinguished, namely (1) the number of resulting clusters must be provided, (2) the algorithm automatically determines the number of clusters or (3) the maximal number of clusters must be provided as an upper bound. Another important property is the type of the cluster membership. Hard cluster algorithms allocate each input object to exactly one cluster, while fuzzy algorithms allow an allocation to multiple clusters at the same time.

With regard to particular cluster objectives, it might be meaningful to include an external validation directly to the cluster approach. E.g. in the context of process mining, the cluster approach of [3, 4] explicitly considers the fitness of the process models to the log data (within a cluster) they are mined from.

In addition to the three mentioned cluster categories, other approaches, especially neuronal networks, are used in the context of clustering process instance data as well.

Implementation. Generally, the implementation of a cluster approach is of high interest as it is the only procedure that allows the application of particular techniques in the context of an evaluation or a real world scenario.

Basically, one may distinguish between whether an approach is implemented or not. Depending on the evaluation objectives and on the general parameters, it may also be important to know the attributes of an implementation, e.g. being publically available, open or closed source or distributed in a free or a commercial manner.

Based on the description of the characteristics above, the morphological box presented in Table 2 describing existing process instance cluster techniques was derived.

3 Selection of Process Instance Cluster Techniques

In order to identify the relevant literature, the databases *Springer, ACM, IEEE Xplore, Ebsco, ISI Web of Science, ScienceDirect, Scopus* and *Google Scholar* were searched for the terms: “trace clustering” AND “process”, “sequence clustering” AND “process”, “clustering” AND “process instances”, “clustering” AND “process mining”, “clustering” AND “BPM”, “clustering” AND “log data”, “clustering” and “log files”. It was desist from further restrictions like a time limit. Moreover, a backwards search was conducted on known journal articles and conference proceeding. The identified

Table 2. Morphological box describing process instance cluster techniques

Process mining specific aspects	**Prime objective**	process identification (5%)	variant identification (40%)	outlier identification (5%)	reducing complexity (55%)	model decomposition (5%)	model hierarchization (10%)
	Trace representation	abstract (75%)			concrete (25%)		
Cluster-theoretical aspects	**Distance measure**	euclid (55%)	jaccard (10%)		cosine (10%)	edit distance (fix costs) (5%)	other (25%)
		hamming (5%)	correlation (5%)		markov chain (20%)	edit distance (variable costs) (10%)	
	Cluster approach	partitioning clustering (60%)	hierarchal clustering (40%)		density clustering (10%)	neuronal network (5%)	
	Cluster assignment	fuzzy (10%)			hard (90%)		
	#Clusters	predefined (40%)	maximum predefined (10%)		undefined (45%)	depending on other parameters (5%)	
Implementation aspects	**Tool distribution**	free (50%)	commercial (0%)		both (5%)	none (45%)	
	Source code	open (50%)		closed (5%)		not available (45%)	

Hint: The percentage values outline the occurrence within existing techniques. See Table 3 for details.

articles were selected concerning whether or not they consider the clustering of business process instance data. In case of more than one article developing a clustering approach (e.g. because of improvements or further developments), generally the newest article was taken into account.

Overall, 20 approaches were identified, which are now characterized using the developed morphologic box. 70% of the approaches name improving the understandability of the resulting models in general as the prime objective and also 55% focus the reduction of the complexity of the resulting model(s) in particular. The identification of different processes and process variants is the prime objective of 45% of the articles.

Furthermore, 75% of all approaches use an abstract trace representation, whereby the most often used distance measure is the Euclidean distance. Considering the cluster category, 60% of the approaches use a partitioning algorithm, even so a hierarchical clustering is applied in 40% (multiple assignments are possible, as some approaches provide the possibility to switch between different cluster algorithms). Furthermore, 45% of the approaches do not require an initial setting of the number of resulting clusters, however 40% do. Thus, only 5% allow the setting of a maximal number of resulting clusters. Moreover, 2 of the 20 approaches work with fuzzy clusters. The characterization of the particular process instance cluster techniques is presented in Table 3.

Within the analyzed papers, 10 different implementations where explicitly named: ProM 5 – DWS Mining & Analysis [2, 9], Microsoft SQL Server [6], ProM 5 – Trace Clustering [21], ProM 5 – Sequence Clustering [27], ProM 5 – Fuzzy Miner [24], reBPMN [17], Markov Cluster Algorithm [7], Medtric Process Mining Studio [20], ProM 6 – ActiTraC [3, 4] and Apromore [5].

Table 3. Literature analysis

			prime objective						trace representation		distance measure									cluster approach: category				cluster approach: features		implementation	
ID	Source	Year	Process Indentification	Variant Identification	Outlier Identification	Understandability: Complexity	Understandability: Decomposition	Understandability: Hierarchization	abstract	concrete	Euclid	Hamming	Jaccard	Correlation	Cosine	Markov chain	Edit-Distance with fixed costs	Edit Distance with variable costs	other	partitioning	hierarchal	density	neuronal network	fuzzy	#cluster [1]	distribution [2]	source [3]
1	[6]	2007	●	●						●						●				●					p	b	c
2	[8]	2009		●		●			●		●									●					m	n	n
3	[11]	2004		●					●		●									●					p	n	n
4	[9]	2006		●					●		●									●					u	f	o
5	[20]	2013		●						●						●				●					m	n	n
6	[16]	2012		●					●		●								●		●				u	n	n
7	[15]	2013		●					●			●	●	●	●						●				u	n	n
8	[19]	2014		●						●							●	●	●		●				u	n	n
9	[7]	2011			●				●							●						●		●	u	f	o
10	[5]	2013				●	●	●	●										●		●				u	f	o
11	[2]	2008				●			●		●									●					p	f	o
12	[21]	2008				●			●		●	●	●							●	●	●	●		d	f	o
13	[13]	2009				●				●								●			●				u	n	n
14	[27]	2010				●			●							●				●					p	f	o
15	[1]	2010				●			●		●										●				u	n	n
16	[24]	2010				●				●									●	●				●	p	f	o
17	[17]	2011				●			●		●				●						●				u	f	o
18	[4]	2012				●			●		●									●					p	f	o
19	[3]	2013				●			●		●								●	●					p	f	o
20	[10]	2005						●	●		●									●					p	n	n

Legend: **1:** p=predefined, u=undefined, m=maximum defined, d=depending on other parameters; **2:** n=none, f=free, c=commercial, b=both; **3:** n=not available, o=open, c=closed

4 Practical Empirical Analysis

4.1 Scenario Selection

The idea of the practical empirical analysis is to get insights on how powerfully existing process instance cluster techniques operate in different application scenarios. Thus, two concrete scenarios where selected, covering the most important objectives in that area. The importance was ascertained based on the number of indications as a prime objective for developing a particular process instance cluster technique.

Nearly 50 percent of the analyzed papers named the identification of different process variants as their prime objective, while one of them explicitly named process identification. Against the background that the differentiation of processes is a special case of differentiating process variants, the first scenario aims at the separation of a log file with respect to different processes. Moreover, 55% of all analyzed papers primarily aim at the reduction of complexity of the mined models. Thus, the second scenario focuses on this aspect. The selection of that scenario allows to address the prime objective of 90% of all identified process instance cluster techniques (Table 3).

4.2 Restriction of Selected Techniques

Since it is necessary to have an implementation of a particular process instance cluster technique in order to analyze its behavior and compare it to others, only those techniques for which a working implementation is available are selected. Under the consideration of these conditions, the following techniques are applied in analysis, as all of them work with the same input data (IDs of table 3 are given in the brackets): DWS Mining and Analysis in ProM 5 (4, 11), Trace Clustering in ProM 5 (12), Sequence Clustering in ProM 5 (14), Fuzzy Miner in ProM 5 (16), ActiTraC in ProM 6 (18, 19). A further limitation leading to a subsequent selection of the cluster techniques may be the vectors representing the single cases, as some of them require further case or task information apart from the regular log requirements of process mining (case, task, timestamp, originator). The only affected cluster technique is that from [21], as additional (partially individual) information on a case is described. However, the approach, as well as its implementation, allows the clustering without these data. Thus, no further restrictions were performed.

4.3 Scenario 1: Process Identification

The first scenario aims at the separation of a log file containing three different processes. The logs from the Incident Management process at RaboBank Group ICT [26], the loan application process at a financial institute from the Netherlands [25] and the translation process at the workflow system ANONYMIZED [23], consisting of 500 instances each and overlapping in time, were randomly extracted and aggregated

Table 4. Applied cluster techniques, number of cluster initially set to 3

		#instances				#instances				#instances		
C		**F**	**I**	**T**		**F**	**I**	**T**		**F**	**I**	**T**
1	**SC**	138	248	137	**TC**	290	0	1	**AT**	0	0	197
2		175	138	190		210	29	499		0	0	263
3		187	114	173		0	471	0		77	0	0

Legend: C=Cluster-No; SC=Sequence Clustering; TC=Trace Clustering; AT=ActiTraC; F=financial traces; I=Incident traces; T=translation traces

to one new log file containing 1,500 instances overall. It was now tried to automatically separate the log file concerning the three different processes.

In a first step, for all cluster techniques allowing an initial setting of the number of resulting clusters (Sequence Clustering, Trace Clustering, ActiTraC), that parameter was set to 3, all other parameters were set to default. Table 4 shows that the three cluster techniques produce substantially different clusters. The Sequence Clustering approach [27] derives three clusters containing a high amount of traces from all three processes. Since increasing the number of clusters to 6, 9, 12, 15 or 18 clusters leaded to the same result, it is concluded that the Sequence Clustering approach is not able to identify the different processes and adequately separate the log file in the intended manner. In contrast to that, the results of the other two approaches seem more promising. The third cluster of the Trace Clustering technique [21] solely contains instances from the Incident process and, except for one outlier, also the first cluster only contains instances from one process – the loan application process. Nevertheless, cluster two contains traces from all processes but it is considerable that the cluster contains (nearly) all traces from the translation process. Increasing the number of clusters for this approach leads to more detailed results as the clusters do not only become clearer with regard to the different processes but also with regard to the identification of different execution variants. A visualization of a clustering leading to 12 clusters is presented in Fig. 1a, where all clusters (except for 1.1) solely contain traces from exactly one process. Furthermore, the resulting clusters from ActiTraC [3, 4] are quite clear, as all three clusters solely contain instances from exactly one process. However, there are two variants of the translation process, while a cluster representing the Incident Management process is missing.

Also the cluster techniques which do not allow the initial setting of the number of resulting clusters were applied to the log files. The DWS Miner [2, 9] produced 12 clusters, whereby 11 of them solely contain traces from exactly one process and only one cluster covers all three processes (F: 65, I: 107, T: 209). These 11 clusters cover 4 variants of the loan application process, 4 variants of the Incident Management process and 3 variants of the translation process. The aggregated binary significance matrix produced by the fuzzy miner (Fig. 1b) very clearly shows the three different processes as clusters with their contained activities.

As a result, it is generally possible to separate a log file with regard to different processes with particular cluster techniques. However, apart from the Fuzzy Miner, none of the used approaches was able to generate three clusters containing the 500 traces of the corresponding processes.

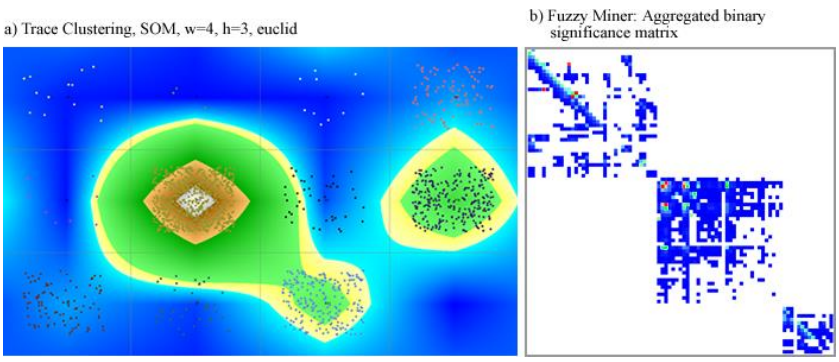

Fig. 1. Process separation with Trace Clustering and Fuzzy Miner

4.4 Scenario 2: Reducing Model Complexity

The second scenario aims at the reduction of complexity of the mined models. Therefore, a log file containing 1,500 traces from the above mentioned load application process were created and used as the data basis for clustering.

Against the background that complexity is often understood as the size of a model in terms of its amount of nodes, edges and the relation between each other, the achievement of that objective is quantified with corresponding metrics. Generally, the HeuristicsMiner of ProM is used for deriving a model G for all generated clusters.

Since the resulting heuristic nets solely contain activity nodes and directed arcs, only complexity metrics considering exactly these constructs can be taken into account. Thus, referring to [18], the metrics presented in Table 5 are selected to quantify the complexity of the resulting models. Indeed, the metrics CNC and CN have a very strong correlation. Though, as CNC represents a relation between arcs and nodes while CN provides an absolute difference, it is decided to calculate both. The corresponding values for the derived model (Fig. 2) from the unclustered log file can be found in the last column of the table.

In contrast to the first application, only 4 (instead of 5) different cluster approaches are considered, which is grounded in the implementation of the Fuzzy Miner (16). Thereby, it is neither possible to look at the concrete clusters nor to mine different models from them. All other cluster techniques are applied with different parameterizations in order to find the best possible solution. An abstract of the aggregated result is presented in Table 5. All generated clusters from the applied cluster techniques lead to models with lower complexity than the unclustered log file in terms of the mentioned complexity metrics. However, there are substantial differences in the clusters produced by the different techniques. For example the Sequence Clustering approach [27], which is applied in 5 different configurations (#c=3,6,9,12,15), produces clusters whose heuristic nets show a significantly higher amount of nodes and arcs than those from the other approaches (with parameterizations leading to a similar number of clusters). This can also be seen in the other metrics. However, the density seems to be quite low, which is grounded in the generally high amount of nodes. Thus, one should interpret these metrics only under the additional consideration of the other metrics.

The DWS [2, 9], the ActiTraC [3, 4] and the Trace Clustering [21] approaches lead to clusters whose models have higher density values than the unclustered one, which is another effect of the lower number of nodes and arcs.

The ActiTraC approach always produces the same clusters in the standard configuration. E.g. if the number of clusters is set to 3 in one run and set to 6 in another, the first three clusters of the 6C-run are equal to the clusters derived with the 3C-run. In that configuration, only equal instances or part-of instances are considered, which are ordered by relevance (number of affected instances). Thus, the 6th cluster of the 6C-run only covers 19 instances, so that it was not meaningful to set the number of clusters to a higher value in the standard configuration. In order to demonstrate the results, the heuristic nets derived from the produced clusters are presented in Fig. 2b (without the 6th cluster, as it covers all other traces). One can see that these 5 simple models (in terms of complexity) describe over 40% of the whole log. In fact, the CN and the CNC values are generally lower than those of the other approaches.

Within the applied settings, the ActiTraC application with an ICS-fitness of 0.95 leads to the best values with c=5. The approach also presents the most promising overall results in terms of the metrics. Another special phenomenon can be observed in the behavior of the Trace Cluster approach [21], which was applied in 7 different configurations (w=1, h=3; w=2, h=3; w=3, h=3; w=4, h=3; w=4, h=4; w=5, h=3;

Table 5. Complexity measurement after log clustering with different approaches

		Seq 3	Seq 6	Seq 9	Seq 12	Seq 15	DWS Std.	DWS 5-5-5-10	ActiTraC 3 Std	ActiTracC 6 Std	ActiTraC 6 0.95 ICS	TC 1-3	TC 2-3 *(equal to unclustered)*
	#c	3	6	9	12	15	4	6	4	7	7	3	1
$\|A\|$	min	94	68	34	54	34	31	10	**2**	**2**	7	**2**	**141**
	avg	101.3	74.5	69.6	65.9	58.8	62.3	41.7	21.5	**15.6**	47.3	50.0	**141.0**
	max	109	89	80	81	86	102	108	77	75	**61**	133	**141**
$\|N\|$	min	35	32	20	27	23	11	6	**3**	**3**	6	**3**	**36**
	avg	35.7	34.7	31.8	32.3	30.9	22.5	19.0	12.0	**10.1**	29.9	16.7	**36.0**
	max	**36**	**36**	**36**	**35**	**35**	**35**	**36**	**36**	**36**	**35**	**36**	**36**
CNC	min	2.611	1.889	1.700	1.806	1.478	2.214	1.667	**0.667**	**0.667**	1.167	**0.667**	**3.917**
	avg	2.842	2.150	2.166	2.039	1.893	2.753	2.015	1.076	**1.062**	1.528	1.908	**3.917**
	max	3.028	2.472	2.438	2.531	2.688	3.182	3.000	2.139	2.083	**1.743**	3.694	**3.917**
CN	min	59	33	15	26	12	18	5	**0**	**0**	2	**0**	**106**
	avg	66.7	40.8	38.8	34.6	28.9	40.8	23.7	10.5	**6.4**	18.4	34.3	**106.0**
	max	67	54	47	50	55	68	73	42	40	**27**	98	**106**
Δ	min	0.075	0.054	0.059	0.056	**0.040**	0.086	0.057	0.061	0.060	0.045	0.106	**0.112**
	avg	0.082	**0.064**	0.071	0.065	**0.064**	0.167	0.160	**0.224**	0.202	0.075	0.192	0.112
	max	0.087	**0.072**	0.089	0.082	0.087	0.318	**0.333**	**0.333**	**0.333**	0.233	**0.333**	0.112

Clustering: Seq x = Sequence Clustering with x clusters, DWS = DWS Miner with standard parameter and with clusters per split = max feature length = max splits = 5 and #features = 10, ActiTraC x Std = ActiTraC with x clusters and standard parameters, TC x-y = Trace Clustering with width=x and height=y; #c=number of resulting clusters; $|A|$= number of arcs; $|N|$= number of nodes; $CNC := \frac{|A|}{|N|}$ coefficient of connectivity; $CN = |A| - |N| + 1$ cyclomatic number; $\Delta := \frac{|A|}{|N|*(|N|-1)}$ density; highest and lowest values of each line are written in bold.

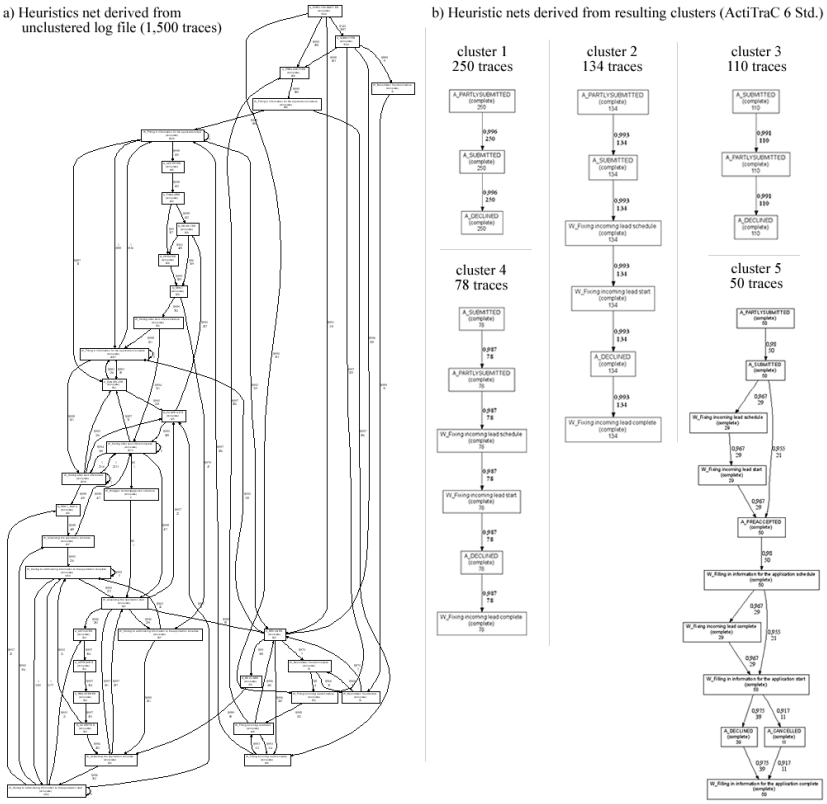

Fig. 2. Heuristics net derived from unclustered log file and from ActiTraC clusters

w=5, h=4). The last four runs produced three clusters each, which are equal in all execution variants - one cluster with 8 arcs and 6 nodes, one with 3 arcs and 3 nodes and one with 132 arcs and 36 nodes. It was not possible to derive more than three clusters with the applied parameters and the configuration w=2 and h=3 was not able to derive any cluster.

In a nutshell, there is a high amount of process instances only containing 2 to 10 activities. All approaches, except for the Sequence Clustering [27], found such a cluster, thus, the approaches are generally suitable for the identification of process variants. However, the results highly depend on the parameterization of the particular cluster techniques. On the contrary, the Sequence Clustering approach [27] produced clusters covering a homogenous amount of instances but with a comparatively high complexity.

5 Limitations

A limitation concerning the practical empirical analysis is the parameterization of the cluster approaches with regard to the different application scenarios. In fact, several

parameterizations were tested, whereby a range leading to the best result in the borders of that parameterization could be identified. However, apart from the Sequence Clustering approach [27] there is an infinite number of possible parameterizations, so that the results of the analysis are only valid in the borders of the ones applied. Thus, it cannot be excluded that other parameterizations lead to other results. Furthermore, although two scenarios with real execution data were conducted, it cannot be guaranteed that the cluster techniques perform differently in other scenarios.

Moreover, the Trace Clustering approach [21] allows a plethora of different configurations, e.g. in varying the vector characteristics, choosing another distance measure or another cluster approach. The paper at hand only analyzed the behavior of the configuration which was identified as the best performing in [21].

6 Conclusion

In summary, the paper at hand identified the currently available techniques for clustering business process instance data, characterized them based on a developed morphological box and analyzed their capabilities in two different application scenarios – (1) separating a log file containing traces from different processes and (2) improving the understandability of the resulting models in terms of reducing their complexity.

As a result of that comparative analysis, some of the available techniques are suitable for the handling of different objectives within bounds – others are less suitable. The Sequence Clustering approach [27] was not able to separate different processes and led to underwhelming results in the context of reducing the complexity of the resulting models. The Trace Clustering approach [21] presents the highest variability as it allows to change the trace characteristics, the distance measure and the clustering approach, thus, it is applicable in manifold contexts. Using the approach in the *most promising* configuration as mentioned in [21], led to good results in separating traces from different processes. At the same time and in comparison to other approaches, it could not convince in reducing the complexity of the resulting models. However, another configuration may lead to much better results with regard to that objective. Within the first scenario, the Fuzzy Miner [24] was the only approach undoubtedly detecting the three different processes, but also ActiTraC [3, 4] and DWS [2, 9] produced promising results in that context. However, they were not able to derive 3 clusters each containing all traces of one process. ActiTraC [3, 4] was also able to reduce the complexity of the mined models (scenario 2) in a higher extent than the other approaches but also DWS seems to be adequate for handling that objective.

A further important finding is the parameterization being the main challenge applying the available techniques in concrete scenarios. Even if the parameters are well known in detail, it is barely possible to set them in an expedient manner. Since that setting also highly depends on the application scenario and on the given data, a general recommendation cannot be provided. Moreover, heuristics predicting the quality of the resulting clusters and determining an optimal configuration are missing.

Thus, there is a need for heuristics recommending promising settings for particular approaches based on the context. E.g. the number of resulting clusters could be set to

the number of blocks resulting from an activity correlation matrix, similar to those in Fig. 1b. Developing such heuristics should be focused in future work and would highly improve the results and the usability of already existing techniques.

References

1. Bose, R.P., van der Aalst, W.M.P.: Trace Clustering Based on Conserved Patterns: Towards Achieving Better Process Models. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) Business Process Management Workshops, vol. 43, pp. 170-181. Springer Berlin Heidelberg (2010)
2. de Medeiros, A.K.A., Guzzo, A., Greco, G., van der Aalst, W.M.P., Weijters, A.J.M.M., van Dongen, B.F., Saccà, D.: Process Mining Based on Clustering: A Quest for Precision. In: ter Hofstede, A., Benatallah, B., Paik, H.-Y. (eds.) Business Process Management Workshops, LNCS 4928, pp. 17-29. Springer, Berlin (2008)
3. De Weerdt, J., van den Broucke, S., Vanthienen, J., Baesens, B.: Active Trace Clustering for Improved Process Discovery. Knowledge and Data Engineering, IEEE Transactions on 25, 2708-2720 (2013)
4. De Weerdt, J., van den Broucke, S.K.L.M., Vanthienen, J., Baesens, B.: Leveraging process discovery with trace clustering and text mining for intelligent analysis of incident management processes. In: Evolutionary Computation (CEC), 2012 IEEE Congress on, pp. 1-8. (Year)
5. Ekanayake, C.C., Dumas, M., García-Bañuelos, L., La Rosa, M.: Slice, Mine and Dice: Complexity-Aware Automated Discovery of Business Process Models. In: Daniel, F., Wang, J., Weber, B. (eds.) Business Process Management, LNCS 8094, pp. 49-64. Springer, Berlin (2013)
6. Ferreira, D., Zacarias, M., Malheiros, M., Ferreira, P.: Approaching Process Mining with Sequence Clustering: Experiments and Findings. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) Business Process Management, vol. 4714, pp. 360-374. Springer Berlin Heidelberg (2007)
7. Folino, F., Greco, G., Guzzo, A., Ponieri, L.: Mining usage scenarios in business processes: Outlier-aware discovery and run-time prediction. Data Knowl. Eng. 70, 1005-1029 (2011)
8. Folino, F., Greco, G., Guzzo, A., Pontieri, L.: Discovering Multi-perspective Process Models: The Case of Loosely-Structured Processes. In: Filipe, J., Cordeiro, J. (eds.) Enterprise Information Systems, vol. 19, pp. 130-143. Springer Berlin Heidelberg (2009)
9. Greco, G., Guzzo, A., Ponieri, L., Sacca, D.: Discovering expressive process models by clustering log traces. Knowledge and Data Engineering, IEEE Transactions on 18, 1010-1027 (2006)
10. Greco, G., Guzzo, A., Pontieri, L.: Mining Hierarchies of Models: From Abstract Views to Concrete Specifications. In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) Business Process Management, vol. 3649, pp. 32-47. Springer Berlin Heidelberg (2005)
11. Greco, G., Guzzo, A., Pontieri, L., Saccà, D.: Mining Expressive Process Models by Clustering Workflow Traces. In: Dai, H., Srikant, R., Zhang, C. (eds.) Advances in Knowledge Discovery and Data Mining, LNAI 3056, pp. 52-62. Springer, Berlin (2004)
12. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc. (2011)

13. Jagadeesh Chandra Bose, R.P., van der Aalst, W.M.P.: Context Aware Trace Clustering: Towards Improving Process Mining Results. Proceedings of the SIAM International Conference on Data Mining, SDM 2009, pp. 401-412, Sparks, Nevada, USA (2009)
14. Jain, A.K., Dubes, R.C.: Algorithms for clustering data. Prentice-Hall, Inc. (1988)
15. Lee, S., Kim, B., Huh, M., Cho, S., Park, S., Lee, D.: Mining transportation logs for understanding the after-assembly block manufacturing process in the shipbuilding industry. Expert Systems with Applications 40, 83-95 (2013)
16. Luengo, D., Sepúlveda, M.: Applying Clustering in Process Mining to Find Different Versions of a Business Process That Changes over Time. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) Business Process Management Workshops, vol. 99, pp. 153-158. Springer Berlin Heidelberg (2012)
17. Marchetto, A., Di Francescomarino, C.: Parameterised trace selection technique for process model recovering. Software, IET 5, 563-575 (2011)
18. Melcher, J.: Process Measurement in Business Process Management – Theoretical Framework and Analysis of Several Aspects. KIT Scientific Publishing, Karlsruhe (2012)
19. Montani, S., Leonardi, G.: Retrieval and clustering for supporting business process adjustment and analysis. Information Systems 40, 128-141 (2014)
20. Rebuge, Á., Ferreira, D.R.: Business process analysis in healthcare environments: A methodology based on process mining. Information Systems 37, 99-116 (2012)
21. Song, M., Günther, C.W., Van der Aalst, W.M.P.: Trace Clustering in Process Mining. In: Ardagna, D., Mecella, M., Yang, J. (eds.) Business Process Management Workshops, LNBIP 17, pp. 109-120. Springer, Berlin (2008)
22. Song, M., Yang, H., Siadat, S.H., Pechenizkiy, M.: A comparative study of dimensionality reduction techniques to enhance trace clustering performances. Expert Syst. Appl. 40, 3722-3737 (2013)
23. Thaler, T., Fettke, P., Loos, P.: Process Mining - Fallstudie leginda.de. HMD Praxis der Wirtschaftsinformatik 293, 56-66 (2013)
24. van Dongen, B.F., Adriansyah, A.: Process Mining: Fuzzy Clustering and Performance Visualization. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) Business Process Management Workshops, vol. 43, pp. 158-169. Springer Berlin Heidelberg (2010)
25. Van Dongen, B.F., Weber, B., Ferreira, D.R.: Business Process Intelligence Challenge (BPIC'12). http://www.win.tue.nl/bpi/2012/challenge (2012)
26. Van Dongen, B.F., Weber, B., Ferreira, D.R., De Weerdt, J.: Business Process Intelligence Challenge (BPIC'14). http://www.win.tue.nl/bpi/2014/challenge (2014)
27. Veiga, G.M., Ferreira, D.R.: Understanding Spaghetti Models with Sequence Clustering for ProM. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) Business Process Management Workshops, vol. 43, pp. 92-103. Springer Berlin Heidelberg (2010)
28. Wang, J., Tan, S., Wen, L., Wong, R.K., Guo, Q.: An Empirical Evaluation of Process Mining Algorithms based on Structural and Behavioral Similarities. In: Ossowski, S., Lecca, P. (eds.) Proceedings of the 27th Annual ACM Symposium on Applied Computing, pp. 211-213. ACM, Trento, Italy (2012)