Process variance analysis and configuration in the Public Administration sector

Flavio Corradini[a], Caterina Luciani[a], Andrea Morichetta[a] and Andrea Polini[a]

[a] University of Camerino, School of Science and Technology, Via Madonna delle Carceri, 9 62032 Camerino (MC) - Italy

Abstract

This paper presents a three-layered methodology to contrast variants of services offered by Municipalities with the main aim of improving their business processes re-engineering as well as other significant phases of the software life cycle, such as configuration and maintenance. The methodology makes it possible to detect discrepancies or alignments among services' variants. It relies on execution logs and applies clustering algorithms to reduce the huge amount of available logs into few clusters of "equivalent" executions. Then variance mining becomes a cornerstone to contrast clusters representatives and enables analysis on the offered services or those a specific Municipality would like to offer. The methodology has been validated on real case studies.

Keywords

Variance analysis, Process variant, Business process, Process comparator

1. Introduction

Every day, Municipalities provide to the citizens a number of different services by means of PAIS (process-aware information system). PAIS is a software system that bases its execution logic on business process models. These "business processes" even though similar in scope, may vary from Municipality to Municipality. The different versioning processes are called *variants*. Just to cite a few examples, there might be differences in the internal management and organisation, such as the human resources involved to carry out specific tasks, or in the process control flow because of different locally-applicable laws, but also in the way services are exposed to citizens, because of the increasing availability of digital services the Public Administrations can rely on.

Variants are part of the Municipalities' information system and as such can provide useful insights. In this paper, we concentrate on the usage of variants to get information useful to contrast their business processes and to improve their re-engineering as well as other phases of the software life cycle such as configuration and maintenance. Just to mention a few examples, our methodology aims at detecting "anomalous" tasks among variants, bottlenecks to be removed to improve services performance, compliance concerning municipalities guidelines or local laws, best practices to be replicated, or trends on the software functionalities depending on the territories or Municipalities' size.

The proposed **3L** methodology depicted in Figure 1 exploits the log files generated by running variants on PAIS systems. Such log

Proccedings of RTA-CSIT 2021, May 2021, Tirana, Albania

✉ flavio.corradini@unicam.it (F. Corradini); caterina.luciani@unicam.it (C. Luciani); andrea.morichetta@unicam.it (A. Morichetta); andrea.polini@unicam.it (A. Polini)
0000-0002-0877-7063 (F. Corradini); 0000-0001-7116-9338 (C. Luciani); 0000-0002-9421-8566 (A. Morichetta); 0000-0002-9421-8566 (A. Polini)

files provide information on data and activities on variants execution and, hence, provide suitable and useful information for our purposes. By contrasting variant log files we detect variants differences/similarities that allow analysis on the offered services or those a specific Municipality would like to offer. Variability mining becomes, hence, a significant cornerstone of our methodology. We exploit suitable techniques for approaching variability and provide a way to deal with many variants because this is the case for our application domain.

The following three-layered architecture describes in more detail our proposal.

LEVEL 1 Rely on the PAIS – process-aware information system – (more details on next section 2) of any Municipality [1] and collect logs regarding variants of specific services.

LEVEL 2 Apply clustering algorithms to the (huge) set of log variants. The clustering has been done on logs exposing the same activities and a "closed" execution flow (within a fixed interval) for the corresponding activities.

In our application domain, Municipalities, the clustering considerably reduces to few clusters (of "equivalent" logs). We elect one representative log for each cluster.

LEVEL 3 Contrast the clusters representatives through algorithms of variance mining. We are actually using the Process Comparator in [2] as a basic algorithm for variant analysis techniques.

The 3L methodology will be evaluated on real data provided by a PAIS software installed in eight thousand Italian municipalities. The software allows users to manage all the processes that can take place in a municipality, from registration at the registry office to change of residence. The software is highly configurable and this gives rise to a great deal of variability. The rest of the paper is organized as follows. The next section contains a brief introduction to PAIS – process-aware information system. Section 3 introduces two variance mining algorithms for comparing two variants. Section 4 describes the validation of Process Comparator algorithm on our data. Section 5 proposes a collection of works on the comparison between variants Section 6 is devoted to concluding remarks and further work.

2. Background

A PAIS (process-aware information system) is a process management and execution software that enables the separation of process logic from application code. The logic is expressed in terms of the process model, in this way, monolithic applications can be broken down into smaller services. This architecture makes it easier to maintain the code, e.g. a service can be modified without having to change the others. PAIS is therefore a tool capable of expressing the flexibility needed to evolve processes and manage exceptions. [1]

PAIS can be observed from different perspectives: functional, behavioural, organisational, operational, and temporal.

The functional perspective concerns the activities that are performed. They constitute the simplest unit of the process model and require human or machine resources to be executed. The behavioural perspective concerns the control flow between activities, i.e. the order in which they are performed. The languages that have been developed to express control flow also allow the expression of notions such as succession, parallel, conditional, and loops. The information perspective concerns data objects and data flow. In data-driven process models it is related to the behavioural perspective. The organisational perspective concerns actors, roles, and organisational units and their relationships. The operational perspective relates to the control

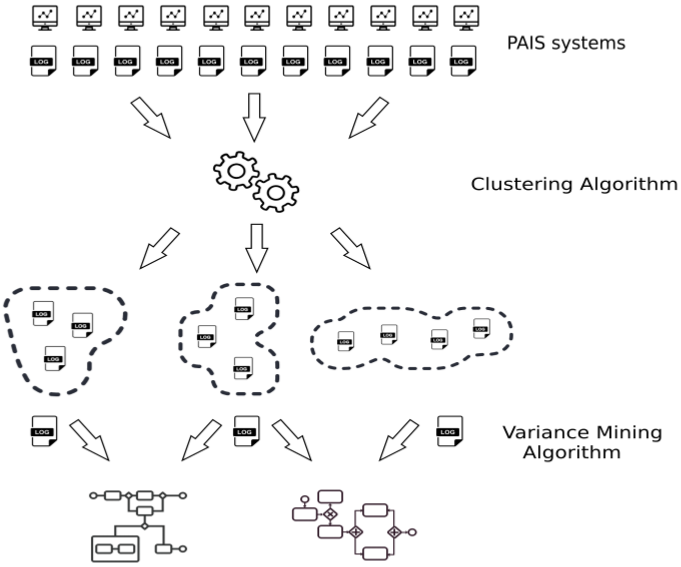


Figure 1: 3L Methodology

flow of activities, where they are considered as black-boxes. The time perspective concerns e.g. activity deadlines, duration, and waiting time between one activity and another.

A business model may present variability according to each of these perspectives. One of the most frequently used techniques for dealing with variability is process mining.

Process mining is a set of applications of data science to process science, where process science is understood as the common field between information technology and management science [3].

Through process mining, business process execution logs can be analysed according to four categories of techniques: automated process discovery (extraction of a model from a log), conformance checking (comparison of a log with the model to identify differences), performance mining (performance monitoring), variant analysis (comparison of variants) [4].

Variant analysis techniques were used in our case study to gain interesting insights.

3. Variance Analysis Algorithms

In literature, there are several approaches to comparing variants. Here below we compare the most used variance analysis algorithms suitable for our methodology.

In [2] Bolt, Leoni, and van der Aalst present a technique and a ProM tool (Process Comparator), for comparing two variants for both control flow and performance. The logs are represented as annotated transition systems, and statistical tests are then performed to identify significant differences between the two models. Consider the log in Fig. 1 and break it down into two sub-logs, where the first two traces belong to sub-log 1 and the third to sub-log 2.

Trace ID	Activity
1	A
1	B
1	C
1	D
2	A
2	C
2	B
2	D
3	A
3	D

Table 1
An example log

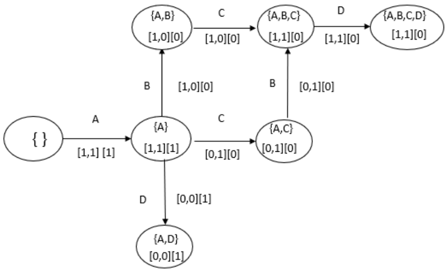


Figure 2: Annotated transition system

The two sub-logs are then represented through an annotated transition system.

As can be seen in Fig. 2 the nodes stand for the states and the arrows show the transitions between them. Annotations appear below states and to the side of transitions. If the trace visits that state (or performs that transition) a 1 will be annotated, otherwise a 0. To determine if the two logs have statistically significant differences in a state (or transition) a "Mann-Whitney U-test" is performed, i.e. a non-parametric test to determine whether two statistical samples come from the same population [5]. If the two states (or the two transitions) turn out to be statistically different, the "Cohen's d" is then measured, which allows us to measure the difference in the sample averages in terms of pooled standard deviation units. The effect size is then translated into a color code.

As can be seen in Fig. 3 the activities in white (and the transitions in black) are those for which no statistically significant difference was found. Colored activities (or transitions), on the other hand, are those whose frequency is higher in one log than another. Shades of red indicate that a state (or transition) is more frequent in the first log, shades of blue indicate the opposite. The colors have a gradation, from lightest to darkest, to indicate the extent of the effect in terms of pooled standard deviations.

The tool also allows to analyze the performance of the two logs by measuring the average activity duration for each log and running the same tests. The frequency of activities and transitions is visually translated with the thickness of arrows and margins.

A similar algorithm capable of visualizing the statistically significant differences of two variants from both control flow and performance perspectives was introduced in [6] by Taymouri, La Rosa, Carmona. They introduce the concept of "mutual fingerprints" that is, a directly-follows graph that shows only the behavior by which the two variants differ from each other.

The method consists of three phases: feature generation, feature selection, and filtering.

The first phase is in itself divided into three parts: binarization, vectorization, and staking. In binarization, traces are represented as time series of 0,1, depending on whether or not an event exists in the given trace. Consider for example the trace $\sigma = e_1e_2e_1e1$ in the event space $\varepsilon = e_1, e_2, e_3$. It can be represented in a vector space in which $f(e_1, \sigma) = 1011$, $f(e_2, \sigma) = 1011$, $f(e_3, \sigma) = 0000$. In the vectorization, the binarized vectors are transformed into the vectors of wavelet coef-

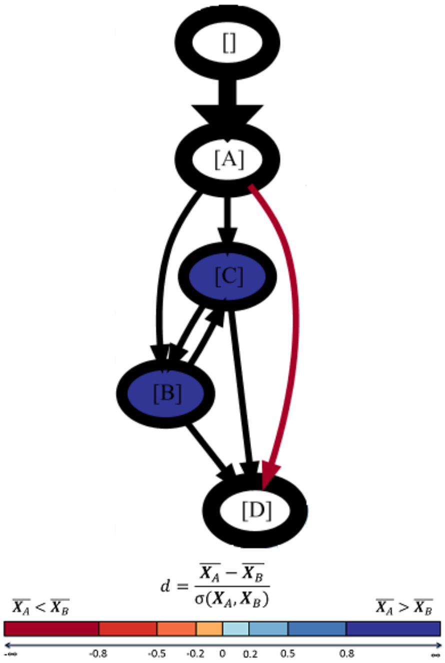

[]
[A]
[C]
[B]
[D]
$d = \frac{\overline{X_A} - \overline{X_B}}{\sigma(X_A, X_B)}$
$\overline{X_A} < \overline{X_B}$
$\overline{X_A} > \overline{X_B}$
-∞
-0.8
-0.5
-0.2
0
0.2
0.5
0.8
∞

Figure 3: Example of two logs analyzed with the Process Comparator. The AB and AC arcs are black because only very high-frequency differences are detected with a few traces.

$$\underbrace{\begin{pmatrix} 0.75 \\ -0.25 \\ 0.5 \\ 0 \end{pmatrix}}_{\mathbf{w}^{(1)}} = \underbrace{\begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & -0.25 & -0.25 \\ 0.5 & -0.5 & 0 & 0 \\ 0 & 0 & 0.5 & -0.5 \end{pmatrix}}_{\mathbf{H}^{-1}} \underbrace{\begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}}_{\mathbf{x}^{(1)}}$$

Figure 4: Transformation in wavelets coefficient vector for vector $f(e_1, \sigma) = 1011$

ficients according to the vector equation $w = H^{-1}x$ where H is the Haar basis matrix (in Fig. 4). In the vectorization they construct the design matrix D, in which rows represent each individual trace and columns are constructed from the concatenation of wavelet coefficient vectors, as represented in Fig. 5.

In the second phase, the feature selection, the augmented design matrix is built, where each track presents the label of its belonging to log 1 or log 2. Then a classifier is trained to select relevant features and the goodness of the classifier is tested with the weighted F1 score.

In the third step, filtering they construct two directly-follow graphs with the traces that contain significant features, one for each variant. This provides a simple interpretation of the results.

Although the formulation of Taymouri et al. performs very well, the algorithm of Bolt et al. allows a very simple visual interpretation, which makes it possible to detect differences between business processes very quickly, even in the case of very large models. For this reason, we preferred to use the Process Comparator in our analysis.

4. Validation

In this section, we apply the proposed 3L methodology on data coming from a large Italian company that provides PAIS systems for about eight thousand Italian municipalities. In particular, we have collected all the logs available for the "Change of residence" service and related to those municipalities with less than 50K inhabitants.

After a clustering phase using the K-medoids [7] algorithm, we identified numerous clusters, which differed from each other in their control flow and activity set. For the sake of space, the discussion on the dimensions of clusters is kept out of this work. Clearly, the result is strongly dependent on the objective defined by the user that has to identify the number of clusters to consider.

For illustration purposes, we selected three clusters and the corresponding medoids. These medoids from here on are indicated according to the dimension of the municipality that generated them. In particular, the following were analysed: one of 7000 inhabitants, one

$$\mathbf{D} = \begin{array}{cccccccccccccc} & & e_1 & e_1 & e_1 & e_1 & e_2 & e_2 & e_2 & e_2 & e_3 & e_3 & e_3 & e_3 \\ \mathbf{w}^{(\sigma_1)} & \Big(& 0.75 & -0.25 & 0.5 & 0 & 0.25 & -0.25 & -0.5 & 0 & 0 & 0 & 0 & 0 & \Big) \\ \mathbf{w}^{(\sigma_2)} & \Big(& 0.5 & -0.25 & 0 & 0 & 0.25 & -0.25 & -0.5 & 0 & 0.25 & -0.25 & 0 & 0.5 & \Big) \end{array}$$

Figure 5: Design matrix

of 10800, and one of 20800.

The log of the municipality of 7000 inhabitants has 386 observations made between 13/01/2014 and 06/02/2020, the log of the municipality of 10800 has 216 observations made between 22/02/2011 and 06/06/2013 and the log of 20800 has 1739 observations made between 29/09/2014 and 11/02/2020. The median process duration is 19.1 days for the municipality of 7000 inhabitants, 19.5 for the municipality of 10800, and 50 seconds for the municipality of 20800. Such a large difference between the first two municipalities and the third can be explained by assuming that in the municipality of 20800 the process executions are computerized only after the process is completed.

As can be seen in Fig. 6 the logs from 7000 and 20800 are very similar to each other, differing significantly from the log of the municipality of 10800 inhabitants (Fig. 7, 8). This shows that in our dataset the control flow of the process is independent of the size of the municipality, in contrast to what intuition would suggest.

Fig. 7 shows the graph of the 10800 municipality compared to the 20800 municipality it can be seen that both processes start with the "Start" activity followed by the "Dossier opening" activity (similarity is represented by a white background). The control flow changes in the transition to the next activity: the 20800 municipality runs the "Waste declaration" activity before running the "Opening printouts" activity, which is why the activity is colored red. The Process Comparator also allows to view the percentage of traces that execute a certain transition or activity. In the case of the municipality of 10800 inhabitants, the "Waste declaration" activity is performed in 0% of the traces, while in the municipality of 20800 it is performed 47.38% of the times. Checking the timestamps of the traces shows that the execution of this activity occurs for the first time in August 2017. This could mean that the activity is the result of a law that went into effect at that time. The 10800 inhabitants log by contrast never performs this activity and this is in line with the argument made, as the data taking ends in 2013, thus before the eventual entry into force of this law. In this case, the variability of the models is a symptom of a temporal evolution of the processes. In future analysis of logs from other municipalities, it will be important to distinguish sources of time-dependent variability in the control flow in order to take into account only the most up-to-date version of the process.

The two models coincide again in the execution of activities "Opening printouts" and "Choice of investigation" that are executed with similar percentages from both processes. As can be seen from "Choice of investigation" the flow is divided into four arcs leading to different activities "End of investigations", "Registration of change", "Prior printouts" and "Investigation". The activities and the arrows in red are only carried out by the municipality with 20800 inhabitants and in blue the activities and jumps carried out by the municipality of 10800. The two processes coincide again in "Dossier closing", while it differs in the next two activities, which are "Action" for

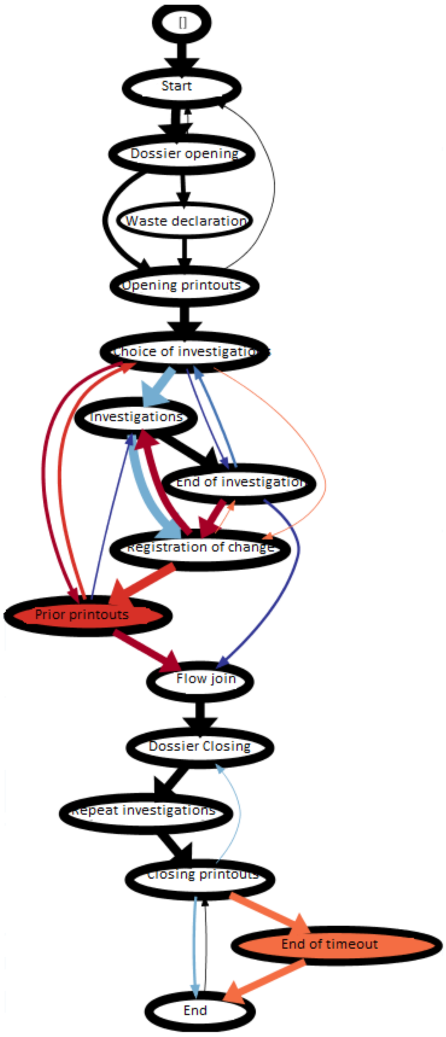


Figure 6: Housing change registration process for two municipalities, one with a population of 7000 and the other with 20800.

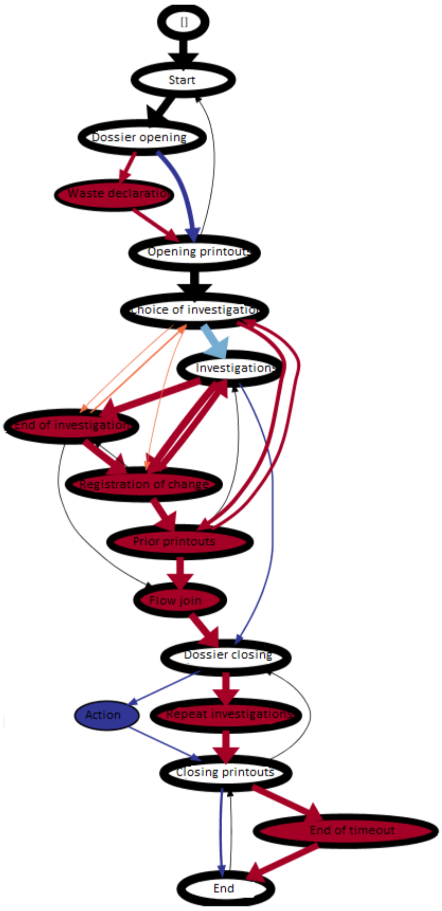


Figure 7: Housing change registration process for two municipalities, one with a population of 10800 and the other with 20800.

the 10800 municipality and "Repeat investigation" for municipality of 20800 inhabitants. The models become overlapping again in the "Closing printouts" activity, while the "End of timeout" activity is only performed by the 20800 municipalities. This activity indicates the presence of a deadline flag for dossiers and is therefore a service implemented in the municipality of 20800 inhabitants that has not been implemented in the municipality of 10800. Then the process terminates with the activity "End".

Concerning the comparison between the municipalities of 7000 and 10800 inhabitants depicted in Fig. 8 we can see that the process of the 7000 inhabitants provides a similar control flows of the 10800 process except for the "Action" activity. The blue activities

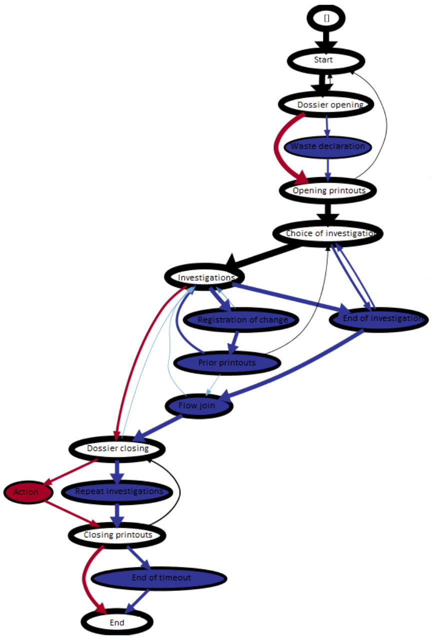
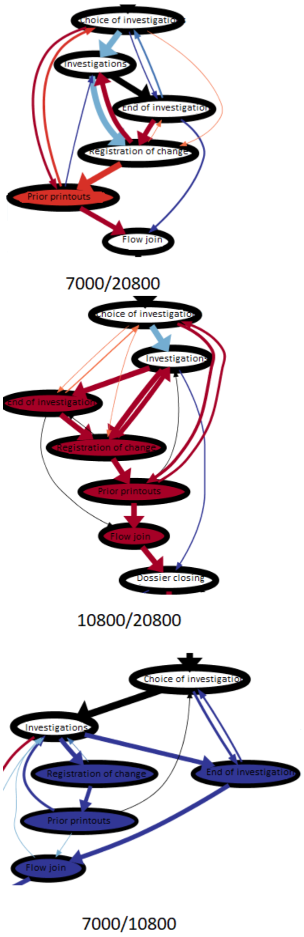


Figure 8: Housing change registration process for two municipalities, one with a population of 7000 and the other with 10800.

belong only to the 7000 municipality and evidence that the municipality of 10800 inhabitants has the same "Change of residence" process installed but with some functionality disabled.

A detail of the main variability of the three processes is given in Fig. 9. The arcs that connect "Choice of investigations" with "Prior printouts" and "Registration of changes" are present only in the log of 20800 inhabitants (observing the detail of the comparison between the municipality of 7000 inhabitants and the one of 10800 it can be seen that only two arcs are present). The flower model-like structure of the 20800 municipality could be a consequence of the almost instantaneousness of the executed actions and could be traced back to a fluctuation in the recording of timestamps.



7000/20800

10800/20800

7000/10800

Figure 9: Detail of process comparator results

5. Related Works

Comparison of process variants is a widely studied problem in the literature.

One of the earliest works on process comparison is [8]. In the paper, the authors present a technique and a tool to compare two models and their process instances. A model is generated by merging the two initial models, annotating the value of the difference between the number of instances of the first process compared to the second. Thus, it will be possible to identify activities that are executed more or less frequently in the second model than in the first.

In [9] Buijs and Reijers use the alignment technique to compare event logs and models from five municipalities. In particular, the alignment between the log of one municipality and the model of another is measured, in order to visualize their differences.

In [10] Nguyen, Dumas, La Rosa and Hofstede use a differential perspective graph that allows to compare two event logs according to each perspective. In this case, decision trees are generated to determine the business rules for each variant. In this case, decision trees are generated to determine the business rules for each variant.

In [5], the work done in [2] is extended: in this case decision trees are generated to determine the business rules for each variant. A variant is then executed using the business rules of the other, to test their exchangeability.

Other authors suggested methods for identifying and use the business rules of a process. In [11] association rule mining is used together with process mining to analyze the deviant cases of a process. The paper presents a case of supervised learning in which traces are labeled as deviant or non-deviant, enriching each trace with a set of relevant attributes. Business rules are then determined that allow the recognition of unlabeled deviant cases.

In [12] Bose and Van der Aalst address the problem of label incompleteness. If the event log has unlabeled instances the k-nearest neighbor approach is used to decide which class the trace belongs to.

In actual reality, it may be the case that data are not labeled as deviant or non-deviant, but have numerical deviation measures, such as risk quantification. In [13] the authors present an algorithm capable of clustering data based on the deviation measure and at the same time extracting rules in a human-readable form.

6. Conclusion and Future Works

This paper contributes to the definition of the 3L methodology to analyse and compare different variants of a business process. Our methodology aims at identifying differences in the control flow, activities, frequencies, and also to identify the causes of these variations.

The 3L methodology permits to simplify and reduce the complexity of the variance analysis approach in order to permit its applicability in contexts where the cardinality of variants is very high like in the public administration domain.

Our methodology aims to reduce the number of comparisons thanks to clustering algorithms that group together logs that have similar control flow and frequencies. Then one representative for each cluster is compared with each other using the process comparator, to highlight the differences between the various variants of the same service.

The proposed methodology is quite modular and we consider for future works to improve and test other clustering and variance analysis algorithms in order to find the best combination of algorithms that permits to reduce the computation effort but at the same time keeping high the reliability. A connected future work concerns the validation of the proposed approach in trusted application domains, in such a field different works aim to implement PAIS systems on the blockchain technologies [14, 15]. Retrieving information from the blockchain permits us to have certified logs and enlarge their availability.

References

[1] M. Reichert, B. Weber, Enabling flexibility in process-aware information systems: challenges, methods, technologies, Springer Science & Business Media, 2012.

[2] A. Bolt, M. de Leoni, W. M. van der Aalst, A visual approach to spot statistically-significant differences in event logs based on process metrics, in: International Conference on Advanced Information Systems Engineering, Springer, 2016, pp. 151–166.

[3] W. Van Der Aalst, Data science in action, in: Process mining, Springer, 2016, pp. 3–23.

[4] F. Taymouri, M. La Rosa, M. Dumas, F. M. Maggi, Business process variant analysis: Survey and classification, Knowledge-Based Systems 211 (2019) 106557.

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

[6] F. Taymouri, M. La Rosa, J. Carmona, Business process variant analysis based on mutual fingerprints of event logs, in: International Conference on Advanced Information Systems Engineering, Springer, 2020, pp. 299–318.

[7] H.-S. Park, C.-H. Jun, A simple and fast algorithm for k-medoids clustering, Expert systems with applications 36 (2009) 3336–3341.

[8] S. Kriglstein, G. Wallner, S. Rinderle-Ma, A visualization approach for difference analysis of process models and instance traffic, in: Business Process Management, Springer, 2013, pp. 219–226.

[9] J. C. Buijs, H. A. Reijers, Comparing business process variants using models and event logs, in: Enterprise, Business-Process and Information Systems Modeling, Springer, 2014, pp. 154–168.

[10] H. Nguyen, M. Dumas, M. La Rosa, A. H. ter Hofstede, Multi-perspective comparison of business process variants based on event logs, in: International Conference on Conceptual Modeling, Springer, 2018, pp. 449–459.

[11] J. Swinnen, B. Depaire, M. J. Jans, K. Vanhoof, A process deviation analysis–a case study, in: International Conference on Business Process Management, Springer, 2011, pp. 87–98.

[12] R. J. C. Bose, W. M. van der Aalst, Discovering signature patterns from event logs, in: 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2013, pp. 111–118.

[13] F. Folino, M. Guarascio, L. Pontieri, A descriptive clustering approach to the analysis of quantitative business-process deviances, in: Proceedings of the Symposium on Applied Computing, 2017, pp. 765–770.

[14] F. Corradini, A. Marcelletti, A. Morichetta, A. Polini, B. Re, F. Tiezzi, Engineering trustable choreography-based systems using blockchain, in: SAC '20: The 35th ACM/SIGAPP Symposium on Applied Computing, ACM, 2020, pp. 1470–1479.

[15] F. Corradini, F. Marcantoni, A. Morichetta, A. Polini, B. Re, M. Sampaolo, Enabling auditing of smart contracts through process mining, in: From Software Engineering to Formal Methods and Tools, and Back, volume 11865 of *LNCS*, Springer, 2019, pp. 467–480.