Detection of statistically significant differences between process variants through declarative rules

Alessio Cecconi[1][0000−0001−5730−6332], Adriano Augusto[2][0000−0001−7970−5246], and Claudio Di Ciccio[3][0000−0001−5570−0475]

[1] Vienna University of Economics and Business, Austria
cecconi@ai.wu.ac.at
[2] The University of Melbourne, Australia
a.augusto@unimelb.edu.au
[3] Sapienza University of Rome, Italy
claudio.diciccio@uniroma1.it

Abstract. Services and products are often offered via the execution of processes that vary according to the context, requirements, or customisation needs. The analysis of such process variants can highlight differences in the service outcome or quality, leading to process adjustments and improvement. Research in the area of process mining has provided several methods for process variants analysis. However, very few of those account for a statistical significance analysis of their output. Moreover, those techniques detect differences at the level of process traces, single activities, or performance. In this paper, we aim at describing the distinctive behavioural characteristics between variants expressed in the form of declarative process rules. The contribution to the research area is two-pronged: the use of declarative rules for the explanation of the process variants and the statistical significance analysis of the outcome. We assess the proposed method by comparing its results to the most recent process variants analysis methods. Our results demonstrate not only that declarative rules reveal differences at an unprecedented level of expressiveness, but also that our method outperforms the state of the art in terms of execution time.

1 Introduction

The execution of a business process varies according to the context in which it operates. The exhibited behaviour changes to adapt to diverse requirements, geographical locations, availability or preferences of the actors involved, and other environmental factors. The alternative enactments lead to the diversification of specialised processes that stem from a general model. Considering some real-world examples, the hospital treatment of sepsis cases follows a different clinical pathway according to their age; the credit collection of road traffic fines typically undergoes an additional appeal to the prefecture when the fine is high.

Recent trends in business process management have led to a proliferation of studies that tackle the automated analysis of process variants [19]. After the initial manual, ad-hoc endeavours on single case studies, the focus has shifted towards the data-driven detection of their behavioural differences from the respective event logs [3,4,10,18]. To

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-030-85440-9_5

Table 1: Description and notation of considered DECLARE constraints

Constraint	Verbal explanation
PARTICIPATION(a)	a occurs at least *once*
ATMOSTONE(a)	a occurs at most *once*
RESPONDEDEXISTENCE(a, b)	If a occurs, then b occurs as well
RESPONSE(a, b)	If a occurs, then b occurs after a
ALTERNATERESPONSE(a, b)	Each time a occurs, then b occurs afterwards, and no other a recurs in between
CHAINRESPONSE(a, b)	Each time a occurs, then b occurs immediately afterwards
PRECEDENCE(a, b)	b occurs only if preceded by a
ALTERNATEPRECEDENCE(a, b)	Each time b occurs, it is preceded by a and no other b recurs in between
CHAINPRECEDENCE(a, b)	Each time b occurs, then a occurs immediately beforehand
COEXISTENCE(a, b)	If b occurs, then a occurs, and vice versa
SUCCESSION(a, b)	a occurs if and only if it is followed by b
ALTERNATESUCCESSION(a, b)	a and b if and only if the latter follows the former, and they alternate each other
CHAINSUCCESSION(a, b)	a and b occur if and only if the latter immediately follows the former

date, however, the explanatory power of the existing techniques is limited: the results are *(i)* exposed as whole graphical models (e.g., directly-follows graphs) leaving the identification of the differences to the visual inspection of the analyst, or *(ii)* expressed as variations within the limited scope of subsequent event pairs (e.g., behavioural profiles).

To overcome this limitation, we propose an approach that infers and describe the differences between variants in terms of behavioural rules. Our declarative approach aims to *(i)* single out the distinct behavioural characteristics leading to the variations observed in the trace, while *(ii)* having a global scope as rules are exerted on the whole process runs. In our pursuit of explainability, we employ and adapt state-of-the-art techniques to guarantee the statistical significance of the inferred differentiating rules and expose the top-ranked distinctive characteristics in the form of natural language. We name our approach Declarative Rules Variant Analysis (DRVA).

In the following, Sec. 2 illustrates existing work in the areas upon which our approach is based. Section 3 describes in details our technique. Section 4 illustrates the results of our tool running on real-world event logs and compares them with state-of-the-art techniques. Finally, Sec. 5 concludes this paper and provides remarks for future research avenues.

2 Background and Related Work

In this section, we provide a brief overview of the use of declarative languages within the process mining area. We discuss the most recent studies on process variants analysis, and we introduce the statistical significance test that we adapted to our context.

2.1 Declarative process specification and mining

A declarative specification represents the behaviour of a process by means of reactive constraints [5], i.e., temporal rules that specify the conditions under which activities can or cannot be executed. For the purposes of this paper, we focus on DECLARE, one of the most well-established declarative process modelling languages to date [12]. DECLARE provides a standard repertoire of templates, namely linear temporal-logic rules parameterised over tasks, here abstracted as symbols of a finite non-empty alphabet Σ. Table 1 illustrates the templates we use in the context of this paper. Declarative

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-030-85440-9_5

rules are typically expressed in an if-then fashion, whereby the "if" part is named *activator* and the "then" part is the *target*. For example, RESPONSE(a, b) requires that if a is executed (activator), then b must be eventually carried out (the target). PRECEDENCE(b, a) imposes that a cannot be executed if b has not occurred earlier in the process instance. RESPONDEDEXISTENCE(a, b) relaxes the condition exerted by RESPONSE(a, b) and PRECEDENCE(b, a) by requiring that if a is executed, then b has to occur in the same trace, regardless of whether it happens before or after a. RESPONSE(a, b) and PRECEDENCE(b, a) thus *entail* RESPONDEDEXISTENCE(a, b) and we say that RESPONDEDEXISTENCE *subsumes* RESPONSE and PRECEDENCE [7]. Subsumption is a partial order on the templates of DECLARE inducing the partial order of entailment on the rules that instantiate those templates. A *declarative specification* $M \ni r$ is a set of rules r that conjunctively define the behaviour of a process. We shall denote the universe of declarative rules as $\mathfrak{R} \supseteq M \ni r$.

The formal semantics of DECLARE rules are rooted in Linear Temporal Logic on Finite Traces (LTL$_f$). A trace $t \in \Sigma^*$ is a finite string of *events*. An event log (or *log* for short) is a multi-set of traces $L : \Sigma^* \to \mathbb{N}$ (in which equivalent elements can occur multiple times). We indicate the universe of event logs as $\mathfrak{L} \ni L$. We shall denote as $|t|$ the length of a trace t, as $|L|$ the cardinality of the event log, and as $|\hat{t}|$ the length of the longest trace in the log. Declarative process discovery tools can assess to what degree constraints hold true in a given event log. To that end, diverse measures $m : \mathfrak{R} \times \mathfrak{L} \to \mathbb{R}$ have been introduced to map a rule $r \in \mathfrak{R}$, and an event log $L \in \mathfrak{L}$ to a real number [5]. Among them, we consider *Support* and *Confidence* here. Their values range from 0 to 1. Support is computed as the fraction of events in which both activator (e.g., the occurrence of a for RESPONSE(a, b)) and target (e.g., the occurrence of b eventually afterwards in the trace) hold true. Confidence is the fraction of the events in which the rule holds true over the events in which the activator is satisfied. With a slight abuse of notation, we shall indicate the measure of r on a single trace t with $m(r, t)$.

2.2 Process variants analysis

The latest literature review on process variants analysis [19] identified more than thirty studies addressing the research problem, clearly showing a growing interest in the topic. Much of the early research work was centred on case studies, providing examples of variant analysis applications that highlighted its potential capabilities. Among the earliest studies, Poelmans et al. [14] combined process mining and data mining techniques to differentiate and analyse the healthcare pathways of patients affected by breast cancer and their response to therapies. Another of the most impactful studies is the one regarding a large Australian insurance company [17], in which Suriadi et al. describe a methodology to pinpoint the weaknesses in the processing of insurance claims that had slow turnaround time. Until recently, the vast majority of variant analysis studies focused on the detection of control flow differences of two or more process variants, with the frequent underlying goal to relate such differences to the outcome of the process variant. Also, proposed methods for variant analysis did not put much attention to the statistical significance of the detected differences, with the risk to catch random differences. Only recently proposed methods for variant analysis [4,10,18] account for statistical significance.

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-030-85440-9_5

Bolt et al. [4] pioneered the introduction of statistical significance in variant analysis, designing a framework to compare process variants in the form of annotated transition systems through the application of statistical tests. Also, the framework allows the analysis of the decision points in different variants and how the underlying decision-making rules differ. The annotations on the transition systems can capture frequencies and elapsed time between the transitions, however, the framework was designed to be extended, with the option to capture new annotations.

Nguyen et al. [10] propose a variant analysis method based on the statistical comparison of *perspective graphs*. A perspective graph is an artifact that captures an attribute of an event log (e.g. activity, resource, etc.), and the relations between the observed attribute values. The two perspective graphs (one per process variant) are then statistically compared and a *differential perspective graph* is generated. The latter is analysed to determine the differences between the variants. By varying the attribute to generate the perspective graphs, the method allows for the analysis of different process perspectives.

Lastly, Taymouri et al. [18] propose a variant analysis method to separate statistically-significant different traces from common traces between the two input process variant logs. Each trace is encoded as a vector, then an SVM is used to classify the traces (assigning them to one of the two logs). A set of trace features (i.e. directly-follows relations) that can statistically discriminate the traces between the two logs is identified, and the logs are filtered by retaining only the traces containing those features. Finally, the two filtered logs are mapped into directly-follows graphs, called mutual-fingerprints, which rely on color-coding to show differences.

Of all the past studies, none is capable of analysing process variants to detect differences in the form of declarative rules [19], which is instead the focus of this paper. However, given that declarative rules can be difficult to interpret, and that variant analysis is of high interest for the practitioners audience, we follow the example of van Beest et al. [3] and provide a natural language translation of the detected differences [1].

2.3 Statistical significance

It is likely that the data at hand captures only a portion of the whole data population, and it can include outliers. Process execution data is no exception. In fact, an event log usually contains only a part of the allowed process behaviour, on one hand, and infrequent behaviour, on the other hand [2]. To draw reliable observations from data samples, statistics provides several methods to assess the likelihood that an observation holds for the whole data population. These methods, known as *statistical significance tests*, estimate the probability that what is observed happened by chance. Traditionally, if such probability is low (below 0.05), the observation is assumed to hold for the whole data population.

Of the many available methods to assess the statistical significance of the discovered declarative rules (which are similar to association rules) [9], we deem the *permutation test* [13] as the one that best fits for our purpose and context. The permutation test (also known as randomization test) is a non-parametric test originally designed to estimate the probability that two numerical series belong to the same population or not. In other words, it provides an answer to the question: "Are the two series statistically different?" The permutation test is carried out as follows. First, the difference of the averages of

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-030-85440-9_5

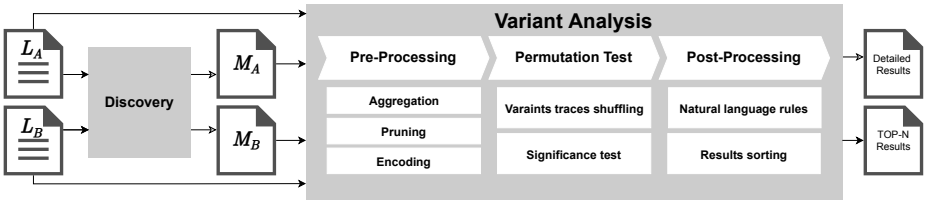


Fig. 1: Schema of our variant analysis approach based on declarative rules.

the two series is calculated (the *delta average*). Then, the elements of the two series are pooled and two new series are drawn at random to compute the delta average again. This step is repeated for all the possible permutations. The likelihood that the two original series are statistically different is assessed as one minus the percentage of delta average that is greater or equal to the original delta average. The original permutation test is computationally heavy because it requires a computation on all the possible permutations of the original series. However, successive studies have proposed approximated [8] and efficient [21] methods to reduce the computational effort and construct permutation tests for any settings [20], from pure maths [13] to medicine [11]. In the following, we show how we adapt the permutation test to address the problem of variant analysis.

3 Approach

In this section, we present in details our variant analysis technique based on declarative process specifications: Declarative Rules Variant Analysis (DRVA). The core idea is to discover the declarative process specifications describing the behaviour of the input variants' event logs and to check, by running a permutation test, whether the difference between the two specifications are statistically significant.To assess the significance, intuitively, we verify whether the differences stemming from the variants' event logs are not due to rules that occur by chance.

Figure 1 presents the overview of the approach. Specifically, the variant analysis is divided into four phases: (1) A discovery phase, to mine the declarative specifications of the variants' behaviour; (2) A pre-processing phase, to prune redundant or irrelevant rules and encode the information for faster computation; (3) A statistical validation phase, in which the permutations and related significance test take place; and (4) A final post-processing phase, to sort the results according to a given relevance criterion and produce a natural language description of the output. In the remainder of this section, we thoroughly explain each step of the approach (illustrated as pseudo-code in Algorithm 1) and its implementation.

Some of the examples in this section refer to the publicly available SEPSIS event log (see Table 2), which records the treatment of patients with sepsis symptoms in a Dutch hospital. The two variants are generated as one containing traces regarding only elderly patients above the age of 70 (L_A) and the other containing young ones below 35 (L_B).

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-030-85440-9_5

Algorithm 1: Computing the set of rules that differ in a statistically significant way. The computational cost of the steps is specified on the right-hand side.

Input : Log variants L_A and L_B, specifications M_A and M_B, a function m, parameters m_{min}, $m_{\text{diff-min}}$, α, π
Result: R, a set of rules exhibiting a statistically significant difference between the variants.

/* Pre-processing */
$M_\cup \longleftarrow M_A \cup M_B$; // Cost: $\mathcal{O}(|M_A| + |M_B|)$
$E_{\text{diff}}, E_A, E_B \longleftarrow \{\}$; // Cost: $\mathcal{O}(1)$
foreach $r \in M_\cup$ **do** // Block cost: $\mathcal{O}(|M_\cup| \times |\hat{t}| \times (|L_A| + |L_B|))$
  $E_A \longleftarrow E_A \cup \{(r, m(r, L_A))\}$; // Cost: $\mathcal{O}(|e_A| + |L_A|)$
  $E_B \longleftarrow E_B \cup \{(r, m(r, L_B))\}$; // Cost: $\mathcal{O}(|e_B| + |L_B|)$
  $E_{\text{diff}} \longleftarrow E_{\text{diff}} \cup \{(r, |m(r, L_A) - m(r, L_B)|)\}$; // Cost: $\mathcal{O}(1)$
foreach $r \in M_\cup$ **do** // Block cost: $\mathcal{O}(|M_\cup|)$
  if $(E_{diff}(r) < m_{diff\text{-}min}) \vee (E_A(r) < m_{min}) \vee (E_B(r) < m_{min})$ **then** // Cost: $\mathcal{O}(1)$
    Remove r from $M_\cup$ and its measurements from E_A, E_B, E_{diff} ; // Cost: $\mathcal{O}(1)$
$M_\cup, E_A, E_B, E_{\text{diff}} \longleftarrow$ hierarchicalSimplification($M_\cup$,E_A,E_B) ; // Cost: $\mathcal{O}(|M_\cup| \times |h|)$
$L^{\text{m}}_A, L^{\text{m}}_B \longleftarrow$ encodeLog(L_A,L_B,$M_\cup$) ; // Cost: $\mathcal{O}(|M_\cup| \times (|L_A| + |L_B|))$
/* Permutation Test */
$C \longleftarrow$ Initialize map such that for all $r \in M_\cup : C(r) = 1$; // Cost: $\mathcal{O}(|M_\cup|)$
for $i \leftarrow 1$ **to** π **do** // Block cost: $\mathcal{O}(|\pi| \times |M_\cup| \times (|L_A| + |L_B|))$
  $L^{\text{m}}_{A_i}, L^{\text{m}}_{B_i} \longleftarrow$ shuffleLog(L^{m}_A,L^{m}_B) ; // Cost: $\mathcal{O}(|L_A| + |L_B|)$
  foreach $r \in M_\cup$ **do** // Block cost: $\mathcal{O}(|M_\cup| \times (|L_A| + |L_B|)$
    if $|m(r, L^m_{A_i}) - m(r, L^m_{B_i})| \geq E_{diff}(r)$ **then** // Cost: $\mathcal{O}(|L_A| + |L_B|)$
      $C(r) \longleftarrow C(r) + 1$ // Cost: $\mathcal{O}(1)$
$R \longleftarrow \{\}$; // Cost: $\mathcal{O}(1)$
foreach $r \in M_\cup$ **do** // Block cost: $\mathcal{O}(|M_\cup|)$
  p-value$(r) \longleftarrow \frac{C(r)}{\pi}$; // Cost: $\mathcal{O}(1)$
  if $p\text{-}value(r) \leq \alpha$ **then** $R \longleftarrow R \cup \{r\}$; // Cost: $\mathcal{O}(1)$
return R

3.1 Declarative specifications discovery

In order to compare the variants through a declarative lens, it is necessary to derive the declarative specifications of their behavior. Specifically, given the input variants event logs L_A and L_B, we want to retrieve the respective declarative specifications M_A and M_B. This can be done by executing existing declarative process discovery techniques [16].

We remark that the criteria used for the discovery influence the subsequent steps of the analysis. For example, discovering only specifications that are highly compliant to the event logs makes the variant analysis consider the most regular and stable behaviour of the processes. Looser specifications encompass also less frequent behaviours.

The discovery step is executed when only the event logs are given as input. Alternatively, our technique can receive as input the declarative specifications already discovered from the two logs, in such a case, the discovery step is skipped and our technique performs only the variant analysis with the input specifications. Hand-crafted or filtered input declarative specifications can be useful when it is desired to test the statistical significance of the differences between two variants according to specific rules. For example, if the analyst is interested only in the behavioural difference involving tasks ER Triage and LacticAcid, she can provide as input two specifications containing only rules involving these tasks (e.g., RESPONSE(ER Triage, LacticAcid), PRECEDENCE(ER Triage, IV Liquid), SUCCESSION(LacticAcid, Admission NC), and so on) together with the variants' logs.

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-030-85440-9_5

3.2 Pre-processing

The variants analysis takes the variants' events logs L_A and L_B as input, together with the respective declarative process specifications M_A and M_B. As the permutation test is a computationally heavy task, it is desirable to keep only the essential information. Thus, to efficiently perform the statistical test, the data must be ① aggregated, to have a common view between the variants, ② cleaned via pruning, to remove redundant or irrelevant information, and ③ encoded, to improve performance.

① **Aggregation.** The declarative specifications M_A and M_B are merged into a unique specification $M_\cup = M_A \cup M_B$. To check their differences, all the rules in M_A must be checked in L_B and vice versa, thus the union of M_A and M_B allows us to consider all and only the distinct rules in both the logs.

The interestingness of each rule $r \in M_\cup$ is measured in each variant log L_A and L_B. We resort to the measurement framework for declarative specifications proposed in [5]. Among the various measures available, we consider Confidence as the best option because it measures the degree of satisfaction of a rule in a log independently from the rule frequency. The comparison of the most suitable measures (or combinations thereof) goes beyond the scope of the paper and paves the path for future work. In our example, $m(\text{RESPONSE}(\text{ER Triage}, \text{LacticAcid}), L_A) = 0.83$ and $m(\text{RESPONSE}(\text{ER Triage}, \text{LacticAcid}), L_B) = 0.53$, which means that $83\,\%$ of the occurrences of ER Triage in L_A are eventually followed by LacticAcid. In L_B, the measure drops to $53\,\%$ for that rule.

Finally, we retain these reference measurements of $M_\cup$ in each variant denoting them as functional relations E_A and E_B. Formally, we define $E_A : M_\cup \to \mathbb{R}$ as $E_A = \{(r, m(r, L_A)) : r \in M_\cup\}$ and $E_B : M_\cup \to \mathbb{R}$ as $E_B = \{(r, m(r, L_B)) : r \in M_\cup\}$. We compute also the reference absolute difference between the variants' measurements E_{diff}. Formally, we define $E_{\text{diff}} : M_\cup \to \mathbb{R}$ as $E_{\text{diff}} = \{(r, |m(r, L_A) - m(r, L_B)|) : r \in M_\cup\}$, where $|x|$ is the absolute value of x. For example, $E_A(\text{RESPONSE}(\text{ER Triage}, \text{LacticAcid})) = 0.83$, $E_B(\text{RESPONSE}(\text{ER Triage}, \text{LacticAcid})) = 0.53$, and their absolute difference is $E_{\text{diff}}(\text{RESPONSE}(\text{ER Triage}, \text{LacticAcid})) = 0.30$.

② **Pruning.** Not all the rules contained in $M_\cup$ are valuable for the statistical test. Specifically, we consider as ignorable those rules that do not meet one of the following criteria.

Minimum difference: If a rule difference between the variants is considered too small by an analyst to be of interest, it can be discarded. For example, rule $r = \text{PRECEDENCE}(\text{ER Registration}, \text{CRP})$ is such that $E_{\text{diff}}(r) = 0.01$ as $E_A(r) = 0.98$ and $E_B(r) = 0.99$. The significance of such a difference is debatable: the difference appears to be negligible although the consideration is subjective and depending on the context of the analysis. Therefore, we allow the user to customise a threshold to this end: $m_{\text{diff-min}}$. According to this criterion, we remove all the rules $r \in M_\cup$ such that $E_{\text{diff}}(r) < m_{\text{diff-min}}$ from $M_\cup$ and all their measurements from E_A, E_B, and E_{diff}.

Minimum interestingness: If, according to an analyst, a rule is not interesting enough in either of the variants to be considered, it can be discarded. For example, rule $r = \text{RESPONSE}(\text{ER Triage}, \text{Release A})$ is such that $E_{\text{diff}}(r) = 0.23$ yet the rule itself is not frequently satisfied in either of the variants as $E_A(r) = 0.41$ and $E_B(r) = 0.64$. Whether this is desirable or not depends on the context of the analysis. Therefore, we

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-030-85440-9_5

define the m_{min} threshold to let the user set the desired minimum value that the rule's measure should be assigned within the variants' logs. In this step, we remove all the rules $r' \in M_{\cup}$ such that $m(r', L_A) < m_{\text{min}}$ and $m(r', L_B) < m_{\text{min}}$ from $M_{\cup}$ and all their measurements from E_A, E_B, and E_{diff}.

No redundancy: If two rules are such that one is logically implied by the other, we do not gather additional information by retaining both in the set of rules under consideration. DECLARE patterns are hierarchically interdependent, as the satisfaction of a rule implies the satisfaction of all the entailed rules. We can exploit the hierarchical relation of the DECLARE templates to prune the redundant rules as in [6]. However, we adapt the original pruning technique by inverting the preference: We keep the most generic rule rather than the strictest one if measurements are the same in at least one of the variants' event logs. We explain the rationale at the core of our design choice with an example from the Sepsis event log.

Figure 2 depicts the partial order that stems from the subsumption relation among DECLARE templates [6] having ALTERNATESUCCESSION(ER Sepsis Triage, IV Antibiotics) as its least element, and RESPONDEDEXISTENCE(ER Sepsis Triage, IV Antibiotics) and RESPONDEDEXISTENCE(IV Antibiotics, ER Sepsis Triage) as its maximal elements. For the sake of space, we shall indicate ER Sepsis Triage and IV Antibiotics with t and v, respectively. ALT.SUCCESSION(t, v) is the strictest rule in the set, as all the other rules can be derived from it. As the associated Confidence is higher in L_A (0.82) than in L_B (0.49) we can claim the following: "In L_A, v follows t and t precedes v with no other t or v occurring in between, more likely than in L_B". Looking at the measurements on L_A and L_B we see that there is no difference between its values and the ones of ALT.RESPONSE(t, v) or SUCCESSION(t, v), and it entails both. Recursively following the subsumption relation, we notice that the same measurements are associated to one of the maximal elements of the induced partial order: RESPONDEDEXISTENCE(t, v). The other maximal element, instead, is such that the associated Confidence is equal to 1.00 in L_A and L_B. This characteristic reverberates along the chain of entailment down to ALT.PRECEDENCE(t, v). With this example, we observe that the least restrictive rules point out more precisely where the cause of the differences between variants lies – in this case, the occurrence of v required by t is at the core of the distinct behaviours. The co-occurrence, order of execution, and lack of internal recurrence are characteristics that ALT.PRECEDENCE(t, v) exhibits although they are evidenced by both variants.

According to this criterion, we thus prune redundant rules only if the measurement between entailing and entailed rules are equivalent in at least one variant. Otherwise, we keep both. Formally, denoting with $\vDash$ the entailment relation, we remove the following subset of rules from $M_{\cup}$ and all their measurements from E_A, E_B, and E_{diff}: $\{r \in M_{\cup} : r \vDash r', r' \in M_{\cup} \setminus \{r\}$ and $m(r, L_A) = m(r', L_A)$ or $m(r, L_B) = m(r', L_B)\}$.

We remark that the first two criteria (minimum difference and minimum interestingness) are generically applicable to declarative rules and measures thereof, whereas the third one (no redundancy) is tailored for templates á-la-DECLARE and measures such as Confidence, as it requires a subsumption hierarchy of the repertoire of templates and the monotone non-decrease of the measure within the subsumption hierarchy (i.e., if $r \vDash r'$, then $m(r, L) \leqslant m(r', L)$) as shown in [6].

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-030-85440-9_5

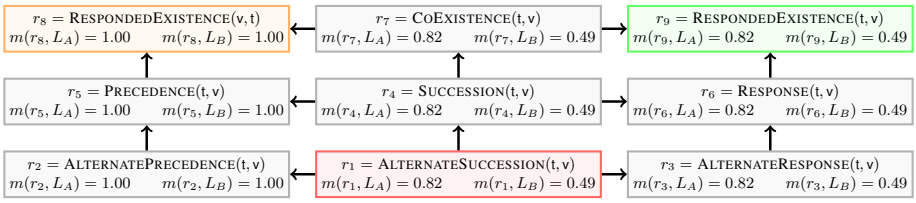


Fig. 2: Example of hierarchical simplification where the less restrictive rule is preferred over the more restrictive one, given equivalent measurements.

③ **Encoding.** The measures of a trace in a log are based on the measurements of the rule in each trace of the log itself [5]. Such trace measurement is independent from the log in which the trace resides, thus moving the trace from one log to another would not change it. We take advantage of this aspect to save computation steps: for each trace t in L_A and L_B, we cache the trace measurement $m(r, t)$ of every rule $r \in M_\cup$ in multi-sets L_A^{m} and L_B^{m}, i.e., $L_A^{\mathrm{m}} = \{m(r, t) : t \in L_A, r \in M_\cup\}$ and $L_B^{\mathrm{m}} = \{m(r, t) : t \in L_B, r \in M_\cup\}$. This step allows us to encode the traces into feature vectors that can be used within the permutation test, as discussed in the next section. For example, given a trace t and a specification $M_\cup = \{\textsc{Resp.}(\mathsf{a}, \mathsf{b}), \textsc{Prec.}(\mathsf{c}, \mathsf{d})\}$, the encoded trace is $t^{\mathrm{m}} = \{(\textsc{Resp.}(\mathsf{a}, \mathsf{b}), m(\textsc{Resp.}(\mathsf{a}, \mathsf{b}), t)), (\textsc{Prec.}(\mathsf{c}, \mathsf{d}), m(\textsc{Prec.}(\mathsf{c}, \mathsf{d}), t))\}$. With a slight abuse of notation, we shall denote with $m(r, L^{\mathrm{m}})$ the measure of a rule r on log L after the encoding of all traces $t \in L$ in L^{m}.

To sum up, at the end of the pre-processing phase we have a unique declarative specification ($M_\cup$) with the measurements in the variants (E_A, E_B, and E_{diff}), and the evaluation of each rule in every trace cached for later reuse (L_A^{m} and L_B^{m}).

3.3 Permutation test

In this phase, we check whether the differences between the declarative rule measurements in the variants are statistically significant. In other words, taken each $r \in M_\cup$, we calculate the likelihood that the absolute difference of its measurements between the variants, $E_{\text{diff}}(r)$, was due to a random factor. If the null hypothesis "$E_{\text{diff}}(r)$ occurred by chance" can be refuted, then the difference is significant. To that extent, we employ an adaptation of the permutation test we introduced in Sec. 2.3. We reshuffle traces between L_A and L_B and observe if the difference in the measures holds as in the original variants' logs. If so, it is likely that its difference was due to chance (i.e., the null hypothesis is confirmed). The rationale of this test is the following: if a difference can be detected by randomly shuffling the variants' traces, this difference has no real discriminative power between the variants.

The acceptance or refutation of the null hypothesis is done in the following two steps (we continue the numbering from the previous section): ④ The reshuffling, in which the data are rearranged and the measures of the rules under the null hypothesis are computed; ⑤ The significance test, in which only the rules that exhibit a statistically significant difference between variants are retained. Notably, thanks to the encoding of the logs

Pre-print copy of the manuscript published by Springer (available at link.springer.com) identified by DOI: 10.1007/978-3-030-85440-9_5

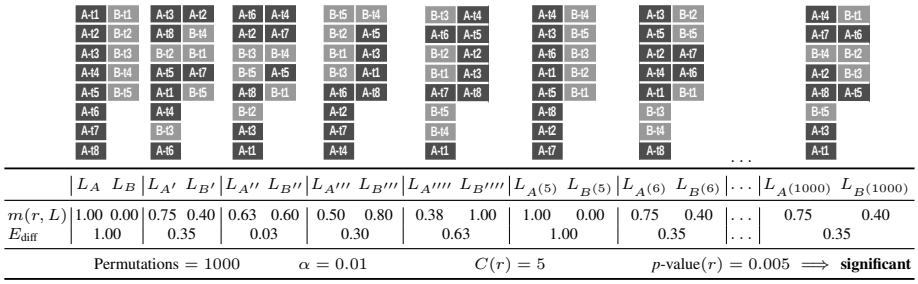


Fig. 3: Visual example of the permutation test. The traces of the original logs L_A and L_B are coloured in dark gray and light grey, respectively.

presented in Sec. 3.2, the trace evaluations of every rule are readily available to compute the measures across the new set of traces. Thus, we can shuffle the evaluations on the traces rather than the traces themselves. Next, we elaborate on these operations.

④ **Reshuffling.** Figure 3 presents a graphical example of the reshuffling. At each iteration, the traces are randomly rearranged between the variants' logs, thereby generating altered logs on which the rule is measured. Specifically, at each iteration i: *(i)* The multi-set of all encoded traces in $L_A^{\text{m}} \cup L_B^{\text{m}}$ is shuffled into two new logs $L_{A_i}^{\text{m}}$ and $L_{B_i}^{\text{m}}$, such that $|L_A^{\text{m}}| = |L_{A_i}^{\text{m}}|$ and $|L_B^{\text{m}}| = |L_{B_i}^{\text{m}}|$ (permutation step); *(ii)* For each $r \in M_\cup$ the measures $m(r, L_{A_i}^{\text{m}})$ and $m(r, L_{B_i}^{\text{m}})$ are computed; *(iii)* Finally, for each rule $r \in M_\cup$ the difference of its measurements is compared to the reference difference of the rule $E_{\text{diff}}(r)$. The number of iterations is set as a user parameter π: according to [8], a suitable value for π is 1000. We denote with $C : M_\cup \to \mathbb{N}$ the function mapping a rule $r \in M_\cup$ to the number of iterations in which its difference is greater than or equal to the reference difference, i.e., $C(r) = \sum_{i=1}^{\pi} [\![|m(r, L_{A_i}^{\text{m}}) - m(r, L_{B_i}^{\text{m}})| \geq E_{\text{diff}}(r)]\!]$ where $[\![\cdot]\!]$ is the indicator function mapping to 1 or 0 if the argument holds true or false, respectively.

⑤ **Significance test.** At the end of the permutations step, the counter $C(r)$ tells us for each rule how frequently a difference greater or equal the reference one is observed. The likelihood to observe the rules difference under the assumption of the null hypothesis is the *p*-value of the test: $p\text{-value}(r) = \frac{C(r)}{\pi}$. The significance level α of the test is the *p*-value threshold below which the null hypothesis should be discarded. It is common to set $\alpha = 0.01$ when the permutation test consists of 1000 iterations [8]. A rule $r \in M_\cup$ has a statistically significant difference between the variants L_A and L_B if and only if $p\text{-value}(r) \leq \alpha$. Figure 3 illustrates an example of such a significance test where a difference of at least $E_{\text{diff}}(r)$ occurred for only 5 permutations out of the 1000 performed, e.g, in permutation $i = 5$. Therefore, $p\text{-value}(r) = 0.005$. As it is less than the significance level of 0.01, it suggest that the difference is statistically significant.

In conclusion, the technique returns the set of rules that determine a statistically significant difference between the variants.

About the computational cost. Algorithm 1 shows the pseudo-code of our approach for the pre-processing and the permutation test phases. The overall computational cost is

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-030-85440-9_5

linear in the input size. Let $|M_A|$, $|M_B|$, and $|M_\cup|$ be the number of rules in, respectively, M_A, M_B, and $M_\cup$, $|L_A|$ and $|L_B|$ the number of traces in L_A and L_B, $|\hat{t}|$ the maximum length of a trace in $L_A \cup L_B$, h the maximum hierarchical level of a rule in $M_\cup$, and π the number of iterations in the permutation test. Summing the costs of each step of Algorithm 1 (on the right side of every line), the overall computation cost of the approach is: $\mathcal{O}(|M_\cup| \times |\hat{t}| \times (|L_A| + |L_B|)) + \mathcal{O}(|M_\cup| \times \pi \times (|L_A| + |L_B|))$.

The overall performance is driven by the cost of the evaluation of the rules on the traces (the encoding phase) and by the permutation tests. We keep the two addenda separated to highlight their contribution. Depending on whether $|\hat{t}|$ is greater or less than π, the first or the second addendum prevails. We remark that the cost of computing the measures in the pre-processing (Lines 4 and 5) stems from the computational cost of the measurement framework [5], while the cost of re-computing the measures during the permutations (Line 16) subtracts the trace evaluation time to that cost due to the encoding presented in Sec. 3.2. For as far as the hierarchical simplification (Line 10), a DECLARE rule from the standard templates set may have at most $h = 11$ (the CHAINSUCCESSION template).

3.4 Post-processing

Once the rules with a statistically significant difference between the variants have been identified, we show them to the final user. All the relevant details are reported, namely the rule r, its p-value, its original measurements in the variants logs $m(r, L_A)$ and $m(r, L_B)$, and their absolute difference $E_{\text{diff}}(r)$. Furthermore, in order to enhance the clarity of the outcome, we perform the following additional steps: *(i)* providing a natural-language description of the output, and *(ii)* sorting the results according to a priority criterion.

⑥ **Natural language.** We report the rules along with a natural language description explaining their behaviour in a concise manner. Indeed, the comprehension of temporal logic formulae is out of the reach for a general audience, and even the DECLARE rules taken by their own are not immediately grasped by non-knowledgeable users [1]. For example, RESPONSE(ER Triage, LacticAcid) is more readable than its LTLp_f formulation, provided that the user has a prior knowledge of DECLARE. To explain differences between variants, we thus provide a description as follows: "In variant A, it is 30% more likely than in variant B that if ER Triage occurs, LacticAcid will occur afterwards".

⑦ **Sorting.** We give higher priority to those rules whose measure differs the most between variants. To ease the discoverability of prominent results, we sort the rules in descending order according to their measurement difference between the variants, $E_{\text{diff}}(r)$, and, in the case of equal difference, the highest measurement in L_A or L_B. For the sake of readability, the user can specify how many of the ranked rules should be initially displayed by means of a dedicated parameter, N.

3.5 Implementation and remarks

We have implemented our approach, Declarative Rules Variant Analysis (DRVA), as a Java command-line tool.[4] Our tool returns two files: *(i)* a textual file containing the

[4] Available at: https://github.com/Oneiroe/Janus

Pre-print copy of the manuscript published by Springer (available at link.springer.com) identified by DOI: 10.1007/978-3-030-85440-9_5

top-N statistically significant differences between the input logs reported in natural language, and *(ii)* one tabular file (CSV format) containing all the statistically-significant differences captured as declarative rules (and in natural language) with the entire quantitative information from the analysis. As input, it takes two event logs (L_A and L_B, in XES, MXML, or CSV formats), two declarative specifications (M_A and M_B, in JSON or XML formats) and the following list of user-defined parameters to tune the algorithm: *(i)* m, the measure to use for the comparison of the variants among those that are supported by the measurement framework described in [5] (the default is Confidence, as explained in Sec. 3.2); *(ii)* m_{min}, the minimum measure threshold that a rule should exceed in at least one variant (0.00 by default); *(iii)* $m_{\text{diff-min}}$, the minimum difference threshold that a rule should exceed between the variants (0.01 by default); *(iv)* π, the number of iterations to perform during the permutation test, set to 1000 by default as explained in Sec. 3.3; *(v)* α, the significance level, namely the maximum p-value to deem a rule as statistically relevant to discriminate the variant (0.01 by default); *(vi)* N, the number of top rules to display in the textual output (see Sec. 3.4), set to 10 by default.

Our tool can be put in pipeline to a declarative discovery algorithm which mines the declarative specifications out of the variants' logs. In the experiment presented in following Sec. 4, we pass the output of the MINERful discovery algorithm [7] to DRVA.

4 Evaluation

Given that no previous work addressed the problem of variant analysis via declarative rules [19], in this section, we provide a qualitative comparison with the latest process variants analysis approach that discovers statistically-significant differences [18], which already demonstrated its advantages [18] with respect to other baselines [4,10]. Henceforth, we will refer to the approach of Taymouri et al. [18] as MFVA (Mutual Fingerprints Variant Analysis). Furthermore, we also consider the work of van Beest et al. [3] as a baseline for comparison given that its output is in natural language statements – henceforth referred to as PESVA (Prime Event Structure Variant Analysis). However, we note that PESVA does not take into account the statistical significance of the detected differences and it is not based on declarative rules either. Instead, PESVA outputs two type of statements: *(i)* frequency-based statements, which highlight differences in branching probabilities of the process decision points; and *(ii)* behaviour-based statements, which highlight differences in directly-follow, and concurrent relations, as well as optional tasks.

Dataset and setup. We reproduce the experimental setup proposed in [18]. The evaluation dataset consists of four publicly available event logs. Each of the logs can be divided into two variants (based on process-instance attribute values). The descriptive statistics characterizing the four logs are shown in Table 2. The original logs (not partitioned by the variants) can be downloaded from the 4TU Research Data Centre.[5] The process variant logs can be downloaded from our online repository together with the full set of results of this evaluation which, due to space limits, we could not report in this section. [6] In the following, we focus our discussion on the most interesting results.

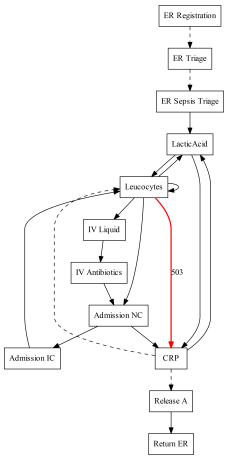
[5] https://data.4tu.nl/search?categories=13503

[6] https://github.com/Oneiroe/DeclarativeRulesVariantAnalysis-static

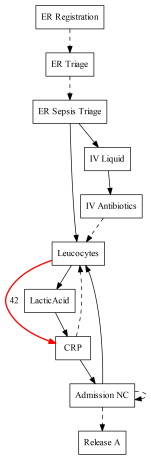
Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-030-85440-9_5

Table 2: Descriptive statistics of the evaluation dataset [18]

Log		Traces		Events		Trace length		
Name and DOI	**Variant**	**Total**	**Distinct**	**Total**	**Distinct**	**Min**	**Avg**	**Max**
BPIC13	Company = A_2	553	25.5%	4,221	3	2	8	53
(10.4121/uuid:a7ce5c55-03a7-4583-b855-98b86e1a2b07)	Company = C	4,417	13.8%	29,122	4	1	7	50
BPIC15	Municipality = 1	1,199	97.6%	36,705	146	2	33	62
(10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460)	Municipality = 2	831	99.6%	32,017	134	1	39	96
RTFMP	Fine Amount $\geq$ 50	21,243	0.7%	91,499	11	2	4	20
(10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5)	Fine Amount $<$ 50	129,127	0.1%	469,971	11	2	4	11
SEPSIS	Patient Age $\geq$ 70	678	85.7%	10,243	16	3	15	185
(10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460)	Patient Age $\leq$ 35	76	67.1%	701	12	3	9	52

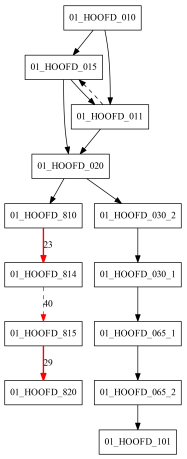


(a) Age $\geq$ 70

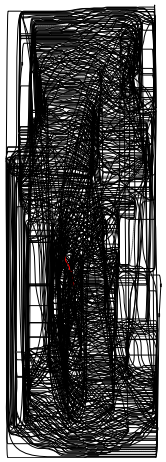


(b) Age $\leq$ 35

Fig. 4: SEPSIS logs, MFVA output [18]



(a) BPIC15 (M1)



(b) BPIC15 (M2)

Fig. 5: BPIC15 logs, MFVA output [18]

In our evaluation, we give each pair of log variants as an input to DRVA, MFVA, and PESVA. While DRVA and PESVA produce natural language statements, MFVA returns graphs named *mutual fingerprints*. Since the different outputs do not allow for a straightforward comparison, we first analyse the results of DRVA and MFVA with the goal to highlight the commonalities and differences in their output, then we compare our output with that of PESVA. We remark that the three tools provide alternative perspectives of the process variants differences. Therefore, they should not be seen as mutually exclusive tools, but rather complementary.

After the qualitative comparison, we also conduct a performance comparison of the three tools, reporting their execution times. All the experiments were run on an Intel Core i7-8565U@1.80GHz with 32GB RAM equipped with Windows 10 Pro (64-bit), with a timeout of 3 hours per variant analysis. The input parameters we used for our DRVA are the default ones explained in Sec. 3.5. The declarative process models are discovered through MINERful [7] with a Support threshold of 0.5 and a Confidence threshold of 0.0, so as to keep the specifications loose enough to capture rare behaviour too. Markedly, the execution times reported for DRVA include the discovery times.

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-030-85440-9_5

Table 3: SEPSIS logs – DRVA output (our approach)

	Statistically significant differences (*Variant A*: Age ≥ 70; *Variant B*: Age ≤ 35)
1	In *Variant A*, it is 37.4% more likely than in *Variant B* that Admission NC occurs in a process instance.
2	In *Variant A*, it is 37.4% more likely than in *Variant B* that if ER Registration occurs, Admission NC will occur afterwards without any other occurrence of ER Registration in between.
3	In *Variant A*, it is 37.4% more likely than in *Variant B* that if ER Sepsis Triage occurs, also Admission NC occurs.
4	In *Variant A*, it is 37.4% more likely than in *Variant B* that if ER Triage occurs, also Admission NC occurs.
5	In *Variant A*, it is 33.9% more likely than in *Variant B* that IV Antibiotics occurs in a process instance.
6	In *Variant A*, it is 33.9% more likely than in *Variant B* that if ER Registration occurs, also IV Antibiotics occurs.
7	In *Variant A*, it is 33.9% more likely than in *Variant B* that if ER Sepsis Triage occurs, IV Antibiotics will occur afterwards without any other occurrence of ER Sepsis Triage in between.
8	In *Variant A*, it is 33.9% more likely than in *Variant B* that if ER Triage occurs, also IV Antibiotics occurs.
9	In *Variant A*, it is 31.2% more likely than in *Variant B* that IV Liquid occurs in a process instance.
10	In *Variant A*, it is 31.2% more likely than in *Variant B* that if ER Registration occurs, also IV Liquid occurs.
11	In *Variant A*, it is 31.2% more likely than in *Variant B* that if ER Sepsis Triage occurs, also IV Liquid occurs.
12	In *Variant A*, it is 31.2% more likely than in *Variant B* that if ER Triage occurs, also IV Liquid occurs.

Table 4: BPIC15 logs – DRVA output (our approach). The list reports the first 10 top-ranked behavioural differences.

	Statistically significant differences (*Variant A*: Municipality $= 1$; *Variant B*: Municipality $= 2$)
1	It happens only in *Variant A* that 01_HOOFD_456 may occur at most once in a process instance.
2	It happens only in *Variant A* that 01_HOOFD_492_1 may occur at most once in a process instance.
3	It happens only in *Variant A* that 01_HOOFD_492_2 may occur at most once in a process instance.
4	It happens only in *Variant A* that if 01_HOOFD_456 occurs, also 01_HOOFD_010 occurs.
5	It happens only in *Variant A* that if 01_HOOFD_456 occurs, also 01_HOOFD_015 occurs.
6	It happens only in in *Variant B* that if 16_LGSD_010 occurs, also 01_HOOFD_490_2 occurs.
7	It happens only in *Variant B* that if 16_LGSD_010 occurs, also 01_HOOFD_495 occurs.
8	It happens only in in *Variant B* that if 16_LGSD_010 occurs, also 02_DRZ_010 occurs.
9	It happens only in in *Variant B* that if 16_LGSD_010 occurs, also 04_BPT_005 occurs.
10	It happens only in in *Variant B* that if 16_LGSD_010 occurs, also 09_AH_I_010 occurs.

Results and discussion. Figure 4 shows the output of MFVA given the variants of the SEPSIS log as an input. In the figure, dashed arcs and red arcs capture a statistically significant difference. The former evidence a change in the processing-time between the traces of the two variants, while the latter capture a difference in frequency (annotated on the arcs). Table 3 shows the output of DRVA given the same logs. We can notice that the difference ranked first in the table (PARTICIPATION(Admission NC) is more likely to occur if the age of the treated patient is over 70) makes explicit a difference that could not be directly inferred by visually comparing Fig. 4(a) with Fig. 4(b), although the alternative paths may suggest a behavioural change. The same holds for the fifth and ninth evidence. We observe that the other statements are related to the three we mentioned (also highlighted in Table 3). A further simplification step to enhance the ranking according to the discriminative power of the statements is an intriguing problem that paves the path for future endeavours building upon this work.

When the behaviour becomes less rigid and more flexible, the variants tend to have a higher percentage of distinct traces, like in the BPIC15 logs. In such cases, the output of MFVA becomes too simplistic (see Fig. 5(a)) or so complex to the extent of being even barely interpretable (see Fig. 5(b)). This result is, in general, typical of approaches

Pre-print copy of the manuscript published by Springer (available at link.springer.com) identified by DOI: 10.1007/978-3-030-85440-9_5

based on graphs depicting processes imperatively [4,10,18]. In fact, the more flexible the behaviour recorded in the log, the larger the output graph (unless filtering is applied). On the other hand, our approach shows the differences as a list of declarative statements, focusing on the differences rather than on the overall picture (see Tables 3 and 4). This is a benefit shared with PESVA, which also outputs natural language statements. However, PESVA suffers from scalability issues (PESVA timed out after running for three hours on BPIC15) and the differences identified by PESVA are limited in scope, which may hamper explanatory power and understandability. On the SEPSIS variants, e.g., PESVA indicates that "Task IV Liquid(4) can be skipped in model 2, whereas in model 1 it is always executed". Such a natural language formulation may be difficult to be interpreted, at times, in fact, task IV Liquid(4) does not refer to any execution of IV Liquid, but to the IV Liquid event that occurs as the fourth one. Our statements, instead, are broader in scope because they are not extrapolated by prime event structures but by declarative rules, therefore they refer to rules exerted on the whole process run rather than on the single occurrence of events. Also, we remind that the differences captured by PESVA are not guaranteed to be statistically significant.

In terms of execution time, DRVA outperforms MFVA and PESVA when analysing the BPIC13, BPIC15, and SEPSIS logs: considering the best execution times between those of MFVA and PESVA, the runs required $81.8\,\mathrm{s}$, $4901.3\,\mathrm{s}$, and $5.1\,\mathrm{s}$, respectively. DRVA required instead $4.9\,\mathrm{s}$, $326.7\,\mathrm{s}$, and $4.6\,\mathrm{s}$, respectively. For the analysis of the RTFMP log, instead, our technique required $38.1\,\mathrm{s}$, thus outperforming MFVA ($1152.9\,\mathrm{s}$) but not PESVA ($21.7\,\mathrm{s}$). Nevertheless, we underline that our approach appears to be more scalable than the MFVA and PESVA: to process the BPIC15 log, indeed, the former took more than an hour and a half, whereas the latter timed out at three hours.

5 Conclusion

In this paper, we proposed a novel method to perform variant analysis and discover statistically-significant differences in the form of declarative rules expressed in natural language. We compared our method with state-of-the-art methods noticing that DRVA provides a different level of expressiveness, easier output interpretation, and faster execution time. Future research endeavours include the extension of our method to encompass the full spectrum of LTLp_f formulae [5] and hybrid models [16] as the rule language in place of standard DECLARE, as well as multi-perspective specifications beyond the sole control-flow structure [15]. Also, we aim at further simplifying the output via redundancy-reduction techniques such as those in [6], to be adapted in order to improve the distinction of variants rather than reducing the rules of a single specifications. Other interesting avenues are the identification of statistically significant performance rules as opposed to behavioural rules, and the enhancement of the statistical soundness of the results addressing the multiple testing problem through p-value correction techniques [9].

Acknowledgments. The work of C. Di Ciccio was partially funded by the Italian MIUR under grant "Dipartimenti di eccellenza 2018-2022" of the Department of Computer Science at Sapienza and by the Sapienza research project "SPECTRA".

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-030-85440-9_5

References

1. van der Aa, H., Balder, K.J., Maggi, F.M., Nolte, A.: Say it in your own words: Defining declarative process models using speech recognition. In: BPM Forum. pp. 51–67 (2020)
2. van der Aalst, W.M.: Process Mining - Data Science in Action. Springer (2016)
3. van Beest, N.R., Dumas, M., García-Bañuelos, L., La Rosa, M.: Log delta analysis: Interpretable differencing of business process event logs. In: BPM. pp. 386–405 (2016)
4. Bolt, A., de Leoni, M., van der Aalst, W.M.: Process variant comparison: using event logs to detect differences in behavior and business rules. Information Systems **74**, 53–66 (2018)
5. Cecconi, A., De Giacomo, G., Di Ciccio, C., Maggi, F.M., Mendling, J.: A temporal logic-based measurement framework for process mining. In: ICPM. pp. 113–120 (2020)
6. Di Ciccio, C., Maggi, F.M., Montali, M., Mendling, J.: Resolving inconsistencies and redundancies in declarative process models. Inf. Syst. **64**, 425–446 (2017)
7. Di Ciccio, C., Mecella, M.: On the discovery of declarative control flows for artful processes. ACM Trans. Management Inf. Syst. **5**(4), 24:1–24:37 (2015)
8. Edgington, E.S.: Approximate randomization tests. The Journal of Psychology **72**(2), 143–149 (1969)
9. Hämäläinen, W., Webb, G.I.: A tutorial on statistically sound pattern discovery. Data Mining and Knowledge Discovery **33**(2), 325–377 (2019)
10. Nguyen, H., Dumas, M., La Rosa, M., ter Hofstede, A.H.: Multi-perspective comparison of business process variants based on event logs. In: ER. pp. 449–459 (2018)
11. Nichols, T.E., Holmes, A.P.: Nonparametric permutation tests for functional neuroimaging: a primer with examples. Human brain mapping **15**(1), 1–25 (2002)
12. Pesic, M., Bosnacki, D., van der Aalst, W.M.: Enacting declarative languages using LTL: avoiding errors and improving performance. In: SPIN. pp. 146–161 (2010)
13. Pitman, E.J.: Significance tests which may be applied to samples from any populations. Supplement to the Journal of the Royal Statistical Society **4**(1), 119–130 (1937)
14. Poelmans, J., Dedene, G., Verheyden, G., Van der Mussele, H., Viaene, S., Peters, E.: Combining business process and data discovery techniques for analyzing and improving integrated care pathways. In: ICDM. pp. 505–517 (2010)
15. Schönig, S., Di Ciccio, C., Maggi, F.M., Mendling, J.: Discovery of multi-perspective declarative process models. In: ICSOC. pp. 87–103 (2016)
16. Slaats, T.: Declarative and hybrid process discovery: Recent advances and open challenges. J. Data Semant. **9**(1), 3–20 (2020)
17. Suriadi, S., Wynn, M.T., Ouyang, C., ter Hofstede, A.H., van Dijk, N.J.: Understanding process behaviours in a large insurance company in australia: A case study. In: CAiSE. pp. 449–464 (2013)
18. Taymouri, F., La Rosa, M., Carmona, J.: Business process variant analysis based on mutual fingerprints of event logs. In: CAiSE. pp. 299–318 (2020)
19. Taymouri, F., La Rosa, M., Dumas, M., Maggi, F.M.: Business process variant analysis: Survey and classification. Knowledge-Based Systems **211**, 106557 (2021)
20. Welch, W.J.: Construction of permutation tests. Journal of the American Statistical Association **85**(411), 693–698 (1990)
21. Wu, J., He, Z., Gu, F., Liu, X., Zhou, J., Yang, C.: Computing exact permutation p-values for association rules. Information Sciences **346**, 146–162 (2016)

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-030-85440-9_5

This document is a pre-print copy of the manuscript
(Cecconi, Augusto, and Di Ciccio 2021)
published by Springer (available at link.springer.com).

The final version of the paper is identified by DOI: 10.1007/978-3-030-85440-9_5

References

Cecconi, Alessio, Adriano Augusto, and Claudio Di Ciccio (2021). "Detection of Statistically Significant Differences Between Process Variants Through Declarative Rules". In: *BPM Forum*. Ed. by Artem Polyvyanyy, Moe Thandar Wynn, Amy Van Looy, and Manfred Reichert. Vol. 427. Lecture Notes in Business Information Processing. Springer, pp. 73–91. ISBN: 978-3-030-85439-3. DOI: 10.1007/978-3-030-85440-9_5.

BibTeX

```
@InProceedings{   Cecconi.etal/BPMF2021:VariantsThroughDeclarativeRules,
  author        = {Cecconi, Alessio and Augusto, Adriano and Di Ciccio,
                   Claudio},
  booktitle     = {BPM Forum},
  title         = {Detection of Statistically Significant Differences Between
                   Process Variants Through Declarative Rules},
  year          = {2021},
  pages         = {73--91},
  crossref      = {BPM2021Forum},
  doi           = {10.1007/978-3-030-85440-9_5},
  keywords      = {Process mining; declarative modelling; permutation test;
                   variant analysis}
}
@Proceedings{     BPM2021Forum,
  title         = {Business Process Management Forum - {BPM} Forum 2021,
                   Rome, Italy, September 06-10, 2021, Proceedings},
  year          = {2021},
  editor        = {Artem Polyvyanyy and Moe Thandar Wynn and Amy Van Looy and
                   Manfred Reichert},
  isbn          = {978-3-030-85439-3},
  publisher     = {Springer},
  series        = {Lecture Notes in Business Information Processing},
  volume        = {427}
}
```