




Multi-perspective Process Analysis: Mining the Association Between Control Flow and Data Objects

Dina Bayomie¹ , Kate Revoredo¹ , and Jan Mendling² 

¹ Vienna University of Economics and Business (WU), Vienna, Austria
{[dina.sayed.bayomie.sobh](mailto:dina.sayed.bayomie.sobh@wu.ac.at),[kate.revoredo](mailto:kate.revoredo@wu.ac.at)}@wu.ac.at

² Humboldt University, Berlin, Germany
jan.mendling@hu-berlin.de

Abstract. Process mining techniques provide process analysts with insights into interesting patterns of a business process. Current techniques have focused by and large on the explanation of behavior, partially by help of features that relate to multiple perspectives beyond just pure control flow. However, techniques to provide insights into the connection between data elements of related events have been missing so far. Such connections are relevant for several analysis tasks such as event correlation, resource allocation, or log partitioning. In this paper, we propose a multi-perspective mining technique for discovering data connections. More specifically, we adapt concepts from association rule mining to extract connections between a sequence of events and behavioral attributes of related data objects and contextual features. Our technique was evaluated using real-world events supporting the usefulness of the mined association rules.

Keywords: Association rules · Process analytic · Multi-perspective process analysis

1 Introduction

Process mining techniques provide process analysts with insights into interesting patterns of a business process [1, 2]. Classical techniques in this area such as automatic process discovery or conformance checking focus on the control-flow perspective to generate process insights based on event logs. These insights are helpful for analysts to understand the performance of a process and root causes of anomalies.

A key challenge of root cause analysis is to identify as many potential explanations for a process issue as possible. Such explanations are not necessarily restricted to the control flow perspective. Indeed, the potential of multi-perspective process mining has been emphasized by several contributions [3–6]. These techniques have focused largely on the explanation of behavior, partially by help of features that relate to multiple perspectives. However, techniques to

provide insights into the connection between data elements of related events have been missing so far. Such connections are particularly relevant for analysis tasks such as event correlation, resource allocation, or log partitioning.

In this paper, we address this research gap by developing a multi-perspective mining technique for the discovery of data connections. More specifically, we adapt concepts from association rule mining with pre- and post-processing [7, 8] to extract connections between a sequence of events and behavioral attributes of related data objects and contextual features. During pre-processing, techniques such as filtering or partitioning can be applied. During post-processing, techniques such as rule comparison or visualization can be used. Our technique was evaluated using real-world event logs. The results show that our method is able to extract association rules that provide useful insights on the event log to support process analysts.

The rest of this paper is organized as follows. In Sect. 2 we review important concepts such as event log and association rule miner and we discuss prior research. In Sect. 3, we describe our method. In Sect. 4 we evaluate our method and discuss the findings. In Sect. 5 we conclude and provide directions for future research.

2 Background

In this section, we discuss the fundamental concepts that are used by our approach and describe some prior work. Section 2.1 introduces the basic notions of event, case and event log. Section 2.2 describes essential concepts of association rule mining. Finally, Sect. 2.3 summarizes related work.

2.1 Preliminaries

We start by introducing the basic notion of event (i.e., the atomic unit of execution) and then discuss the notions of event log and case in turn.

Definition 1 (Event, attribute). *An event e represents the execution of a process activity. An event has a set of attributes (\mathfrak{A}), that provides information about context data objects, e.g. activity (Act), timestamp (Ts), resource, cost, ..., etc. An attribute $Attr \in \mathfrak{A}$ has a non-empty set of domain values $Dom(Attr)$, such that each event is mapped to one of the attribute's domain values. We indicate the value mapped by $Attr$ to an event e by using a dot notation, i.e., $e.Attr$.*

We assume the mapping of Ts to be coherent with \preceq , i.e., if $e \preceq e'$ then $e.Ts \preceq e'.Ts$. Considering the total ordering as a mapping from a convex subset of integers, we can assign to every event a unique integer index (or *event id* for short), induced by \preceq on the events. We shall denote the index i of an event e as a subscript, i.e., e_i .

Definition 2 (Case). A case $\sigma = \langle e_{\sigma_1}, \dots, e_{\sigma_m} \rangle$ is a finite sequence of length m of events e_{σ_i} with $1 \leq i \leq m$ induced by \leq , i.e., such that $e_{\sigma_i} \leq e_{\sigma_k}$ for every $i \leq k \leq m$. We assume every case to be assigned a unique case identifier (case id for short), namely an integer in a convex subset.

Definition 3 (Event log). An event log $L = \{\sigma_1, \dots, \sigma_n\}$ is a finite non-empty set of non-overlapping cases, i.e., if $e \in \sigma_i$, then $e \notin \sigma_j$ for all $i, j \in [1 \dots n]$, $i \neq j$.

Figure 1 depicts an example of an event log L in a tabular representation. L has three cases grouped over the *case id* attribute. The case σ_1 defined by the case id 1 is $\langle e_1, e_2, e_3, e_4 \rangle$. Notice that it preserves the order of the events within the case. L contains five attributes that describe the event, such that $\mathfrak{A} = \{\text{Activity, Timestamp, Resource, Type, Supervisor}\}$. For example, the event e_1 indicates that activity $e_1.\text{Act} = \text{Applicationsubmit}$ was executed by resource $e_1.\text{Resource} = \text{C1}$ and finished at time $e_1.\text{Ts} = \text{“01/06/2020”}$. $e_1.\text{Type} = \text{Car}$ and $e_1.\text{Supervisor} = \text{R1}$ represent additional data objects associated with the event.

Case Id	Activity	Timestamp	Resource	Type	Supervisor
1	Application submit	01/06/2020	C1	Car	R1
1	Review application	02/06/2020	R1	Car	R3
1	Accept application	03/06/2020	R3	Car	R3
1	Send notification	03/06/2020	R1	Car	R3
2	Application submit	03/06/2020	C2	House	R2
2	Review application	04/06/2020	R2	House	R3
2	Reject application	08/06/2020	R3	House	R3
2	Send notification	08/06/2020	R2	House	R3
3	Application submit	04/08/2020	C4	House	R2
3	Review application	04/08/2020	R2	House	R3
3	Request documents	07/08/2020	R2	House	R3
3	Review application	07/08/2020	R2	House	R3
3	Accept application	11/08/2020	R3	House	R3
3	Send notification	11/08/2020	R2	House	R3

Fig. 1. An event log sample

2.2 Association Rule Mining

Association rule mining is a rule-based machine learning method that searches in a transaction database (also called transaction table) for relevant relations, i.e., frequently occurring patterns, correlations, or associations, between pairs of attribute and its value [8,9]. An association rule R represents the influence of a set of pairs of attributes and their values, called *antecedent* of the rule, in another set of pairs of attributes and their values, called *consequent* of the rule, i.e. $R : \text{IF } \textit{antecedent} \text{ THEN } \textit{consequent}$.

The mining algorithm computes different measures to rank the discovered rules, such as support, confidence and lift [10]. The support measures how frequently a rule R appears in the transaction database T :

$$support(R) = \frac{|(antecedent \cup consequent) \subseteq T|}{|T|} \quad (1)$$

The confidence measures how often the rule is satisfied in the transaction database:

$$confidence(R) = \frac{support(antecedent \cup consequent)}{support(antecedent)} \quad (2)$$

The lift is an objective interestingness measure that assesses the performance of a rule at classifying the transaction database. It measures the deviation of the support of the whole rule from the support expected under independence given the supports of the antecedent and the consequent:

$$lift(R) = \frac{support(antecedent \cup consequent)}{support(antecedent) * support(consequent)} \quad (3)$$

2.3 Prior Work

There have been various works on finding correlations between process events [11]. They have focused on different tasks such as identifying process instances, i.e. cases, for event log generation [12], also considering object-centric perspective [13] and middleware [14], for discovering a process model [15] or enriching an event log with sensor data [16]. However, the area has not received much attention from the perspective of analyzing the correlation of events and data objects behavior within an event log. Our work is most closely related to prior work in this direction.

In [6], association rules with respect to control flow, resources, and temporal process execution behaviour are learned to monitor the process for anomaly detection. The control rules indicate flow patterns, i.e. expected sequence of events. The temporal rules concern the duration of the activities. The resource rules constrain which resources execute the activities. They are split into two groups, i.e., rules covering all activities that must be executed by different resources, and rules covering all activities that must be executed by the same resource. In this work, we evaluate other data objects besides resource and also how their behavior correlate with the events.

In [3], data objects are used to revise the set of declare rules extracted from an event log. With the consideration of data objects it was possible to identify more precise correlations between the events, eliminate constraints that were not meaningful, and provide more understanding to the remaining constraints. This approach works in a broad way searching for general patterns considering the control-flow information and uses the data object to refine the patterns found.

In [5], the authors propose a multi-perspective trace clustering approach that uses the data objects to compute the case distance measure. Using the control-flow, resource and data objects perspectives improve the homogeneity of the cases within the same cluster. In this work, we focus on finding commonalities among cases by investigating associations between control-flow and data objects behavior.

In [4], the authors use multiple perspectives to compare different processes. They visualize the difference by help of the process model and data objects information available in the event log. The work provides three comparative views. The first view shows the difference in the process performance and resource perspectives over the process model. The second view focuses on showing the activity similarity between the processes. The last view focuses on comparing the time perspective. In this work, we evaluate other data objects besides resource, performance and time and also it focuses on identifying the association patterns between the control-flow and the data objects perspectives of one event log.

3 Method

In this section, we describe our method *Event Log Rule Miner*, henceforth called *EL-RM*, which extracts association rules between the control-flow and data objects behavior within an event log. It is inspired on the knowledge discovery in database (KDD) process [17], being composed of three main steps: preparation of the event log, the mining itself and a post processing of the association rules discovered. Figure 2 presents an overview of *EL-RM*.

EL-RM receives as input an event log and returns as output association rules and information about these rules. It is composed of four steps. The first step prepares the event log data based on the process analyst objectives. The second step encodes the pre-processed event log as a transaction table. In the third step, the association rule miner is applied on the transaction table to discover the association rules that define the association relation between the control flow and the data objects behavior. In the fourth step the association rules are post-processed based on the business analyst’s objectives. The following sections detail these four steps.

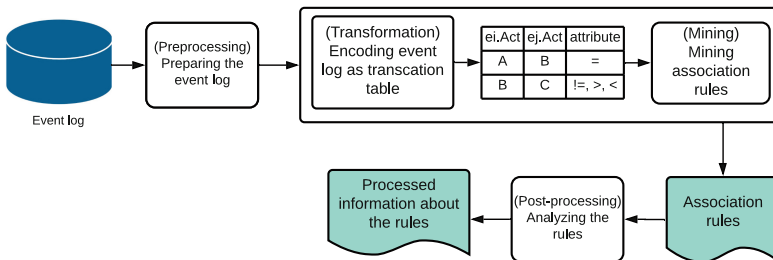


Fig. 2. EL-RM method overview

3.1 Preparing the Event Log

The main two goals of this step are to guarantee the quality of the event log and its alignment with the process analyst's analysis objectives. For addressing the former, pre-processing techniques such as data cleaning [18] are applied and for addressing the latter filtering [19] or partitioning [20] of the event log are used.

Filtering techniques select cases in the event log, i.e., $L' \subset L : L' = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$, based on criteria to reach the business analysts objectives. For example, selecting the non-conformed cases with the goal to investigate the reasons for this non-conformance, or selecting cases that have cycle time above a certain threshold (e.g., longest cycle time) to analyze the root cause of the delay within these cases.

Partitioning techniques split the event log into groups with common behavior allowing the process analyst to investigate behavioral differences. Various partitioning criteria can be used, for instance based on time window interval, on process variations, cycle time duration or the number of events per case. For example, partitioning the event log depicted in Fig. 1 using time window of one month as the partition criterion returns the partitioning $\{\{\sigma_1, \sigma_2\}, \{\sigma_3\}\}$, while using the cycle time duration with three days as a split criterion yields the partition $\{\{\sigma_1\}, \{\sigma_2, \sigma_3\}\}$.

3.2 Encoding the Event Log as a Transaction Table

The association rule miner runs on a transaction table. In this section, we describe the technique that we developed to encode an event log as a transaction table.

We define a transaction as the behavior of data attributes over sequences of events. We investigate the change behavior of data objects from two perspectives: *atomic* and *complex* perspectives. The former represents the behavior of a single attribute, i.e., $\text{Attr} \in \mathfrak{A}$ over the events, while the latter represents the behavior of a pair of attributes that share domain values, i.e., $(\text{Attr}, \text{Attr}') \in \mathfrak{A} \times \mathfrak{A} : \text{Dom}(\text{Attr}) \cap \text{Dom}(\text{Attr}') \neq \phi$ over the events. For example, considering the event log in Fig. 1, the *atomic* perspective based on the attribute *Resource* investigates how the *Resource* changes over the events and the *complex* perspective based on the *Resource* and *Supervisor* investigate how they changed over the events. Definition 4 formally defines a transaction in *EL-RM*.

Definition 4 (Transaction). *A transaction $t = \langle e_i.\text{Act}, e_j.\text{Act}, s_{\text{Attr}_1}, \dots, s_{\text{Attr}_m}, s_{(\text{Attr}, \text{Attr}')_0}, \dots, s_{(\text{Attr}, \text{Attr}')_l} \rangle$ is an n -tuple with the two first terms related to the control-flow perspective, i.e., activities $e_i.\text{Act}$ and $e_j.\text{Act}$, the following $[3, k]$ terms related to the status of behavior of the attributes in atomic perspective where k is bounded to the number of data attributes in the event log ($k - 3 = |\mathfrak{A}|$) and the remaining terms related to the status of behavior of the attributes in the complex perspective, if exists. The status of the attributes behavior is computed based on their values as stated in Eq. (4). Note that for the atomic perspective ($\text{Attr} = \text{Attr}'$).*

$$\text{status}(e_i, e_j, \text{Attr}, \text{Attr}') \begin{cases} = & \text{if } e_i.\text{Attr} = e_j.\text{Attr}' \\ \neq & \text{if } e_i.\text{Attr} \neq e_j.\text{Attr}' \text{ and } \text{Dom}(\text{Attr}) \text{ is discrete} \\ > & \text{if } e_i.\text{Attr} > e_j.\text{Attr}' \text{ and } \text{Dom}(\text{Attr}) \text{ is numeric} \\ < & \text{if } e_i.\text{Attr} < e_j.\text{Attr}' \text{ and } \text{Dom}(\text{Attr}) \text{ is numeric} \end{cases} \quad (4)$$

Definition 5 (Transaction table). A transaction table $T = \{t_1, t_2, \dots, t_n\}$ is a set of transactions represented in a tabular structure where each item of the tuple of the transaction corresponds to a column in the table.

Algorithm 1 presents our algorithm for encoding an event log into a transaction table.

Algorithm 1. Create transaction table from an event log

Require: Event Log $L = \{\sigma_1, \dots, \sigma_n\}$, Attributes \mathfrak{A} , Decision attributes D ,

Ensure: Transaction table T

- 1: $C = \{(\text{Attr}, \text{Attr}') \mid \text{Attr}, \text{Attr}' \text{ in } \mathfrak{A} \wedge \text{Dom}(\text{Attr}) \cap \text{Dom}(\text{Attr}') \neq \emptyset\}$
 - 2: $T = \text{new table}(e_i.\text{Act}, e_j.\text{Act}, L.\mathfrak{A}, C)$
 - 3: **for all** $\sigma \in L$ **do**
 - 4: $T.\text{append}(\text{Create_Transactions}(\sigma, \mathfrak{A}, C))$ Algorithm 2
 - 5: **for all** d **in** D **do**
 - 6: $G = \text{partitions of } \sigma \text{ based on decision attribute } d \text{ values}$
 - 7: **for all** $g \in G$ **do**
 - 8: **if** $|g| \geq 2$ **then**
 - 9: $T.\text{append}(\text{Create_Transactions}(g, \mathfrak{A}, C))$ Algorithm 2
 - 10: **end if**
 - 11: **end for**
 - 12: **end for**
 - 13: **end for**
-

It receives an event log L , the set of attributes observed in the log \mathfrak{A} and an optional set of attributes D indicated by the process analyst to be considered when building the transactions. The algorithm starts by searching for the pairs of attributes that represent the complex perspective (see line 1). For example, considering the event log in Fig. 1, both “Resource” and “Supervisor” attributes share domain values, so a new complex attribute is added to represent this relation. The following steps concern with creating transactions for each case σ in L following two alternatives for event pairs selection. The first one is based on the direct successor relation between the events within a case, i.e., considering $e_i.\text{Act}$ and its direct successor event $e_{i+1}.\text{Act}$ (line 4). The second one is based on selecting the pairs of successors events i.e., considering $e_i.\text{Act}$ and its successor event $e_j.\text{Act}$ where $j > i$, based on the decision attributes chosen by the analyst (D) (lines 5-11). In this approach, the events in σ are grouped based on the values of the decision attributes D . In each group the events share the same value of $d \in D$ (lines 6). Then, for each group with at least 2 events, Algorithm 1 assesses

the behavior of the atomic and complex attributes over selected subset of events g , by calling Algorithm 2 (line 9). For example, in Fig. 3 the fourth row in the transaction table is created by grouping σ_1 over “Resource” attribute as input decision attribute.

Algorithm 2. Create_Transactions(E, A, C): Create transactions over events in a case $E \in \sigma$

Require: Set of Events E , Attributes \mathfrak{A} , Complex attributes C

Ensure: *setoftransactionstransactions*

```

1: transactions =
2: row = 1
3: for (i = 1; i < |E| - 1; i = i + 1) do
4:   ei = E[i], ej = E[i + 1]
5:   transactions[row][1] = ei.Act
6:   T[row][2] = ej.Act
7:   col = 3
8:   for all a in  $\mathfrak{A}$  do
9:     transactions[row][col] = status(ei, ej, a, a)
10:    col = col + 1
11:  end for
12:  for all (a, a') in C do
13:    if status(ei, ej, a, a') ∈ {=, >, <} then
14:      transactions[row][col] = "ei." + a + status(ei, ej, a, a') + "ej." + a'
15:    else if status(ei, ej, a', a) ∈ {=, >, <} then
16:      transactions[row][col] = "ei." + a' + status(ei, ej, a, a') + "ej." + a
17:    else
18:      transactions[row][col] = status(ei, ej, a, a')
19:    end if
20:    col = col + 1
21:  end for
22:  row = row + 1
23: end for

```

Algorithm 2 shows how we take the decision of attributes change behavior over a set of events. For each pair of events e_i and e_j in the input set of events E where e_i are followed by e_j in E , we first check the atomic attribute behavior over e_i and e_j by using $status(e_i, e_j, a, a)$ as per Eq. (4) (lines 8–11). Then, the second step checks the complex attribute (a, a') behavior over e_i and e_j by assessing if both events have the same value over a and a' by using $status(e_i, e_j, a, a')$ as per Eq. (4) (line 12–23). Figure 3 depicts the transaction table generated by Algorithm 1 when receiving a filtering of the event log showed in Fig. 1.

3.3 Mining Association Rules

The third step executes the association rule miner over the transaction table, considering only the rules that sustain the control-flow perspective.

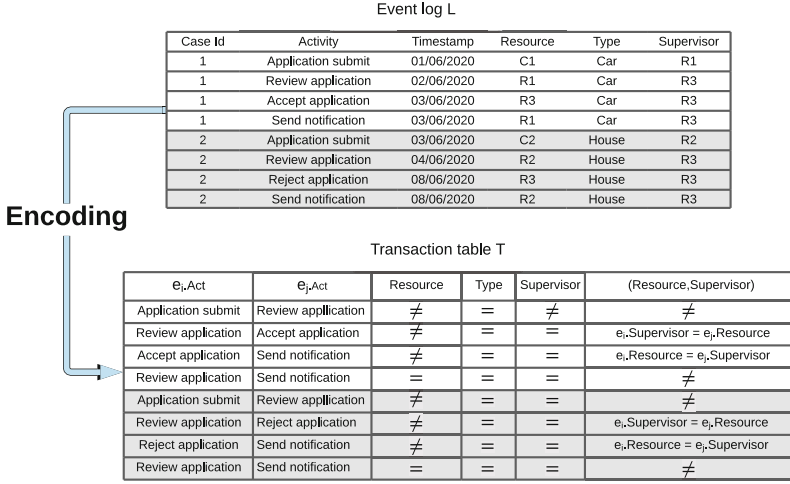


Fig. 3. Encoding the event log L into the transaction table T

Definition 6 (Association rule). *EL-RM establishes the association rules with a control flow information in the antecedent, and the consequent represents the data attributes behavior as follows:*

$$R : IF e_i.Act = a \wedge e_j.Act = b THEN e_i.Attr \leq e_j.Attr'$$

$$\leq : < \mid > \mid = \mid \neq$$

$R_1 : IF e_i.Act = "Application submit "$ and $e_j.Act = "Review Application "$ THEN $e_i.Resource \neq e_j.Resource$	$R_{14} : IF e_i.Act = "Review Application "$ and $e_j.Act = "Send notification "$ THEN $e_i.Resource = e_j.Resource$
$R_2 : IF e_i.Act = "Application submit "$ and $e_j.Act = "Review Application "$ THEN $e_i.Type = e_j.Type$	$R_{15} : IF e_i.Act = "Review Application "$ and $e_j.Act = "Send notification "$ THEN $e_i.Type = e_j.Type$
$R_3 : IF e_i.Act = "Application submit "$ and $e_j.Act = "Review Application "$ THEN $e_i.Supervisor = e_j.Supervisor$	$R_{16} : IF e_i.Act = "Review Application "$ and $e_j.Act = "Send notification "$ THEN $e_i.Supervisor = e_j.Supervisor$
$R_4 : IF e_i.Act = "Application submit "$ and $e_j.Act = "Review Application "$ THEN $e_i.Resource \neq e_j.Supervisor$	$R_{17} : IF e_i.Act = "Review Application "$ and $e_j.Act = "Send notification "$ THEN $e_i.Resource \neq e_j.Supervisor$
$R_5 : IF e_i.Act = "Application submit "$ and $e_j.Act = "Review Application "$ THEN $e_j.Resource \neq e_i.Supervisor$	$R_{18} : IF e_i.Act = "Review Application "$ and $e_j.Act = "Send notification "$ THEN $e_j.Resource \neq e_i.Supervisor$
$R_6 : IF e_i.Act = "Review Application "$ and $e_j.Act = "Accept Application "$ THEN $e_i.Resource \neq e_j.Resource$	$R_{19} : IF e_i.Act = "Review Application "$ and $e_j.Act = "Reject Application "$ THEN $e_i.Type = e_j.Type$
$R_7 : IF e_i.Act = "Review Application "$ and $e_j.Act = "Accept Application "$ THEN $e_i.Type = e_j.Type$	$R_{20} : IF e_i.Act = "Review Application "$ and $e_j.Act = "Reject Application "$ THEN $e_i.Resource \neq e_j.Resource$
$R_8 : IF e_i.Act = "Review Application "$ and $e_j.Act = "Accept Application "$ THEN $e_i.Supervisor \neq e_j.Supervisor$	$R_{21} : IF e_i.Act = "Review Application "$ and $e_j.Act = "Reject Application "$ THEN $e_i.Supervisor = e_j.Supervisor$
$R_9 : IF e_i.Act = "Review Application "$ and $e_j.Act = "Accept Application "$ THEN $e_i.Supervisor = e_j.Resource$	$R_{22} : IF e_i.Act = "Review Application "$ and $e_j.Act = "Reject Application "$ THEN $e_i.Supervisor = e_j.Resource$
$R_{10} : IF e_i.Act = "Accept Application "$ and $e_j.Act = "Send notification "$ THEN $e_i.Resource \neq e_j.Resource$	$R_{23} : IF e_i.Act = "Reject Application "$ and $e_j.Act = "Send notification "$ THEN $e_i.Resource \neq e_j.Resource$
$R_{11} : IF e_i.Act = "Accept Application "$ and $e_j.Act = "Send notification "$ THEN $e_i.Type = e_j.Type$	$R_{24} : IF e_i.Act = "Accept Application "$ and $e_j.Act = "Send notification "$ THEN $e_i.Type = e_j.Type$
$R_{12} : IF e_i.Act = "Accept Application "$ and $e_j.Act = "Send notification "$ THEN $e_i.Supervisor = e_j.Supervisor$	$R_{25} : IF e_i.Act = "Reject Application "$ and $e_j.Act = "Send notification "$ THEN $e_i.Supervisor = e_j.Supervisor$
$R_{13} : IF e_i.Act = "Accept Application "$ and $e_j.Act = "Send notification "$ THEN $e_i.Resource = e_j.Supervisor$	$R_{26} : IF e_i.Act = "Reject Application "$ and $e_j.Act = "Send notification "$ THEN $e_i.Resource = e_j.Supervisor$

Fig. 4. The association rules extracted from transaction table T in Fig. 3

Figure 4 depicts the association rules mined from the transaction table T in Fig. 3. For example, the discovered rule R_1 states that different resources executed the activities “Application submit” and “Review Application” when they were executed in this order.

3.4 Analyzing the Rules

In the final step, EL-RM prepares the extracted association rules to align with the business analyst objectives. There are many possible post analyses to be performed to improve the usages, interpretation, and visualization of the association rules. In this paper, we discuss three post analyses options (a) combining the rules, (b) ranking the rules, and (c) Comparing the rules. These analyses can be used separately or together based on the analyst objectives.

Combine the Rules. The first option is compressing the extracted rules to improve the rules’ visualization. EL-RM aggregates the rules based on common antecedent and common consequent. We propose three possible aggregations. In the first step, rules with the same antecedent are aggregated. The new consequent is the conjunction of the individual consequents. In the second step, rules with the same consequent are aggregated. The new antecedent is the disjunction of the individual antecedents. In the third step, the aggregated rules that share the same consequent or antecedent are aggregated.

For instance, antecedent aggregation applied on the association rules shown in Fig. 4 reduces the number of rules from 26 to 6 rules. R_A in Fig. 5, for example, combines rules R_1 , R_2 , R_3 , R_4 , and R_5 . Consequent aggregation reduces the rules from 26 to 8 rules. R_C in Fig. 5, for example, combines rules R_1 , R_6 , R_{10} , and R_{22} . Using both aggregation reduces the rules from 26 to 4 rules. R_B in Fig. 5, for example, combines rules R_1 , R_2 , R_6 , R_7 , R_{10} , R_{11} , R_{23} , and R_{24} . By aggregating the rules, EL-RM reduces the number of the rules without losing any information. Also, it helps the analysts to focus on rules relevant to their analysis.

R_A : IF e_i . Act = " Application submit " and e_j . Act = " Review Application " THEN e_i . Resource \neq e_j . Resource \wedge e_i . Type = e_j . Type \wedge e_i . Supervisor = e_j . Supervisor \wedge e_i . Resource \neq e_j . Supervisor \wedge e_j . Resource \neq e_i . Supervisor
R_C : IF (e_i . Act = " Application submit " and e_j . Act = " Review Application ") \vee (e_i . Act = " Review Application " and e_j . Act = " Accept Application ") \vee (e_i . Act = " Accept Application " and e_j . Act = " Send notification ") \vee (e_i . Act = " Reject Application " and e_j . Act = " Send notification ") THEN e_i . Resource \neq e_j . Resource
R_B : IF (e_i . Act = " Application submit " and e_j . Act = " Review Application ") \vee (e_i . Act = " Review Application " and e_j . Act = " Accept Application ") \vee (e_i . Act = " Accept Application " and e_j . Act = " Send notification ") \vee (e_i . Act = " Reject Application " and e_j . Act = " Send notification ") THEN e_i . Resource \neq e_j . Resource \wedge e_i . Type = e_j . Type

Fig. 5. Example of combine rules from rules in Fig. 4

Ranking and Filtering. The second option is ranking and filtering the rules. The rules can be ranked based on the confidence or lift as per defined in Sect. 2.2. The lift measure is a well known objective interestingness measure, thus, the higher the lift is, the more interesting is the rule. In this step, it is also possible for the process analyst to filter the rules based on the activity represented in the antecedent or on the data objects behavior represented in the consequent to focus on specific analysis.

Comparing the Rules. If during the preparation of the event log, the process analyst decided to partitioning the event log, then in this step it is possible to compare the rules discovered in each of the partitioning. EL-RM induces various sets of rules from the extracted rules. The *All* rule set is the union of the set of rules extracted from each of the partitions (Rs) without duplicate rules:

$$All(Rs) = \bigcup_{i=1}^{|Rs|} Rs_i \quad (5)$$

The *Common* rules set is the intersection of the set of rules extracted for each of the partitions (Rs):

$$Common(Rs) = \bigcap_{i=1}^{|Rs|} Rs_i \quad (6)$$

We use the difference set operation to extract the distinct rules that distinguish each partition. We compare the rules per each partition against the common rules:

$$Difference_{com}(Rs_i, Common) = Rs_i - Common \quad (7)$$

For more distinctive rules, we compare the rules per each partition against the union set of the other partitions:

$$Difference_{partition}(Rs_i, Rs) = Rs_i - \bigcup_{\substack{j=0 \\ j \neq i}}^{|Rs|} Rs_j \quad (8)$$

Consider that time window of one month was used as the partition criterion in the event log shown in Fig. 1. The mining algorithm discovered 32 rules in total over the two partitions ($\{\{\sigma_1, \sigma_2\}, \{\sigma_3\}\}$). From these 32 rules, 16 rules are common among the partitions. The common rules spotlight the log's general rules, which exist at the intersection of all the partitions. The individual rules per each partition are found using the difference set operation with the common rules and the rest of the partitions. For example, rules in Fig. 6 distinguish the second partition as they are not satisfied by the first partition.

IF e_i . Act = " Review Application " and e_j . Act = " Request documents " THEN e_i . Type = e_j . Type
IF e_i . Act = " Review Application " and e_j . Act = " Request documents " THEN e_i . Resource = e_j . Resource
IF e_i . Act = " Request documents " and e_j . Act = " Review Application " THEN e_i . Type = e_j . Type
IF e_i . Act = " Request documents " and e_j . Act = " Review Application " THEN e_i . Resource = e_j . Resource
IF e_i . Act = " Request documents " and e_j . Act = " Review Application " THEN e_i . Supervisor = e_j . Supervisor
IF e_i . Act = " Request documents " and e_j . Act = " Review Application " THEN e_i . Resource \neq e_j . Supervisor
IF e_i . Act = " Review application " and e_j . Act = " Request documents " THEN e_i . Supervisor = e_j . Supervisor
IF e_i . Act = " Review application " and e_j . Act = " Request documents " THEN e_i . Resource \neq e_j . Supervisor
IF e_i . Act = " Review application " and e_j . Act = " Request documents " THEN e_j . Resource \neq e_i . Supervisor
IF e_i . Act = " Request documents " and e_j . Act = " Review Application " THEN e_j . Resource \neq e_i . Supervisor

Fig. 6. $Difference_{partition}(Rs_2, Rs)$

4 Evaluation

We evaluated our method using a prototypical implementation. The main steps shown in Fig. 2 were implemented as follows. For filtering and partitioning of the event log we used Disco¹ process mining tool. To encode the event log into a transaction table we implemented a script in R². For the mining step we used apriori from *arules* package³. And for the post analysis we implemented a script in R.

We conducted three exploratory experiments to explore the usefulness of EL-RM. Section 4.1 illustrates the experiments setup over the three datasets. Then, Sect. 4.2 elaborates about the experiments finding. And finally, we discuss the usefulness of EL-RM in the lights of the experiments findings in Sect. 4.3.

4.1 Experiment Setup

We conducted three exploratory experiments with different analysis objectives to explore the usefulness of our method and the effectiveness of the association rules in understanding the relationship between the multi-perspectives observed in the event log. Table 1 summarizes the quantitative information about the three real datasets and the targeted analysis we used on each log. In all three experiment, we considered a confidence threshold of 90% and a support threshold of 2%. We used low support because of variant in the process execution behavior that leads to low number of occurrences of a sequence of events (*antecedent*) in the transaction table.

¹ <https://www.fluxicon.com/disco/>.

² <https://github.com/DinaBayomie/EL-RM>.

³ <https://www.rdocumentation.org/packages/arules/versions/1.6-8>.

Table 1. Summary of the datasets and the objective of the analysis

Experiment	Dataset	#Cases	#Attributes	#Activities	Analysis objective
1	BPIC-2017	31509	16	26	Pattern analysis
2	BPIC-2020 (prepaid travel)	2099	21	21	Process drift analysis
3	Road traffic fine	150370	13	11	Variant analysis

First Experiment. We used the BPIC-2017 dataset⁴, which contains the events of the loan application process of a Dutch financial institute. The events are generated from three sub-processes, i.e., application, offer and workflow. The log contains cases that started at the beginning of 2016 until the 1st of February 2017. The objective of the analysis in this experiment is exploring the association patterns within the log to understand the data object behavior over execution of the three sub-processes within the log. We follow EL-RM as in Fig. 2. For the preparation step, we did not perform any filtering or partitioning and used the entire event log to get prior insights about the whole log. For the encoding of the event log as a transaction table, we used thirteen attributes as decision attributes, i.e., all the available attributes except the attributes case id, activity and timestamp, in order to explore the different possible patterns generated by the combinations between the events. For the post analysis operations, we first ranked the rules based on the lift measure to show the most interesting rules and then we combined the rules using antecedent aggregation.

Second Experiment. We used BPIC-2020 (prepaid travel) dataset⁵, which contains the events of the prepaid travel request process at Eindhoven University of Technology (TU/e). The log covers the cases from the beginning of 2017 till the 21st of February 2019. The objective of the analysis in this experiment is exploring the process evolution to understand how the process behavior change over time (from 2017 to 2018). We follow EL-RM as in Fig. 2. For the preparation step, we partitioned the log into two partitions based on the time window of 1 year. Thus, the first partition covers cases that started in 2017, and the second partition covers cases that started in 2018. For the encoding step, we used eighteen attributes as decision attributes, i.e., all the available attributes except the attributes case id, activity and timestamp, in order to explore the different possible patterns generated by the combinations between the events. For the post analysis step we compared the rules to identify the uniqueness patterns over the two partitions.

Third Experiment. We used the road traffic fine management process dataset⁶, which contains the events of the road traffic fines process. The cases have a diverse cycle time duration behavior. The shortest cycle time is counted in days (less than a month), while the longest cycle time is 11.8 years. The objective

⁴ <https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f>.

⁵ <https://doi.org/10.4121/uuid:5d2fe5e1-f91f-4a3b-ad9b-9e4126870165>.

⁶ <https://doi.org/10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5>.

of the analysis in this experiment is exploring the process variants between the shortest cycle time cases and the longest cycle time cases to understand how the process behavior change. We follow EL-RM as in Fig. 2. For the preparation step, we partition the log based on the cycle time and filter the shortest and longest cycle time partitions. Thus, the first partition covers cases with short cycle time (max 1 month cycle time), and the second partition covers cases that longest cycle time (more than 3 years cycle time). For the encoding step, we used ten attributes as decision attributes, i.e., all the available attributes except the attributes case id, activity and timestamp, in order to explore the different possible patterns generated by the combinations between the events. For the post analysis step we compared the rules to identify the common and uniqueness patterns over the two partitions.

4.2 Findings

Table 2 summarizes the quantitative findings of our experiments through different steps of EL-RM method. For each experiment, we show information about the encoded transaction table from the entire log (as in experiment one) or the partitions (as in experiments two and three). First, we show the number of encoded transactions that reflects the number of pairs of events that were used for exploring the patterns and the number of the atomic and complex attributes, which represents the number of columns within the transaction table. Second, we show information about the discovered rules. The resulting number of rules after the combination is show in table. We assumed that the analyst wanted to investigate patterns from the perspective of the control flow, and therefore the rules were combined using the antecedent perspective. Finally, the range of confidence and lift measures over the rules are shown.

First Experiment. Figure 7a shows a summary of the distribution of the rules discovered. We carried out a pattern analysis by exploring the discovered rules to understand the relationship between the events and the data objects behavior. Also, it uses the data objects to understand the execution behavior of the events with the three sub-processes and how the three sub-processes interact. For example we found the following three rules:

Table 2. Quantitative summary of the three experiments

Datasets	# Transactions		Attributes		# Discovered rules		# Combined rules		Confidence		Lift	
			Atomic	Complex					Min	Max	Min	Max
BPIC-2017	4101108		16	33	751		15		0.94	1	0.95	13.63
BPIC-2020	30004	223410	18	39	204	306	12	18	0.99	1	1	8.90
Road traffic fine	173102	306925	13	66	78	429	4	16	0.99	1	1	31.07

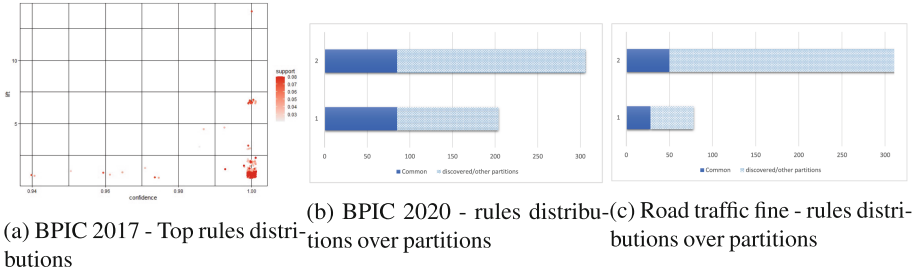


Fig. 7. Summary of findings for the three experiments

R_1 : IF $e_i.Act = "O_Created"$ and $e_j.Act = "O_Sent(mail\ and\ online)"$
 THEN $e_i.Offer\ ID = e_j.Offer\ ID$
 R_2 : IF $e_i.Act = "O_CreateOffer"$ and $e_j.Act = "O_Created"$
 THEN $e_i.EventID = e_j.OfferID$
 R_3 : IF $e_i.Act = "A_Complete"$ and $e_j.Act = "W_Validateapplication"$
 THEN $e_i.Resource = e_j.Resource$

R_1 and R_2 help in explaining the correlating between the events that were executed by offer sub-process based on the data objects perspective. R_1 helps in understanding the correlation between 14% of the events within BPIC17 just that they should have the same offer ids, where there are 6872 offer ids. R_2 emphasis the correlation relation between the ‘Event ID’ and ‘Offer ID’ data objects attributes. For the interaction between application sub-process and workflow sub-process, we found R_3 that shows the resource behavior between the two sub-processes.

Second Experiment. Figure 7b shows the distribution of the association rules over the two partitions. 20% of the rules are common between the two partitions. However, 72% of the rules discovered from the second partition, i.e., cases in 2018, differ from the first partition, i.e., in 2017. For instance, the following two rules were found.

R_1 : IF $e_i.Act = "Permit\ approved\ by\ administration"$ and $e_j.Act = "Permit\ approved\ by\ budget\ owner"$
 THEN $e_i.Resource = e_j.Resource$ and $e_i.org : role! = e_j.org : role$
 R_2 : IF $e_i.Act = "Permit\ approved\ by\ supervisor"$ and $e_j.Act = "Permit\ final.approved\ by\ director"$
 THEN $e_i.(case)_{org}organizationalEntity = e_j.(case)_{P}ermit_{org}organizationalEntity$

R_1 applies for 24% of the cases that occurred in 2018, while it does not apply for the cases executed in 2017, while R_2 applies in cases executed in 2017 but does not apply for cases in 2018.

Third Experiment. Figure 7(c) shows the distribution of the association rules over the two variants, 6% of the rules are common between the two partitions.

We compared the rules to understand the different behavior between the longest and shortest cycle time cases. For example, the following three rules were found.

```

R1 : IF  $e_i.Act = \text{"Insert Fine Notification"}$  and  $e_j.Act = \text{"Insert Date Appeal to Prefecture"}$ 
THEN  $e_i.NotificationType \neq e_j.NotificationType$ 
R2 : IF  $e_i.Act = \text{"Create Fine"}$  and  $e_j.Act = \text{"Payment"}$ 
THEN  $e_i.NotificationType = e_j.NotificationType$ 
R3 : IF  $e_i.Act = \text{"Create Fine"}$  and  $e_j.Act = \text{"Send Fine"}$ 
THEN  $e_i.Resource \neq e_j.Resource$ 

```

Cases with long cycle time used different type of notification as in R_1 . While, in the short cycle time cases, they tend to use the same notification type as in R_2 . Also, the longest cycle time cases used different resources for creating the fine and sending it as in R_3 , unlike in the cases with shortest cycle time they used the same resource.

4.3 Discussion

The three exploratory experiments showed that EL-RM is a useful method that provides insights to understand the association relations between the events and data objects behavior perspectives. EL-RM explores the correlation relation between the data object attributes through attributes complex view and highlights these attributes behavior over the control-flow perspective for the analyst. Moreover, our exploratory experiments showed that the association rules support pattern analysis, process drift analysis and variant analysis.

Our work contributes to research on process mining, as it shows how association rules can represent multi-perspective patterns over the event log. Our work also has a potential impact on industry by providing a new tool that supports the process analysts in their decision making process.

5 Conclusion

In this paper, we proposed a multi-perspective mining technique for the discovery of data connection. Our method uses association rules to represent the relation between the control-flow perspective and its impacts on the behavior of the data objects perspective. Our method has a pre-processing step that allows the analysts to prepare the data for their analysis by applying several techniques such as filtering or partitioning techniques. Moreover, our method has a post-processing step that allows the analysts to improve the usages, interpretation and visualization of the association rules such as combining, comparing and ranking the rules. The results of our evaluation showed the potential of the approach to extract relevant insights about the change behavior of the attributes over the events.

As future work, we will investigate methods to discover further correlation patterns to improve the interest of the rules and to measure their interestingness.

References

1. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: *Fundamentals of Business Process Management*, 2nd edn. Springer, Heidelberg (2018). <https://doi.org/10.1007/978-3-662-56509-4>
2. van der Aalst, W.: *Process Mining - Data Science in Action*, 2nd edn. Springer, Heidelberg (2016). <https://doi.org/10.1007/978-3-662-49851-4>
3. Bose, R.P.J.C., Maggi, F.M., van der Aalst, W.M.P.: Enhancing declare maps based on event correlations. In: Daniel, F., Wang, J., Weber, B. (eds.) *BPM 2013*. LNCS, vol. 8094, pp. 97–112. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_9
4. Pini, A., Brown, R., Wynn, M.T.: Process visualization techniques for multi-perspective process comparisons. In: Bae, J., Suriadi, S., Wen, L. (eds.) *AP-BPM 2015*. LNBIP, vol. 219, pp. 183–197. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19509-4_14
5. Jablonski, S., Röglinger, M., Schönig, S., Wyrтки, K.M.: Multi-perspective clustering of process execution traces. *Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model.* **14**, 2:1–2:22 (2019). <https://doi.org/10.18417/emisa.14.2>
6. Böhmer, K., Rinderle-Ma, S.: Mining association rules for anomaly detection in dynamic process runtime behavior and explaining the root cause to users. *Inf. Syst.* **90**, 101438 (2020)
7. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB*, vol. 1215, pp. 487–499. Citeseer (1994)
8. Dongre, J., Prajapati, G.L., Tokekar, S.V.: The role of apriori algorithm for finding the association rules in data mining. In: *International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT) 2014*, pp. 657–660 (2014)
9. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: *SIGMOD Conference*, pp. 207–216. ACM Press (1993)
10. Hornik, K., Grün, B., Hahsler, M.: arules—a computational environment for mining association rules and frequent item sets. *J. Stat. Softw.* **14**(15), 1–25 (2005)
11. Diba, K., Batoulis, K., Weidlich, M., Weske, M.: Extraction, correlation, and abstraction of event data for process mining. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **10**(3), e1346 (2020)
12. Bayomie, D., Di Ciccio, C., La Rosa, M., Mendling, J.: A probabilistic approach to event-case correlation for process mining. In: Laender, A.H.F., Pernici, B., Lim, E.-P., de Oliveira, J.P.M. (eds.) *ER 2019*. LNCS, vol. 11788, pp. 136–152. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33223-5_12
13. Li, G., de Carvalho, R.M., van der Aalst, W.M.P.: Configurable event correlation for process discovery from object-centric event data. In: *ICWS*, pp. 203–210. IEEE (2018)
14. Bala, S., Mendling, J., Schimak, M., Queteschner, P.: Case and activity identification for mining process models from middleware. In: Buchmann, R.A., Karagiannis, D., Kirikova, M. (eds.) *PoEM 2018*. LNBIP, vol. 335, pp. 86–102. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02302-7_6

15. Pourmirza, S., Dijkman, R.M., Grefen, P.: Correlation miner: mining business process models and event correlations without case identifiers. *Int. J. Cooperative Inf. Syst.* 26(2), 1742002:1–1742002:32 (2017)
16. Senderovich, A., Rogge-Solti, A., Gal, A., Mendling, J., Mandelbaum, A.: The ROAD from sensor data to process instances via interaction mining. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) *CAiSE 2016*. LNCS, vol. 9694, pp. 257–273. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39696-5_16
17. Han, J., Kamber, M., Pei, J.: *Data Mining: Concepts and Techniques*, 3rd edn. Morgan Kaufmann, Burlington (2011)
18. Wynn, M.T., Sadiq, S.: Responsible process mining - a data quality perspective. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) *BPM 2019*. LNCS, vol. 11675, pp. 10–15. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_2
19. Vidgof, M., Djurica, D., Bala, S., Mendling, J.: Interactive log-delta analysis using multi-range filtering. *Softw. Syst. Model.* 1–22 (2021). <https://doi.org/10.1007/s10270-021-00902-0>
20. de Leoni, M., van der Aalst, W.M.P., Dees, M.: A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. *Inf. Syst.* **56**, 235–257 (2016)