Multi-perspective Analysis of Approval Processes based on Multiple Event Logs

Dorina Bano[1], Maximilian Völker[1], Simon Remy[1], Henrik Leopold[1,2], and Mathias Weske[1]

[1] Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
dorina.bano|simon.remy|maximilian.voelker|mathias.weske@hpi.de
[2] Kühne Logistics University, Hamburg, Germany
henrik.leopold@the-klu.org

Abstract. Over the years, the Business Process Intelligence (BPI) Challenge has become integral part of the activities of the process mining community. In this paper, we report on our results for the BPI Challenge 2020. This year's challenge contains data about the process related to travel expense claims. A notable characteristic of this data set is that it consists of five different event logs. To successfully analyze this data set, we used a variety of techniques and tools including Python together with the PM4Py framework, SQL Server, and the process mining tools Disco and ProM. The analysis we conducted includes all three main process mining perspectives, i.e., discovery, conformance, and performance. Among others, we identified concept drift, performance variations during holiday seasons, and potential conformance issues related to double payments and reimbursements without a permit.

Keywords: Business Process Intelligence, Process Mining, Event Log, Concept Drift, Business Process Management, Process Discovery

1 Introduction

Over the years, the Business Process Intelligence (BPI) Challenge has become integral part of the activities of the process mining community. This year's challenge is concerned with a data set about travel expense claims from the TU/e located in the Netherlands [5]. A notable characteristic of this data set is that it consists of five different event logs. The proper identification and consideration of the links among these event logs is, therefore, one of the key obstacles for a comprehensive analysis.

To successfully analyze this year's data set, we used a variety of techniques and tools. Most importantly, we used Python together with the PM4Py framework [2] for developing scrips, SQL Server for storing and analyzing event log data, and the process mining tools Disco [7] and ProM [6] for analyzing the processes captured in the event logs. Our analysis includes all three main process mining perspectives, i.e., discovery, conformance, and performance [1]. Our key findings are as follows. First, we detected a *concept drift* from 2017 to 2018, which is reflected in a threefold increase in the number of cases and events. A possible explanation for this concept drift might be a digitization initiative that resulted in a higher availability of data on the processed travel expense claims. Second, we detected that several performance related aspects, such as mean case duration and waiting time, vary throughout the year. Our analysis indicates that this might be explained by a lower availability of staff during typical holiday seasons in summer and winter. Third, we identified several potential conformance issues including double payments and reimbursements without a permit. While those might be also explained by data quality issues, they represent an important point for further investigation.

The remainder of this paper elaborates on the details of our analysis and our results. Section 2 reports on high-level insights, statistics, and the pre-processing steps we performed. Section 3 presents the results related to process discovery including insights on batch processing and the activity name pattern. Section 4 discusses the performance perspective and, among others, findings about concept drift and seasonality. Section 5 reflects on the general operational perspective and is concerned with aspects such as rerouting and resubmissions. Section 6 finally presents the results of the conformance analysis before Section 7 concludes the paper. In this project, we did not only rely on standard process mining software but also implemented our own functionality to further analyze the event logs. This code is publicly available on a Git repository[3].

2 Data and Log Understanding

In this section, we present high-level insights about the event logs, report on data quality issues, and the pre-processing steps we performed to deal with them. Moreover, a data model that visualizes the shared attributes of the different event logs is presented.

2.1 Statistical Insights

This year's challenge comprises five event logs. Based on the number of events, Table 1 shows that the `Permit` log is the largest in this collection, containing about 81,000 events, followed by the `InternationalDeclarations` log with about 70,700 events. With around 10,000 cases, the `DomesticDeclarations` log contains the highest number of process instances. A clear difference between the event logs is the number of unique traces, i.e., trace variants. While the `RequestForPayment` log only contains 77 different trace variants (out of 6,760 cases), the `Permit` log contains 1,401 different trace variants (out of 6,426 cases). Also, the number of activities differs across the logs. This mostly results from using different abstraction levels. In some logs, the role is used to further differentiate a particular activity, which is then reflected in the activity label accordingly (see

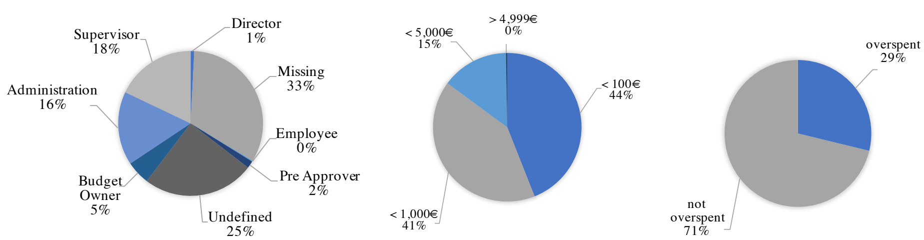
[3] https://github.com/bptlab/bpi-challenge-2020

Table 1: Overview of the five event logs, including the number of events and cases, after the pre-processing. The number of attributes does not include activity name, timestamp, and case ID.

	Requests for Payment	Domestic Declarations	Prepaid Travel Cost	International Declarations	Travel Permits
Events	36,073	55,629	17,512	70,732	81,092
Cases	6,760	10,357	2,007	6,323	6,426
Case variants	77	89	189	727	1,401
Activities	16	14	29	33	49
Min. trace length	1	1	1	3	3
Max. trace length	20	24	21	27	81
Mean trace length	5	5	9	11	13
Num. of attributes	12	8	20	19	163

Section 3.3). Therefore, a higher number of activities allows to better understand the interactions among the process stakeholders. In three out of five logs, the shortest cases only contain a single event, while the others include at least three events per case. Due to the low complexity of the underlying process, the mean trace length ranges from 5 to 13 events and is, therefore, relatively small.

Concerning the number of attributes, the `Permit` log differs from the other event logs. With 163 attributes, this log has about eight times more attributes than the log with the second highest number. However, the actual information richness is limited since for only 24 of the attributes (i.e., about 14%), there is a value for at least 50% of the events.



(a) Ratio of executed activities by roles

(b) Ratio of requested amounts per declaration

(c) Ratio of overspent declarations

Fig. 1: Proportion of three attributes across the event logs.

Figure 1a provides further insights into the event logs. Measured by the number of events per role, *Employee* executed most of the activities (32.8%), followed by *Supervisor* (17.9%). However, this information is missing for 24.9% of the events. As depicted in Fig. 1b, 44% of the requested reimbursements are below 100 €. In about 41% of the cases, applicants requested not more then 1,000 €, across all event logs. Nonetheless, in less than 30% of all cases, the initially requested budget was overspent, as shown in Fig. 1c.

2.2 Log Pre-processing

Data quality issues can have a tremendous impact on process model discovery. Therefore, they have to be analyzed and addressed prior to any discovery activities in order to obtain a model that is conducive for decision making. We have identified the following data quality issues and performed a number of pre-processing steps in the provided event logs:

Duplicate attribute names: The `InternationalDeclarations` log contains two attributes that only differ with respect to capitalization: *Permit ID* and *Permit id*. After realizing that the corresponding attribute values are different, we decided to rename the attribute *Permit id* to *Permit id2*. After solving this issue, we were able to import the event log to a SQL Server application, which applies unique attribute name constraints.

Identical attribute values: In the `InternationalDeclarations` log, there are four attributes having the same value for all events: *AdjustedAmount*, *Amount*, *OriginalAmount*, and *RequestedAmount*. To simplify the event log and remove redundancy, we decided to discard three of them and keep only the *Amount* attribute.

Incomplete cases: We are treating a case as complete if 1) all its events fall into the time frame between the 9th of January 2017 and 9th of January 2019 and 2) if there are at least 10 cases in the event log that end with the last event of the considered trace. We use the first condition to make sure that we can compare cases across event logs. We observed that, for instance in the `InternationalDeclarations` log, there are cases with timestamps up to May 2020 (although the event log was published in March 2020). Therefore, we discard cases that do not fall into the defined time frame. We use the second condition to filter cases with infrequent end events. For example, if we see in the entire event log that only one case ends with *Request for Payment REJECTED by Supervisor* event than we exclude this case from our analyzes. The rationale is that such cases have probably occurred due to errors or other rare events and would only distort our analysis.

One-time activities: To simplify the discovered process model, we only consider events that occur more than once in the event log. Whether such events result in deleting the event only or the entire case depends on the timestamp and the event class of the last event in the respective case. If the event class of the last event is not *Payment Handled* and the timestamp is end 2017 (the time the concept drift occurs) or 2018 (when the database is dumped), we delete the complete case. If the event class of the last event is *Payment Handled*, we only delete the event.

In Table 2, we list the events that occur only once in the event logs together with the corresponding event log and case ID. The last column shows the decision taken for each event.

Table 2: Unique activities collected from all event logs and the decision made upon each activity (case/event deletion).

Log	Activity name	Case ID	Operation
`RFP`	*RFP FINAL APPROVED by BUDGET OWNER*	181880	Delete case
`RFP`	*RFP FOR_APPROVAL by SUPERVISOR*	183392	Delete case
`RFP`	*RFP FOR_APPROVAL by ADMINISTRATION*	159579	Delete event
`ID`	*Permit Rejected by DIRECTOR*	63785	Delete event
`DD`	*Declaration FOR_APPROVAL by ADMINISTRATION*	108210	Delete event
`DD`	*Declaration FOR_APPROVAL by PRE APPROVER*	96530	Delete case
`DD`	*Declaration FOR_APPROVAL by SUPERVISOR*	89887	Delete case
`PL`	*Permit FOR APPROVAL by ADMINISTRATION*	43924	Delete case
`PL`	*Permit FOR APPROVAL by SUPERVISOR*	13082	Delete case

2.3 Data Model

The data model in Fig. 2 shows a selection of attributes for each of the five logs, along with the occurring attribute values and links among the logs.

From this data perspective, there are three main types of logs sharing a similar set of attributes. First, the `DomesticDeclarations` and `InternationalDeclarations` logs are very much alike, except for the identifiers, which are log specific (e.g., *Permit id2*). Additionally, their *ID* values do not overlap. The same applies to the `RequestForPayment` and `PrepaidTravelCost` logs with `PrepaidTravelCost` log having permit related attributes, and their unique *Rfp_id* values. The complex and extensive `Permit` log constitutes the third type.

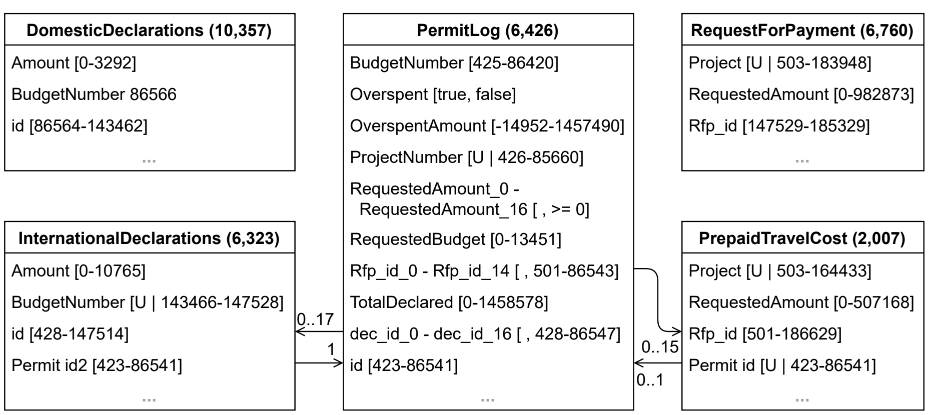


Fig. 2: Simplified data model of the data set, depicting the relations between the event logs.

In terms of relations among the logs, `DomesticDeclarations` and `RequestForPayment` logs are not connected to any other log, which corresponds to the given description that for "*domestic trips, no prior permission is needed*" [8] and requests for payment should not be associated with travels [8].

The remaining logs, however, are connected, with the `Permit` log being the central entity. Each travel permit in the `Permit` log can be associated with several international declarations via the *dec_id_** attributes in the `Permit` log (up to 17 declarations for one permit were observed in the data). Analogous, a travel permit can refer to multiple requests for reimbursing prepaid travel costs recorded in the `RequestForPayment` log via the *Rfp_id_** attributes (in the data, 15 was found to be the maximum number of requests for one travel permit).

2.4 Time wrapping

The processes of this year's challenge are of administrative nature. Therefore, we cannot assume shift working, which would result in a 24 hours working day, seven days a week. Having realistic assumptions about the working time has notable impact on the process's performance as it, for example, affects the effective waiting time between two activities. In order to obtain a realistic picture of these aspects in our further analysis, we applied a time wrapper function to the event data. We agreed on the following work schedule for our analysis: Monday - Friday from 8:00 - 18:00. Figure 3 provides an intuition of the approach. It visualizes how the time wrapping affects the waiting time. In the remainder, our calculations will be only based on the (assumed) working time.

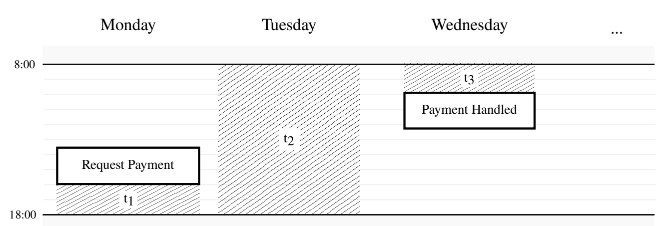


Fig. 3: Due to time wrapping, only the actual working time is considered in further analysis. Therefore, the waiting time (wt) between the two activities is equal to the sum of the three highlighted periods: $wt = \sum_{i=1}^{3} t_i$.

3 Process Models

In this section, we illustrate the described process as well as the discovered models from all events logs as BPMN (Business Process Model and Notation) models [10]. Moreover, we describe the activity pattern and batch processing behavior we identified in the process.

3.1 Described Process Model

To visualize the process as described in the challenge's description [8], we use BPMN (see Fig. 4) as a language of choice. The following behavior applies to all kind of requests, like declarations, request for payments, and permits. The process starts with the request submission performed by the *Employee*. Next, the *Administration* has to approve it. In consequence, the approved request can be approved by the *Budget Owner* or the *Supervisor*. The last role that can approve the submitted request is the *Director*. If it is not approved by one of the roles (e.g., *Budget Owner* or the *Supervisor*), the *Employee* can either resubmit the request or can reject it. Otherwise, the *Employee* can request the payment, which eventually results in handled payment.

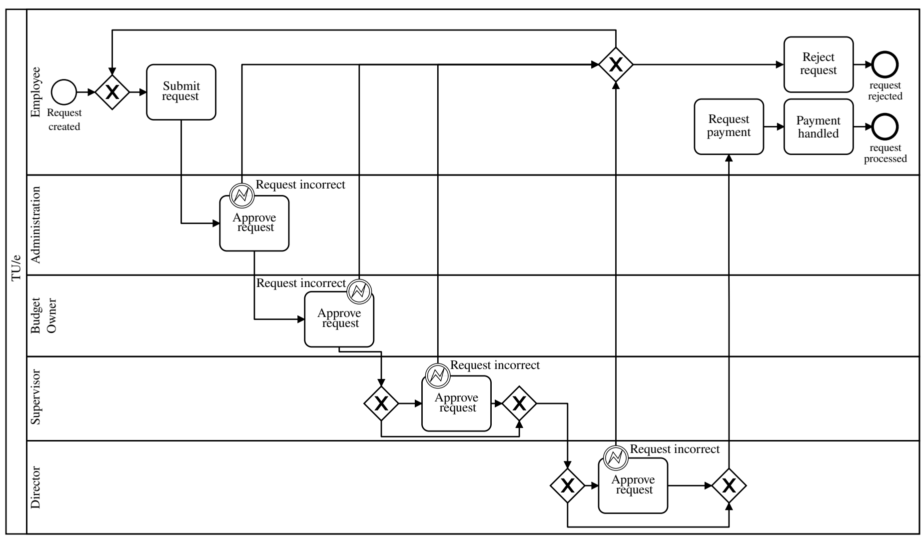


Fig. 4: Described business process model represented as BPMN.

3.2 Discovered Process Model

To obtain a clear picture of the processes captured in the event logs, we simplified the event logs. More specifically, we focus on the operations (e.g., *Request for Payment*, *Permit*, and *Declaration*) and the activities that are executed by the system (*Request Payment*, *Payment Handled*, *Send Reminder*) except the *Start Trip*, and *End Trip* activities. For all activities that match the *[Operation + State + Role]* pattern, we remove the state (e.g., *submitted* or *approved*) as well as the role (e.g., or *Supervisor*) from the activity name. For example, we transform the activity *Request for payment SUBMITTED by EMPLOYEE* into *Request for Payment*. We performed a respective SQL query to implement these

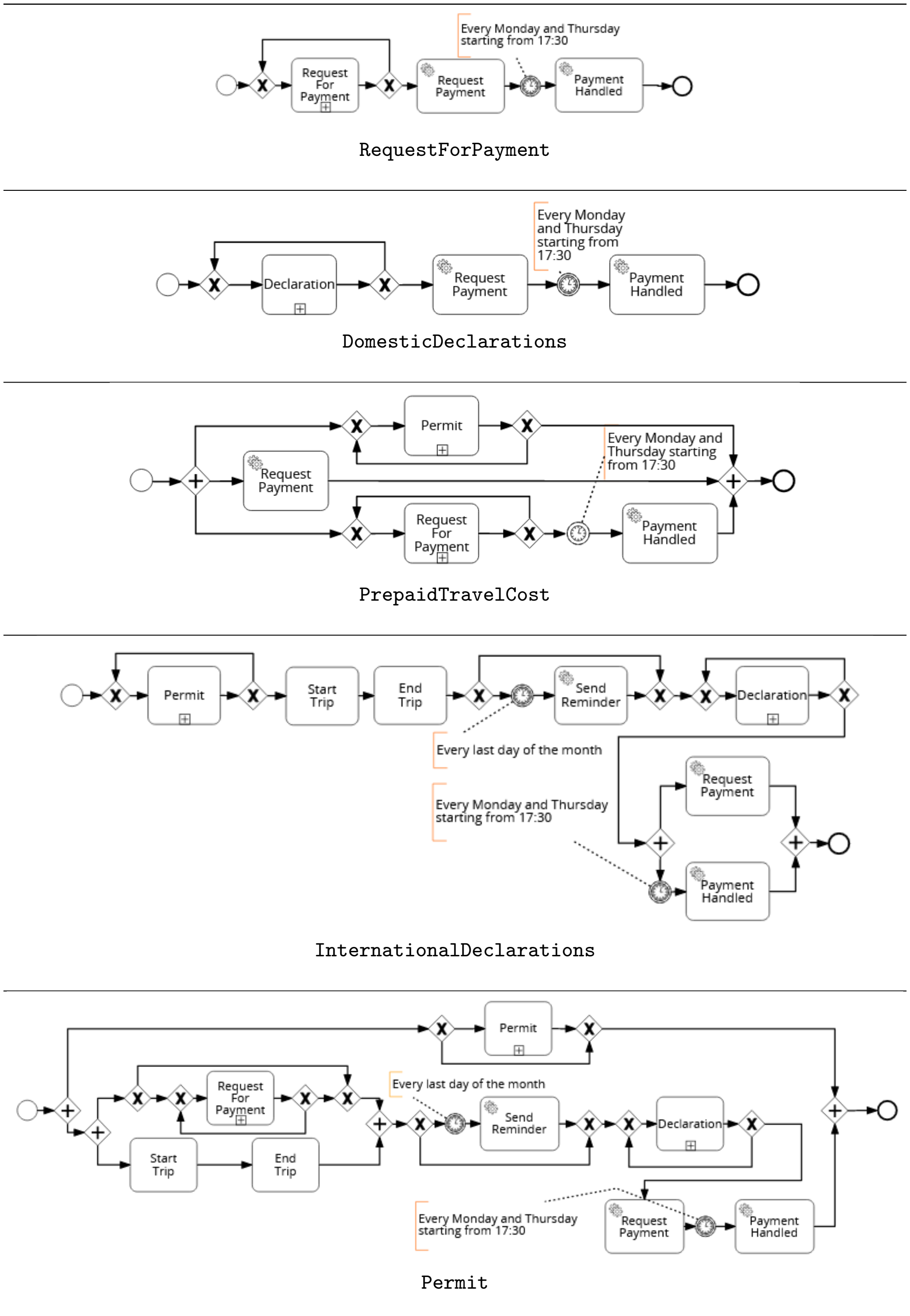


Fig. 5: Discovered BPMN models after applying the Inductive Visual Miner.

changes in the event log. To discover the business processes, we applied the Inductive Visual Miner [9] (as a ProM plug-in [6]) on the event logs. We set the filter configuration to 100% for activities and to 80% for paths. The discovered BPMN models are shown in Fig. 5. Below, we give a detailed explanation of each model:

- `RequestForPayment` log: This log contains events regarding the request for payment and starts when the employee submits such a request for payment. Once the request is submitted, different roles (e.g., *Administration*, *Supervisor*) can change the status of the request by following a similar process to the one illustrated in Fig. 4. To abstract from the details of this status change procedure (presented in Fig. 4), we represent it as a sub process in the model. After the request for payment has been processed the employee requests the payment (activity *Request Payment*). The activity *Payment Handled* marks the end the process.
- `DomesticDeclarations` log: This log contains events regarding domestic declarations. Following the same logic as the `RequestForPayment` log, declarations can run through different states from different roles. Therefore, it is represented as a sub process as well. Afterwards, we observe the same activities as in the `RequestForPayment` log. The process is finished, once the payment has been handled.
- `PrepaidTravelCost` log: The process discovered from the `PrepaidTravelCost` log slightly differs from the two previously considered processes. Most importantly, it contains the activity *Permit*, which is required for completing the process.
- `InternationalDeclarations` log: This log contains events regarding the permit and the declaration. The process starts with the *Permit* activity, which can go through different states performed by different roles. The following *Start Trip* and *End Trip* activities represent the beginning and the end of the respective trip of the employee. Once the trip has ended, the employee has to submit the declaration. If this is not the case, a reminder is sent by the system (*Send Reminder*) every last day of the month. After the declaration has been submitted, the payment is requested (*Request Payment*) and handled (*Handle Payment*).
- `Permit` log: This log contains activities from all previous discovered models such as: *Request for Payment*, *Permit*, and *Declaration*. Moreover, it also contains the activities *Start Trip*, *End Trip*, *Request Payment*, and *Payment Handled* from the `InternationalDeclarations` log.

3.3 Activity Name Pattern

To better understand the results presented in this paper (and the discovered business processes in Fig. 5) it is important to reflect on how the activity names are defined in the event logs. As Fig. 6 illustrates, we concluded that all activities that are executed by the staff member (represented as resource in the event log) follow the *[Operation + State + Role]* pattern except for the *Start Trip*

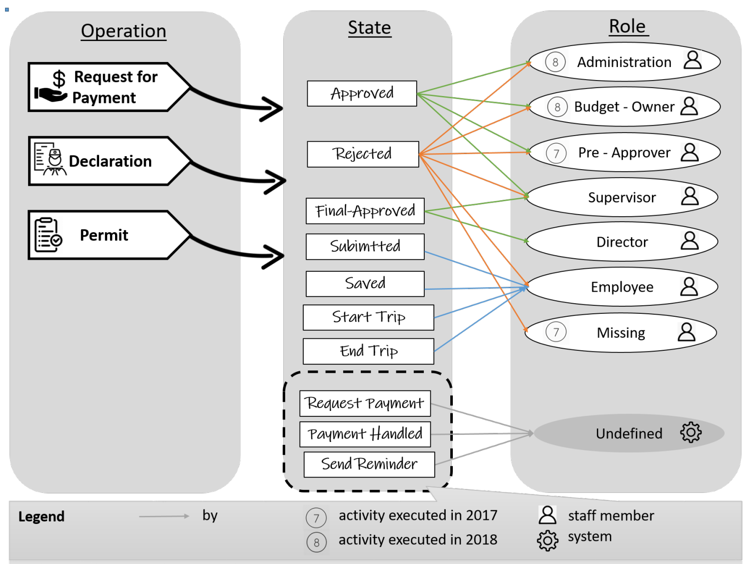


Fig. 6: Visualization of the activity name pattern, present in all event logs.

and *End Trip* activities. For example, the *Employee* can request for a payment *Request for Payment*, declare (*Declare*), or ask for a permit (*Permit*). Each operation can run through different states such as *Approved*, *Rejected*. Each operation state can be executed by different roles. For example, the *Request for Payment* can be submitted by an *Employee* (*Request for Payment SUBMITTED by EMPLOYEE*) or the declaration can be final approved by the *Supervisor* (*Declaration FINAL_APPROVED by SUPERVISOR*). Each role can belong to one of the resources (represented in Fig. 6 with a small icon next to each role): *staff member* or *system* which are not part of the activity name.

Activities executed by the *Undefined* role (located inside the rectangle with the dashed border in Fig. 6) and the *Start Trip* and *End Trip* activities follow a slightly different pattern. They follow the verb-object pattern, where the verb represents an action and the noun a business object. Note that the *Undefined* role (which belongs to the system resource) shown in a dark gray ellipse in Fig. 6 is not part of the activity name.

The BMPN model in Fig. 4 shows which operation and which system task is involved in which event log. There are only a few combinations that can be derived from the introduced activity pattern that are not part of the event log, for example:

– In the `DomesticDeclarations` log, a declaration is never *FINAL_APPROVED* by the *Supervisor* or *Director*. Therefore, we do not have any activity called

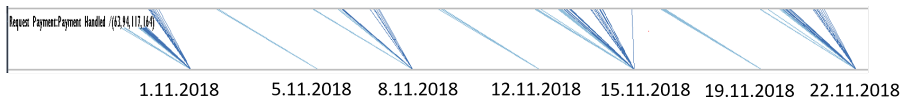


Fig. 7: Batch processing of the *Payment Handled* activity.

  Declaration FINAL_APPROVED by SUPERVISOR or *Declaration FINAL_APPROVED by Director*.
- In the `PrepaidTravelCost` log, we never see an activity where the *Employee* is saving the permission (*Permit SAVED by EMPLOYEE*). The *Employee* just submits the permission.
- In the `Permit` log, we never see the *Request for Payment REJECTED by BUDGET OWNER* activity and *Declaration REJECTED by SUPERVISOR*.

When roles have a number (7 or 8) as a prefix, it means that they perform activities only in 2017 or only in 2018 respectively. For example, an operation (*Request for Payment*, *Permit*, and *Declaration*) is *APPROVED* or *REJECTED* by *Administration* or *Budget Owner* only in 2018 and never in 2017. The opposite holds for an operation *APPROVED* or *REJECTED* by *Pre-Approver*, which is executed only in 2017 and not in 2018. The same holds if an operation is rejected by *Missing*. This is explained by the *drift concept* (see Section 4.2) happening in the end of 2017. The roles that do not have any number as prefix execute activities during both years.

3.4 Batch Processing

In all provided event logs, we have observed that certain activities are executed as a batch across process instances. To identify these activities, we used the *Performance Spectrum Miner* [4] provided as a ProM plug-in. We have represented these activities as *Intermediate Timer Event* in Fig. 5 and screenshots of the performance spectrum algorithm are provided in Fig. 7 and Fig. 8.

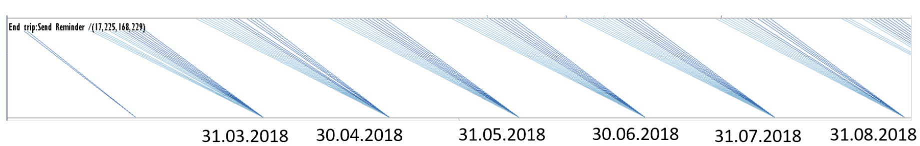


Fig. 8: Batch processing of the *Send Reminder* activity.

The following activities are executed as a batch in all provided event logs:

- *Payment Handled*: The *Payment Handled* activity is processing the requested payments every Monday and Thursday starting from 17:00.

– *Send Reminder*: After the trip has finished (represented by the *End Trip* activity in Fig. 5) the system sends every last day of the month a reminder to the *Employee* to submit his/her declaration regarding the trip. In addition, we noted that several reminders can be sent for the same declaration.

4 Performance Analysis

In this section, we summarize our findings regarding the performance dimension. We analyzed several aspects including concept drift, seasonality, waiting times, bottlenecks, and rejection rates.

4.1 Overview

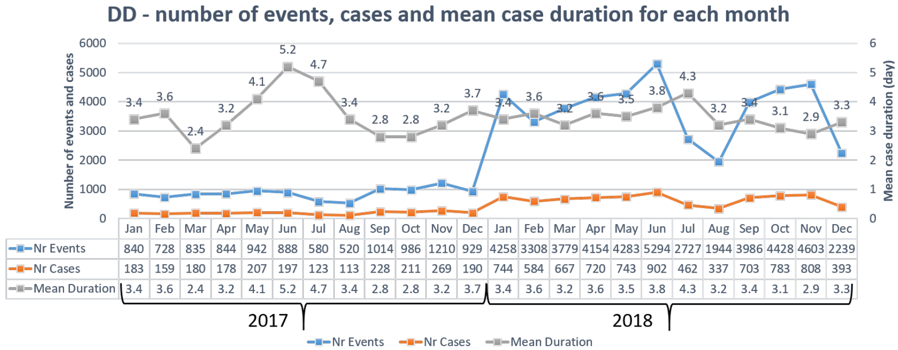
To analyze the performance of the process, we applied the following filter settings:

– *Filter incomplete cases*: We selected all cases where the activity *Declaration SUBMITTED by EMPLOYEE* is eventually followed by the *Payment Handled* activity (i.e., 96% of the cases). This way, we removed incomplete cases.
– *Remove events by attributes*: In the `InternationalDeclarations` log, we excluded all events which are related to the activities *Permit*, *Start Trip* and *End Trip* since they are happening before a declaration takes place (see BPMN model presented in Fig. 5). This is necessary because we are only interested in the cases that contain events related to declarations and payments.
– *Started in time frame*: We selected all cases that started in a specific month but not necessary have finished with the *Payment Handled* activity within the same month.
– *Time wrap*: We applied time warping based on working days (from Monday to Friday, 08:00-18:00 o'clock) as explained in Section 2.4.

Figure 9 illustrates the throughput of the `DomesticDeclarations` and `InternationalDeclarations` logs regarding the number of cases, events, and mean case duration (in terms of days) for each month. In the following subsections, we take a closer look at these results.

4.2 Concept Drift

A brief look at Fig. 9b and Fig. 9a reveals that there is a considerable difference between the years 2017 and 2018. Most notably, the number of cases and events in 2018 is roughly three times bigger compared to the number of cases and events in 2017. Interestingly, the mean case duration is not affected by this increase in the number of cases and events. While we can only speculate about the exact reasons behind this phenomenon, it is clear that we observe a so-called *concept drift*, i.e., a significant change in the process [3]. We can imagine that regulations



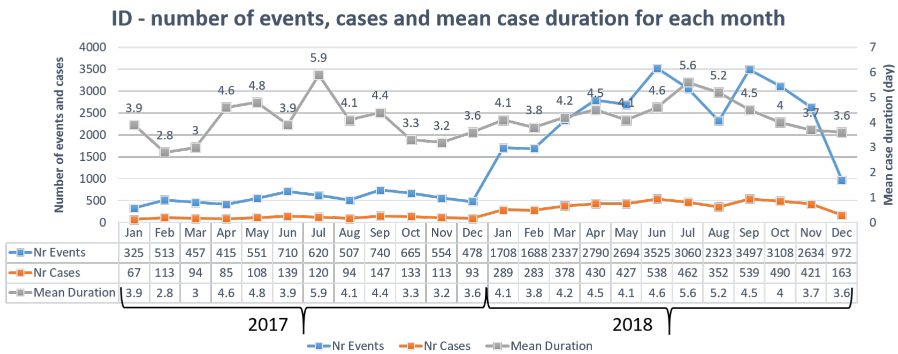
2017

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Nr Events	840	728	835	844	942	888	580	520	1014	986	1210	929
Nr Cases	183	159	180	178	207	197	123	113	228	211	269	190
Mean Duration	3.4	3.6	2.4	3.2	4.1	5.2	4.7	3.4	2.8	2.8	3.2	3.7

2018

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Nr Events	4258	3308	3779	4154	4283	5294	2727	1944	3986	4428	4603	2239
Nr Cases	744	584	667	720	743	902	462	337	703	783	808	393
Mean Duration	3.4	3.6	3.2	3.6	3.5	3.8	4.3	3.2	3.4	3.1	2.9	3.3

(a) `DomesticDeclarations` log



2017

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Nr Events	325	513	457	415	551	710	620	507	740	665	554	478
Nr Cases	67	113	94	85	108	139	120	94	147	133	113	93
Mean Duration	3.9	2.8	3	4.6	4.8	3.9	5.9	4.1	4.4	3.3	3.2	3.6

2018

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Nr Events	1708	1688	2337	2790	2694	3525	3060	2323	3497	3108	2634	972
Nr Cases	289	283	378	430	427	538	462	352	539	490	421	163
Mean Duration	4.1	3.8	4.2	4.5	4.1	4.6	5.6	5.2	4.5	4	3.7	3.6

(b) `InternationalDeclarations` log

Fig. 9: Number of events, cases and mean case duration (in days) per month for 9 Jan '17 - 9 Jan '19.

were changed from 2017 to 2018 such that all cases had to be filed digitally from thereon. Obviously, any kind of paper-based declaration is not visible because these process instances did not leave digital traces. A full digitization from 2017 to 2018 could also explain the stable mean case duration. The personnel was already available, but partially working on paper-based cases. Another possible explanation is that only selected instances were recorded in 2017.

4.3 Seasonality

Besides the concept drift explained above, we detected several seasonal performance changes. In the `DomesticDeclarations` log, we observe that June 2017 has the longest mean time duration (5.2 days) followed by July 2017 (4.3 days). In the `InternationalDeclarations` log, the same holds for July 2017 (5.9 days) followed by July 2018 (5.6 days). The shortest mean case duration in the `DomesticDeclarations` log can be observed in March 2017 (2.4 days). Similarly, in the `InternationalDeclarations` log, February 2017 is the month with the shortest mean case duration (2.8 days), followed by March (3 days). From this data, we conclude that the mean case duration is mainly affected by holidays. The summer holidays in June and July seem to result in a longer mean case duration, which makes sense considering the more limited availability of staff. The shorter mean case duration in February and March could be explained by full staff availability in these months. While we did not have empirical proof, we do know from our own experience that February and March are not typical holiday months in the Netherlands.

Another aspect that is noteworthy, are the months due to highest and lowest number of events. In the `InternationalDeclarations` log, September (for 2017 and 2018) is the month with the highest number of events. In the `DomesticDeclarations` log, November is the month with the highest number of events (for 2017 and 2018). In the `DomesticDeclarations` log, August (for 2017 and 2018) is the month with the lowest number of events. In the `InternationalDeclarations` log, January 2017 and December 2018 are the months with the lowest number of events. This again might be explained with holidays (summer and Christmas). Since holidays might lead to a lack of resources, fewer events (such as status changes) can be observed.

4.4 Waiting Times

So far we have focused on the `InternationalDeclarations` and `DomesticDeclarations` event logs. However, for the analysis of waiting time, it is interesting to conduct a comparison among all event logs. Again we distinguish in our analyzes between 2017 and 2018. Table 3 shows the mean waiting times in hours between the individual approval steps for declarations (`DomesticDeclarations` (`DD`) and `InternationalDeclarations` (`ID`)), requests for payments (`RequestForPayment` (`RFP`) and `PrepaidTravelCost` (`PTC`)) as well as permits (`Permit` log (`PL`)). Note that the times are adjusted to the assumed working times and days (cf. Section 2.4).

Table 3: Average waiting times between approval steps in hours for cases completed in 2017 and 2018, adjusted for working times.

		`DD`		`ID`		`RFP`		`PTC`		`PL`	
From	To	'17	'18	'17	'18	'17	'18	'17	'18	'17	'18
Submit by EMP	APP by ADM [4] [5]	8.8	7.3	8.3	8	2.7	4.3	1.5	2.7	0.5	2.6
APP by ADM	APP by BO [6]		11		20.1		16.3		13.1		11.1
APP by ADM	F_APP by SUP	12.3	11.1	12.6	18.8	11.9	17.2	13.8	13.3	12.6	11.6
APP by BO	F_APP by SUP		20.4		21.7		21.8		21		22
F_APP by SUP	Request Payment	26.8	21.5	34.2	21.1	27.6	24.1	33.6	21	N/A	N/A
Request Payment	Handle Payment	29.8	28.4	26.8	27.8	28.8	28.5	27.6	27.1	N/A	N/A

The data from Table 3 shows that even though the number of cases significantly increases in 2018 (cf. Figs. 9a and 9b), the average waiting times between the process steps roughly remains the same. Furthermore, the waiting times are similar across the different types of requests, e.g., waiting for the final approval by the *Supervisor* after the *Budget Owner* has approved the request takes nearly as much time as for each other type (20.4 hours to 22 hours). This especially applies to direct-payment related tasks in 2018 since these activities are handled by the system and regularly executed (for more information see Section 3.4).

However, there is a notable difference between the waiting times of the individual approval steps in the `DomesticDeclarations` and `InternationalDeclarations` logs. In the `InternationalDeclarations` log, each non-payment related step involves longer waiting times, starting from a few minutes for the first approval by the *Administration* up to 9 hours for the approval by the *Budget Owner*, which nearly corresponds to a whole working day. Another interesting finding is that as soon as the *Budget Owner* is involved in the approval procedure, the waiting time for the final approval by the *Supervisor* increases. For example, in the `DomesticDeclarations` log, the final approval is usually given 11.1 hours after the *Administration* approved the request. If the *Budget Owner* approved the request too, the waiting time between the *Budget Owner* approval and the final approval increased to 20.4 hours. Similar behavior can be observed in the other logs, especially in the `PrepaidTravelCost` and `Permit` logs.

4.5 Bottleneck Analysis

Besides the throughput of a process or certain stages, another frequently imposed performance metric is the cycle-time, and with that, the question about potential bottlenecks. Parts of a process may require more time to execute compared to others, which delays the processing of the current instance. In the worse case, this may also influence other process instances due to resource limitations. Reasons for bottlenecks can be manifold, like complex and time-consuming tasks, resource limitations, or untrained workers, i.e., systemic constraints of the process.

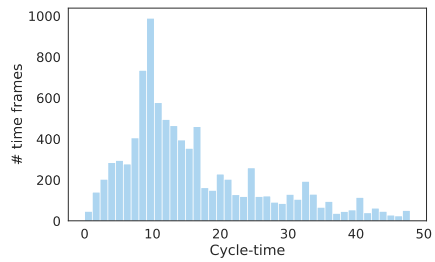
To investigate a single task's processing time, the point in time for its beginning and termination must be available. Unfortunately, this is not the case in the provided event logs, since only the activity's termination event is reported. Because of this, we can only investigate the time between the termination of two activities, which in essence is the sum of the waiting time between the activities and the runtime of the subsequent one.

In the first part of the bottleneck analysis, we focused on the processing of declarations. First, we identified potential bottlenecks with the help of the process mining tool Disco [7]. It is important to note that we only considered

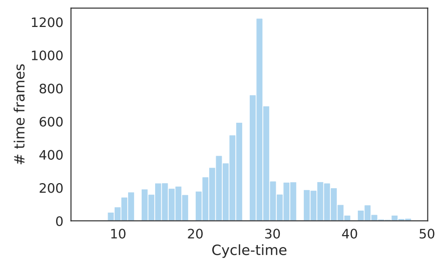
[4] In 2017, there is no *Administration* but a *Pre-Approver* role instead (cf. Section 3.3).

[5] The median values belonging to the reported means indicate that there actually are many very short durations (range of seconds) and therefore also several very long durations.

[6] In 2017, there is no *Budget Owner* role.



(a) Cycle-time of *Declaration FINAL_APPROVED by SUPERVISOR* and *Request Payment* in 91% of the cases.



(b) Cycle-time of *Request Payment* and *Payment Handled* in 95% of the cases.

Fig. 10: Distribution of the cycle-times of two potential bottlenecks for `DomesticDeclarations`.

frequent behavior, since this analysis aims to find systemic bottlenecks. For the following steps, we considered two metrics: the total duration and mean duration. For the latter, we found only one anomaly concerning the events *Declaration FINAL_APPROVED by SUPERVISOR* (*DFAS*) and *Declaration REJECTED by MISSING* (*DRM*) for the `DomesticDeclarations` log. In the mean, the cycle-time of those activities is about 3.1 days. However, cases that are affected by this started only in 2017 and are no longer observed in 2018. We assume this is due to the concept drift in 2018, as explained in Section 4.2. Hence, this issue has no longer an impact on the process's performance.

Regarding the same log and the total duration, which is the sum of all cycle-times of all process instances, we identified two subsequent potential bottlenecks: *DFAS* $\rightarrow$ *Request Payment* (*RP*) and *RP* $\rightarrow$ *Payment Handled* (*PH*). For the first one, we observe a total duration of 25.5 years, while the second one has a total duration of about 32 years. Based on the available data, this might be because of two reasons. First, the total cycle-time is comparably high, since the affected activities are among the most frequent in the log; thus, the single occurrences add up. Second, as described in Fig. 7, the activity *Payment Handled* is batched. Because of this, a slightly higher waiting time has to be expected. However, as shown in Fig. 10, the impact of both potential bottlenecks is very low. While in only 9.35% of all cases, the payment is requested later than 48 hours, the handling of the payment took in only 4.74% of the cases longer than 48 hours.

Concerning international declarations (`InternationalDeclarations` log), we did not identify any potential bottlenecks. Nevertheless, it is only notable that the cycle-time of activities before starting the trip is comparably high, which makes sense since such trips usually require some planning upfront.

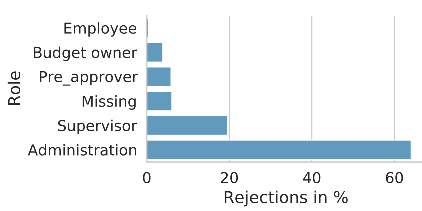
Regarding the processing of travel permits, we observed a similar behavior. After a permit was finally approved by the *Supervisor* (*Director*), it takes on average 13.7 days (9.6 days) until the applicant started the trip. However, we found one potential bottleneck that occurs after the applicant returns from the trip; the average time between the two activities *End Trip* and *Send Reminder* is 20.7 days. As already reported in Section 3.4, the reason for this is the batch processing for the latter activity, which only takes place on the last day of the month.

To summarize the bottleneck analysis, we did not identify any systemic bottlenecks in the process of travel declarations. However, the process might benefit from sending reminders multiple times per month to reduce delays in the processing of travel permits.

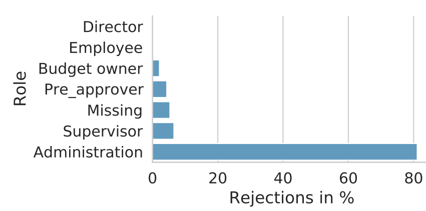
4.6 Rejection Rate

Another aspect we investigated during this challenge was the rejection rate of declarations at various stages in the process. According to the general process, as depicted in Fig. 4, we defined one stage per decision-maker, e.g., *Supervisor*, *Budget Owner*, etc., in the process. It is notable that every time a declaration is rejected at some stage in the process, the *Employee* who submitted the request also has to reject it. This might be due to the circumstance that after a declaration got rejected, the *Employee* can resubmit a reworked version. Because of this, we only considered rejections by employees, if declarations in such cases were not rejected at any other stage in the process. Also, we considered the possibility that declarations could be resubmitted within the same case [8].

Figure 11 depicts the rejection rates at the various stages of the process in the `DomesticDeclarations` and `InternationalDeclarations` logs. In both cases, we found that most requests were rejected by the *Administration* (`DomesticDeclarations` log: 64%, `InternationalDeclarations` log: 81%). We assume that this is due to formal errors or other unspecific reasons. The second most common reason for rejecting declarations are the *Supervisor*. While this is the case for about 20% of the cases for domestic declarations, it happens in less than 10% for international declarations. In general, there are just a few cases



(a) In the `DomesticDeclarations` log about 80% of the declaration got rejected within the first two stages.



(b) In the `InternationalDeclarations` log 81% of the declarations got rejected at the first stage of the process.

Fig. 11: Ratio of rejections per stage in the process. In both logs, the majority of declarations got rejected by the *Administration*, followed by the *Supervisor*.

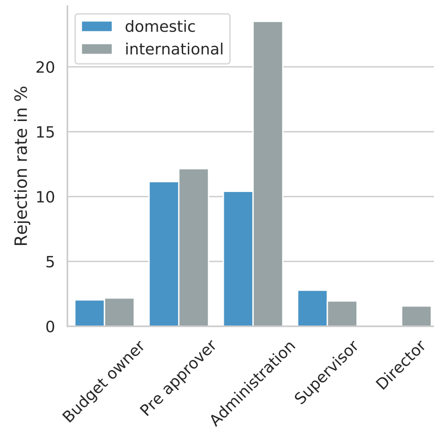


Fig. 12: Rejection rate per role for domestic and international declarations. It is notable, that the role *Pre-Approver* is only present in cases in 2017. Also, there are no domestic declarations processed by the *Director*.

(less than 1%) in both event logs, where *Employee* reject their requests without any other validation. Also, in less than 1% of all cases (e.i., 4 cases) international declarations got rejected by the *Director*. This corresponds to the small number of cases that have to be approved by the *Director* in general. In two cases, the requested amount (5,397.81 € and 4,000.10 €) are way beyond the average of 771.78 € for international declarations. We assume that those cases where extraordinary costs incurred, for example, for business class flights or expensive hotels.

When comparing the number of rejected and accepted declarations per stage, the general rejection rate is relatively low. With 23.5% of the international declarations (10.4% for domestic declarations), the *Administration* has by far the highest rejection rate. Followed by the *Pre-Approver* with 11.2% for domestic and 12.2% for international declarations, for cases in 2017. In all other stages of the process, the rejection rates are below 4% for both declaration types. Lastly, 419 (4.05%) domestic and 21 (0.33%) international declarations were never approved. This means that they were never finally approved at any stage in the process.

5 General Operational Analysis

In this section, we report on general operational aspects of the approval processes, like the rerouting and resubmission of applications and characteristics of international declarations.

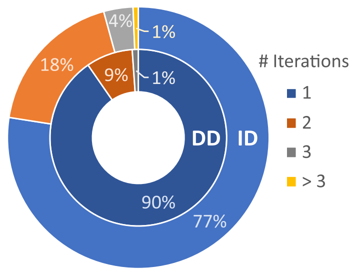


Fig. 13: Number of iterations per declaration for `DomesticDeclarations` and `InternationalDeclarations`

5.1 Declarations

An important aspect of the travel expense reimbursement process at the TU/e are declarations that can be submitted after travel to obtain a refund.

From the 6,078 international declarations included in the `International-Declarations` log that actually involve a payment, i.e., are approved and settled, roughly 64 percent (3,878 cases) are assigned to a specific project. A prominent finding related to projects is that 440 declarations are booked on project 426, which is an exceptionally high number (e.g., the second most frequent project number only appears 50 times). A possible explanation for this will be given in Section 6.1.

Resubmissions The approval process of declarations (domestic, as well as international) at the TU/e allows that declarations that have been rejected can be resubmitted by the *Employee* [8], most likely in a slightly altered version addressing the reasons for rejection. This implies that for a single declaration, several submissions and rejections can be recorded until the declaration is eventually approved and settled (*Payment Handled* was executed).

In Fig. 13, the distribution of number of resubmissions in the `DomesticDeclarations` and `InternationalDeclarations` logs for eventually paid declarations is shown. The great majority of domestic declarations (90%) are approved and settled after the first submission, i.e., no modification and re-submission was necessary at all. 9% of the declarations were revised once before they were approved. In case of international declarations, this ratio is slightly worse, with only about three-quarters of applications being approved in the first round and 18% after one correction.

In general, however, the majority of domestic as well as international declarations that are eventually settled are not subject to more than one revision.

Rerouting Another potential problem in the processing of declarations are delays in the approval phase. According to the procedure employed by the TU/e, travel declarations that are not handled by the *Budget Owner* (BO) within seven days should automatically be delegated to the *Supervisor* [8]. In the following, cases that were potentially subject to such a rerouting will be analyzed.

Based on the described approval process (cf. Section 3.1), the *Budget Owner* is involved between the approval by the *Administration* and the final approval by the *Supervisor*. However, in case the BO and the *Supervisor* are the same person, only the latter step is executed. That means, there are also "normal" cases where the BO approval is skipped legitimately. We therefore concentrated on the set of 293 cases, where there are at least seven days between the approval by the *Administration* and the *Supervisor*. However, the mean and median waiting times between these two steps (8.1 days and 7 days respectively) indicate, that there are also many cases with unusually long waiting times (max. 41 days), which might not be explainable by a potential rerouting.

As described in Table 3, the *Supervisor* usually takes about 22 working hours to approve open requests in the `InternationalDeclarations` log after the *Budget Owner* approved them. To also remove the cases which took exceptionally long, we filter for waiting times of more than seven days (until the "BO rerouting" takes places) but not more than the seven days plus the average time the *Supervisor* usually needs to approve a declaration. This results in a set of 161 cases where the approval by the *Supervisor* is given within the usual 22 working hours time frame after the assumed rerouting has taken place.

The remaining cases might be exceptional for other reasons than the *Budget Owner* time out (e.g. there are 109 cases where the approval by the *Supervisor* takes more than 30 working hours).

Late Applications Next to the rerouting of declarations to the *Supervisor* after seven days, another time constraint plays an important role. Employees are asked to file their declarations within two months after their trip has finished [8]. To identify the number of declarations that are submitted two months after the trip has finished, we first filter the event logs (`InternationalDeclarations` and `Permit` logs) for all cases where the *End Trip* is eventually followed by *Declaration SUBMITTED by EMPLOYEE* activity and where the time difference between these activities is more than 2 months (60 days). In both event logs there are 5% (`InternationalDeclarations` log: 378 cases, `Permit` log: 345 cases) of the total number of cases where the employee submitted the declaration more than 60 days after the trip ended.

The employee can resubmit the declaration because it is rejected before by one of the roles (e.g., *Administration*, *Supervisor*). Therefore, we have to identify, on the results of previous findings, all the cases that contain at least two declaration submissions by the employee because a rejection is happening in between. In the `InternationalDeclarations` log there are in total 141 cases, 2% of the total number of cases or 37% of the cases where the employee submitted the first declaration more than 60 days after the end of the trip and afterwards at least another resubmission took place. In the `Permit` log, there are 149 cases out of 345. In addition, as Fig. 14 illustrates (inner circle), out of 141 number of cases, 99 (70%) are submitted two times, 28 (20%) are submitted 3 times, 9

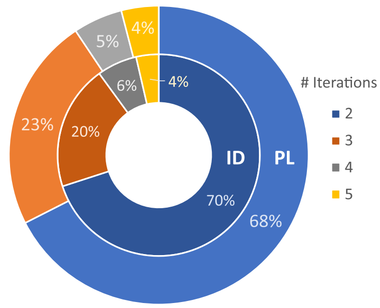


Fig. 14: Proportion of declarations submitted at least two times and where the first declaration was submitted at least 60 days after the end of the trip.

(6%) are submitted 4 times and only 5 (4%) are submitted 5 times. We have realized that before the employee resubmits an application he/she first has to reject it and afterwards resubmit it again. We are assuming that this is a process flow restricted by the software or the university's policy.

Regarding the `Permit` log (Fig. 14, outer circle), out of 149 number of cases, 101 (68%) are submitted 2 times, 33 (23%) are submitted 3 times, 8 (5%) are submitted 4 times, and 6 (4%) declaration are submitted 5 times. There is only one declaration submitted 10 times which is not shown in Fig. 14 because of the small number of occurrence.

In the `InternationalDeclarations` log, we found that before an employee can resubmit an application he/she first has to reject it and afterwards resubmit it again. This was not always true for the `Permit` log. We are assuming that this is an university's policy for international declarations.

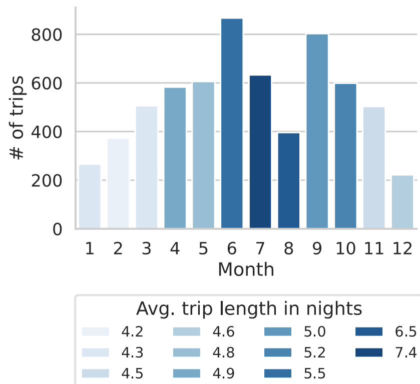
As it is pointed out above in the `InternationalDeclarations` log, most of the declarations happening in 2017 are rejected by *Missing*, *Pre-Approver* or *Supervisor*. Because of the *drift concept* the flow has completely changed in 2018 where *Missing* and *Pre-Approver* have never rejected a declaration. However, most of the declarations in 2018 first are rejected by the *Administration* and, if it was approved, the next resource responsible to approve or reject a declaration was the *Budget Owner* or *Supervisor*.

5.2 International Trips

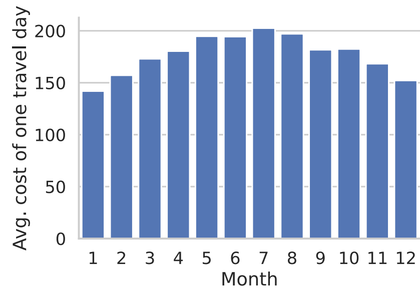
In the data set, information on individual trips and related payments are contained. This section provides some insights on trips approved and paid by the TU/e.

For the following analysis, 6,357 trips, recorded in the `Permit` log, that did not exceed 70 days were considered (69 trips were excluded). In general, about 90 percent of all trips do not take longer than 10 days and nearly every second trip is shorter than five days.

Considering the distribution of the trips over the year, significant differences can be noticed. Figure 15a shows, on the one hand, that the number of trips



(a) Number of trips started per month with their average trip length



(b) Average cost of a travel day per month, based on the *TotalDeclared* attribute

---

One exceptional expensive trip in July was excluded (Permit 54518, see also Section 6.2).

Fig. 15: Distribution of trips not exceeding 70 days over the months based on the `Permit`log.

increase during spring and reach their maximum in June. The amount of trips then decreases rapidly until August, reaches a second peak in September and then decreases again in the course of autumn. On the other hand, Fig. 15a shows the average trip length for each month. In winter, apparently most trips are rather short (on the average about 4 to 4.5 nights), whereas in the summer months not only the amount of trips, but also their average duration increases. On average, the longest trips take place in July and August with 7.4 nights and 6.5 nights respectively.

Another aspect are the expenses per trip, shown in Fig. 15b. To be able to compare the different months, the average costs of a single travel day are considered. Trips during the summer months seem to be slightly more expensive – trips in January cost on the average 142 € per day, while in July the costs per day rise to 203 €, which corresponds to a plus of 43 percent.

6 Conformance Analysis

This section describes and illustrates some anomalies identified in the provided event logs such as declarations for unapproved travels and possibly inadvertent double payments.

6.1 Declarations for Unapproved Travels

In order to be allowed to file a declaration, the corresponding travel permit must have been approved [8]. All cases recorded in the `Permit` log adhere to this rule. There are no declarations without a previously approved travel permit.

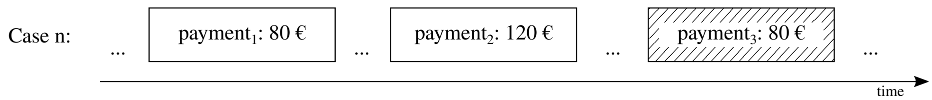


Fig. 16: Visualization of a double payment, for a case with three payment activities. Since the value of the first and third payment are equal, the latter one is classified as potential double payment. The estimated loss in this case would be 80 €.

However, in the `InternationalDeclarations` log, there are 425 declarations (6%) which are final approved by either the director or the supervisor but have neither an approved permit nor any permit related activity recorded at all. These seem to form a special group of declarations, as in these cases the attributes *Permit ID* and *Permit id2* (renamed attribute, see Section 2.2) do not match. While the *Permit ID*s all have values above 86541 (which are not included in the `Permit` log itself), the value for *Permit id2* is the same for all 425 cases (permit ID 423). The 425 cases also inherited all attributes from this single permit, such as the project number (426), causing the high frequency of this project number pointed out in Section 5.1. The permit 423 itself is included in the `Permit` log, but the corresponding case refers to only one international declaration and also has an approved permit. Therefore, it is very likely that these declarations being linked to the same, unrelated permit, is caused by a data recording or data processing problem and might not indicate a compliance issue.

The same applies to the `PrepaidTravelCost` log, where for 178 cases the request was approved without any preceding permit related activities. Again, the *Permit id2* attribute is either UNKNOWN, or 423, just as in the `InternationalDeclarations` log. Apart from this group of cases, all declarations and requests are preceded by an approved permit.

6.2 Double Payments

Since the data set at hand is from the TU/e reimbursement process, the correct handling of payments plays an essential role in the process. One aspect, therefore, are potential double payments. As a disclaimer, the findings of this part of the analysis should be interpreted with caution, since the amounts mentioned in the event logs are not exact values [8].

For the following, we define double payments as the occasion of multiple payment activities with the same amount in one case, as depicted in Fig. 16. While there are no cases in the domestic and international declaration logs that meet this definition, there are 20 suspicious cases in the `Permit` log. Because of this, we extracted a subset of the event log, which only contained those cases and conducted further analyses. To estimate the potential financial damage, caused by the double payments, we added up the total amount of double paid expenses. For example, in the case shown in Fig. 16, the potential loss would be 80 €, if we assume that there are no other payment activities with the same amount later

Table 4: The top five suspicious cases in the `Permit` log ordered by the estimated loss.

Permit ID	Trace length	Num. of payments	Potential loss
54518	24	3	475,704.95 €
72935	23	3	2,726.80 €
58250	24	3	1,397.14 €
83423	14	3	1,394.16 €
78577	19	3	667.06 €

in the case. Table 4 shows an excerpt from the top five cases, measured on the value of potential losses due to double payments. With more than 475,000 € the permit "54518" has by far the highest potential loss and should be investigated further by the process owner.

Based on our initial findings, we further analyzed the payments in the `Permit` log. In the first step, we compared the sum of the single requested amounts, and the designated total declared amount per permit. We found that in 2,625 cases (40.8%), the total declared amount was higher. On average, the difference was about 205 €, while at maximum, the stated amounts differed by 1,747.76 €.

Further, we compared those two numbers with the requested budget and the permit's indicator, whether the budget was overspent or not. However, we couldn't find a clear relation either between the sum of requested amounts and the requested budget or between the requested budget and the total declared amount. Therefore we assumed that if either the total declared amount or the sum of the requested amounts is higher than the requested budget, the case should be marked as overspent. Following this assumption, we found that about 13% (594) cases were not correctly marked as overspent. However, as already stated in the introduction of this subsection, in some cases this might be caused by the inaccurate amounts in the log.

7 Conclusion

In this paper, we reported on our findings for the BPI Challenge 2020. Our analyses included the three main process mining perspective discovery, conformance, and performance. The findings yielded several relevant insights about the process. We detected concept drift from 2017 to 2018, performance variations during summer and winter holiday seasons, and potential conformance issues related to double payments and reimbursements without a permit. Discovering and transforming the underlying processes into BPMN process models supported the knowledge discovery and highlighted characteristics, such as batch processing. The models will provide a common ground for the process owner and other stakeholders to further analyze the processes in the future. One of the main challenges we faced was identifying the links among the five event logs. Furthermore, we found it difficult to provide clear reasons for several phenomena we observed, such as potential double payments. This emphasizes that a process mining analysis is an important starting point but certainly not the end of a process analysis. Several answers can only be provided by involving domain experts and by going back into the original data.

References

1. van der Aalst, W.: Process Mining - Data Science in Action, Second Edition. Springer (2016), https://doi.org/10.1007/978-3-662-49851-4
2. Berti, A., van Zelst, S.J., van der Aalst, W.: Process mining for python (PM4Py): Bridging the gap between process- and data science. In: Proceedings of the ICPM Demo Track 2019. pp. 13–16 (2019), http://ceur-ws.org/Vol-2374/
3. Bose, R.P.J.C., van der Aalst, W., Zliobaite, I., Pechenizkiy, M.: Handling concept drift in process mining. In: Advanced Information Systems Engineering International Conference, CAiSE. Lecture Notes in Computer Science, vol. 6741, pp. 391–405. Springer (2011), https://doi.org/10.1007/978-3-642-21640-4_30
4. Denisov, V., Fahland, D., van der Aalst, W.: Unbiased, fine-grained description of processes performance from event data. In: Business Process Management - 16th International Conference (BPM). Lecture Notes in Computer Science, vol. 11080, pp. 139–157. Springer (2018), https://doi.org/10.1007/978-3-319-98648-7_9
5. van Dongen, B.: BPI Challenge 2020 (2020), https://doi.org/10.4121/uuid:52fb97d4-4588-43c9-9d04-3604d4613b51, accessed: 2020-08-17
6. van Dongen, B.F., de Medeiros, A.K.A., Verbeek, H.M.W., Weijters, A.J.M.M., van der Aalst, W.: The ProM Framework: A New Era in Process Mining Tool Support. In: Applications and Theory of Petri Nets 2005, 26th International Conference, ICATPN. Lecture Notes in Computer Science, vol. 3536, pp. 444–454. Springer (2005), https://doi.org/10.1007/11494744_25
7. Fluxicon BV: Disco, https://fluxicon.com/disco/, version: 2.6.6
8. ICPM 2020: Business Process Intelligence Challenge (2020), https://icpmconference.org/2020/bpi-challenge/, accessed: 2020-08-17
9. Leemans, S.J.J., Fahland, D., van der Aalst, W.: Process and deviation exploration with inductive visual miner. In: Proceedings of the BPM Demo Sessions Co-located with the 12th International Conference on Business Process Management (BPM). CEUR Workshop Proceedings, vol. 1295, p. 46. CEUR-WS.org (2014)
10. Weske, M.: Business Process Management - Concepts, Languages, Architectures, Third Edition. Springer (2019), https://doi.org/10.1007/978-3-662-59432-2