BPI challenge 2020

Nela Bulavova

V08973

Business Information Systems
Prof. Paolo Ceravolo
a.a. 2021 - 2022

Table of Contents

Introduction

The BPI Challenge 2020 [1] provides participants with a real-life event log from Eindhoven University of Technology (TU/e) presenting procedures for arranging travels of staff members to conferences, other universities, etc., and the procedures for the reimbursement of the expenses.

The competition challenges its participants to analyse these data using any available tools and techniques, focusing on the questions of interest.

Description of the case study

Two types of travels of the staff members can be distinguished - domestic and international. An employee can attend domestic trips without any prior permission and can ask for payment of the costs afterwards. For taking part in an international trip a permission from the supervisor needs to be obtained, to get a permission an employee must fill in a travel-permit document that is then passed to the superiors for assessment. To receive money for the expenses a claim is filed. Expenses can be filed as soon as costs are actually paid for or within two months after the trip.

The process flow of the data was described by the challenge [1] where two process flows are outlined.

First one is similar for all various declaration documents. After an employee submits a request, it is sent for approval to the travel administrator. If it is approved, it is sent to the budget owner and after that to the supervisor. In case when the budget owner and the supervisor are the same person, only one of these steps is taken. In some cases, the director needs to approve the request. If the request is rejected an employee can either resubmit the request or reject the request. If the request is approved in all the steps, the payment is requested and made.

The second process flow is followed by the travel permits, in comparison to the first one this process flow does not involve payment. After all the steps are approved a trip can take place. Before the trip starts an employee can ask for reimbursement of pre-paid travel costs. After the trip an employee is reminded to submit a travel declaration.

Data description

The data are already extracted from the information systems of the TU/e. They are provided to the challenge's participants in the form of five datasets - request for payment, prepaid travel costs, travel permit data, international declarations, domestic declarations. Datasets are available in the XES format, which is an XML-based standard for event logs. [2]

Datasets contain data from 2017 and 2018. The year 2017 was a pilot year and the data come from only two departments, whereas from the year 2018 onwards the data are from the whole university. The organizers of the challenge provide us with the process flow which describes the process for 2018.

The data are anonymized, no internal IDs are visible throughout the log. Therefore, no staff members can be identified in the data, instead for each step the role of the person who executed the step is recorded (system, staff member, unknown, missing).

Organisational goals

In this project the focus will be given on following questions of interest purposed by the challenge itself. Answering the questions asked by the TU/e provides value for the organization and is therefore considered of significant importance.

1. What is the throughput of a travel declaration from submission (or closing) to paying?
2. Is there any difference in throughput between national and international trips?
3. Where are the bottlenecks in the process of a travel declaration?
4. How many travel declarations get rejected in the various processing steps and how many are never approved?
5. How many travel declarations are booked on projects?
6. How many corrections have been made for declarations?
7. Are there any double payments?

Knowledge Uplift Trail

Each given log provides different perspective on the overall process; however, not all of them were used in the analysis. With the aim to answer business questions provided by the TU/e regarding International Declaration and Domestic declaration, this project will mainly perform analysis on these two datasets. The results of these analyses might help to understand the overall process better, which leads to potential improvement of them (reducing bottlenecks, understanding of data anomalies).

To acquire new knowledge from the event logs the following steps were taken using Disco software [3] and PM4PY Python library [4].

Table 1 Knowledge uplift trail

	Input	Analytics/model	Type	Output
Step 1	5 datasets (Total of 270211 events)	Observing and presenting the most important insights	Descriptive	Statistics
Step 2	2 datasets (Domestic and international declarations)	Filtering on time using PM4PY	Prescriptive	Time filtered data
Step 3	Step 2	Filtering on activities	Prescriptive	Filtered data
Step 4	Step 3	Filtering on frequent variants	Prescriptive	Two filtered datasets
Step 5	Step 4	Using process discovery algorithms	Prescriptive	Alpha miner Inductive miner Heuristic miner

Step 6	Step 5 - alpha miner	Visualizing petri net out of alpha miner	Descriptive	Petri net
Step 7	Step 5 - inductive miner	Visualizing Process tree out of inductive miner	Descriptive	Process tree
Step 8	Step 7	Visualizing Petri net	Descriptive	Petri net
Step 9	Step 5 - heuristic miner	Visualizing Petri net	Descriptive	Petri net
Step 10	Step 5	Measuring Fitness, Precision, Generalization, Simplicity	Descriptive	Measures

Project Results

Firstly, the Disco software was used to better understand the provided datasets. Starting with RequestForPayment dataset which contains 6,886 cases and 36,796 events, 89 variants were found, the dataset consists of 19 activities. This dataset refers to the expenses that are not travel related, employees might request money for purchasing hardware for work, etc. Each dataset consists of events with the respect to the characteristics. DomesticDeclarations dataset is composed of the events related to the declarations of the domestic trips. InternationalDeclarations dataset consists of the events related to the declarations of international trips and the application of travel permits. PrepaidTravelCost dataset contains the events related to the claims of reimbursement of the prepaid travel costs and the application of travel permits. Permitlog dataset has all the the events related to international trips.

In Disco we can get the initial overview, the table below shows some statistics of the datasets. For each dataset there is recorded the number of cases, events, variants, and activities.

A case is a process instance (a collection of events that relate to the same process execution). An event is an activity, or a task executed at a particular time for a specific case. A variant is a collection of cases that follow the same sequence of executed activities.

Table 2 BPI Challenge datasets

Dataset	Cases	Events	Variants	Activities
RequestForPayments	6886	36796	89	19
DomesticDeclarations	10500	56437	99	17
InternationalDeclarations	6449	72151	753	34
PrepaidTravelCost	2099	18246	202	29
PermitLog	7065	86581	1478	51

Using Fluxicon Disco we were provided with the visualization of 100 % activities and 100% paths of the raw data. As it can be seen from the visualizations below, filtering is necessary for drawing conclusions from the data.

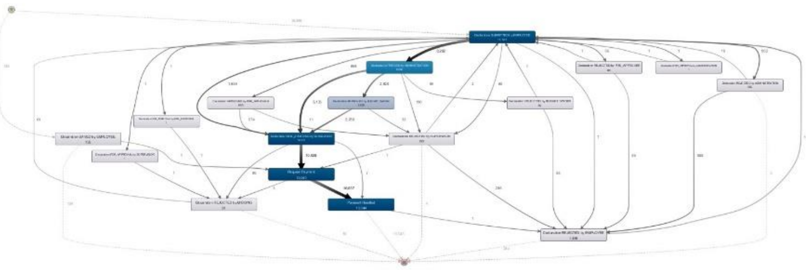


Figure 1 Domestic Declarations

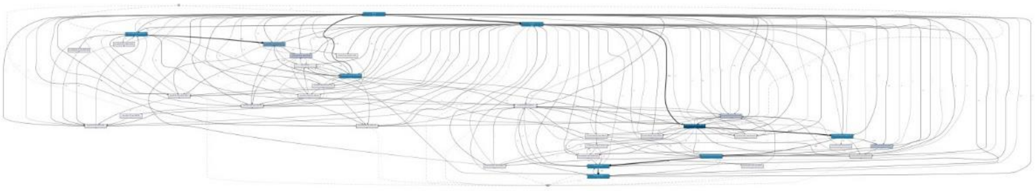


Figure 2 International declarations

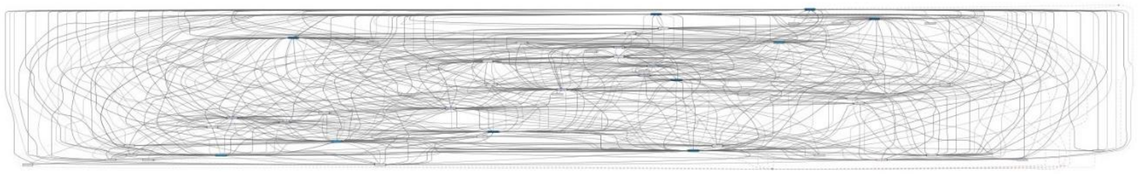


Figure 3 Permit log

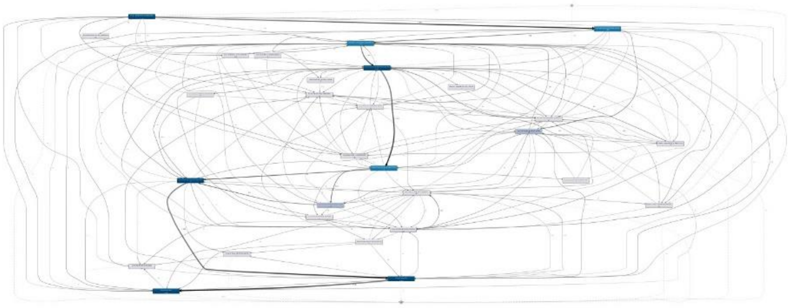


Figure 4 Prepaid travel cost

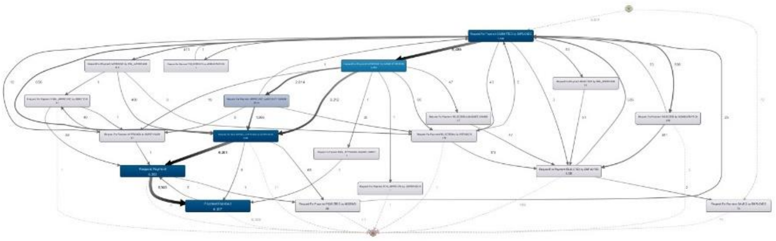


Figure 5 Request for payment

As was mentioned before, the year 2017 was a pilot year when the process was still not settled. Therefore, for analysis the cases that started before year 2018 are filtered out.

The impact of this filtering can be seen in the table below, where each dataset was reduced by approximately 19% cases and 16% events.

Table 3 Datasets timeframe

Dataset	2017		2018+		Total	
	Cases	Events	Cases	Events	Cases	Events
RequestForPayments	1108	4976	5778	31820	6886	36796
DomesticDeclarations	2240	10062	8260	46375	10500	56437
InternationalDeclarations	1497	14239	4952	57912	6449	72151
PrepaidTravelCost	323	2442	1776	15804	2099	18246
PermitLog	1467	15497	5598	71084	7065	86581

Table 4 Datasets reduction

Dataset	Reduction in %	
	Cases	Events
RequestForPayments	16.09	13.52
DomesticDeclarations	21.33	17.83
InternationalDeclarations	23.21	19.74
PrepaidTravelCost	15.39	13.38
PermitLog	20.76	17.9

As was purposed in the organizational goals, the focus in this project is given on two datasets - domestic declarations and international declarations.

Analysis of the domestic declarations' dataset

1. Filtering

Besides filtering out the cases that started before year 2018 we can filter out the cases that do not commence with the submission of the declaration by the employer. As can be seen in Figure 6, the activity "Declaration SAVED by EMPLOYEE" starts the process, but no activity follows this one. For starting the process of domestic declaration, it is necessary that it starts with the activity "Declaration SUBMITTED by EMPLOYEE". This removal corresponds to the reduction of 100 cases. After submission of the employee the cases can go to "Declaration FOR_APPROVAL by ADMINISTRATION" which in this event log took only one case, might be interesting to get to know what was this special case, having a closer look at this case I found that this case was five times resubmitted by the employee and many times rejected when then after the step "Declaration FOR_APPROVAL by ADMINISTRATION" it finally positively flowed through the diagram to a successful end "PAYMENT HANDLED". As it might be interesting for further analysis, for the creation of the process model it can be filtered out. Every declaration must include at least two activities - the two mandatory ones are the submission of the declaration and the approval or rejection of the declaration.

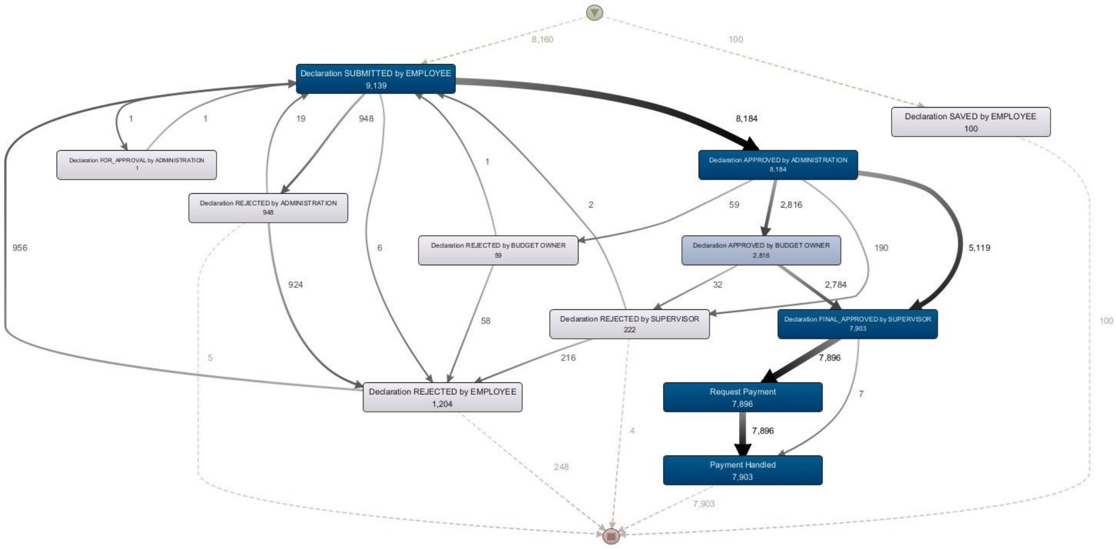
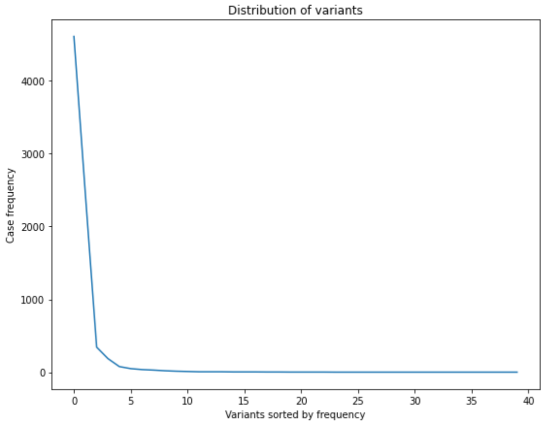
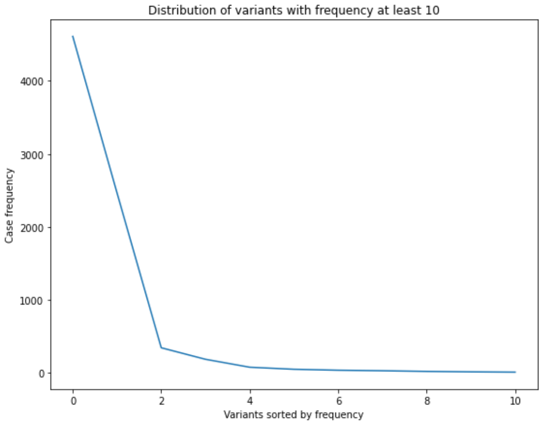

8,160
100
Declaration SUBMITTED by EMPLOYEE
9,139
8,184
Declaration SAVED by EMPLOYEE
100
1
1
19
948
Declaration FOR_APPROVAL by ADMINISTRATION
1
Declaration APPROVED by ADMINISTRATION
8,184
1
Declaration REJECTED by ADMINISTRATION
948
2
59
2,816
956
6
Declaration REJECTED by BUDGET OWNER
59
Declaration APPROVED by BUDGET OWNER
2,816
190
5,119
924
32
2,784
58
Declaration REJECTED by SUPERVISOR
222
Declaration FINAL_APPROVED by SUPERVISOR
7,903
100
5
216
7,896
Declaration REJECTED by EMPLOYEE
1,204
Request Payment
7,896
7
4
7,896
248
Payment Handled
7,903
7,903

Figure 6 Domestic declarations diagram, using DISCO

The end activities are 'Declaration REJECTED by ADMINISTRATION', 'Declaration REJECTED by EMPLOYEE', 'Declaration REJECTED by SUPERVISOR', 'Payment Handled'. We can be interested in both successful declarations that end with 'Payment Handled' and unsuccessful ones. Let's assume that we are only interested in the successful ones as they provide us with useful information about the ideal way of the system.
After all the filtering the number of cases got reduced from 10500 to 7902, the number of variants is 40.

Filtering on variants
Using Python, it was possible to illustrate the distribution of variants - case frequency against the variants sorted by frequency.



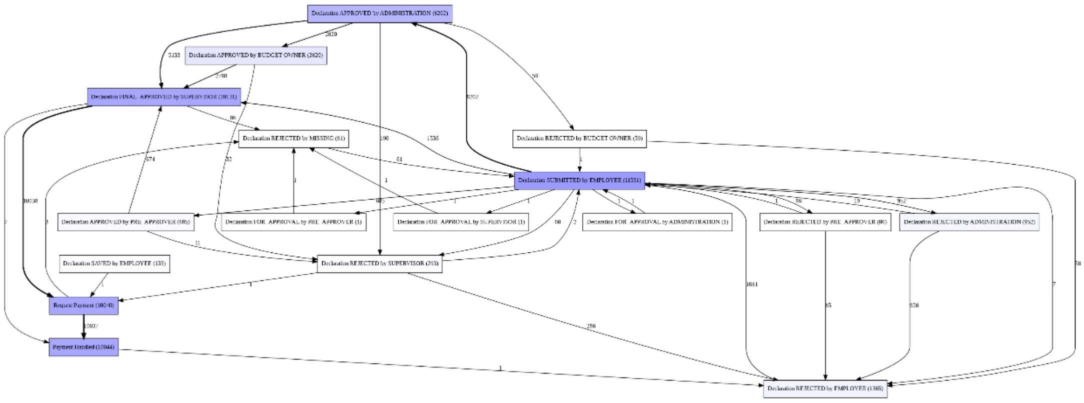


As can be seen from the graphs (on the left side a graph of all 40 variants, on the right side a graph of only those variants that have at least 10 cases), a few variants describe considerably larger number of cases than the rest of the variants. The frequencies of variants let us distinguish between the typical process execution patterns and the exceptions. It is possible to look at the less frequent variants and detect what is the reason for not following the standard procedure. The following analysis can focus on either the mainstream variants or the exceptions. As to answer question n.2, we will focus on the more frequent variants (that are followed by at least 10 cases). Number 10 was chosen based on the fact that there are 29 variants each with a frequency of maximum 6 cases, all together 29 variants containing only 61 cases.

2. Process discovery

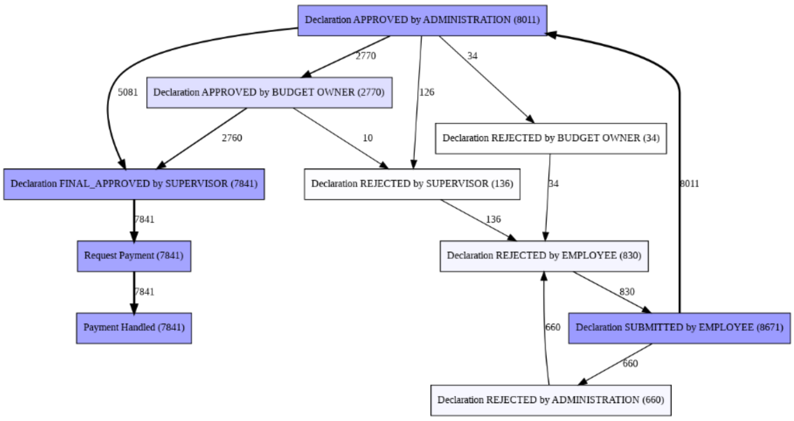
To construct a representation of a business process several techniques can be used. A few algorithms that take the filtered event log as an input and provide the corresponding process model will be presented.

In the first step the unfiltered and filtered log was modelled using Directly-followed graphs.



Declaration APPROVED by ADMINISTRATION (8202)
2820
5133
Declaration APPROVED by BUDGET OWNER (2820)
2780
59
8202
Declaration FINAL_APPROVED by SUPERVISOR (10131)
86
Declaration REJECTED by MISSING (91)
190
1536
Declaration REJECTED by BUDGET OWNER (59)
674
32
61
1
1
Declaration SUBMITTED by EMPLOYEE (11531)
10038
685
1
1
1
1
1
86
19
952
Declaration APPROVED by PRE_APPROVER (685)
Declaration FOR_APPROVAL by PRE_APPROVER (1)
Declaration FOR_APPROVAL by SUPERVISOR (1)
80
2
Declaration FOR_APPROVAL by ADMINISTRATION (1)
Declaration REJECTED by PRE_APPROVER (86)
Declaration REJECTED by ADMINISTRATION (952)
31
Declaration SAVED by EMPLOYEE (135)
Declaration REJECTED by SUPERVISOR (293)
1
1
1081
85
928
7
58
Request Payment (10040)
10037
286
Payment Handled (10044)
1
Declaration REJECTED by EMPLOYEE (1365)

Figure 7 DFG unfiltered log



Declaration APPROVED by ADMINISTRATION (8011)
2770
34
5081
Declaration APPROVED by BUDGET OWNER (2770)
126
2760
10
Declaration REJECTED by BUDGET OWNER (34)
Declaration FINAL_APPROVED by SUPERVISOR (7841)
Declaration REJECTED by SUPERVISOR (136)
34
8011
7841
136
Request Payment (7841)
Declaration REJECTED by EMPLOYEE (830)
7841
830
Payment Handled (7841)
660
Declaration SUBMITTED by EMPLOYEE (8671)
660
Declaration REJECTED by ADMINISTRATION (660)

Figure 8 DFG filtered log

Then we applied three algorithms on the filtered dataset to obtain a process model.

a) Alpha miner

Alpha miner algorithm reconstructs causality from a set of sequences of events (observing the relations between two succeeding activities). It was one of the first algorithms being able to discover concurrency. When applying the alpha miner, the focus is given on the order of the activities within a particular case.

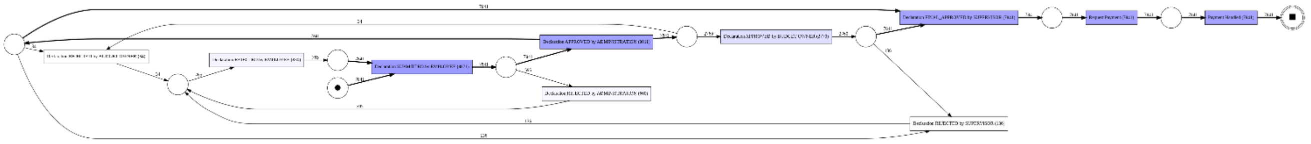

Declaration FINAL_APPROVED by SUPERVISOR (7841)
Request Payment (7841)
Payment Handled (7841)
Declaration APPROVED by ADMINISTRATION (8011)
Declaration APPROVED by BUDGET OWNER (2770)
Declaration REJECTED by BUDGET OWNER (34)
Declaration REJECTED by EMPLOYEE (830)
Declaration SUBMITTED by EMPLOYEE (8671)
Declaration REJECTED by ADMINISTRATION (660)
Declaration REJECTED by SUPERVISOR (136)
7841
34
136

Figure 9 Petri net, using alpha miner

b) Heuristic miner

This miner improves the alpha algorithm by considering frequencies, it can filter out infrequent behavior.

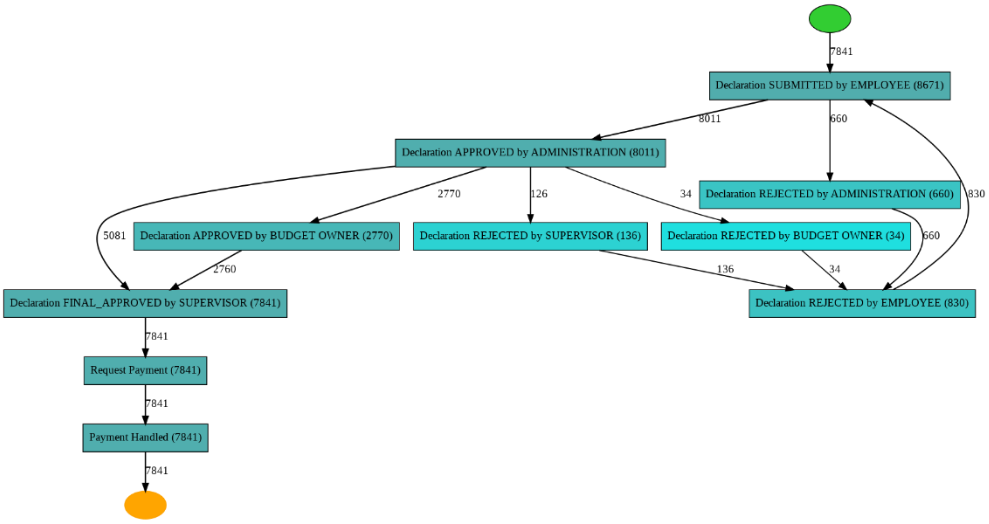

7841
Declaration SUBMITTED by EMPLOYEE (8671)
8011
660
Declaration APPROVED by ADMINISTRATION (8011)
2770
126
34
Declaration REJECTED by ADMINISTRATION (660)
830
5081
Declaration APPROVED by BUDGET OWNER (2770)
Declaration REJECTED by SUPERVISOR (136)
Declaration REJECTED by BUDGET OWNER (34)
660
2760
136
34
Declaration FINAL_APPROVED by SUPERVISOR (7841)
Declaration REJECTED by EMPLOYEE (830)
7841
Request Payment (7841)
7841
Payment Handled (7841)
7841

Figure 10 Heuristic net, using heuristic miner

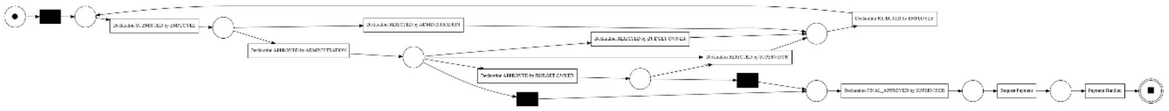

Declaration SUBMITTED by EMPLOYEE
Declaration REJECTED by ADMINISTRATION
Declaration REJECTED by EMPLOYEE
Declaration REJECTED by BUDGET OWNER
Declaration APPROVED by ADMINISTRATION
Declaration REJECTED by SUPERVISOR
Declaration APPROVED by BUDGET OWNER
Declaration FINAL_APPROVED by SUPERVISOR
Request Payment
Payment Handled

Figure 11 Petri net, using heuristic miner

c) Inductive miner

This algorithm takes the event log as an input and recursively decomposes it into sub-logs. The sequence of activities is represented as a process tree.

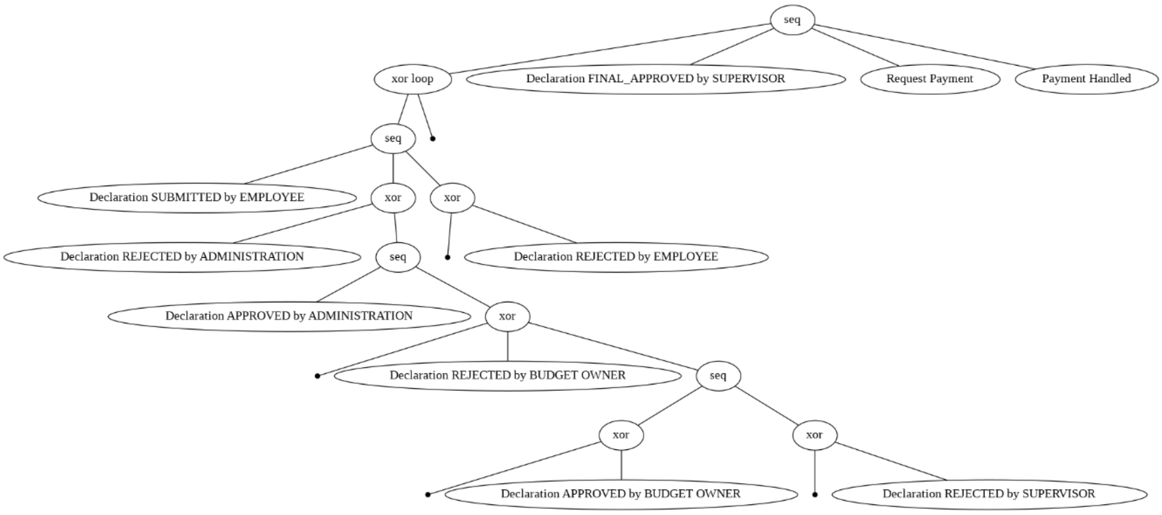

seq
xor loop
Declaration FINAL_APPROVED by SUPERVISOR
Request Payment
Payment Handled
seq
Declaration SUBMITTED by EMPLOYEE
xor
xor
Declaration REJECTED by ADMINISTRATION
seq
Declaration REJECTED by EMPLOYEE
Declaration APPROVED by ADMINISTRATION
xor
Declaration REJECTED by BUDGET OWNER
seq
xor
xor
Declaration APPROVED by BUDGET OWNER
Declaration REJECTED by SUPERVISOR

Figure 12 Process tree, using inductive miner

The process tree was converted into a Petri net.

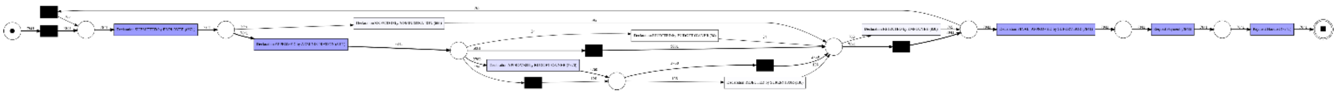


Figure 13 Petri net, using inductive miner

3. Conformance checking

Conformance checking is used to check if the process model is the correct reflection of the real process and vice versa.

As can be seen in table below the measures - fitness, precision, generalization, and simplicity were evaluated for all three algorithms. In this way the algorithms can be compared using the four dimensions.

Fitness of value 1 indicates that all the behavior in the event log is possible according to the model. A model should be precise - not allow all kinds of behavior unrelated to the event data, avoiding underfitting. Generalization assesses the extent to which the obtained process model can capture future behavior. A model should be general enough, avoiding overfitting. Simplicity is also an important measure capturing the complexity of the process model.

Table 5 Quality criteria

Measures	Algorithms		
	Alpha miner	Heuristic miner	Inductive miner
Fitness	0.85	1	0.99
Precision	1	0.75	0.98
Generalization	0.96	0.96	0.97
Simplicity	0.66	0.68	0.74

It is difficult to choose the best model, as one model can be better in terms of fitness but much worse in terms of other measures. The measures can represent axis in a 4-dimensional graph, the model can be chosen on the so-called Pareto front. From table 4 it is visible that inductive miner seems to be evaluated better than other two miners.

<u>Analysis of the international declarations' dataset</u>

1. Filtering

As in the dataset domestic declarations, firstly the cases that started before the year 2018 are filtered out.

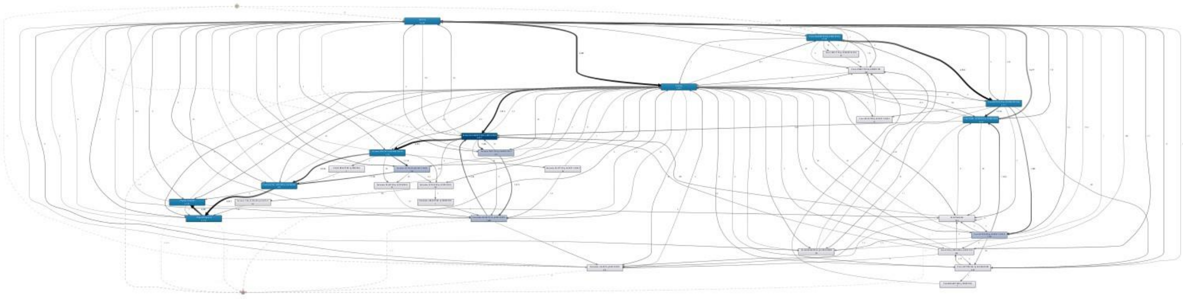
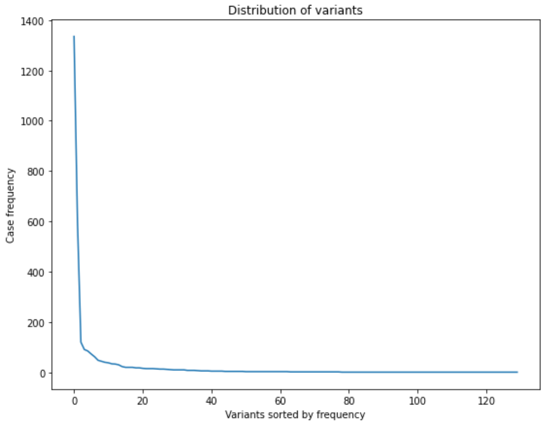
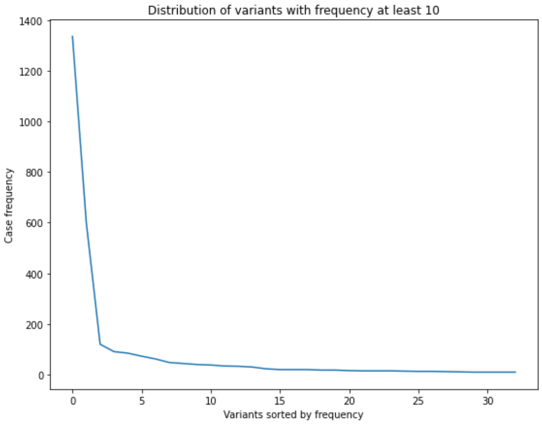


Figure 14 Domestic declarations after timestamp filter

Assuming that an employee can submit the declaration after being granted the permission to travel, we can focus on the cases which include payment handled activity. To understand the best-case scenario of an international declaration, the cases that were rejected are filtered out.

After this filtering the number of variants was reduced to 130. Compared to the domestic declarations' dataset, the international declarations' dataset has more variants, many of which are represented by a single case. Observing the frequencies of each variant, graphs below, only variants with at least 10 cases were kept, like in domestic declarations' dataset.





2. Process discovery

Directly-followed graphs were used to model unfiltered and filtered data.

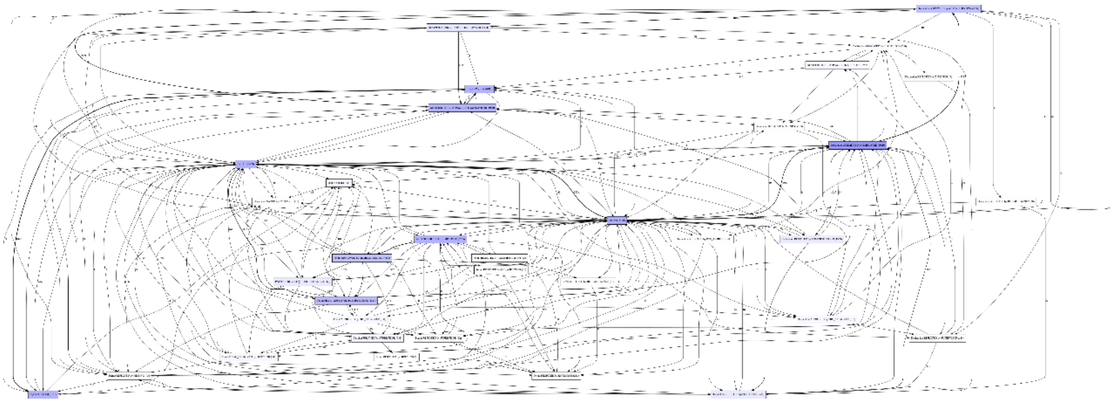


Figure 15 DFG unfiltered log

On filtered data the limitation of 14 most frequent edges was set to simplify the graph.

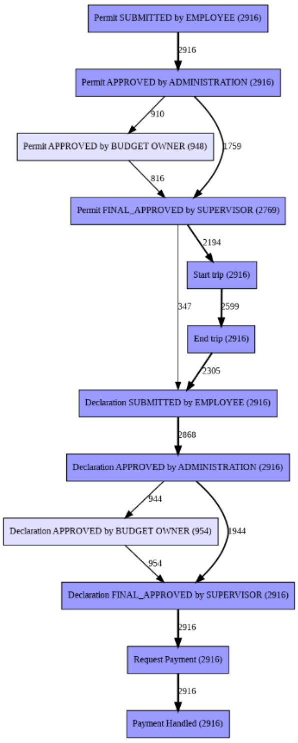

Permit SUBMITTED by EMPLOYEE (2916)
2916
Permit APPROVED by ADMINISTRATION (2916)
910
1759
Permit APPROVED by BUDGET OWNER (948)
816
Permit FINAL_APPROVED by SUPERVISOR (2769)
2194
Start trip (2916)
347
2599
End trip (2916)
2305
Declaration SUBMITTED by EMPLOYEE (2916)
2868
Declaration APPROVED by ADMINISTRATION (2916)
944
1944
Declaration APPROVED by BUDGET OWNER (954)
954
Declaration FINAL_APPROVED by SUPERVISOR (2916)
2916
Request Payment (2916)
2916
Payment Handled (2916)

Figure 16 DFG filtered log

a) Alpha miner

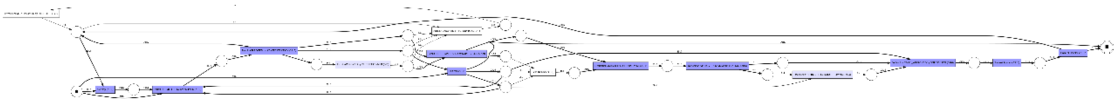


Figure 17 Petri net, using alpha miner

b) Heuristic miner

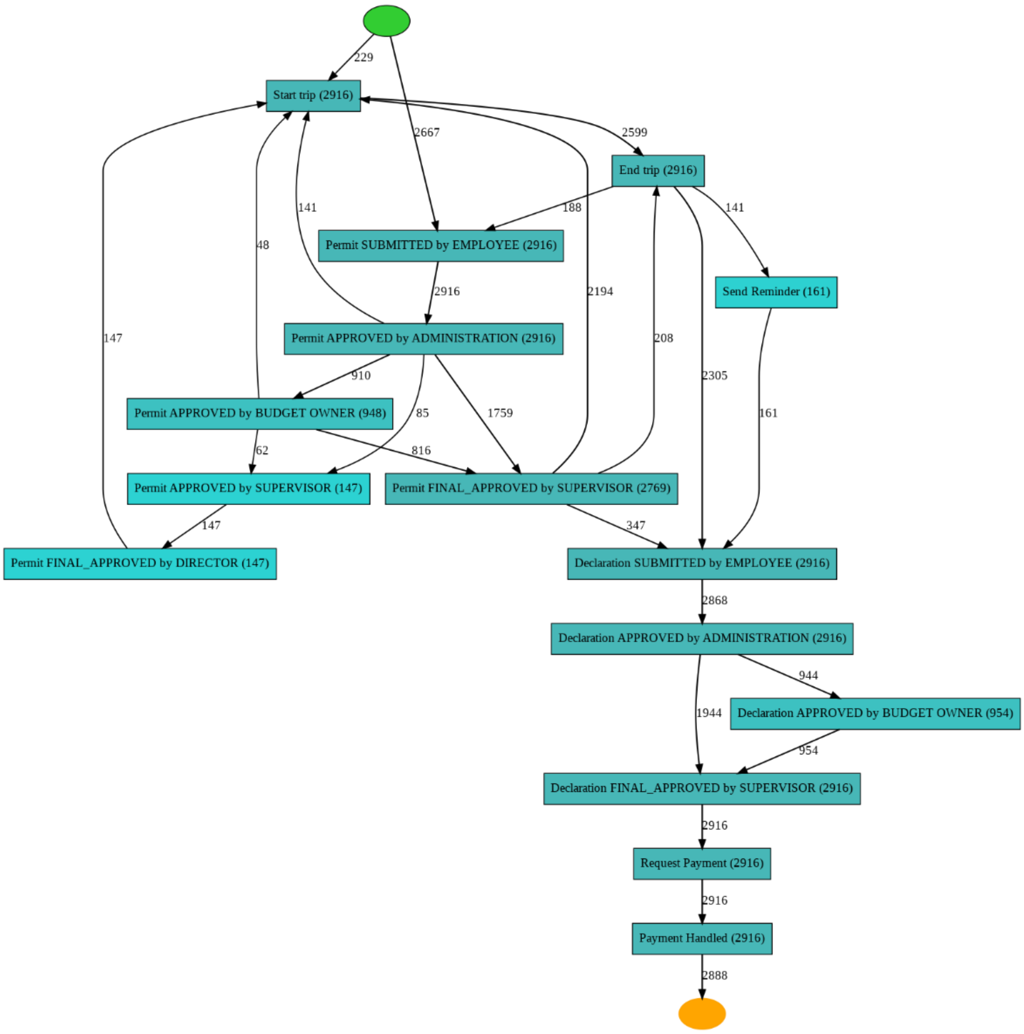

Start trip (2916)
229
2667
2599
End trip (2916)
141
188
141
Permit SUBMITTED by EMPLOYEE (2916)
Send Reminder (161)
48
2916
2194
Permit APPROVED by ADMINISTRATION (2916)
147
208
2305
910
Permit APPROVED by BUDGET OWNER (948)
85
1759
161
62
816
Permit APPROVED by SUPERVISOR (147)
Permit FINAL_APPROVED by SUPERVISOR (2769)
147
347
Permit FINAL_APPROVED by DIRECTOR (147)
Declaration SUBMITTED by EMPLOYEE (2916)
2868
Declaration APPROVED by ADMINISTRATION (2916)
944
1944
Declaration APPROVED by BUDGET OWNER (954)
954
Declaration FINAL_APPROVED by SUPERVISOR (2916)
2916
Request Payment (2916)
2916
Payment Handled (2916)
2888

Figure 18 Heuristic net, using heuristic miner

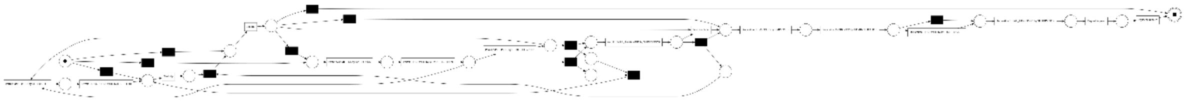


Figure 19 Petri net, using heuristic miner

c) Inductive miner

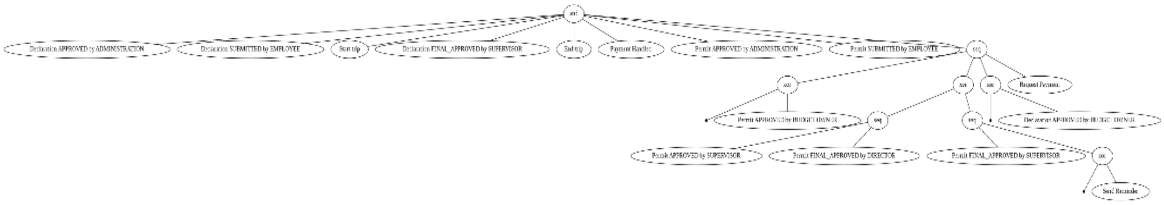

and
Declaration APPROVED by ADMINISTRATION
Declaration SUBMITTED by EMPLOYEE
Start trip
Declaration FINAL_APPROVED by SUPERVISOR
End trip
Payment Handled
Permit APPROVED by ADMINISTRATION
Permit SUBMITTED by EMPLOYEE
seq
xor
xor
xor
Request Payment
Permit APPROVED by BUDGET OWNER
seq
seq
Declaration APPROVED by BUDGET OWNER
Permit APPROVED by SUPERVISOR
Permit FINAL_APPROVED by DIRECTOR
Permit FINAL_APPROVED by SUPERVISOR
xor
Send Reminder

Figure 20 Process tree, using inductive miner

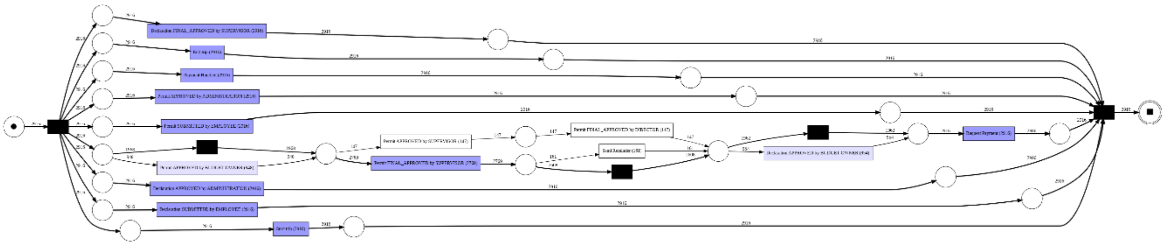

Declaration FINAL_APPROVED by SUPERVISOR (2916)
End trip (2916)
Payment Handled (2916)
Permit APPROVED by ADMINISTRATION (2916)
Permit SUBMITTED by EMPLOYEE (2916)
Permit APPROVED by BUDGET OWNER (426)
Permit APPROVED by SUPERVISOR (147)
Permit FINAL_APPROVED by SUPERVISOR (2769)
Permit FINAL_APPROVED by DIRECTOR (147)
Send Reminder (181)
Declaration APPROVED by BUDGET OWNER (954)
Request Payment (2916)
Declaration APPROVED by ADMINISTRATION (2916)
Declaration SUBMITTED by EMPLOYEE (2916)
Start trip (2916)

Figure 21 Petri net, using inductive miner

3. Conformance checking

Table 5 Quality criteria

Measures	Algorithms		
	Alpha miner	Heuristic miner	Inductive miner
Fitness	0.67	0.91	1
Precision	1	0.99	0.29
Generalization	0.97	0.89	0.97
Simplicity	0.51	0.67	0.67

As can be seen from the table above the alpha miner's simplicity measure is quite low, that implies that the graph is complex to read. Low precision in inductive miner shows that the process model allows all kinds of behavior. In this case the heuristic miner presents the data most accurately.

Business questions

1. What is the throughput of a travel declaration from submission (or closing) to paying?

From description of the datasets, there are two types of travel declarations. The throughput time (the time it takes a case to be completed from start to finish) was computed for both types, before filtering, after filtering not based on variant analysis and after filtering on variant analysis.

Before filtering the data, the Disco software provided us with the mean and median throughput time for the domestic declaration as: mean is 11,5 days and median is 7.3 days. After filtering not based on variants the following values were computed: mean of 11 days and 15 hours, median of 8 days and one hour. After filtering on variants, the mean is approximately 11 days and 10 hours and median of 8 days and 41 minutes. As can be seen the mean hasn't changed significantly, but the median increased after the data filtering.

In international declarations the values of mean and median for the unfiltered dataset are as follows: the mean case duration is 14 days and 4 hours, the median is 10 days and 4 hours. The calculated values for filtered dataset are - mean is 12 days 20 hours, median is 9 days 9 hours. After filtering based on most frequent variants mean is 12 days 6 hours, median 9 days 7 hours. Both mean and median with the application of filters decreased.

2. Is there any difference in throughput between national and international trips?

The table below shows the differences in throughput time between national and international trips, values are approximately recalculated to days for clarity.

Table 6 Throughput time

	Domestic trips		International trips	
	Mean	Median	Mean	Median
Unfiltered data	11.5	7.3	14.2	10.2
Filtered data	11.6	8	12.8	9.4
Filtered data based on variant analysis	11.2	8	12.3	9.3

As mean is not as robust to outliers as median, one could say that in case of international trips, the cases with long execution times are removed using filtering. Comparing the times of both types of declarations, on average, it takes longer to process international declaration from submission to paying than to process domestic declaration.

One may ask what the ratio of successful cases is in each declaration. Working on raw data, in domestic declarations 10044 cases were paid for, out of 10500 cases. In international declarations 6187 cases out of 6449 cases were successfully handled. Recounted to the percentage it is 95.66% of domestic declarations and 95.94% of international declarations. The remaining cases were either never submitted or always rejected. As can be seen, there is no significant difference between the percentages of successfully handled cases in both types of declarations.

In Disco software the two datasets were processed, filtering only on timestamp. Might be worth noticing the decrease in number of events during the summertime, especially in August. As also the approval time slightly increases it might be due to a holiday season.

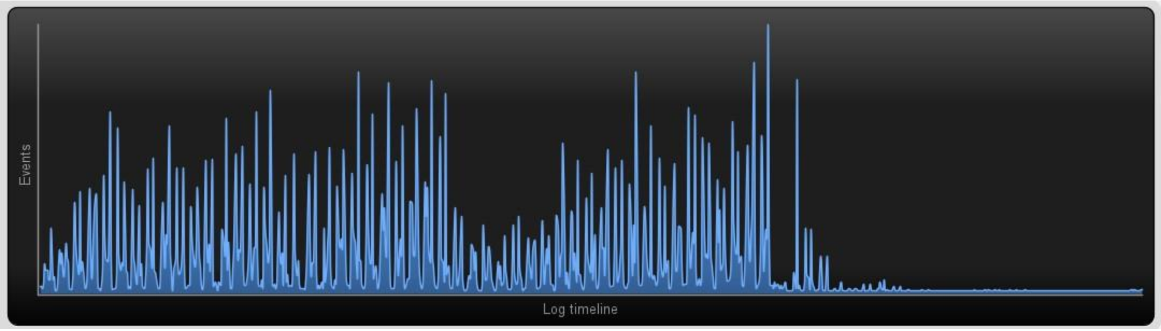


Figure 22 Disco, global statistics of domestic declarations

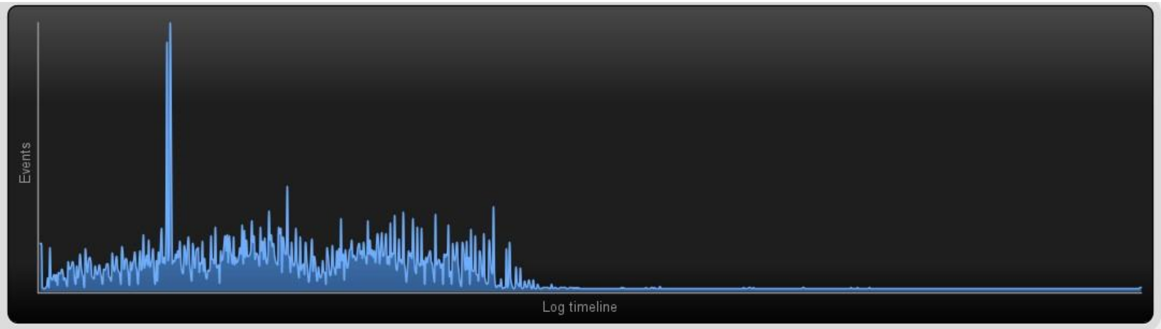


Figure 23 Disco, global statistics of international declarations

Observing the process models, specifically a Petri net obtained using Inductive miner for domestic declarations and a Petri net derived using Heuristic miner for international declarations. One can see the sequence of typical activities that contribute to successful handling of the cases. For both declaration procedures the sequence of activities is the same, just as it was indicated in the description of the process flows.

3. Where are the bottlenecks in the process of a travel declaration?

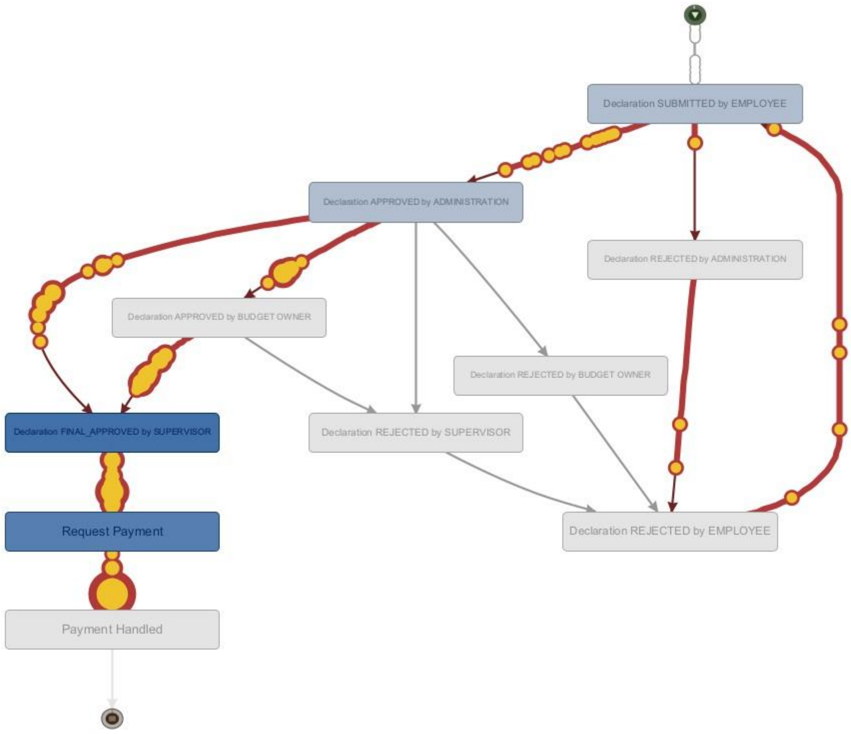

Declaration SUBMITTED by EMPLOYEE
Declaration APPROVED by ADMINISTRATION
Declaration REJECTED by ADMINISTRATION
Declaration APPROVED by BUDGET OWNER
Declaration REJECTED by BUDGET OWNER
Declaration FINAL_APPROVED by SUPERVISOR
Declaration REJECTED by SUPERVISOR
Request Payment
Declaration REJECTED by EMPLOYEE
Payment Handled

Figure 24 Bottlenecks, domestic declarations

Using Disco software on filtered domestic declarations dataset it was possible to visually identify the bottlenecks. A case is represented by yellow balls, the larger the number of balls the more cases are waiting for an activity to execute. It can be identified that the flow of the documents is probably not completely electronic, as for instance the activity "Request Payment" would be completed much faster. This might be one of the bottlenecks the TU/e wants to analyze in order to speed up the process. Observing the "Request Payment" and "Payment Handled" activities, I would like to know what the reason is for taking it so much time to execute these activities, as they are fully operated by SYSTEM, presumably without any human intervention. This might be for further analysis of the TU/e organization, as no more information is provided regarding this issue.

From the animation it can be seen that cases that after the activity "Declaration APPROVED by ADMINISTRATION" flow to "Declaration APPROVED by BUDGET OWNER" and then to "Declaration FINAL_APPROVED by SUPERVISOR" take much longer than those cases that don't undertake the "Declaration APPROVED by BUDGET OWNER" step. From the description of the challenge, it is apparent that supervisor and budget owner are sometimes the same people. As it takes less time to process the case if a budget owner and a supervisor are the same people, one might reconsider the job duties of these two people to improve the process.

To support the statement that cases not being processed by a budget owner take less time, in the following table the duration of cases were computed using filtered data in Python.

Table 7 Budget owner entity, domestic declarations

	With the activity "Declaration APPROVED by BUDGET OWNER"	Without the activity "Declaration APPROVED by BUDGET OWNER"
Number of cases	2760	5081
Mean	13.6	10.2
Median	10.3	7.2

The same table was constructed for international declarations, where the same bottlenecks were found.

Table 8 Budget owner entity, international declarations

	With the activity "Declaration APPROVED by BUDGET OWNER"	Without the activity "Declaration APPROVED by BUDGET OWNER"
Number of cases	954	1962
Mean	14.2	11.3
Median	12	7.9

4. How many travel declarations get rejected in the various processing steps and how many are never approved?

This question will be answered on data filtered by timestamp in view of the fact that in the time before the year 2018 the process wasn't fully established, and the results might not be of importance for the process that is now in operation. By using this filter, we reduced the number of activities from 17 to 12, reducing activities 'Declaration REJECTED by PRE-APPROVAL', 'Declaration REJECTED by missing', etc.

The table below shows the number of submitted declarations, the number of never approved cases, and the case frequency of different kinds of rejections.

	Domestic declarations	International declarations
Declaration submitted by employee	8160	4895
Number of cases never approved	257	154
Declaration rejected by employee	1063	1345
Declaration rejected by administrator	842	1260
Declaration rejected by supervisor	217	91
Declaration rejected by budget owner	58	39
Declaration rejected by director	-	1

Table 9 Rejection proportion

From the table 7 can be seen that the ratio of rejections is much higher for international declarations than for domestic declarations. It seems that it is much more difficult to correctly fill in the forms for declaring international trips.

5. How many travel declarations are booked on projects?

It was unfeasible to find if a domestic declaration is booked on a project, since no attribute in the domestic declarations' dataset contains such values. However, in international declarations dataset is an attribute 'Permit ProjectNumber' that can be considered the right attribute to look at when answering this question.

In Python, filtering was used on the raw data that are recording the international declarations. The number of cases that were declared and had the project number is 4075, the remaining 2300 cases are not tied to any projects and the value is 'UNKNOWN'. The project n. 426 is booked by highest number of cases: 454. Next in the order of frequency is project n. 3442 with 52 cases and project n. 8761 with 47 cases.

6. How many corrections have been made for declarations?

Corrections might be defined on cases that were successfully handled in the end (containing activity 'Payment handled') and that happened to be at least once rejected. Timestamp filtering will be just as in question 4 performed on the datasets.

In Disco software for domestic declarations there are 40 variants and 7903 cases ending with activity 'Payment handled'. Out of 40 variants 36 variants include at least one rejection, there 36 variants are followed by 815 cases, which is approximately 10% out of the total number of cases. Most of these cases were rejected 1-2 times, the rest was resubmitted even 5-6 times.

7. Are there any double payments?

For each declaration number only one execution of payment handling was observed.

Conclusions

In this project the analysis of the tenth International Business process intelligence process challenge was presented. Focus was given on questions proposed by the organizers, especially on questions regarding the domestic and international declarations, as these two event logs contained interesting data for further analysis. Answering the questions of interest might be of importance for the TU/e organization to put more emphasis on highlighted issues.

To recapitulate the analysis and its most important findings.

Conducting filtering, process discovery and conformance checking on both domestic and international declarations the following miners provided the best process models. For domestic declarations it was the inductive miner and for international declarations the heuristic miner.

International declarations had more different traces than domestic declarations. The throughput time for international declarations is longer, perhaps because the rejection rate is higher and many declarations need to be resubmitted. Both in domestic declarations and international declarations the major bottlenecks were found, one of which is related to the approval of budget owner. Having the declarations approved by a budget owner, who is not a supervisor at the same time, adds additional time.

Some recommendations on how to make the process more efficient were offered.

References

[1] B. Challenge, 23 March 2020. [Online]. Available: https://www.tf-pm.org/competitions-awards/bpi-challenge/2020.

[2] C. W. G. a. E. Verbeek, 2014. [Online]. Available: https://pure.tue.nl/ws/portalfiles/portal/3981980/692728941269079.pdf.

[3] CS Disco, "Fluxicon Disco," 2012.

[4] "PM4PY library documentation," [Online]. Available: https://pm4py.fit.fraunhofer.de/.

[5] W. v. d. Aalst, "Process Mining: Data science in Action," [Online]. Available: https://www.coursera.org/learn/process-mining.

Source code: https://github.com/Nela-B/Project