BPIC Challenge 2012

Alireza Zarghamnezhad

Business Information Systems

Prof. Paolo Ceravolo

a.a. 2021 – 2022

Table of Contents

Case Study Description

A financial organization located in the Netherlands that deals with loan and overdraft approval processes offers online access to its customers who can submit their applications through a website and provide additional details via phone. After the application is submitted, automated evaluations are performed. If the customer is deemed eligible, they will receive an offer by email, otherwise, the financial institution will get in touch with the customer regarding any incorrect information. Finally, a comprehensive assessment is done after the application is accepted.

The Second International Business Processing Challenge provides participants with a real event log and encourages them to analyze the data using any available techniques. Participants have the option to either:

1. Concentrate on a specific aspect of interest and delve into it thoroughly, for instance, control-flow models, social network models, performance models, or predictive models.
2. Cover a wider range of aspects without having to delve into detail. The report submitted under this category will be judged based on its comprehensiveness of analysis and its practical value for real-life business improvement.

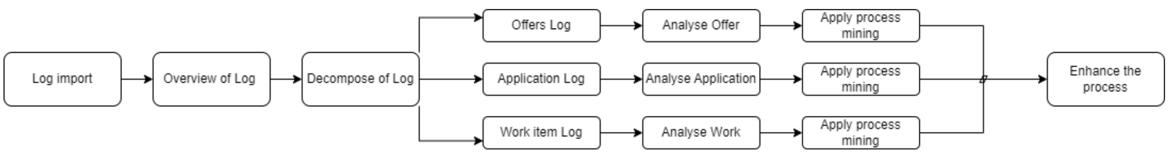
The event log provided to participants is anonymized and formatted in XES, with a size of 3.3MB. It is a real-life log collected from a financial institute based in the Netherlands that handles loan and overdraft approval processes. The log consists of 262,200 events and 13,087 cases, each of which contains a widespread case attribute, "AMOUNT-REQ", which represents the amount requested by the customer.

The event log is a combination of three sub-processes that can be easily distinguished by the first letter of each task. The first sub-process, represented by event types starting with "A_" (Application Events), depicts the states of the application events. The second sub-process, represented by states starting with "O_" (Offer Events), indicates the states of the offer events that belong to the application. Finally, the third sub-process, represented by states starting with "W_" (Work Item Events), depicts the states of the work item events that are related to the application.

Each sub-process is a critical component of the overall loan and overdraft approval process, and the ability to understand and analyze them is essential for improving the efficiency and effectiveness of the process. The use of different sub-processes, as well as the case attribute "AMOUNT-REQ", provides a comprehensive view of the loan and overdraft approval process, allowing participants to gain insights into the various stages of the process and identify areas for improvement.

Goals

To gain a deeper understanding of the Dutch Financial Institute's loan and overdraft approval process, a more in-depth analysis of the event log should be performed. This can be done by first examining the overview of the whole log and each subprocess. Then, the business process can be thoroughly understood by applying process mining techniques such as variant analysis, filtering, process discovery techniques, and conformance checking to the event log and breaking it down into its subprocesses. Through this detailed analysis, the process can be explored in greater depth, and by identifying any issues and redundancies, the organization can potentially improve the overall process in terms of both cost and time efficiency.



Knowledge Uplift Trail

The steps described are a process for analyzing an event log to uncover new process models. The steps are as follows:

1. File Extraction: The first step involves extracting the file that contains the event log data.
2. Decompose to Sub-logs: In this step, the event log is decomposed into smaller sub-logs for easier analysis.
3. Variant Analysis: In this step, a variant analysis is conducted on the sub-logs to uncover patterns and correlations in the data.
4. Filtering: To further refine the data, a filtering step is applied to the data frame to only include relevant data.
5. Process Discovery: In this step, process discovery algorithms are applied to the filtered log to uncover new process models.
6. Conformance Checking: The final step involves conducting conformance checks on the process models, using algorithms such as token-replay or alignment, to assess their quality and conformance. The output of this step is a set of metrics that provide an assessment of the models.

Steps	Input	Analytic	Model	Output
Step 1	File	Extraction	Descriptive	Event log
Step 2	Event Log	Decompose to sub logs	Descriptive	Sub Logs
Step 3	Sub Logs	Variant Analysis	Descriptive	Variant
Step 4	Data frame	Filtering	Descriptive	Filtered log
Step 5	Filtered log	Process Discovery	Prescriptive	Model
Step 6	Model	Conformance checking	Prescriptive	Metrics assessment

Overview of Log

The original log, prior to being decomposed, was a complicated set of data that made it difficult to identify bottlenecks and areas of rework. However, it was apparent that there was an excessive amount of rework in specific work items. To provide further insight, the table below lists all the activities in the work item sub-log, along with their corresponding English translations, as they were not originally in the English language, and their respective descriptions.

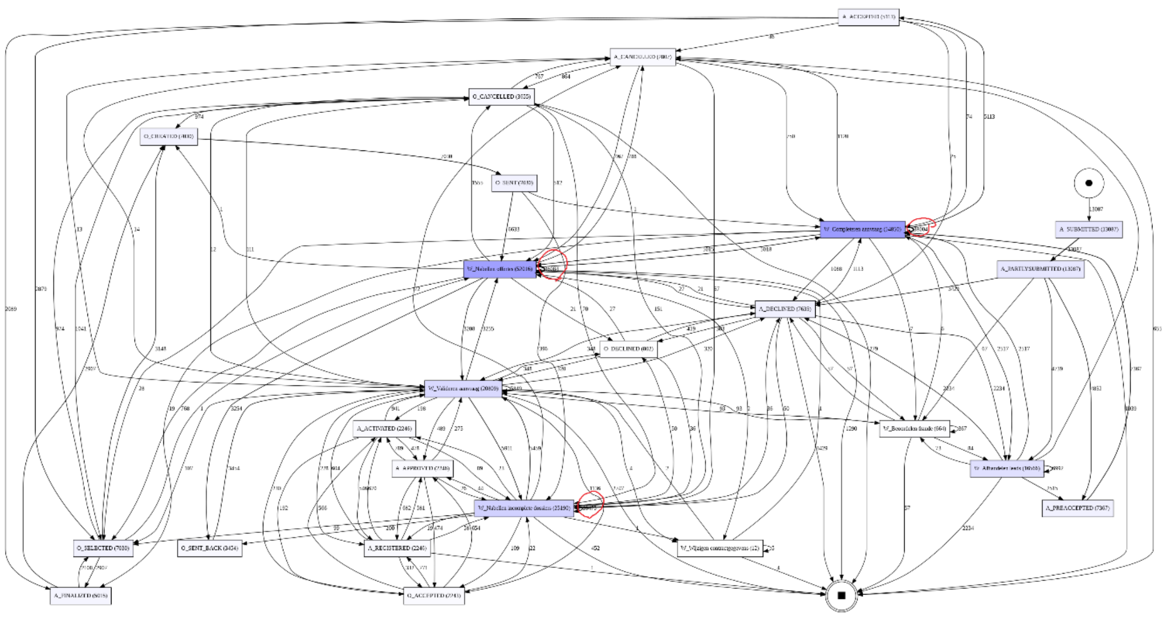
Activity	Description
W_Completeren aanvraag	W_Filling in information for the application
W_Afhandelen leads	W_Fixing incoming lead
W_Nabellen offertes	W_Calling after sent offers
W_Beoordelen fraude	W_Assess fraud
W_Valideren aanvraag	W_Assessing the application
W_Nabellen incomplete dossiers	W_Calling to add missing information to the application
W_Wijzigen contractgegevens	W_Change contract details

The table below illustrates all of the activities recorded in the log, including the number of times each activity occurred.

Activity	Count
A_SUBMITTED	13087
A_PARTLYSUBMITTED	13087
A_PREACCEPTED	7367

W_Completeren aanvraag	54850
A_ACCEPTED	5113
O_SELECTED	7030
A_FINALIZED	5015
O_CREATED	7030
O_SENT	7030
W_Nabellen offertes	52016
O_SENT_BACK	3454
W_Valideren aanvraag	20809
A_REGISTERED	2246
A_APPROVED	2246
O_ACCEPTED	2243
A_ACTIVATED	2246
O_CANCELLED	3655
W_Wijzigen contractgegevens	12
A_DECLINED	7635
A_CANCELLED	2807
W_Afhandelen leads	16566
O_DECLINED	802
W_Nabellen incomplete dossiers	25190
W_Beoordelen fraude	664

And we utilized pm4py to extract the direct-follow graph of the process, which is presented in below:



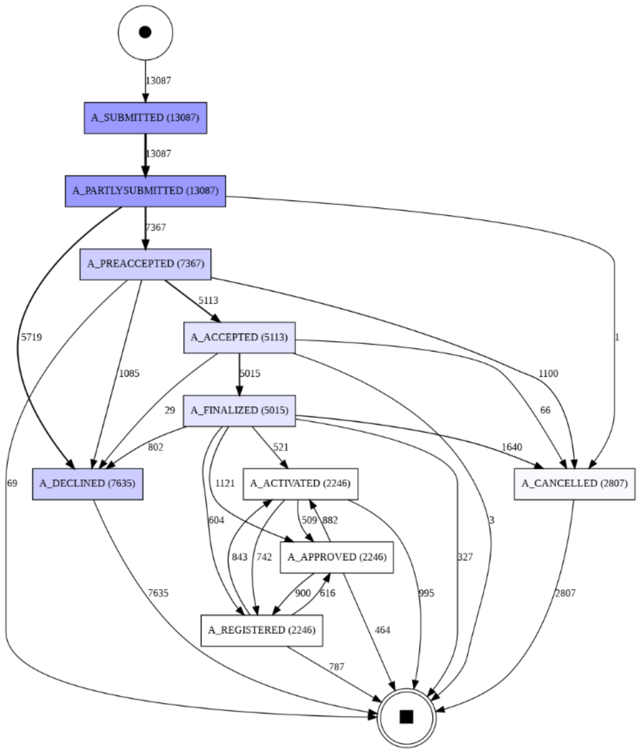
It is evident from the direct-follow graph (DFG) generated by pm4py that there is a significant amount of rework for three specific work items: *W_Completeren aanvraag*, *W_Nabellen incomplete dossiers*, and *W_Nabellen offertes*. This is further highlighted by the data in the table, which shows the number of occurrences of each event in the log. We can see later each of the subprocess in more detail and by analyzing the direct-follow graph of the process, we can gain a deeper understanding of the bottlenecks and rework that occur within the subprocesses. With this information, we will be able to identify areas for improvement and make more informed decisions to optimize the overall process.

Decomposition

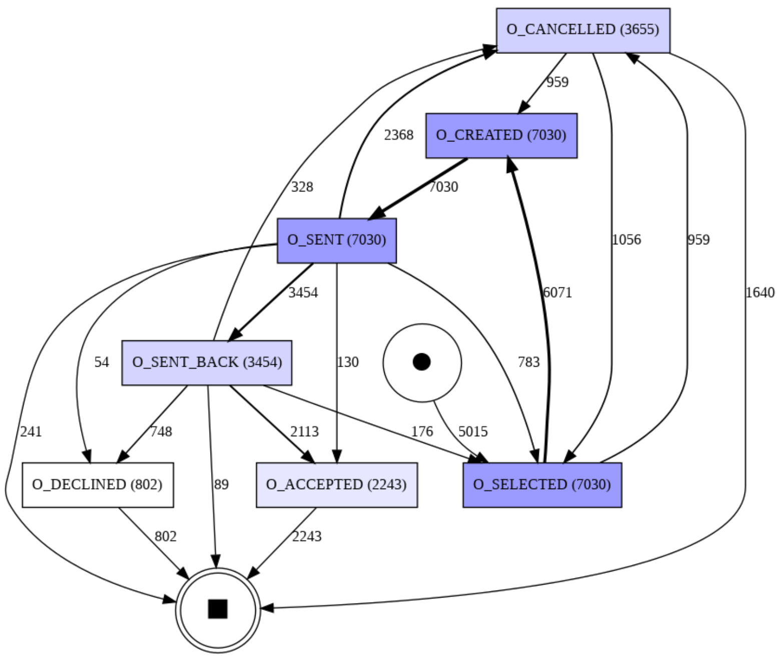
By utilizing the power of Pandas, we have successfully decomposed the event log into three distinct sub-process logs, each of which specifically deals with either the offer, application, or work item aspects of the loan and overdraft approval process. With this separation, we can delve deeper into the analysis of rework and potential bottlenecks, and leverage process mining techniques to their fullest potential. This will ultimately lead to a more comprehensive and in-depth analysis that can inform business improvements and streamline the overall process for better efficiency, cost-effectiveness, and speed.

After decomposition the DFG for each sub-process has been depicted bellow.

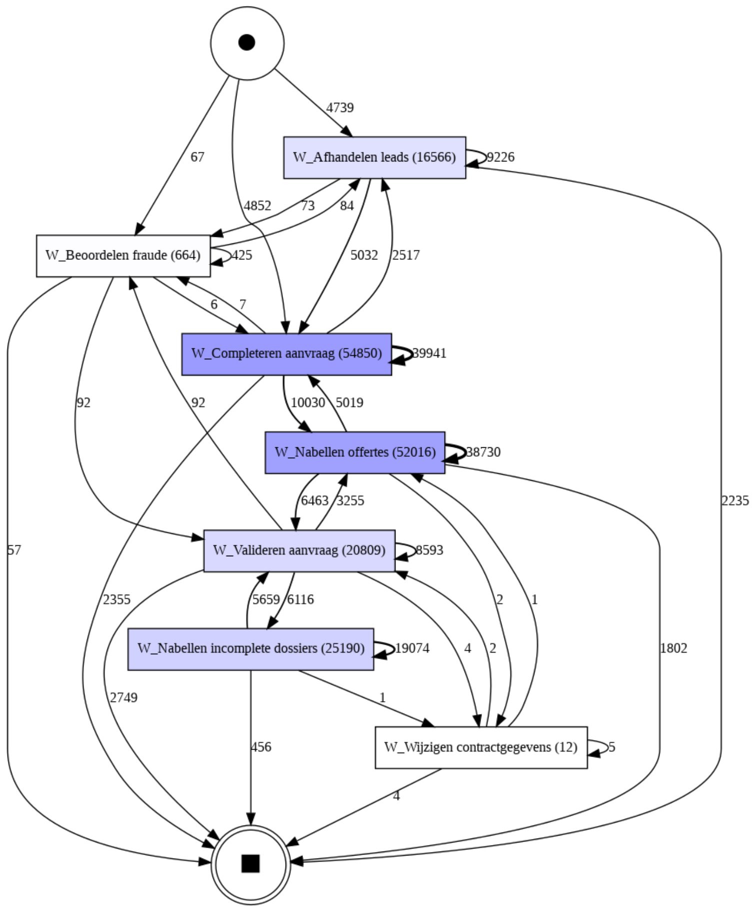
For applications log:



For Offers log:



For work items log:



Also, we extract from the logs that all the cases started with A_SUBMITTED and followed by A_PARTLYSUBMITTED.

Analysis

The function analyze_process has been implemented to provide a comprehensive analysis of each process. The function will print the start date and end date of the first and last activities in the event log, along with the unique number of activity types and resources used. Additionally, it will provide the total number of cases and events in the log, as well as the mean duration of the process. The mean duration is calculated both with and without considering events with a duration of 0 days. The output of the function is presented below.

The result for Application sub-process:

```
Start Timestamp: 2011-09-30 22:38:44.546000+00:00
End Timestamp: 2012-03-14 14:33:57.651000+00:00
Number of Activity Types: 10
Number of Resources: 61
Number of Cases: 13087
Number of Events: 60849
Mean Duration (excluding 0 day filtered): 8 days 01:55:14.860649805
Mean Duration (0 day filtered): 17 days 13:08:00.343700820
```

The result for Offer sub-process:

```
Start Timestamp: 2011-10-01 08:44:40.725000+00:00
End Timestamp: 2012-03-14 14:50:59.683000+00:00
Number of Activity Types: 7
Number of Resources: 60
Number of Cases: 5015
Number of Events: 31244
Mean Duration (excluding 0 day filtered): 17 days 04:20:24.846146959
Mean Duration (0 day filtered): 18 days 00:27:56.279726417
```

The result for Work Item sub-process:

```
Start Timestamp: 2011-09-30 22:39:38.875000+00:00
End Timestamp: 2012-03-14 15:04:54.681000+00:00
Number of Activity Types: 7
Number of Resources: 59
Number of Cases: 9658
Number of Events: 170107
Mean Duration (excluding 0 day filtered): 11 days 16:26:31.640076517
Mean Duration (0 day filtered): 17 days 20:54:58.011957562
```

Process Mining

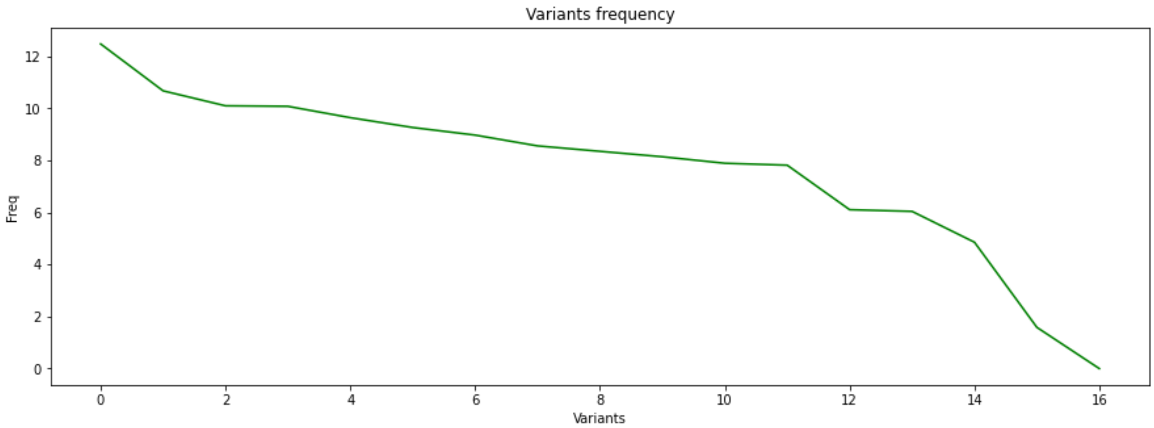
Process mining involves using data from event logs to understand and depict the flow of processes. It combines aspects of data science, such as process discovery and process conformance analysis, to uncover inefficiencies and deviations in a process. The purpose of process mining is to gain a comprehensive and impartial view of a process in order to make improvements in efficiency, lower costs, and enhance customer satisfaction.

For each sub-process, the process mining techniques like variant analysis and process discovery and conformance checking were applied later.

Application Log

Variant Analysis

The frequency distribution of the 17 variants in the Application log is displayed below, which is considered to be a satisfactory number of variants in this log.



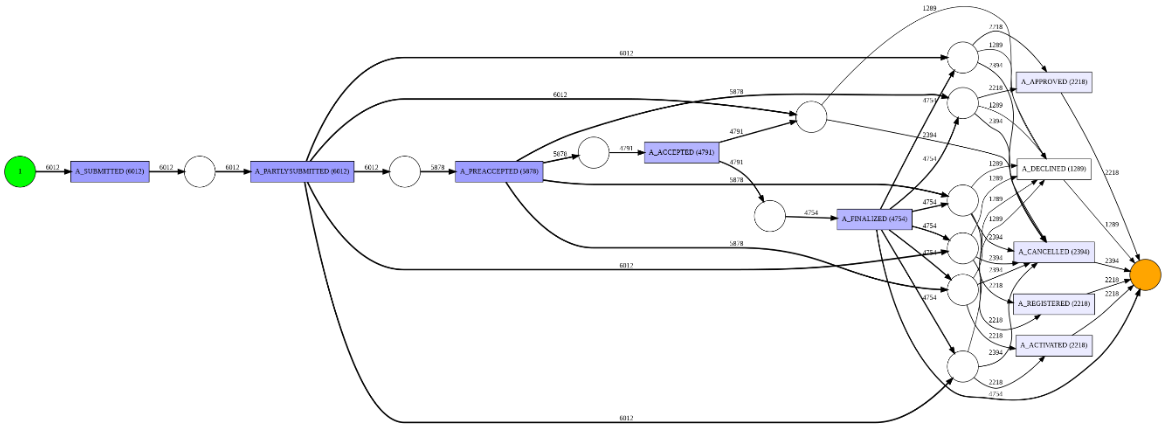
Filtering

The frequency distribution of the 17 variants in the Application log is shown below, which is deemed to be a suitable number of variants in this log. A filter was applied to only include cases with a duration greater than 1 day, resulting in a decrease in the number of cases from 13087 to 6012.

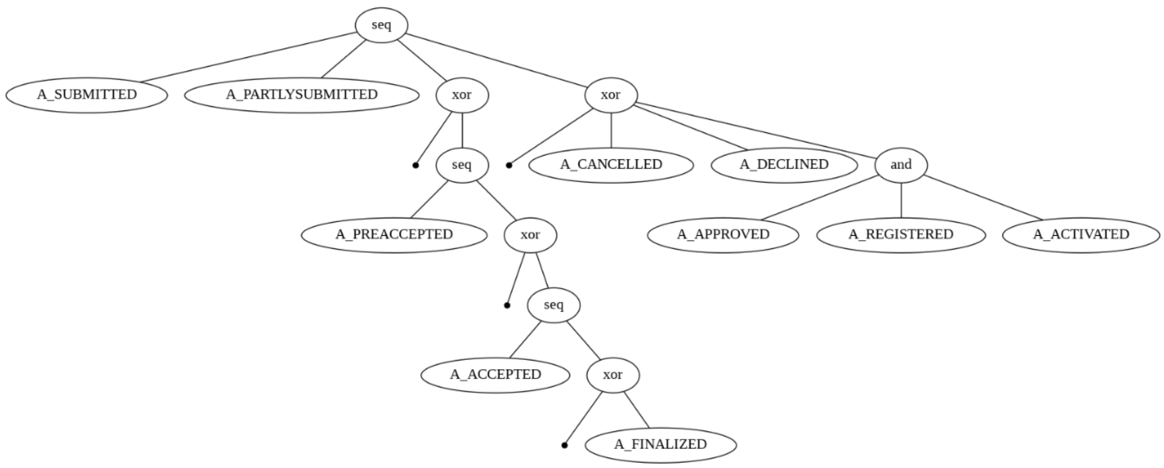
Process Discovery

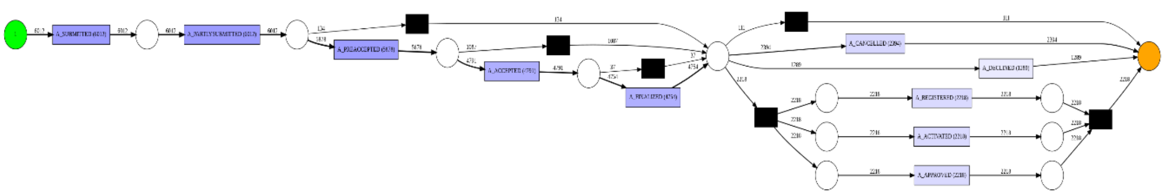
The process discovery techniques were applied using various algorithms such as alpha-miner, inductive miner, and heuristic miner, and the resulting visualization of the model was presented using Petri-Net, Directly-Follows Graph (DFG), and process tree representations.

The following illustration showcases the application of the alpha-miner algorithm on application, demonstrated through its representation in a Petri net created with Pm4Py functions. This algorithm allows for the detection of parallel activities within the process.

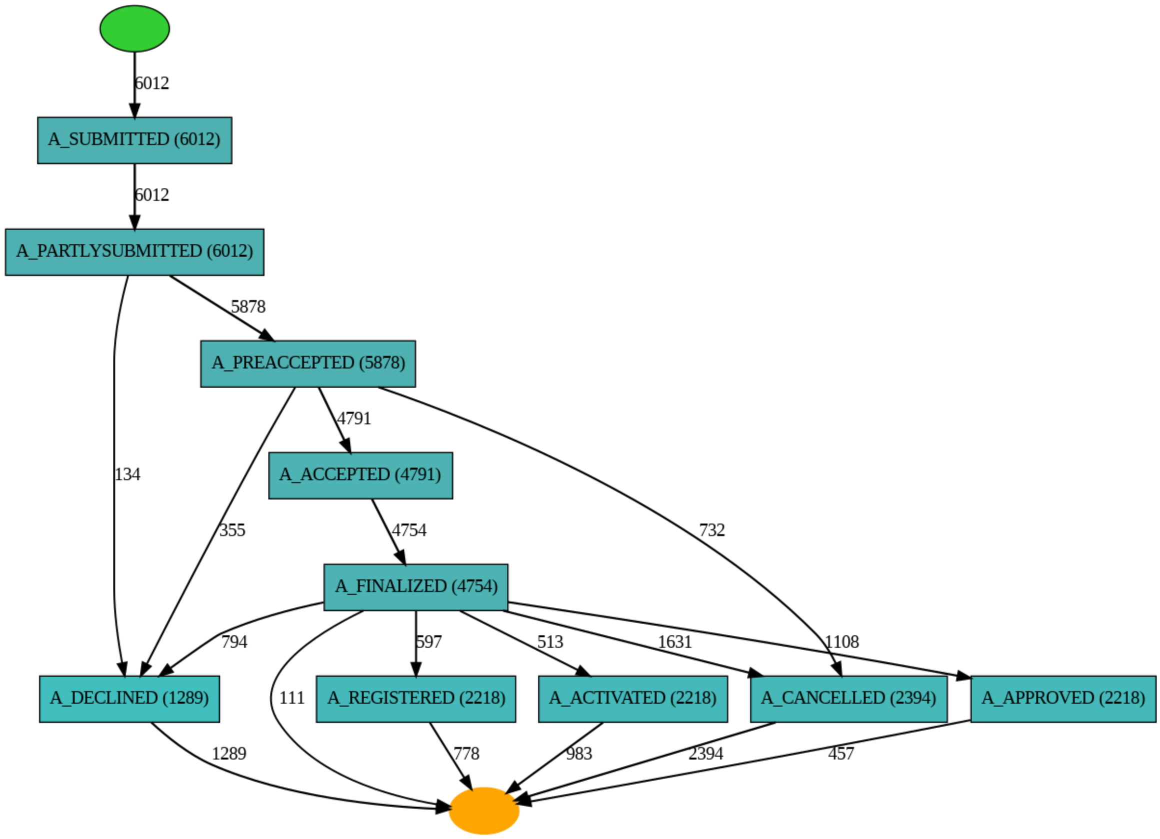


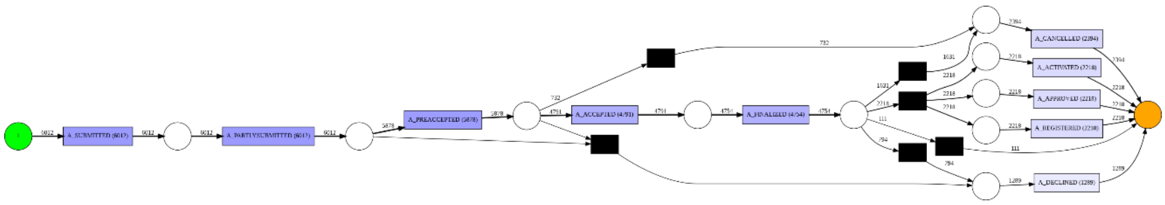
The Pm4Py documentation mentions that the Inductive Miner (IM) algorithm is designed to detect cuts in the log, such as sequential, parallel, concurrent, and loop cuts. The IM algorithm then operates recursively on the sub-logs obtained by applying the cuts until a base case is reached. The visualization of the Inductive Miner applied to the application is presented below.





The Heuristic Miner (HM) algorithm was also applied using Pm4Py functions, which operates on Directly-Follows Graphs (DFG). The output of the HM algorithm is a Heuristic Net, and its visualization was created to show the sub processes.





Quality Metrics

The evaluation of the performance of each process discovery model (Alpha-miner, Inductive miner, Heuristic miner) was conducted by using quality metrics such as Precision, Fitness, Generalization, and Simplicity, which were implemented through Pm4Py functions.

Model	Fitness	Precision	Generalization	Simplicity
AM	0.81	0.62	0.98	0.30
IM	1.0	0.60	0.96	0.67
HM	0.95	1.0	0.91	0.66

Conformance Checking

On heuristic the percentage of anomalous traces was 70% while in inductive they are 0%.

On Heuristic Threshold 0.90%

```
replaying log with TBR, completed variants :: 100%  17/17 [00:00<00:00, 114.27it/s]
REPLAY
Number of traces 13087
173688, 173691, 173694, 173697, 173700, 173706, 173718, 173724, 173727, 173730, 173733, 173736, 173739, 173742, 173751, 17
Number of anomalous traces  9218
Percentage of anomalous traces 70.43631084282111 %
aligning log, completed variants :: 100%  17/17 [00:00<00:00, 84.43it/s]
ALIGNMENTS
Number of traces 13087
['173688', '173691', '173694', '173697', '173700', '173706', '173718', '173724', '173727', '173730', '173733', '173736', '1
Number of anomalous traces  9218
Percentage of anomalous traces 70.43631084282111 %
```

On Inductive Miner

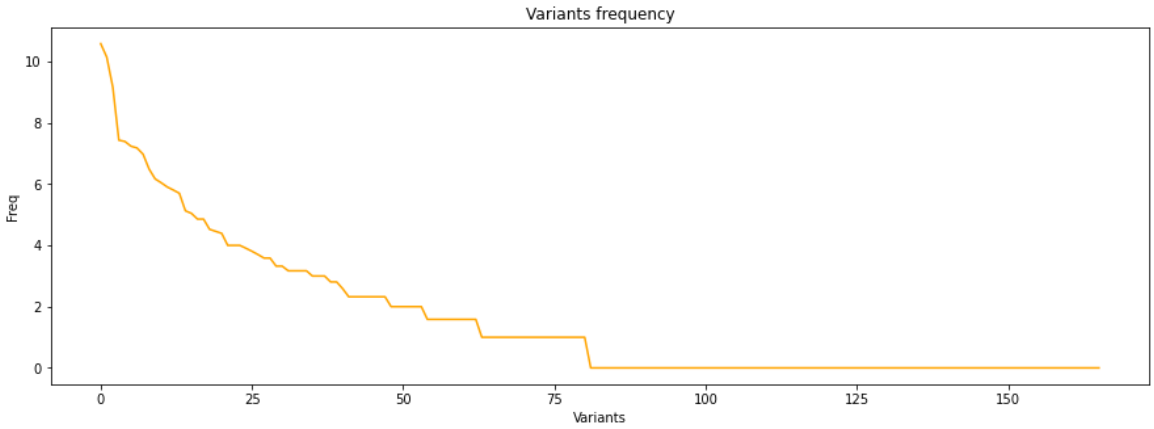
```
replaying log with TBR, completed variants :: 100%  17/17 [00:00<00:00, 108.97it/s]
REPLAY
Number of traces 13087

Number of anomalous traces  0
Percentage of anomalous traces 0.0 %
aligning log, completed variants :: 100%  17/17 [00:00<00:00, 147.34it/s]
ALIGNMENTS
Number of traces 13087
[]
Number of anomalous traces  0
Percentage of anomalous traces 0.0 %
```

Offer Log

Variant Analysis

The frequency distribution of the 166 variants in the Offer log is displayed below:



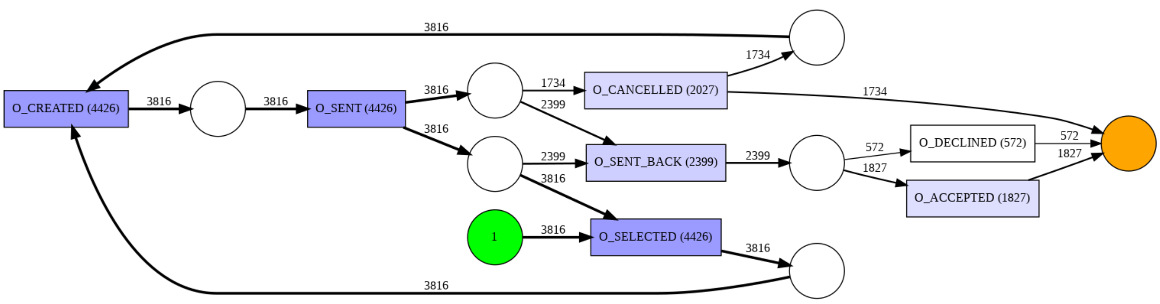
Filtering

The filtering was carried out on the top 8 variants, and it only included cases where the offer duration was greater than 1 day. Cases with a duration less than a day were not considered. This filtering resulted in a decrease of the number of cases from 5015 to 3816.

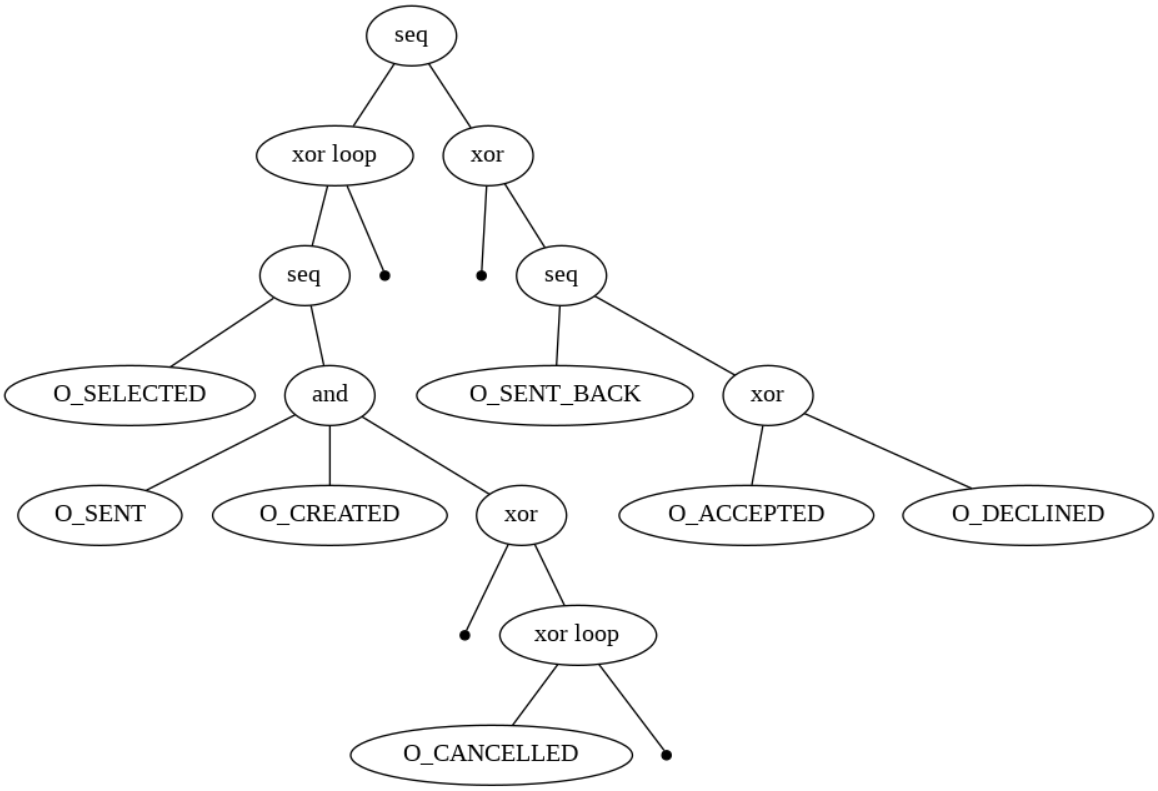
Process Discovery

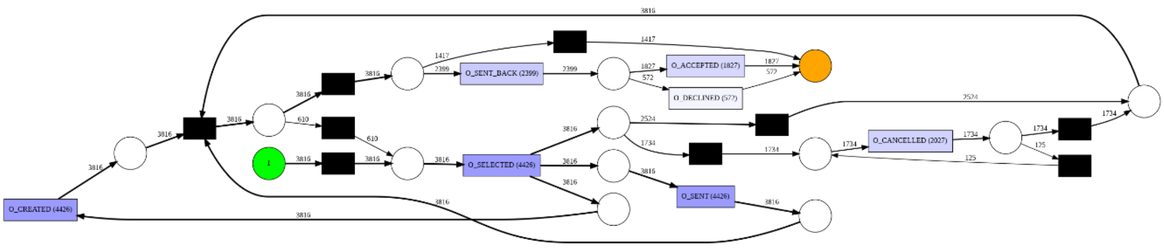
The techniques of process discovery have also been applied to the Application process, and the results are displayed below.

Alpha Miner:

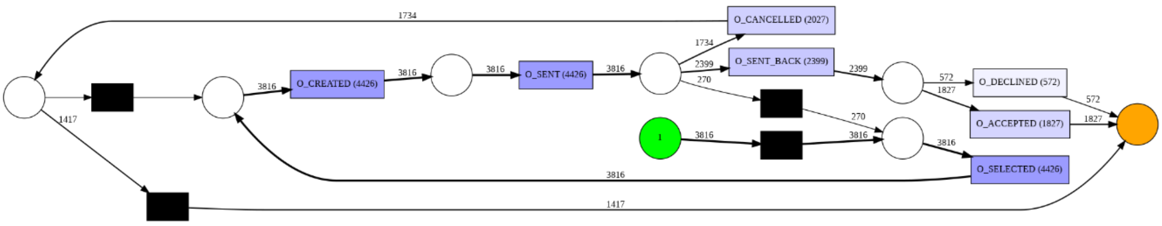


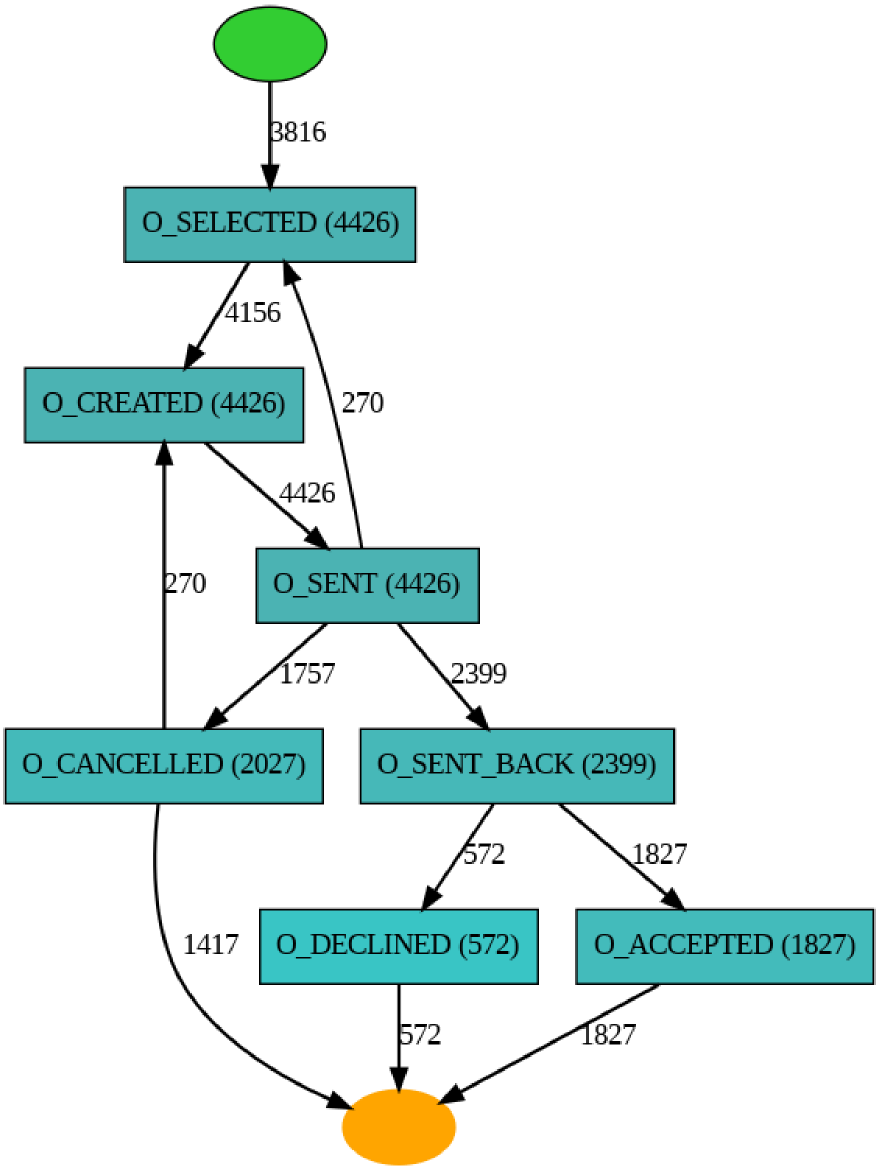
Inductive Miner:





Heuristic Miner:





Quality Metrics

Model	Fitness	Precision	Generalization	Simplicity
AM	0.82	1.0	0.97	0.65
IM	1.0	0.66	0.97	0.71
HM	0.98	0.97	0.88	0.75

Conformance Checking

On Alpha-Miner:

```
replaying log with TBR, completed variants :: 100% 166/166 [00:00<00:00, 481.72it/s]
REPLAY
Number of traces 5015
173736, 173748, 173880, 174096, 174036, 174132, 174168, 174337, 174358, 174382, 174761, 174815, 174872, 175015, 174944, 174764, 174
Number of anomalous traces  623
Percentage of anomalous traces 12.422731804586242 %
aligning log, completed variants :: 100% 166/166 [00:06<00:00, 30.27it/s]
ALIGNMENTS
Number of traces 5015
['173736', '173748', '173880', '174096', '174036', '174132', '174168', '174337', '174358', '174382', '174761', '174815', '174872',
Number of anomalous traces  623
Percentage of anomalous traces 12.422731804586242 %
```

On Inductive Miner:

```
replaying log with TBR, completed variants :: 100% 166/166 [00:00<00:00, 366.31it/s]
REPLAY
Number of traces 5015
173718, 173691, 173736, 173748, 173787, 173817, 173694, 173880, 174060, 174012, 173928, 174096, 174141, 174
Number of anomalous traces  1772
Percentage of anomalous traces 35.33399800598205 %
aligning log, completed variants :: 100% 166/166 [00:03<00:00, 46.11it/s]
ALIGNMENTS
Number of traces 5015
['173718', '173691', '173736', '173748', '173787', '173817', '173694', '173880', '174060', '174012', '17392
Number of anomalous traces  1772
Percentage of anomalous traces 35.33399800598205 %
```

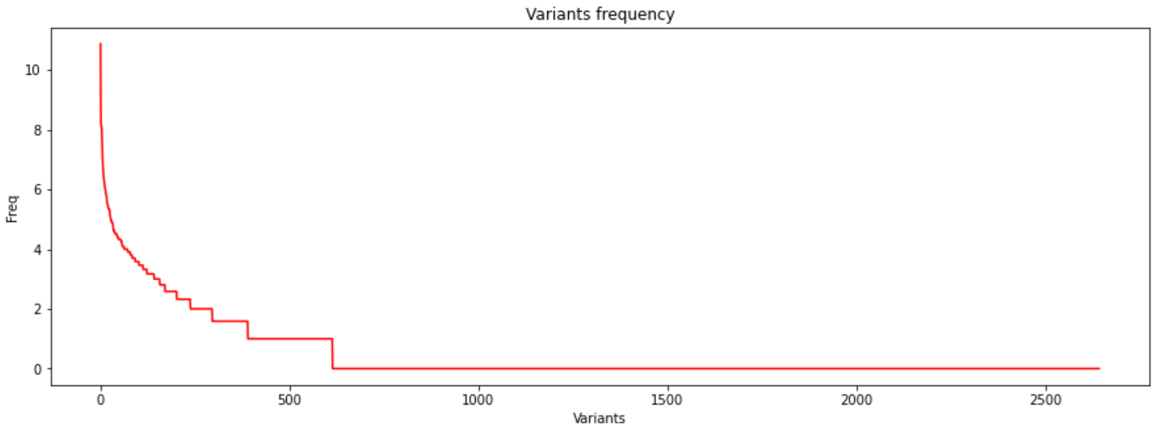
On heuristic:

```
replaying log with TBR, completed variants :: 100% 166/166 [00:00<00:00, 630.74it/s]
REPLAY
Number of traces 5015
173718, 173691, 173736, 173748, 173787, 173817, 173694, 173880, 174060, 174012, 173928, 174096, 174141, 174069, 174036, 174132,
Number of anomalous traces  1772
Percentage of anomalous traces 35.33399800598205 %
aligning log, completed variants :: 100% 166/166 [00:04<00:00, 31.92it/s]
ALIGNMENTS
Number of traces 5015
['173718', '173691', '173736', '173748', '173787', '173817', '173694', '173880', '174060', '174012', '173928', '174096', '17414
Number of anomalous traces  1772
Percentage of anomalous traces 35.33399800598205 %
```

Work Item Log

Variant Analysis

The log holds a large number of variants, totaling 2643, which is considered a significant amount and suggests the presence of rework. Upon examining the diagram, it becomes evident that the vast majority of these variants consist of cases with unique trace events. This high number of variants may require a revision of the current processes in order to address the rework and improve efficiency.

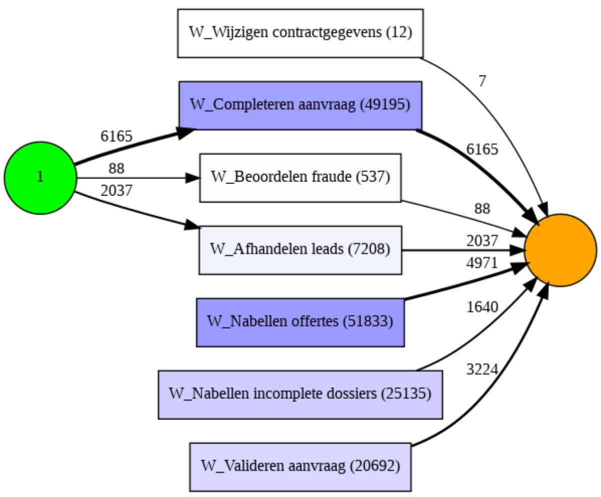


Filtering

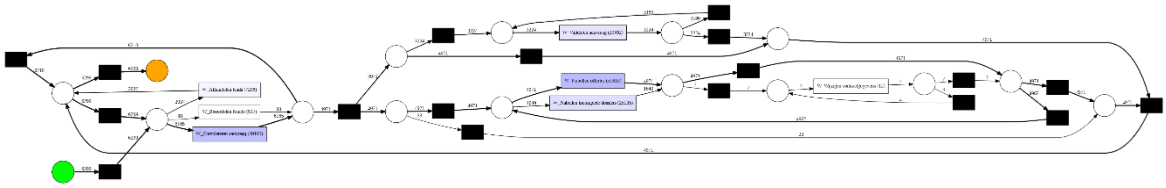
Due to the high degree of rework in this log, filters were applied to only include cases that took more than one day to complete. which reduces the number of cases from 9658 to 6299.

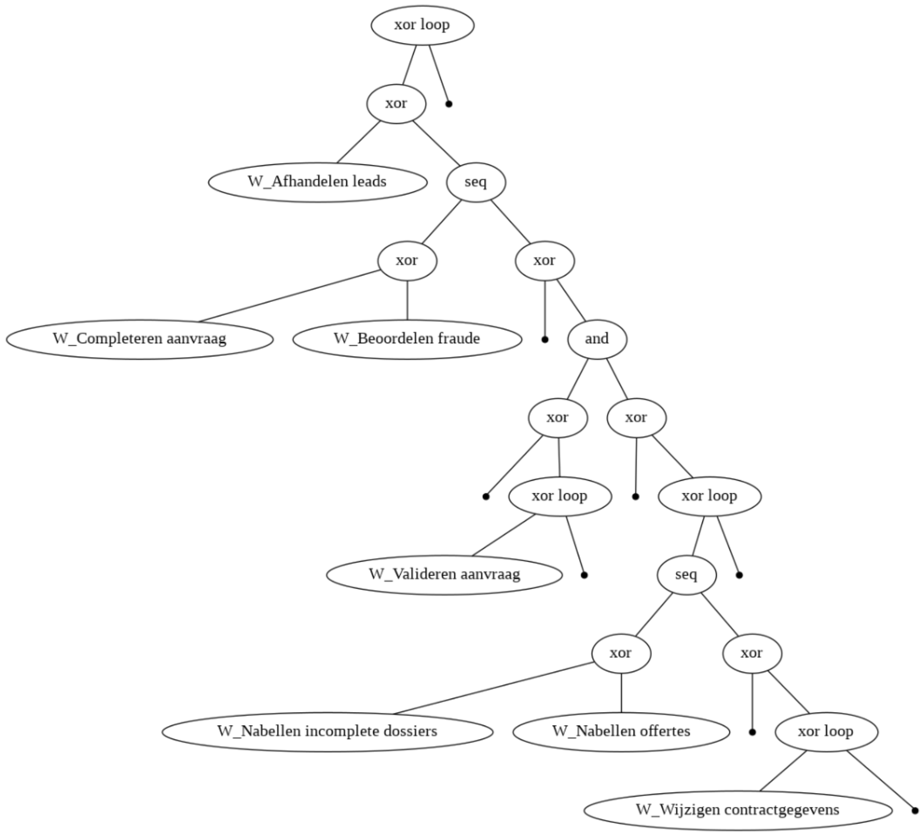
Process Discovery

Alpha Miner:

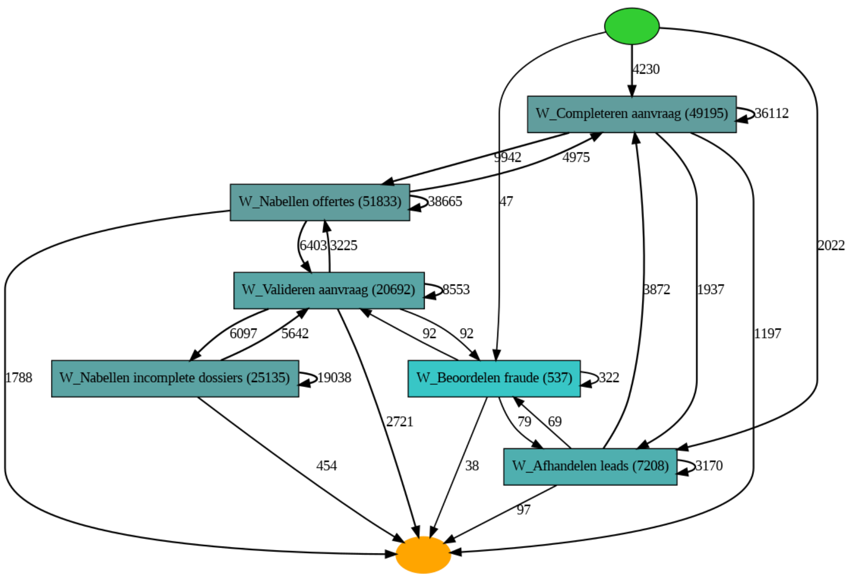


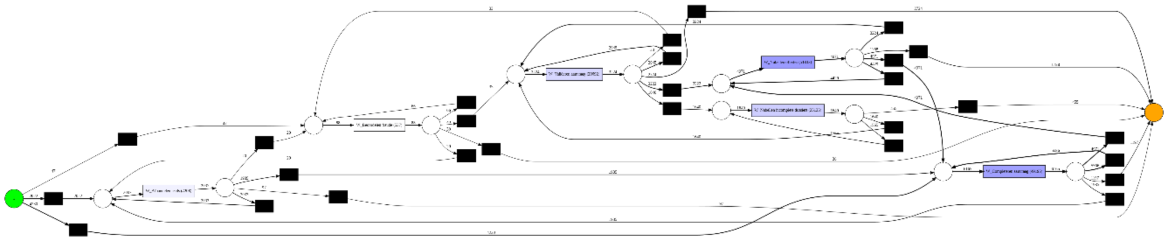
Inductive Miner:





Heuristic Miner:





Quality Metrics

Model	Fitness	Precision	Generalization	Simplicity
AM	0.20	0.27	0.94	0.81
IM	1.00	0.28	0.92	0.65
HM	0.99	0.52	0.96	0.55

Conformance Checking

On Heuristic Miner:

```
replaying log with TBR, completed variants :: 100% 2643/2643 [00:15<00:00, 215.42it/s]
REPLAY
Number of traces 9658
173754, 177206, 195082
Number of anomalous traces  3
Percentage of anomalous traces 0.031062331745703043 %
aligning log, completed variants :: 100% 2643/2643 [01:40<00:00, 47.36it/s]
ALIGNMENTS
Number of traces 9658
['173694', '173754', '175756', '177206', '178843', '181447', '191464', '191737', '195082', '204523']
Number of anomalous traces  10
Percentage of anomalous traces 0.10354110581901013 %
```

Conclusions

To provide a more comprehensive overview, we first conducted an analysis of the financial log of a Dutch institute. This log was created by merging three sub-logs, which were then decomposed and subjected to further examination. During our analysis, we looked at crucial data for each of the sub-logs and made a determination to apply filters to each log based on metrics such as the duration of the cases. Our goal was to uncover new models through the use of algorithms like alpha miner, inductive miner, and heuristic miner, and to represent these models through pertinent, DFG, and process tree formats.

We then compared the quality metrics of these models, including fitness, precision, simplicity, and generalization, and conducted conformance checks using algorithms such as token-replay and alignment. The results of these evaluations allowed us to make an assessment of the quality and conformance of each model.

One sub-log in particular, the work item sub-log, was found to have a high degree of rework, which could be addressed through improved filtering. To that end, we noted the availability of the new rework filter in the PM4Py library, which could be used to identify a better process for this sub-log. In conclusion, our analysis highlights the importance of filtering and the use of appropriate algorithms and metrics to uncover new models and improve processes.

https://github.com/Alireza-Zarghamnezhad/BIS2023.git