BPIC 2012

Seyedalireza Mirzaghavami

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

Description of Case Study

A financial institute in Netherlands which handles the loan and overdraft approval processes is provided online facilities where customer applications can be submitted through a webpage, additional information will be given from the customer by phone.

After application complementation, some automatic checks are performed, in case of customer become eligible, offers is sent via email, otherwise they will contact the customers about the misleading information. After application approval final assessment is done.

Second international business processing challenge[1] provides participants with a real-life event log and asks them to analyze these data using whatever techniques available and document them in one of two ways:

1) The participants can focus on a specific aspect of interest and analyze this aspect in detail. Here, one can choose for example to focus on specific models, such as control-flow models, social network models, performance models, predictive models.
2) The participants may report on a broader range of aspects, where each aspect does not have to be developed in full detail. The report submitted in this category will be judged on its completeness of analysis and usefulness for the purpose of a real-life business improvement setting.

Data and Log Description

The event log is given with anonymization in two different formats: XES and MXML with the size of 3.3MB and 5.3MB, respectively.

The log is a real-life event log from a Dutch Financial Institute which handles the loan and overdraft approvals process.

The event log consists of 262,200 events and 13,087 cases (Table 1). A widespread case attribute AMOUNT-REQ is the amount which is requested by the customers. Every case consists of this case attribute.

Table 1. Event Log Description

Data	#Events	#Cases	Size
financial_log.xes	262,200	13,087	70.6 MB
financial_log.mxml	262,200	13,087	140 MB

[1] Full description at https://www.win.tue.nl/bpi/doku.php?id=2012:challenge

The event log is a combination of three sub-processes which their source can be identified by the first letter of each task. The sub processes can be distinguishable from their first letters.

Table 2 illustrates the abstraction of three types of events.

Table 2. Abstraction of Three Event Types

Event Type	Meaning
States starting with 'A_' (Application Events)	States of the application events
States starting with 'O_' (Offer Events)	States of the offer events belonging to the application
States starting with 'W_' (Work Item Events)	States of the work item events belonging to the application

Generally, each event consists of a time with a resource and a life cycle transition.

Organizational Objectives

By understanding the business process in detail and apply process mining techniques such as variant analysis, filtering, process discovery techniques, and conformance checking to the event log and decompose them by the sub processes and analyze each one, the process could be analyzed in more details, then by identifying the issues and redundancy of the process, the organization might be used to improve the overall process in terms of cost and time.

Knowledge Uplift Trail

Table 3. Knowledge Uplift Trail

	Input	Acquired Knowledge		Output
		Analytics/Models	**Type**	
Step 1	Through a webpage application submits	Automatic checks	Descriptive	Submitted Application
Step 2	a) step 1 b) employee calls for additional information	Conversation	Descriptive	Complemented application
Step 3	a) step 2 b) check eligibility	Information matching	Descriptive	a) Eligible client contacted b) Offer created
Step 4	Find the client email	List match	Prescriptive	Offer sent by email
Step 5	Offer received by client	In-form information checks	Prescriptive	Registered application

Event Log Analyzing

After importing the event log using Pm4Py, the complete event types were extracted which are counted by 24 unique types, are depicted with their relative major types.

Table 4 shows complete event types for application events and offer events while Table 5 depicts event types for work item events.

The process event log starts from the October 2011 to February of 2012 which is a period of six months

Table 4. Complete Application and Offer Event Types

Event Type	Description
A_SUBMITTED	Application submitted
A_PARTLYSUBMITTED	Application partly submitted
A_PREACCEPTED	Application pre-accepted, but not admission
A_ACCEPTED	Application accepted
A_FINALIZED	Application finalized
A_REGISTERED	Application registered
A_ACTIVATED	Application activated
A_APPROVED	Application approved
A_DECLINED	Application rejected
A_CANCELLED	Application canceled
O_SELECTED	Offer selected
O_CREATED	Offer created
O_SENT	Offer sent
O_SENT_BACK	Offer sent back
O_ACCEPTED	Offer accepted
O_CANCELLED	Offer cancelled
O_DECLINED	Offer declined

Table 5. Work item events with their English translation

Event Type	English Translation
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

Noticed that there is another attribute which is life cycle transition of work items, shows in Table 6 bellow:

Table 6. Life Cycle Events for Each Work Item

Event Type	Meaning
SCHEDULE	The work item (of type 'W_') is created in the queue (automatic step following manual actions)
START	The work item (of type 'W_') is obtained by the resource
COMPLETE	The work item (of type 'W_') is released by the resource and put back in the queue or transferred to another queue (SCHEDULE)

By importing the event log using Pm4Py library and convert to pandas data frame, noticed that there are 262,200 events each one accompanying with 6 attributes which are *org:resource*, *lifecycle:transition*, *concept:name*, *time:timestamp*, *case:REG_DATE*, *case:concept:name*, and *case:AMOUNT_REQ* and they start from the first of October 2011 to the 28th of February 2012.

After filtering the event log by start activities and end activities using Pm4Py methods, revealed that all 13,087 cases start with the event *A_SUBMITTED* which means 100% of the cases requires to start with the mentioned event type (Table 7). In contrast, by end activities, cases were ended by 11 different event types which are depicted in the following Figure 2 with their relative percentages.

Table 7. Filter by Start Activity

Event	Count
A_SUBMITTED	13087

As can be seen from the bar plot, the A_DECLINED with 26.2% has the top percentage among the others, while O_CANCELLED with 2.13% is the lowest. Those three activities which are lower than 2.13% can be neglected due to their inadequate amount. Complete percentage of each activity can be seen in the table below. (Table 8)

But with this assumption we can say that there is no deterministic end activity and need to perform further analysis in more details

Table 8. Filtering by End Activities

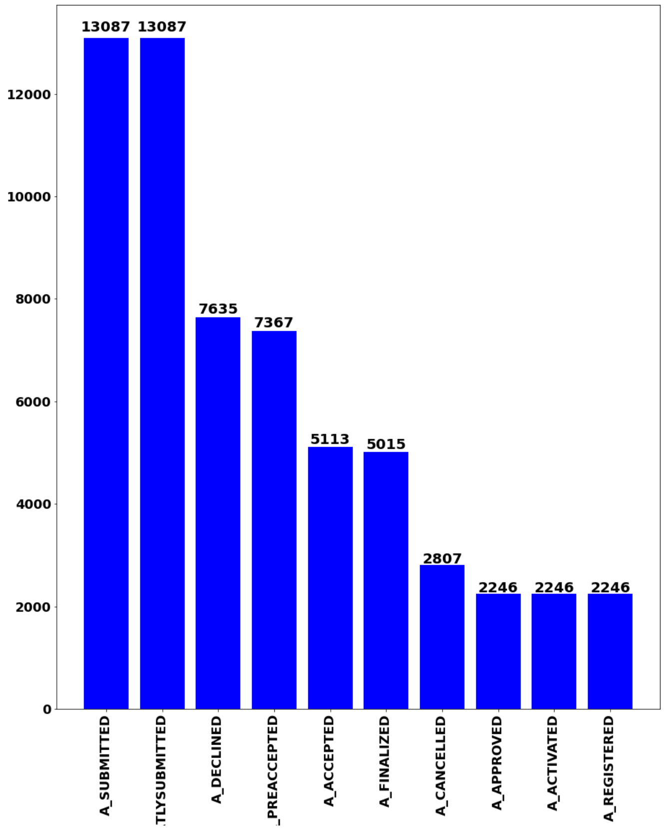
Event	Count
A_DECLINED	3,429
W_Valideren aanvraag	2,747
W_Afhandelen leads	2,234
W_Completeren aanvraag	1,940
W_Nabellen offertes	1,291
A_CANCELLED	653
W_Nabellen incomplete dossiers	452
O_CANCELLED	279
W_Beoordelen fraude	57
W_Wijzigen contractgegevens	4
A_APPROVED	1

By performing filtering on application events, we can find the statistics of the application submissions and to understand the flow of general loan application process. By looking at the bar chart below (Figure 2), we can say that the event *A_SUBMITTED* and

A_PARTLYSUBMITTED has been appear in all the cases together, above we observed that all the cases start with the event *A_SUBMITTED* (Table 7), and now we observe that without performing variant analyzing that 100% of the cases the event *A_PARTLYSUBMITTED* comes after the *A_SUBMITTED*. Also, we can say that the events *A_APPROVED*, *A_REGISTERED*, *A_ACTIVATED* can be considered as the last and successful states of the loan application approval process.

We can see that there are 5,113 cases that has been accepted at least one time and there are 7,635 rejections in the process.

Figure 2. Application States Statistics



Variant Analysis

Variant analysis was performed using Pm4Py and the result of the analysis on the financial log and 4,366 variants were extracted which over 29% of the cases are in the first variant which is *A_SUBMITTED,A_PARTLYSUBMITTED,A_DECLINED*, that means over 3,429 cases has been declined instantly after the automatic checks performed by the system.

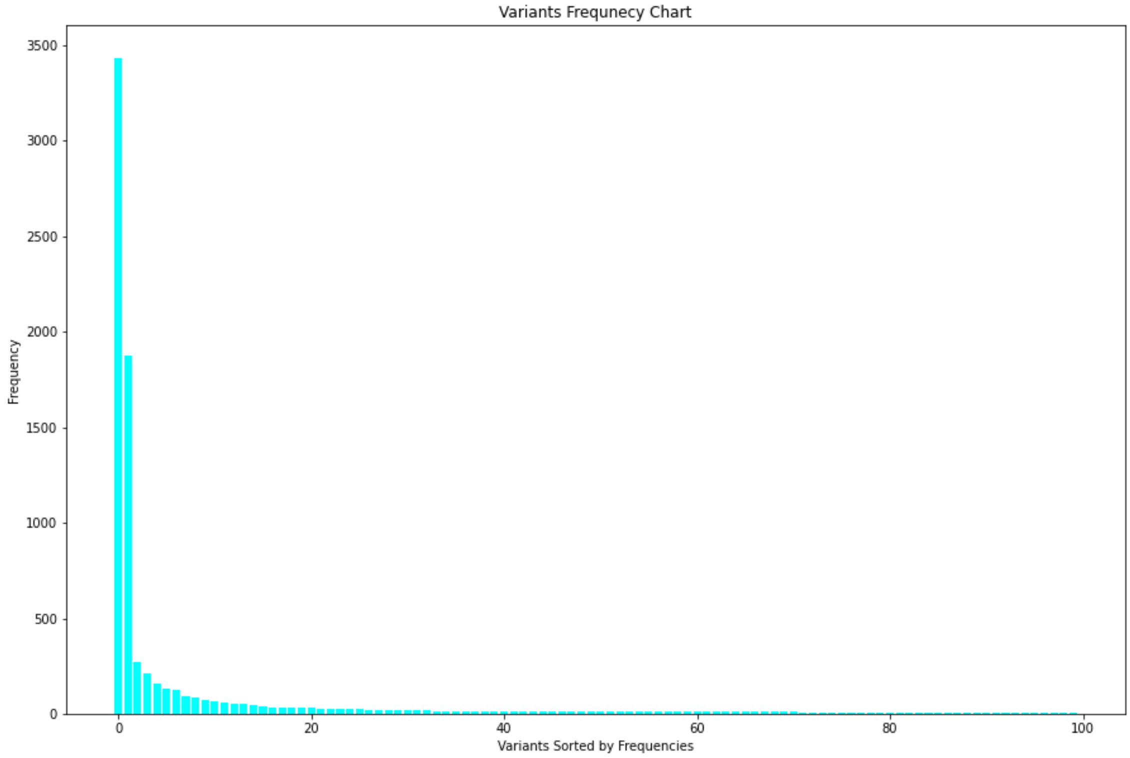
The top 5 popular variants are shown in Figure 3.

Figure 3. Top 5 Popular Variants

index	variant	count
0	A_SUBMITTED,A_PARTLYSUBMITTED,A_DECLINED	3429
1	A_SUBMITTED,A_PARTLYSUBMITTED,W_Afhandelen leads,W_Afhandelen leads,A_DECLINED,W_Afhandelen leads	1872
2	A_SUBMITTED,A_PARTLYSUBMITTED,W_Afhandelen leads,W_Afhandelen leads,W_Afhandelen leads,W_Afhandelen leads,A_DECLINED,W_Afhandelen leads	271
3	A_SUBMITTED,A_PARTLYSUBMITTED,W_Afhandelen leads,W_Afhandelen leads,A_PREACCEPTED,W_Completeren aanvraag,W_Afhandelen leads,W_Completeren aanvraag,A_DECLINED,W_Completeren aanvraag	209
4	A_SUBMITTED,A_PARTLYSUBMITTED,A_PREACCEPTED,W_Completeren aanvraag,W_Completeren aanvraag,A_DECLINED,W_Completeren aanvraag	160

To understand the frequency and distribution of variants, the variant frequency chart is provided. (Figure 4)

Figure 4. Variant Frequency



After extracting variants, more analysis was performed to extract more information reported in Table 9.

Table 9. Analyzing Cases in More Details

Info	**Count**
#Cases at least one time approved	7,367
#Cases never approved	5,720
#Calling more than 3 times	4,420
#Calling for additional information	1,647
#Suspecting fraud cases	108
#Needs to modify approved contracts	7

In further sections, each sub process is separated from the merged event log and process mining (PM) algorithms, process discovery techniques and conformance checking to each of them is applied. At the end of the report the process discovery of the general loan approval process and conformance checking is applied.

Analyzing Sub Processes

Each sub process was analyzed in more details and process mining algorithms were performed.

There are three sub processes consists of Offer Sub Process (OSP), Application Sub Process (ASP) and Work Item Sub Process (WISP).

Event Log Analyzing for Sub Processes

As the challenge mentioned the states that start with the letter “O_” are belonging to the OSP which indicates the state of the offer belonging to the application. And the events which are started by the letter “A_” are belonging to the application states (ASP) and those with the letter “W_” are belonging to the work item states (WISP). The sub processes were extracted by using panda’s library after the conversion of the log to the data frame.

The details of these sub processes are reported in the Table 10 , Table 11 and Table 12 below:

Table 10. OSP Preliminaries Statistics

Stat	**Value**
Number of cases	5,015
Number of events	31,244
First event	October 1st 2011
Last event	March 14th 2012
Mean	17 days, 4:20 h
Mean with 0-day filter	18 days, 0:27 min
Number of event types	7
Number of resources	60

All cases in OSP starts by O_ SELECTED activities while they end in different activities.

All cases in ASP were started by the *A_SUBMITTED* event.

Table 11. ASP Preliminaries Statistics

Stat	Value
Number of cases	13,087
Number of events	60,849
First event	September 30st 2011
Last event	March 14th 2012
Mean	8 days, 1:55 h
Mean with 0-day filter	17 days 13:08 h
Number of event types	10
Number of resources	61

In WISP the cases are started in a more different variations which is 4,852 cases starts by the event W_Completeren aanvraag, which is completing pre accepting application or by the event W_Afhandelen leads for 4,739 cases which is for completion of the incomplete initial submission, and a few amounts of the cases are started by the event W_Beoordelen fraude which is for investigating suspect fraud cases.

Table 12. WISP Preliminaries Statistics

Stat	Value
Number of cases	9,658
Number of events	170,107
First event	September 30st 2011
Last event	March 14th 2012
Mean	11 days, 16:26 h
Mean with 0-day filter	17 days 20:54 h
Number of event types	7
Number of resources	59

Variant Analysis on Sub Processes

To find and extract more details we need to perform variant analysis, the implementation was with Pm4Py and the top 5 variants of each sub process as Pandas data frame is shown in the notebook.

Number of variants for each of the sub processes is shown in Table 13.

Table 13. Number of Variants

Sub-Process	Count
WISP	2,643
OSP	166
ASP	17

The variant frequency of OSP is reported in Figure 5, frequency of variances for ASP is Figure 6 and variant frequency of WISP is shown in Figure 7:

Figure 5. Variant Frequency of OSP

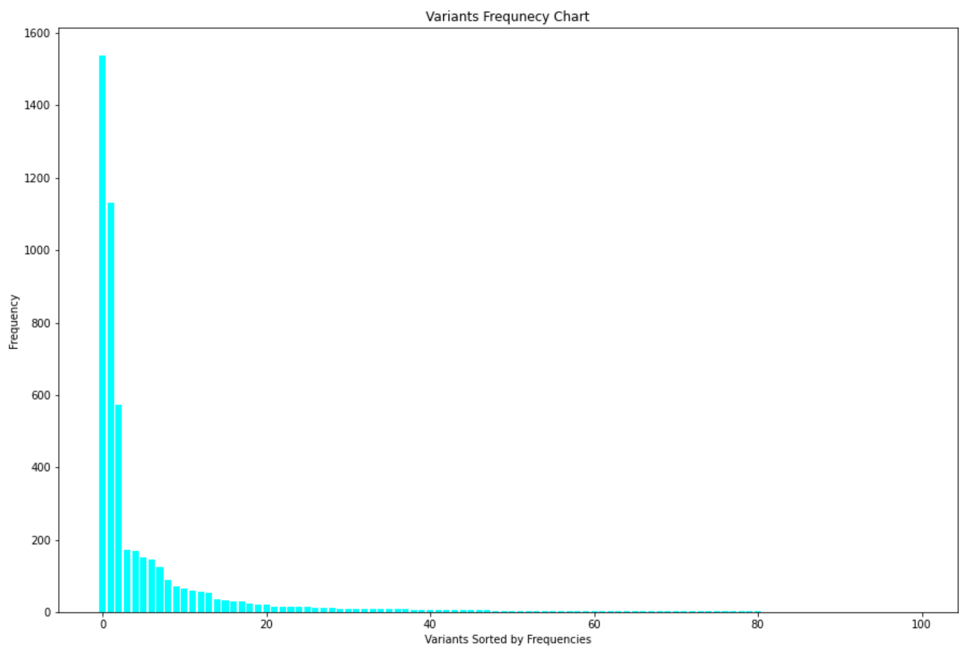


Figure 6. Variant Frequency of ASP

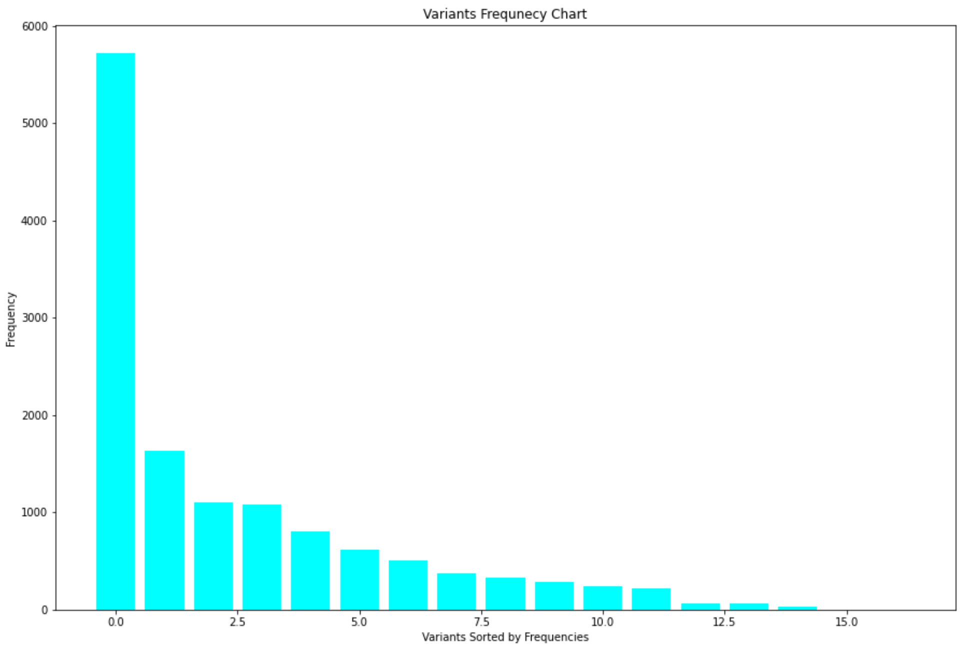
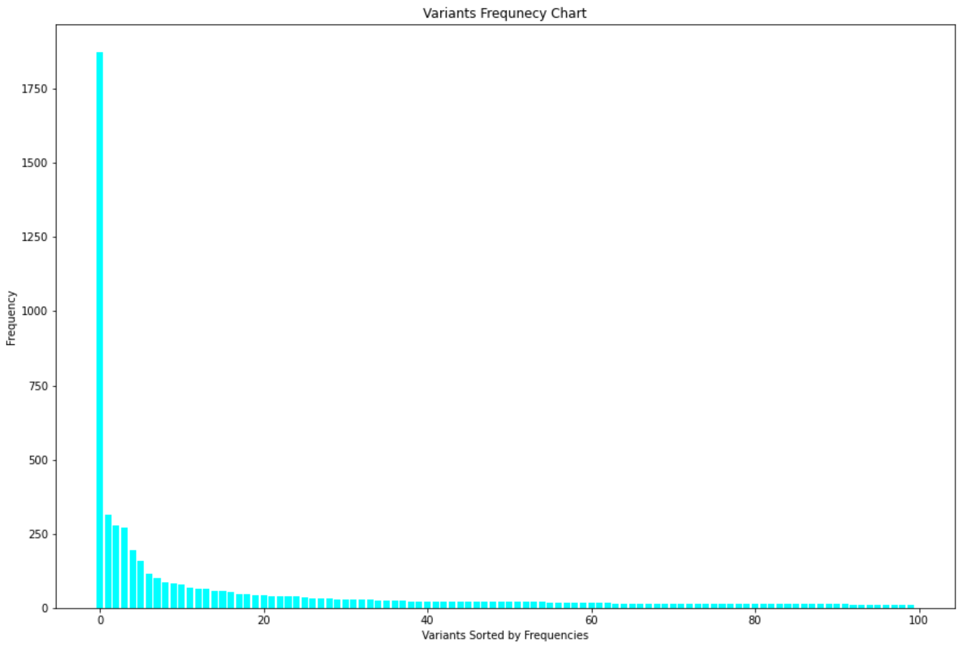


Figure 7. Variant Frequency of WISP



The more detail statistics of each sub processes are reported in the following Tables 14, 15, 16:

Table 14. More Detail Information on OSP

Stat	Value
#Offers created	5,015
#Offers answered	3,254
#Accepted offers	2,772
#Incomplete offers	211

Table 15. More Detail Information on ASP

Stat	Value
#Never Accepted Applications	7,367
#Finalized Application	5,015
#Pre-accepted Application	2,806
#Accepted Application	2,246

Table 16. Number of Activities on WISP

Stat	Value
#W_Completeren aanvraag	54,850
#W_Nabellen offertes	52,016
#W_Nabellen incomplete dossiers	25,190
#W_Valideren aanvraag	20,809
#W_Afhandelen leads	16,566
#W_Beoordelen fraude	664
#W_Wijzigen contractgegevens	12

Filtering on Sub Processes

For each sub process a unique filtering has performed, for ASP, only the variants which are passed the automatic check declined and could lead to pre-accepted state are considered, while for OSP only top 5 variants are considered and filtered, and for WISP, the 200 top variants are extracted and filtered.

The number of cases before and after filtering are reported in Table 17.

Table 17. Count Cases After and Before Filtering

Sub-process	Before Filtering	After Filtering
ASP	13,087	7,367
OSP	5,015	3,584
WISP	9,658	6,475

Process Discovery on Sub Processes

Using directly-follow graph we can see the flow of this sub processes with their frequency and with their performance. The DFG algorithm and visualization were implemented using Pm4Py methods.

It should be noticed that parallel activities cannot be detected with DFG.

Figure 8. DFG Frequency for ASP

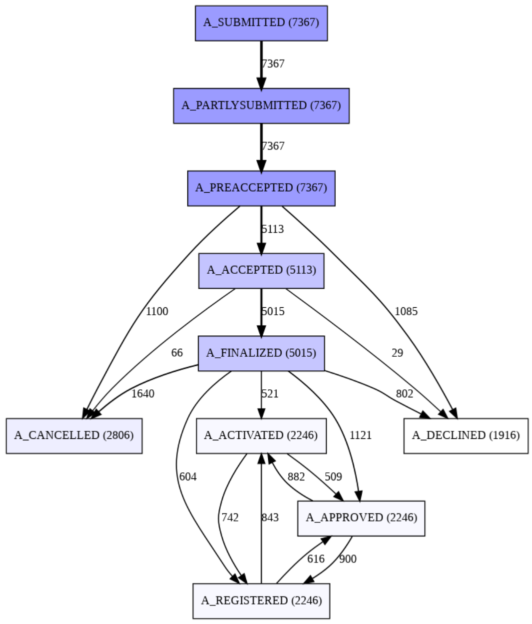


Figure 9. DFG Frequency for OSP

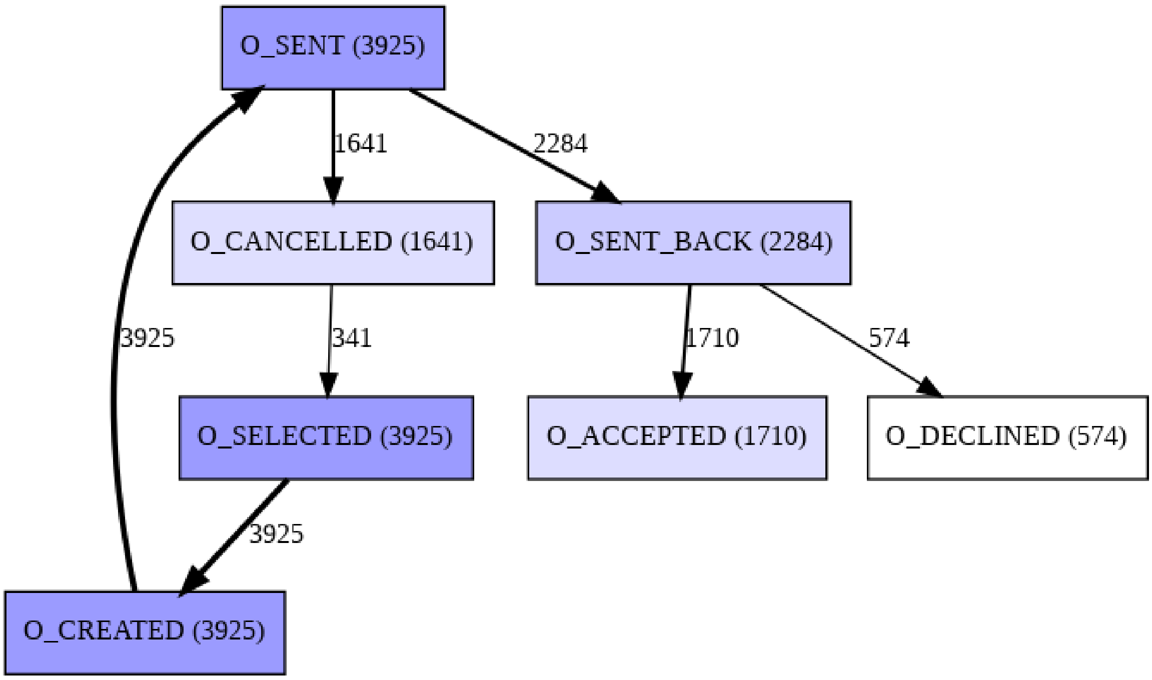
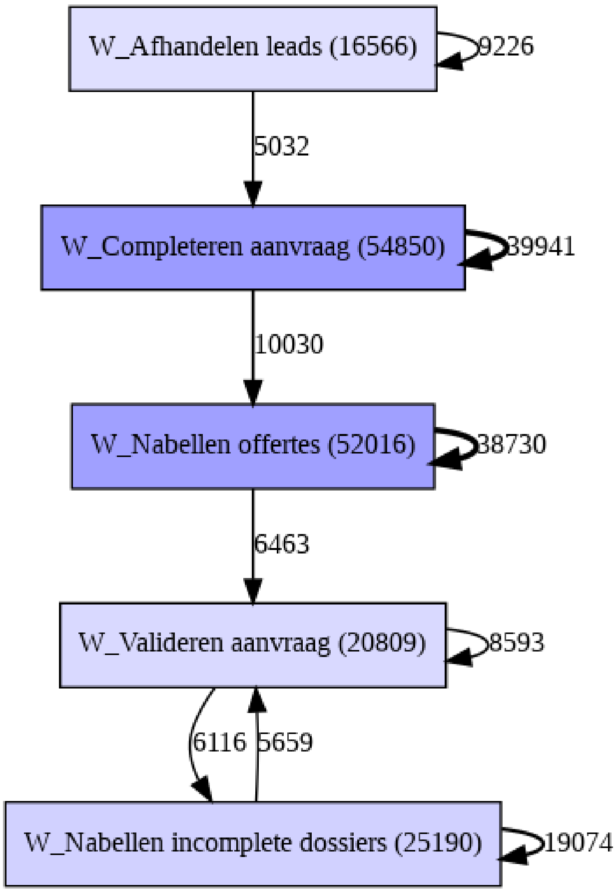


Figure 10. DFG Frequency for OSP



With alpha-miner we can detect parallel activities, in the following diagram AM was illustrated in petri net by using Pm4Py functions on sub processes.

Figure 11.AM on ASP [FREQUENCY]

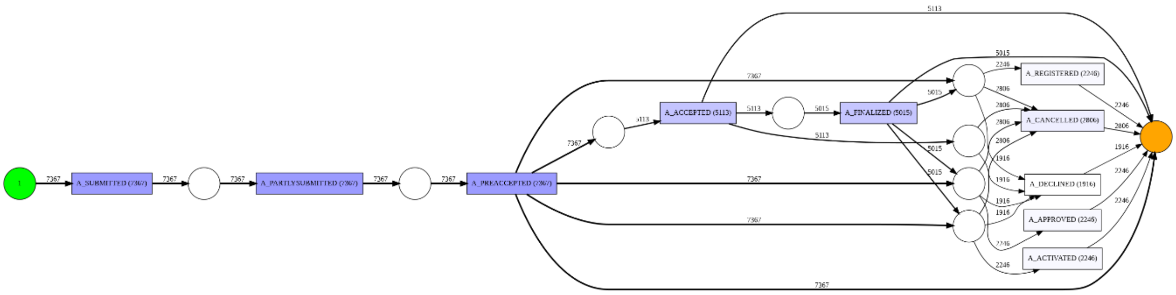


Figure 12.AM on ASP [PERFORMANCE]

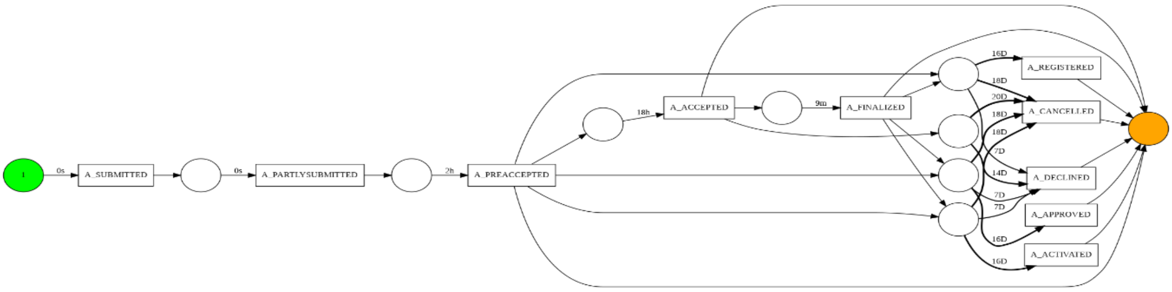


Figure 13. AM on OSP

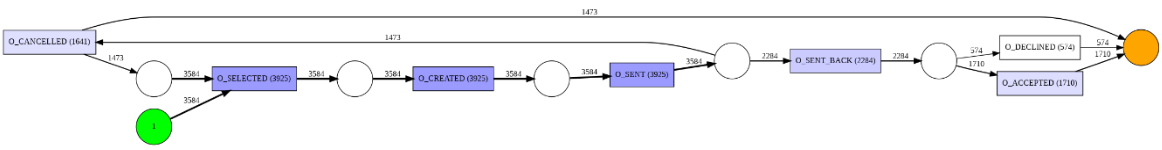
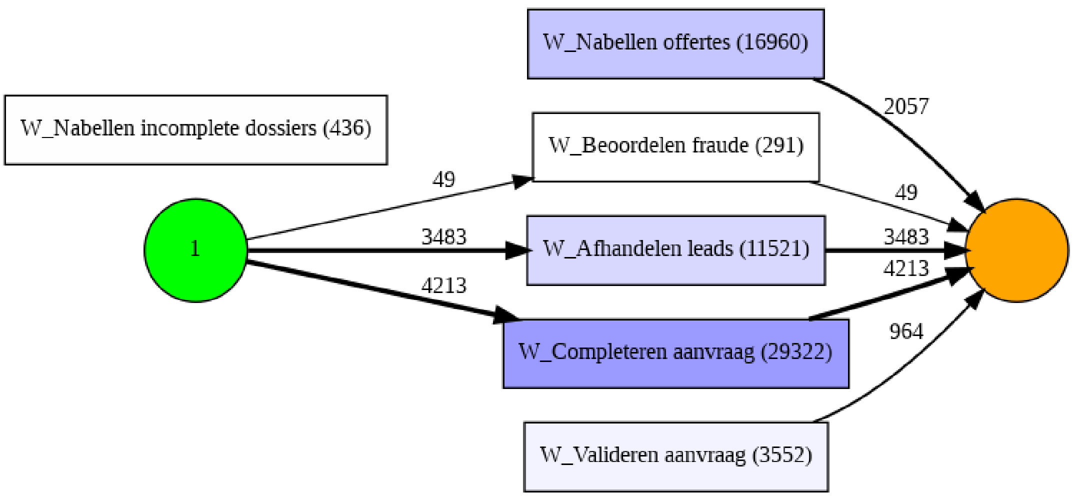


Figure 14. AM on WISP



As Pm4Py documentation mentioned the idea of inductive miner (IM) Is about detecting a "cut" in the log, with cut we mean sequential cut, parallel cut, concurrent cut and look cut. Then recur on sub logs, which were found applying the cut, until a base case is found.
The inductive miner of the sub processes is shown below:

Figure 15. IM on Process-Tree (PT) of ASP

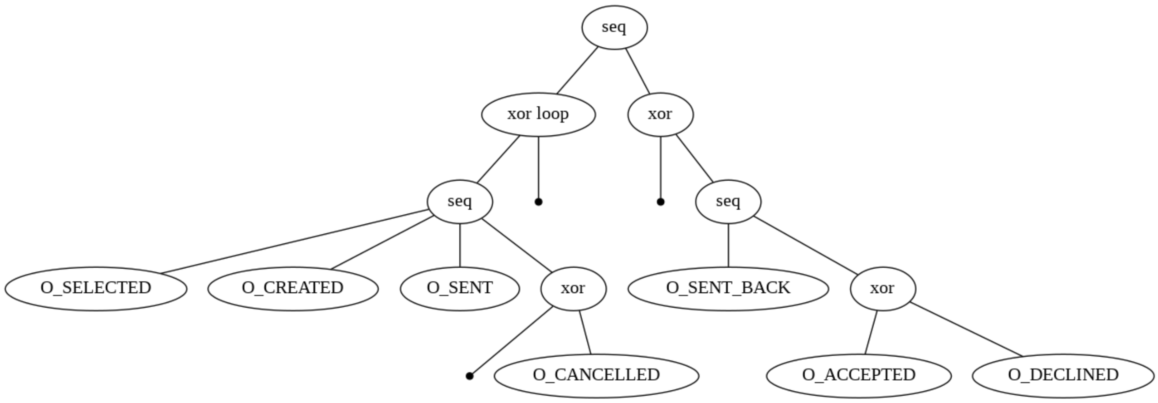


Figure 16. IM on Process-Tree (PT) of OSP

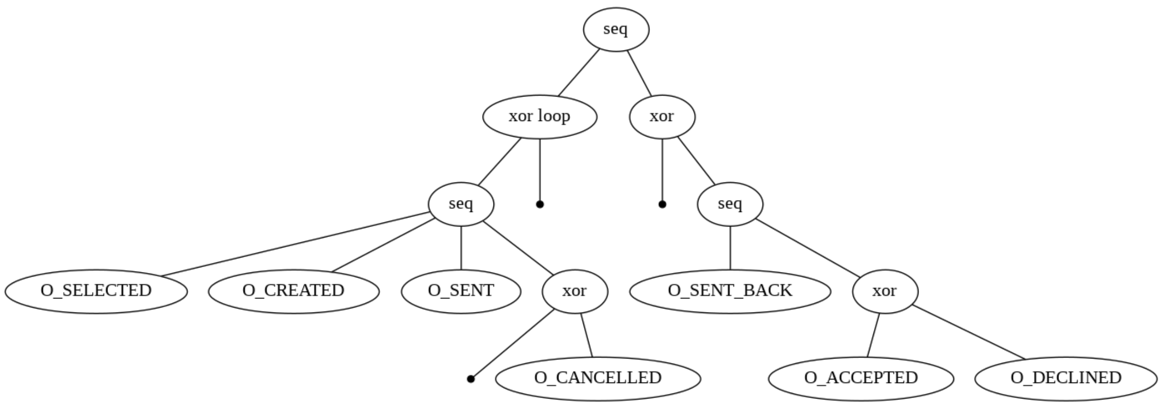
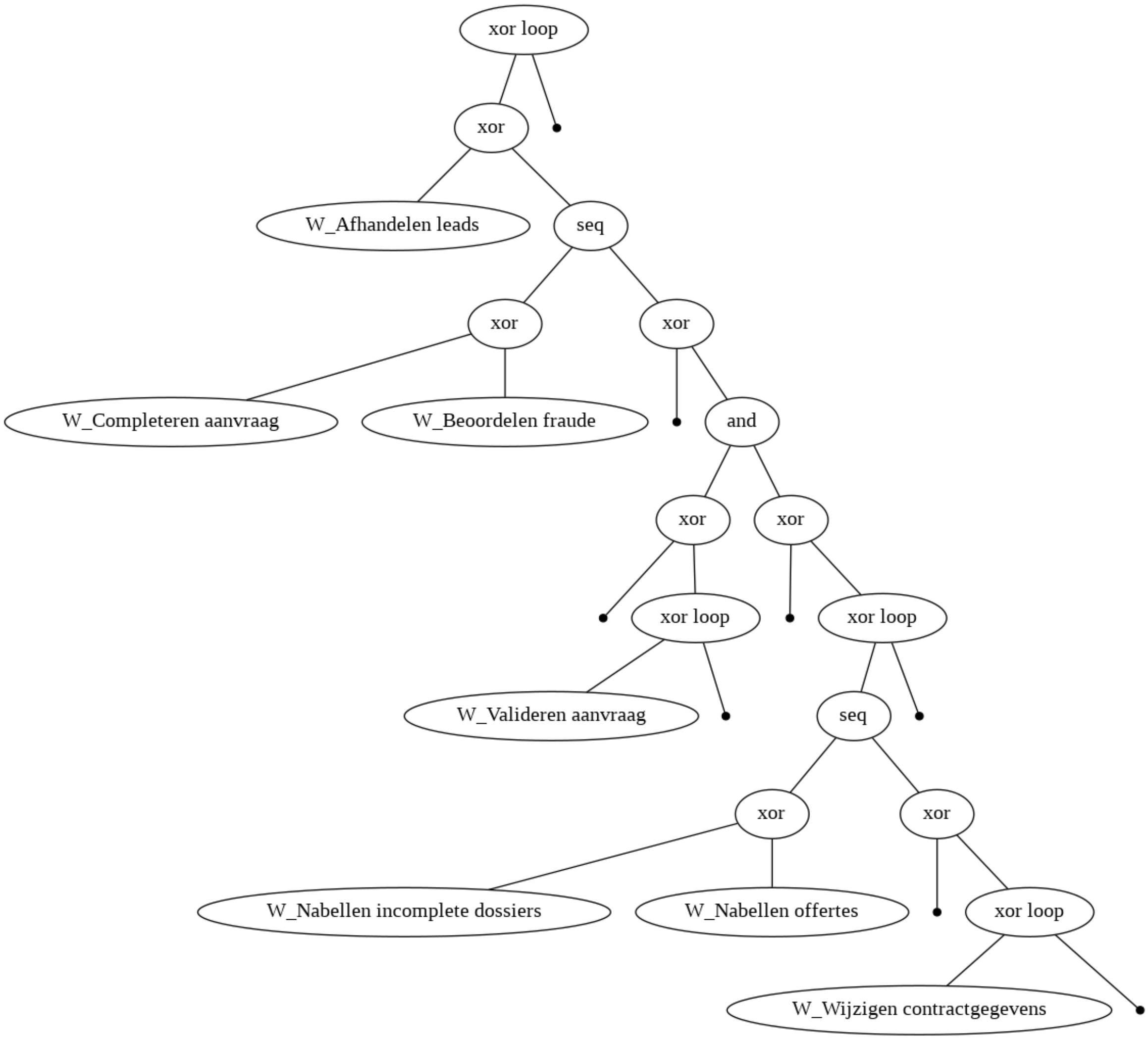


Figure 17. IM on Process-Tree (PT) of WISP



The petri net of the IM was performed by converting the tree to petri net using petri net converter in Pm4Py. The result of the petri net both in terms of variants of frequency can be seen in figures below:

Figure 18. IM on Petri-Net of ASP

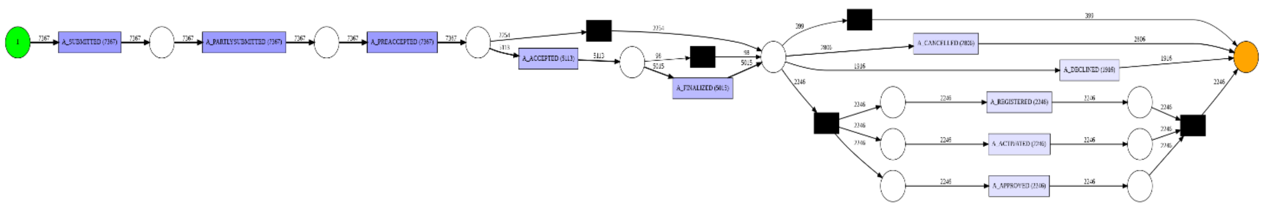


Figure 19. IM on Petri-Net of OSP

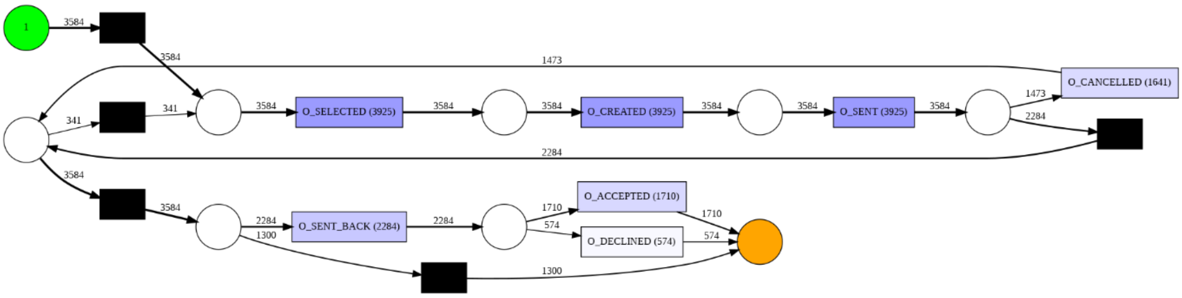
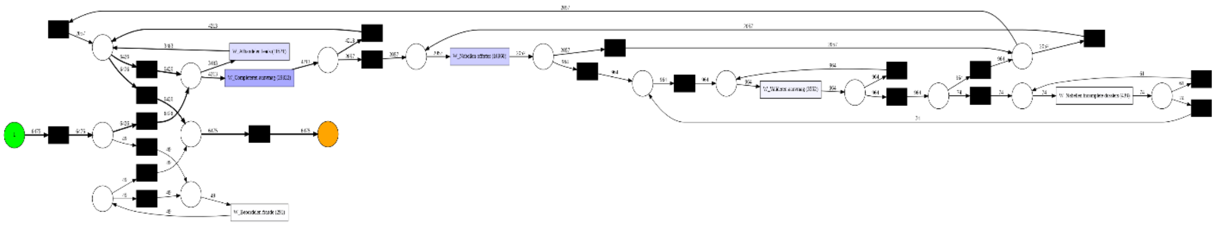


Figure 20. IM on Petri-Net of WISP



The last process discovery algorithm is heuristic miner (HM). The HM algorithm and visualizations were implemented using Pm4Py functions. HM is an algorithm that acts on DFG and the output of HM is a heuristic net.

Figure 21. HM on DFG for ASP

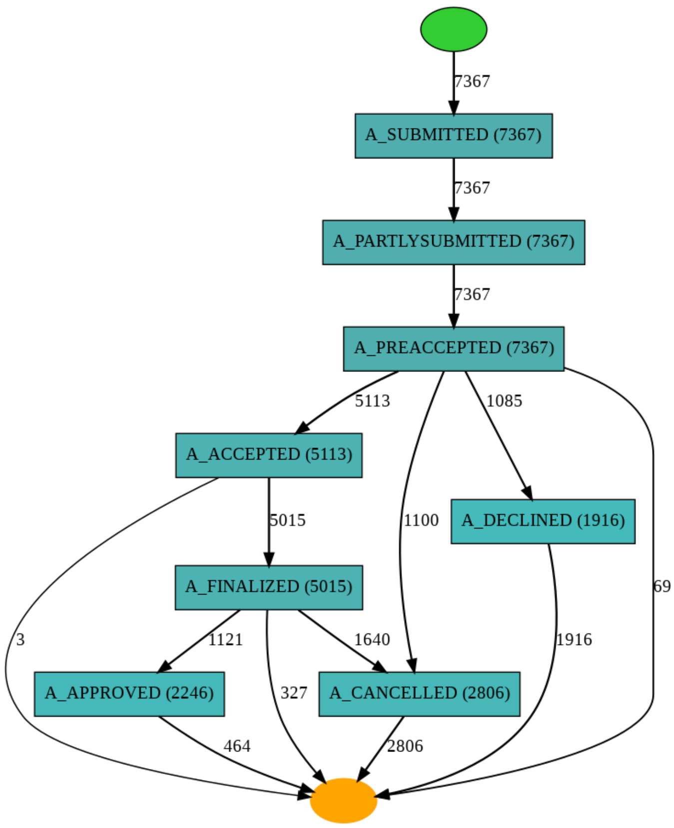


Figure 22. HM on Petri-Net for ASP

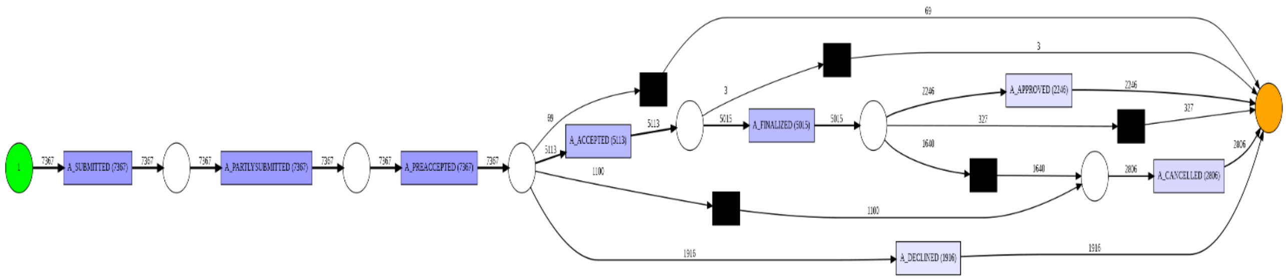


Figure 23. HM on DFG for OSP

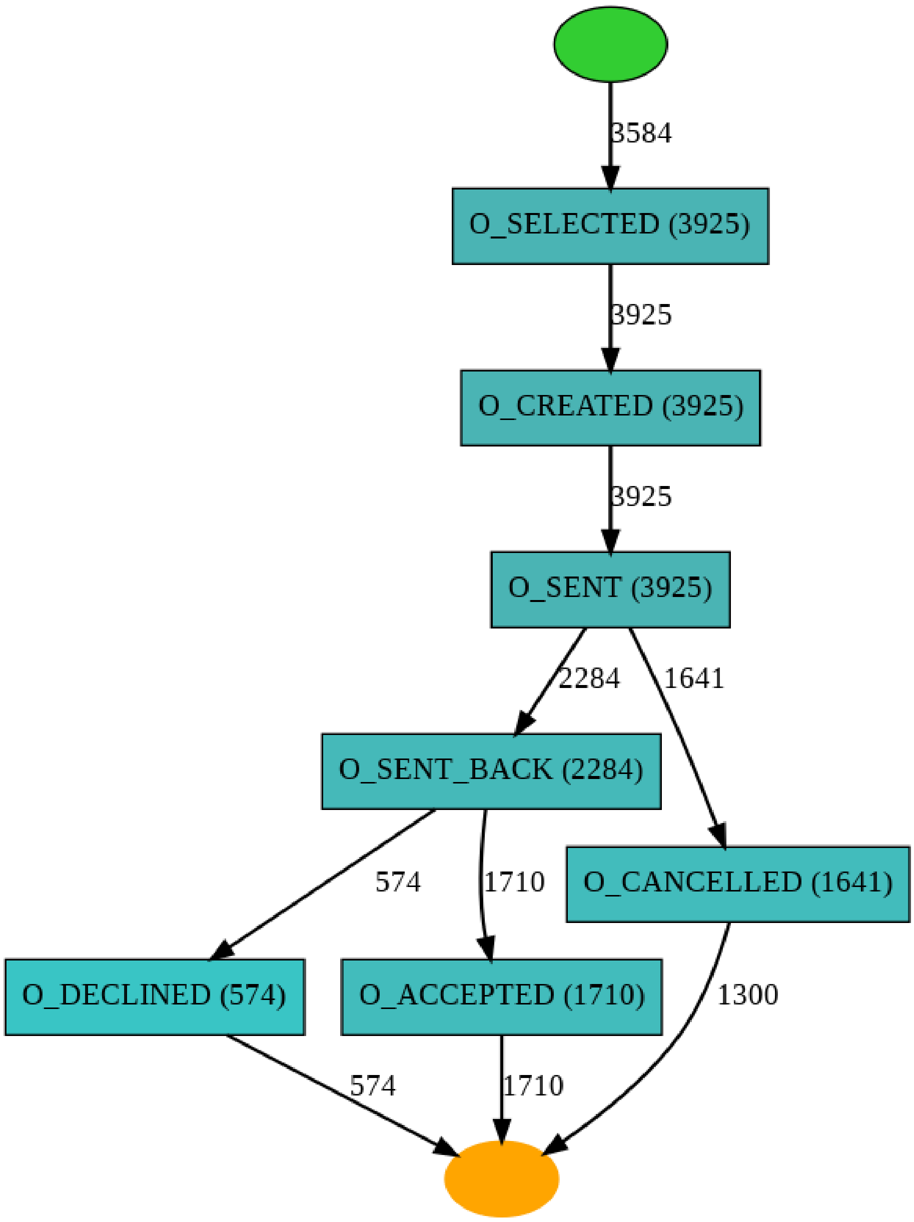


Figure 24. HM on Petri-Net for OSP

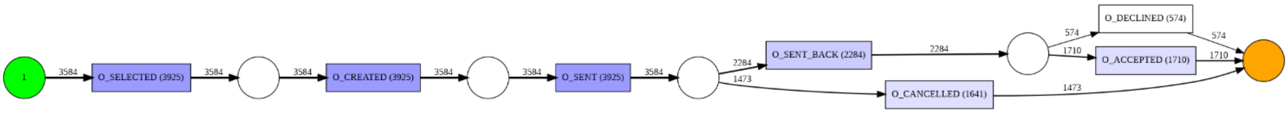


Figure 25. HM on DFG for WISP

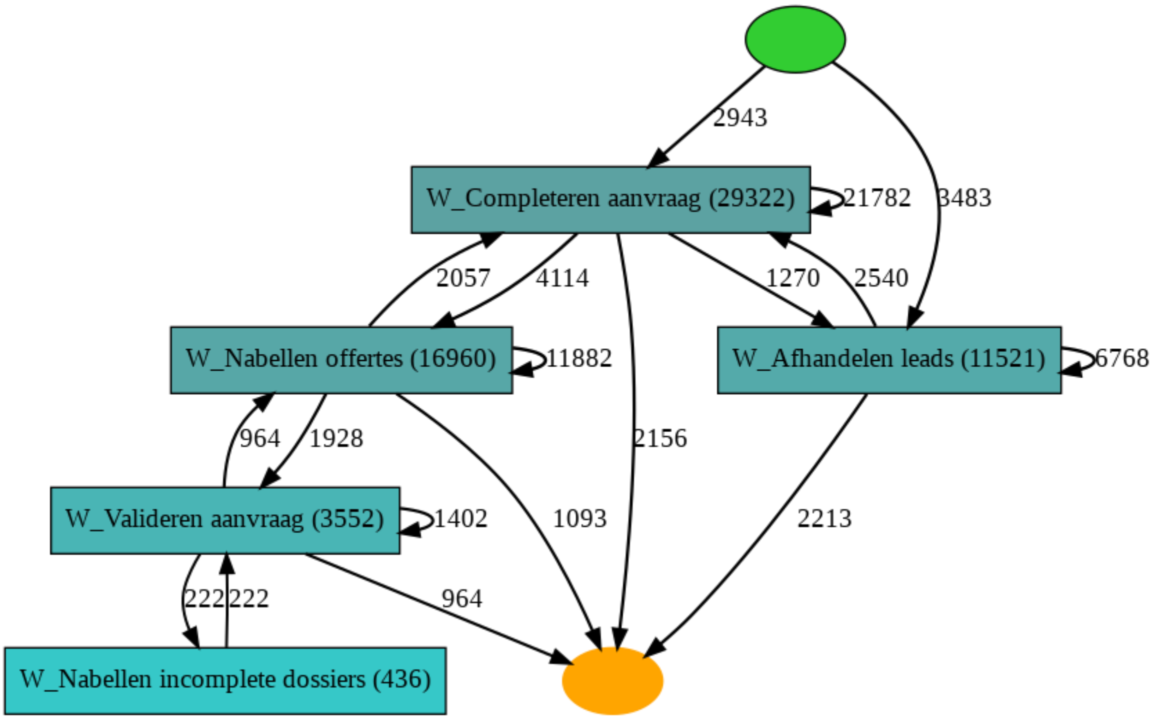
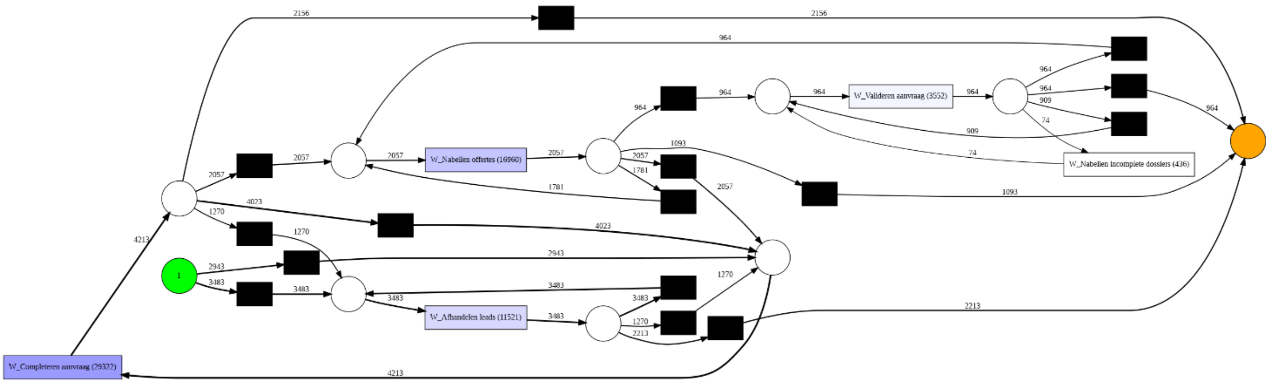


Figure 26. HM on Petri-Net for WISP



Quality Metrices of The Models

The quality metrices for each model (Precision, Fitness, Generalization, Simplicity) are performed using Pm4Py for each model behind its section in code and the result of them are shown also in Table 17.

Table 17. Results of Quality Metrices

Sub process	Model	Fitness	Precision	Generalization	Simplicity
ASP	AM	0.79	0.58	0.98	0.40
ASP	IM	1.0	0.70	0.97	0.70
ASP	HM (D_T=0.998)	0.98	0.97	0.92	0.67
OSP	AM	0.89	1.0	0.97	0.77
OSP	IM	1.0	0.84	0.97	0.77
OSP	HM (D_T =0.998)	0.98	1.	0.97	0.86
WISP	AM	0.32	0.33	0.97	1
WISP	IM	1.0	0.49	0.96	0.71
WISP	HM	0.99	0.53	0.97	0.58

Conformance Checking

Using Token-Based replay and alignment in Pm4Py for best models are performed to check the efficiency of the model

Figure 27. Token-Based Replay and Alignment for IM of ASP

```
replaying log with TBR, completed variants :: 100%
REPLAY
Number of traces 13087
173697, 173700, 173727, 173733, 173757, 173763, 173769
Number of anomalous traces  5720
Percentage of anomalous traces 43.70749598838542 %
aligning log, completed variants :: 100%
ALIGNMENTS
Number of traces 13087
['173697', '173700', '173727', '173733', '173757', '17
Number of anomalous traces  5720
Percentage of anomalous traces 43.70749598838542 %
```

Figure 28. Token-Based Replay and Alignment for HM of ASP

```
REPLAY
Number of traces 13087
173697, 173700, 173715, 173721, 173727, 173733, 17374
Number of anomalous traces  6617
Percentage of anomalous traces 50.5616260411095 %
aligning log, completed variants :: 100%
ALIGNMENTS
Number of traces 13087
['173688', '173691', '173694', '173697', '173700', '1
Number of anomalous traces  8863
Percentage of anomalous traces 67.72369527011539 %
```

Figure 29. Token-Based Replay and Alignment for IM of OSP

```
REPLAY
Number of traces 5015
173718, 173736, 173748, 173880, 174060, 174012, 1740
Number of anomalous traces  1145
Percentage of anomalous traces 22.831505483549353 %
aligning log, completed variants :: 100%
ALIGNMENTS
Number of traces 5015
['173718', '173736', '173748', '173880', '174060', '
Number of anomalous traces  1145
Percentage of anomalous traces 22.831505483549353 %
```

Figure 30. Token-Based Replay and Alignment for HM of OSP

```
REPLAY
Number of traces 5015
173718, 173691, 173736, 173748, 173787, 173817, 173
Number of anomalous traces  1772
Percentage of anomalous traces 35.33399800598205 %
aligning log, completed variants :: 100%
ALIGNMENTS
Number of traces 5015
['173718', '173691', '173736', '173748', '173787',
Number of anomalous traces  1772
Percentage of anomalous traces 35.33399800598205 %
```

Figure 31. Token-Based Replay and Alignment for IM of WISP

```
REPLAY
Number of traces 9658
173721, 173754, 173955, 174036, 174084, 174168, 1742
Number of anomalous traces  507
Percentage of anomalous traces 5.2495340650238145 %
aligning log, completed variants :: 100%
ALIGNMENTS
Number of traces 9658
['173694', '173721', '173754', '173955', '174036', '
Number of anomalous traces  513
Percentage of anomalous traces 5.311658728515221 %
```

Figure 32. Token-Based Replay and Alignment for HM of WISP

```
REPLAY
Number of traces 9658
173718, 173721, 173730, 173736, 173739, 173784, 1738
Number of anomalous traces  1623
Percentage of anomalous traces 16.804721474425346 %
aligning log, completed variants :: 100%
ALIGNMENTS
Number of traces 9658
['173694', '173718', '173721', '173730', '173736', '
Number of anomalous traces  1659
Percentage of anomalous traces 17.177469455373785 %
```

Conclusion

In conclusion, as reported the major bottleneck of the process handle by this company is in the work item states sub process where there are amount of rework and self-loops, as we observed in the direct-follow graph simply, it takes a long time in average to pass from some events in that process, calling after sent offers and filling information for completion of the application are the most obvious ones, and there is an extensive diversity of variants in the sub-process which makes it difficult to find a suitable model. After inductive miner algorithm were performed in overall good model for WISP sub-process. For other process, OSP and ASP, these two processes are very dependent in on their activities, especially ASP, the flow of the model can be obvious as reported in the event log analysis section, there are not variants as much as WISP and some events were not repeated more than once in one variant. In OSP the overall data at first was not that much redundant but preferred to perform filtering and process discovery as the nature of process mining. Therefore, there are some redundancies in each sub process, and some followed by traditional ways such as calling by phone and send offers by email, they could be observed from models, the organization can change the business management and using high-tech developments to improve the efficiency and flexibility.

Full Notebook and Code on Github Repository

1. https://github.com/mirzaghavami/BPI-Challenge-2012