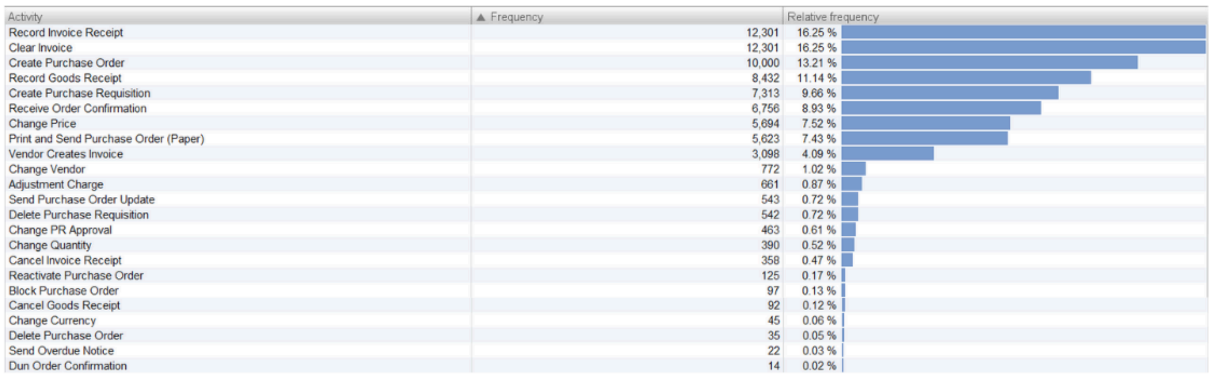
PROCESS MINING PROJECT

Camilla Colanero 2086977

OVERVIEW

Before starting with the tasks assigned, we are going to briefly analyze an overview of the process and of the event Dataset.
Using the Disco tool we can see that the Dataset is composed of 75677 events that create 10000 cases. We have 23 different activities, that we list below along with their frequency:



Activity	▲ Frequency	Relative frequency
Record Invoice Receipt	12,301	16.25 %
Clear Invoice	12,301	16.25 %
Create Purchase Order	10,000	13.21 %
Record Goods Receipt	8,432	11.14 %
Create Purchase Requisition	7,313	9.66 %
Receive Order Confirmation	6,756	8.93 %
Change Price	5,694	7.52 %
Print and Send Purchase Order (Paper)	5,623	7.43 %
Vendor Creates Invoice	3,098	4.09 %
Change Vendor	772	1.02 %
Adjustment Charge	661	0.87 %
Send Purchase Order Update	543	0.72 %
Delete Purchase Requisition	542	0.72 %
Change PR Approval	463	0.61 %
Change Quantity	390	0.52 %
Cancel Invoice Receipt	358	0.47 %
Reactivate Purchase Order	125	0.17 %
Block Purchase Order	97	0.13 %
Cancel Goods Receipt	92	0.12 %
Change Currency	45	0.06 %
Delete Purchase Order	35	0.05 %
Send Overdue Notice	22	0.03 %
Dun Order Confirmation	14	0.02 %

Figure 1. List of activities and their frequency

The most frequent variant (416 cases) is made of 7 events:
Create Purchase Requisition, Create Purchase Order, Print and Send Purchase Order (Paper), Receive Order Confirmation, Record Goods Receipt, Record Invoice Receipt, Clear Invoice.

If we filter out the events that do not end with the activity *Clear Invoice,* that should be the last in order to complete the process, we remain with 8704 cases.

The dataset consists of an event log that refers to executions of instances of the purchase-order process as carried on by a SAP system at a German company starting from the 3rd of January 2008 and ending the 9th of February 2010. The mean case duration is 51.5 days but there are a few cases that take more than 1 year.

DESIGN AND VALIDATION OF A PROCESS MODEL

Using the description of the process that we have been given, for the construction of our first model we are going to use only 13 activities out of the 23. The first Petri net that we draw, using the tool WoPeD, is the following:

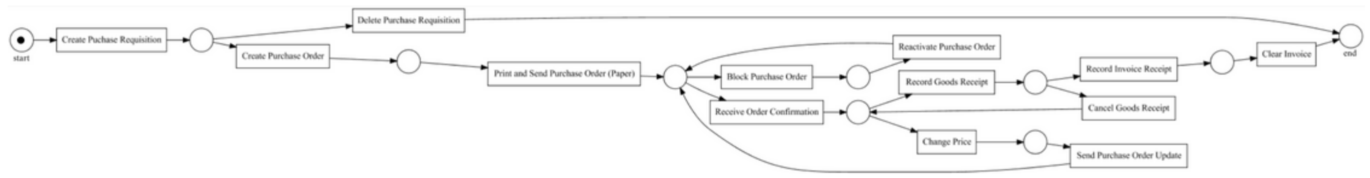

start
Create Puchase Requisition
Delete Purchase Requisition
Create Purchase Order
Print and Send Purchase Order (Paper)
Block Purchase Order
Receive Order Confirmation
Reactivate Purchase Order
Record Goods Receipt
Change Price
Record Invoice Receipt
Cancel Goods Receipt
Send Purchase Order Update
Clear Invoice
end

Figure 2. Visualization of the first Petri net through the Graphviz Petri net visualization of ProM

In order to find out if the model is satisfactory, i.e. it has a fitness score of at least 0.8 and the value of precision is at least 0.9, we are going to check its conformance. To do so we use the tool ProM: we import our Event Log and the Petri net that we want to study. We use the action

Replay a Log on Petri Net for Conformance Analysis that takes in input a Event Log and a Petri net.
From the visualization *Model Projected with Alignments* we can see that we have some asynchronous moves, that are numerous in four transitions: *Create Purchase Requisition, Print and Send Purchase Order (Paper), Receive Order Confirmation* and *Record Goods Receipt.*

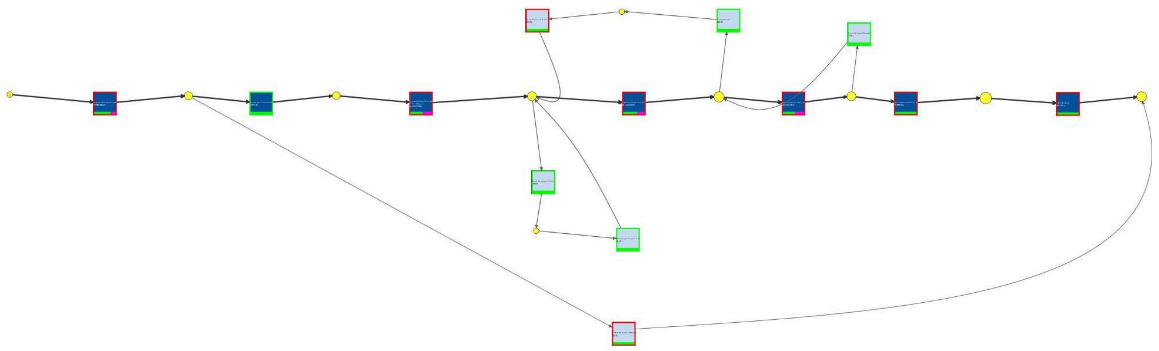
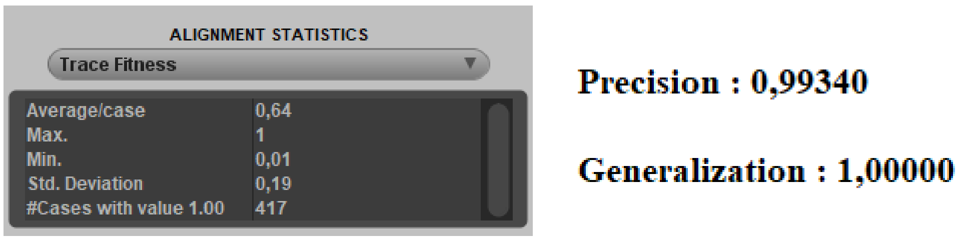


Figure 3. Model Projected with Alignments visualization

We do not filter our Event Log because the ProM tool classifies all events as complete. We have tried all the following discussions also on the filtered log and the results were similar, therefore in the following dissertation we will compute the measures of fitness and precision on the entire Event Log.
We can check the fitness of this model by changing the visualization to Project Alignment to Log. We get a fitness score of 0.64, as shown in the first picture in figure 4.


ALIGNMENT STATISTICS

Trace Fitness

Average/case	0,64
Max.	1
Min.	0,01
Std. Deviation	0,19
#Cases with value 1.00	417

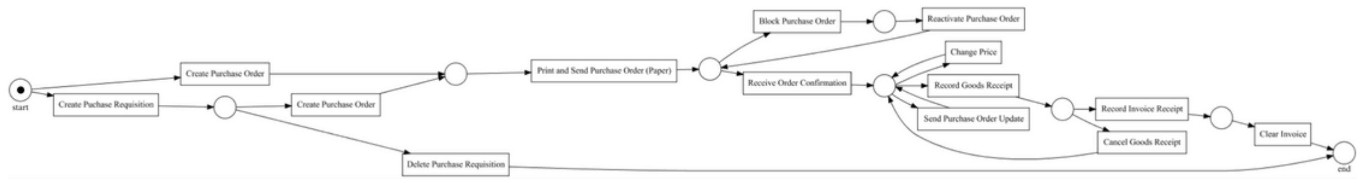
Precision : 0,99340

Generalization : 1,00000

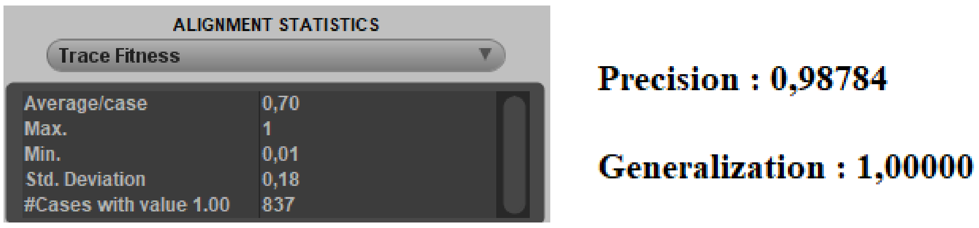
Figure 4. Fitness score and Precision score for the first model

To compute the precision, we use the action *Measure Precision/Generalization* that takes in input the Event Log, the Petri net and the Alignment that we previously created. The precision score for this first model is 0.99.

From these results, we can see that the first model that we designed does not fit the process very well but performs very well in terms of precision. In order to improve our model, we can take a deeper look at figure 3. As already mentioned, the asynchronous moves are particularly present in four transitions. If we focus on the first transition we can see that, if we go back to the description of the process, with this Petri net we do not model the process for small orders (the description says that for small orders, the activity *Create Purchase Requisition* can be often skipped). We also realize that the two activities *Change Price* and *Send Purchase Order Update* do not need to happen consequently. The second model that we are going to check is therefore the following:


start
Create Puchase Requisition
Create Purchase Order
Create Purchase Order
Delete Purchase Requisition
Print and Send Purchase Order (Paper)
Block Purchase Order
Reactivate Purchase Order
Receive Order Confirmation
Change Price
Record Goods Receipt
Send Purchase Order Update
Record Invoice Receipt
Cancel Goods Receipt
Clear Invoice
end
Figure 5. Visualization of the second Petri net through the Graphviz Petri net visualization of ProM

With the same steps described before, we compute the fitness and precision scores. We arrive at a fitness of 0.7 and precision of 0.987, as shown in figure 6.


ALIGNMENT STATISTICS

Trace Fitness

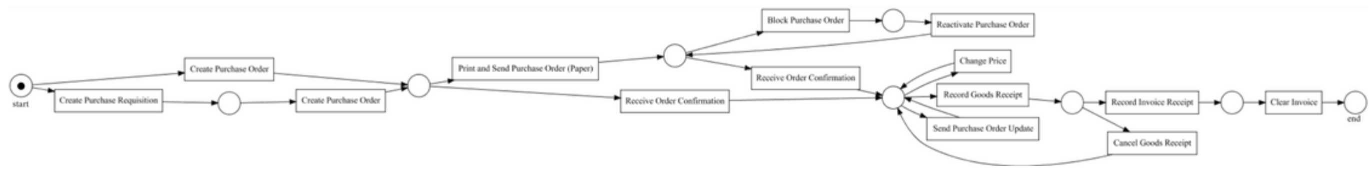
Average/case	0,70
Max.	1
Min.	0,01
Std. Deviation	0,18
#Cases with value 1.00	837

Precision : 0,98784

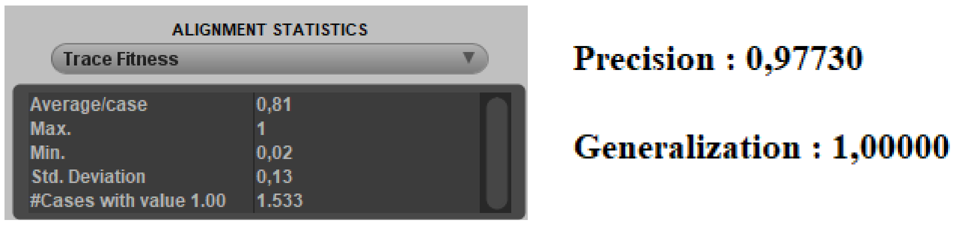
Generalization : 1,00000

Figure 6. Fitness score and Precision score for the second model

We are thus in the right direction, but we need to improve even more the fitness of our model. After some tests, we discover that if we remove the *Delete Purchase Requisition* transition, the fitness score greatly increases and we do not lose much precision. The final adjustment that allows us to arrive at a satisfactory model is to create an option to skip the activity *Print and Send Purchase Order (Paper)* after *Create Purchase Order* and go directly to the activity *Receive Order Confirmation.* Our final model is the following:


start
Create Purchase Requisition
Create Purchase Order
Create Purchase Order
Print and Send Purchase Order (Paper)
Receive Order Confirmation
Block Purchase Order
Reactivate Purchase Order
Receive Order Confirmation
Change Price
Record Goods Receipt
Send Purchase Order Update
Record Invoice Receipt
Cancel Goods Receipt
Clear Invoice
end
Figure 7. Visualization of the final Petri net through the Graphviz Petri net visualization of ProM

Again, with the usual procedure, we compute the fitness and precision scores for this model:


ALIGNMENT STATISTICS

Trace Fitness

Average/case	0,81
Max.	1
Min.	0,02
Std. Deviation	0,13
#Cases with value 1.00	1.533

Precision : 0,97730

Generalization : 1,00000

Figure 8. Fitness score and Precision score for the final model

We have therefore found a satisfactory model for the process described.

PROCESS DISCOVERY

The new goal is to discover models using different miners and find a model that balances precision and fitness at best. The best model that we aim to find needs to score at least 0.8 in fitness and at least 0.8 in precision. The miners that we are going to use for this investigation are the following:

- Interactive Data-aware Heuristic Miner
- Inductive Visual Miner
- Alpha Miner

Now we will delve into each of the models obtained with the different miners.

INTERACTIVE DATA-AWARE HEURISTIC MINER

For the discovery of the model using the Heuristic Miner, we upload to the ProM tool the Event Log and we perform the action *Interactive Data-aware Heuristic Miner* on it. The program asks us to select the event classifier to be used and we select the *Event Name* option. To find the best model we changed the values of frequency and dependency: doing that allows us to alter the resulting Petri net. If we increase the value of frequency, then we will have only the paths that are more frequent, while decreasing the value of frequency results in a more precise Petri net. After changing the parameters, we export the Petri net and we use the procedure explained in the previous paragraph to compute fitness and precision.
We report in this table the tries that we made in order to achieve the best model. We did not arrive at a fitness score of at least 0.8 but we got very close.

FREQUENCY	DEPENDENCY	FITNESS	PRECISION
0.268	0.719	0.771	0.95
0.208	0.812	0.775	0.83
0.268	0.822	0.775	0.95
0.201	0.794	0.777	0.74
0.2	0.9	0.78	0.74
0.268	0.058	0.788	0.96
0.248	0.06	0.795	0.84

The Petri net that corresponds to the last set of parameters is the following:

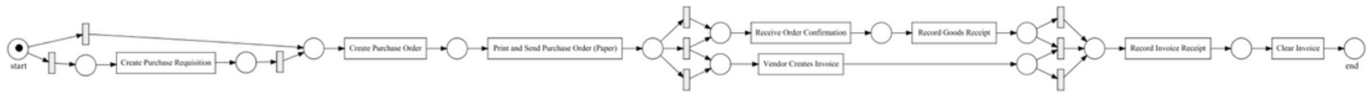
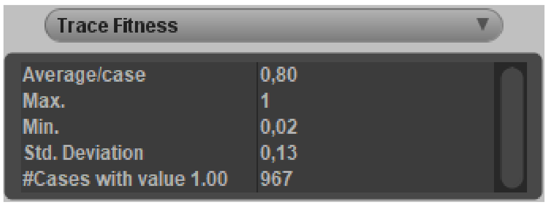


Figure 9. Heuristic Miner model through the Graphviz Petri net visualization of ProM

The fitness and precision scores are shown in figure 10.



Average/case	0,80
Max.	1
Min.	0,02
Std. Deviation	0,13
#Cases with value 1.00	967

Precision : 0,84326

Generalization : 0,99999

Figure 10. Fitness score and Precision score for the Heuristic Miner model

INDUCTIVE VISUAL MINER

In order to find the best model through the Inductive Visual Miner, we apply the action *Mine with Inductive Visual Miner* to the Event log and we change the parameters of activities and paths (the miner is the default miner IMf). If we lower the parameter of activities, we lower the number of activities that will result in the Petri net, while if we decrease the parameter of paths, we lower the connections between activities in the resulting Petri net. If we try to set both parameters to high values, we will have some problems in the computation of the precision score, because the action *Measure Precision/Generalization* does not stop working. We start to see some results when we set the activities parameter to 0.5 with a not too high value of the paths parameter.
We achieve the best model by setting the activities parameter to 0.309 and the paths parameter to 0.565. The following is the resulting Petri net:

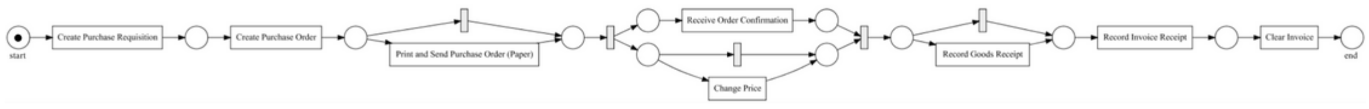
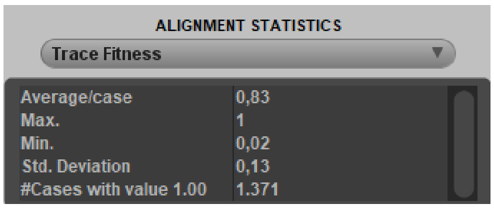


Figure 11. Inductive Visual Miner model through the Graphviz Petri net visualization of ProM

The fitness and precision scores are computed with the usual procedure and are shown in figure 12.



Average/case	0,83
Max.	1
Min.	0,02
Std. Deviation	0,13
#Cases with value 1.00	1.371

Precision : 0,86990

Generalization : 1,00000

Figure 12. Fitness score and Precision score for the Inductive Visual Miner model

ALPHA MINER

At last we discover a model using the Alpha Miner. In order to do so we perform on the Event log the action *ILP-Based Process Discovery:* we select the basic level of configurations and we set the miner to Alpha. The action returns the Petri net shown in figure 13:

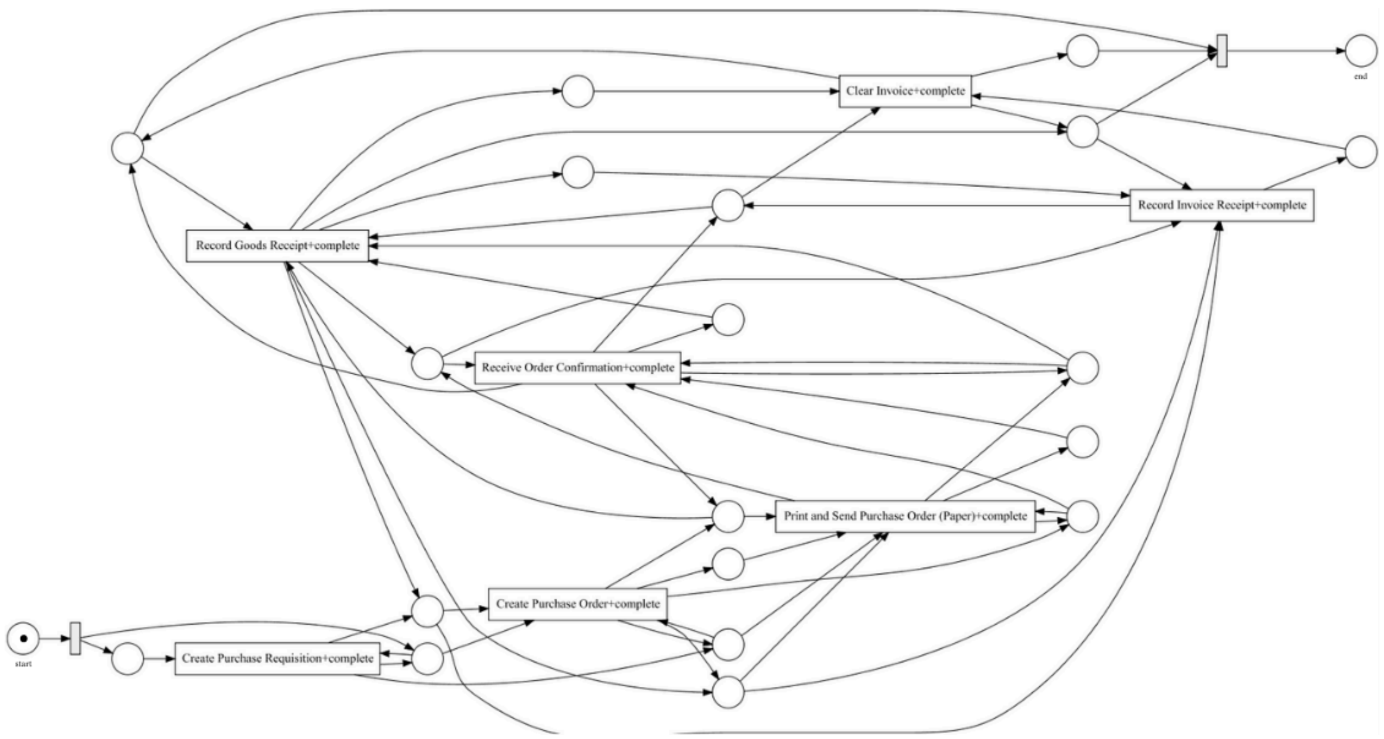
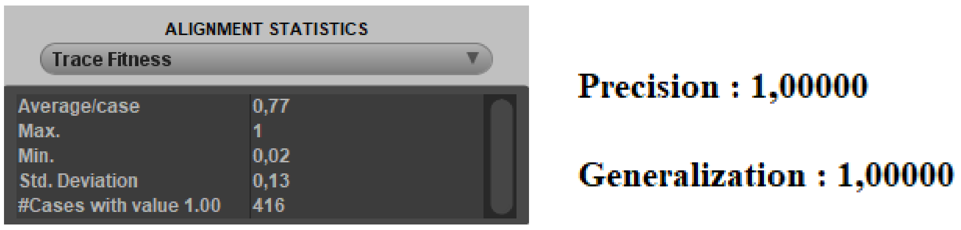

Clear Invoice+complete
Record Invoice Receipt+complete
Record Goods Receipt+complete
Receive Order Confirmation+complete
Print and Send Purchase Order (Paper)+complete
Create Purchase Order+complete
Create Purchase Requisition+complete
start
end

Figure 13. Alpha Miner model through the Graphviz Petri net visualization of ProM

The fitness score and precision score are computed in the usual way, and can be found in figure 14.



ALIGNMENT STATISTICS

Trace Fitness

Average/case	0,77
Max.	1
Min.	0,02
Std. Deviation	0,13
#Cases with value 1.00	416

Precision : 1,00000

Generalization : 1,00000

Figure 14. Fitness score and Precision score for the Alpha Miner model

As we can see, the Alpha Miner model has a very high precision, but not a good enough fitness score.

CONCLUSIONS

The best model among the ones that we studied is the model discovered through the Inductive Visual Miner, with a score of 0.83 for fitness and of 0.87 for precision.
If we compare this last mentioned model with the model that we designed in the previous paragraph (see figure 7) we can notice some differences that are the reasons why we have, in this last model obtained through the Inductive Visual Miner, a better fitness score but a worse precision score.
We will refer to the model that we designed as model 1 and the model discovered with the Inductive Visual Miner as model 2 for an easier notation.
We list here the differences between the two models:

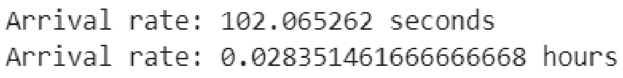
- In model 1 we have the option to skip the *Create Purchase Requisition* activity and go directly to the *Create Purchase Order Activity* (that is what is done for small orders as indicated in the description of the process) while in model 2 we don't have this possibility.
- In model 2 there is not the possibility to further investigate the purchase order since the activities *Block Purchase Order* and *Reactivate Purchase Order* are not present, instead in model 1 this eventuality is represented.
- In model 2 we can perform the activity *Change Price* before the activity *Receive Order Confirmation* because the two transitions are in parallel, while in model 1 we can perform the *Change Price* activity only after the *Receive Order Confirmation*.
- Another activity that is not in model 2 while is present in model 1 is the *Send Purchase Order Update* and the same applies to the *Cancel Goods Receipt* activity.
- In model 2 we have the option to skip the activity *Record Goods Receipt* while in model 1 there is not such a possibility.

SIMULATION PARAMETERS AND SIMULATION

For the next task, we delve into the details related to resources and their roles. Using the ProM tool and PM4Py library, we compute the case arrival rate, the branching probabilities, the tasks durations and the resource roles and calendars. Then we use the information extrapolated to create a simulation model through the BIMP simulator.

CASE ARRIVAL RATE

Using PM4Py we compute the mean of the daily arrival rate, that is the interval between two started process instances. We report here the results obtained:



```
Arrival rate: 102.065262 seconds
Arrival rate: 0.028351461666666668 hours
```

Figure 15. Daily arrival rate in seconds and hours

For a better visualization, we plot the distribution of case arrivals throughout the week. As we can see from figure 16, on the weekend we have very few case arrivals, so we hypothesize that the business week runs from Monday to Friday.

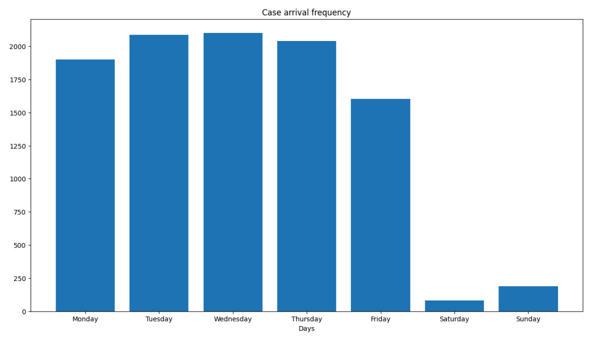
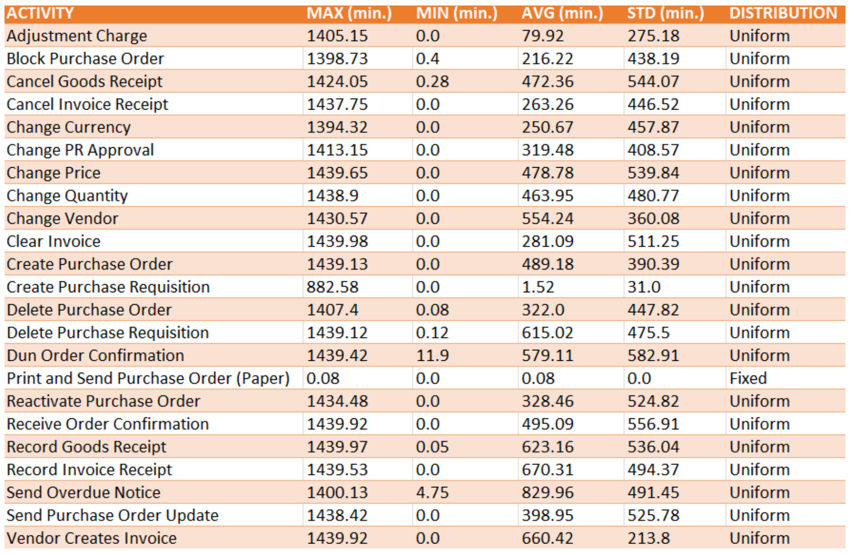


Figure 16. Case arrival distribution over the week

TASKS DURATION

To understand the duration of each task, we printed the statistics in minutes of each activity, that includes the maximum, minimum, average and standard deviation of the duration of each task. We are then going to use these results for the simulation's parameters of BIMP. We report in figure 17 the outcomes of the statistics. The last column of the table represents the type of distribution of each activity, that we found out through the plotting of the distributions.

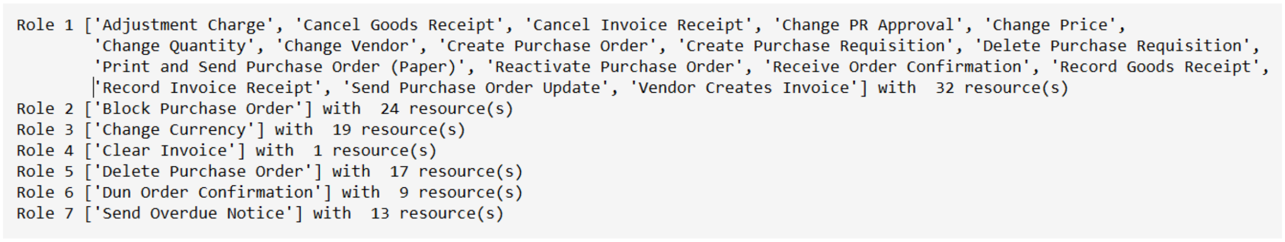


ACTIVITY	MAX (min.)	MIN (min.)	AVG (min.)	STD (min.)	DISTRIBUTION
Adjustment Charge	1405.15	0.0	79.92	275.18	Uniform
Block Purchase Order	1398.73	0.4	216.22	438.19	Uniform
Cancel Goods Receipt	1424.05	0.28	472.36	544.07	Uniform
Cancel Invoice Receipt	1437.75	0.0	263.26	446.52	Uniform
Change Currency	1394.32	0.0	250.67	457.87	Uniform
Change PR Approval	1413.15	0.0	319.48	408.57	Uniform
Change Price	1439.65	0.0	478.78	539.84	Uniform
Change Quantity	1438.9	0.0	463.95	480.77	Uniform
Change Vendor	1430.57	0.0	554.24	360.08	Uniform
Clear Invoice	1439.98	0.0	281.09	511.25	Uniform
Create Purchase Order	1439.13	0.0	489.18	390.39	Uniform
Create Purchase Requisition	882.58	0.0	1.52	31.0	Uniform
Delete Purchase Order	1407.4	0.08	322.0	447.82	Uniform
Delete Purchase Requisition	1439.12	0.12	615.02	475.5	Uniform
Dun Order Confirmation	1439.42	11.9	579.11	582.91	Uniform
Print and Send Purchase Order (Paper)	0.08	0.0	0.08	0.0	Fixed
Reactivate Purchase Order	1434.48	0.0	328.46	524.82	Uniform
Receive Order Confirmation	1439.92	0.0	495.09	556.91	Uniform
Record Goods Receipt	1439.97	0.05	623.16	536.04	Uniform
Record Invoice Receipt	1439.53	0.0	670.31	494.37	Uniform
Send Overdue Notice	1400.13	4.75	829.96	491.45	Uniform
Send Purchase Order Update	1438.42	0.0	398.95	525.78	Uniform
Vendor Creates Invoice	1439.92	0.0	660.42	213.8	Uniform

Figure 17. Table of statistics of the duration of each activity

RESOURCE ROLES AND CALENDARS

We want to find out the roles of the process and the number of resources that work in each one. With some code, we find out that there are 7 different roles, that are represented in figure 18, along with the number of resources for each one.



```
Role 1 ['Adjustment Charge', 'Cancel Goods Receipt', 'Cancel Invoice Receipt', 'Change PR Approval', 'Change Price',
        'Change Quantity', 'Change Vendor', 'Create Purchase Order', 'Create Purchase Requisition', 'Delete Purchase Requisition',
        'Print and Send Purchase Order (Paper)', 'Reactivate Purchase Order', 'Receive Order Confirmation', 'Record Goods Receipt',
        'Record Invoice Receipt', 'Send Purchase Order Update', 'Vendor Creates Invoice'] with  32 resource(s)
Role 2 ['Block Purchase Order'] with  24 resource(s)
Role 3 ['Change Currency'] with  19 resource(s)
Role 4 ['Clear Invoice'] with  1 resource(s)
Role 5 ['Delete Purchase Order'] with  17 resource(s)
Role 6 ['Dun Order Confirmation'] with  9 resource(s)
Role 7 ['Send Overdue Notice'] with  13 resource(s)
```

Figure 18. Roles with the associated number of resources

We also compute the work schedule of each role. All the roles have a very similar work timetable, except for role 4 that is slightly different. In figure 19 we show the work schedule for role 1 (that also represents the other roles) and role 4, both in the form of daily working time and working hours.

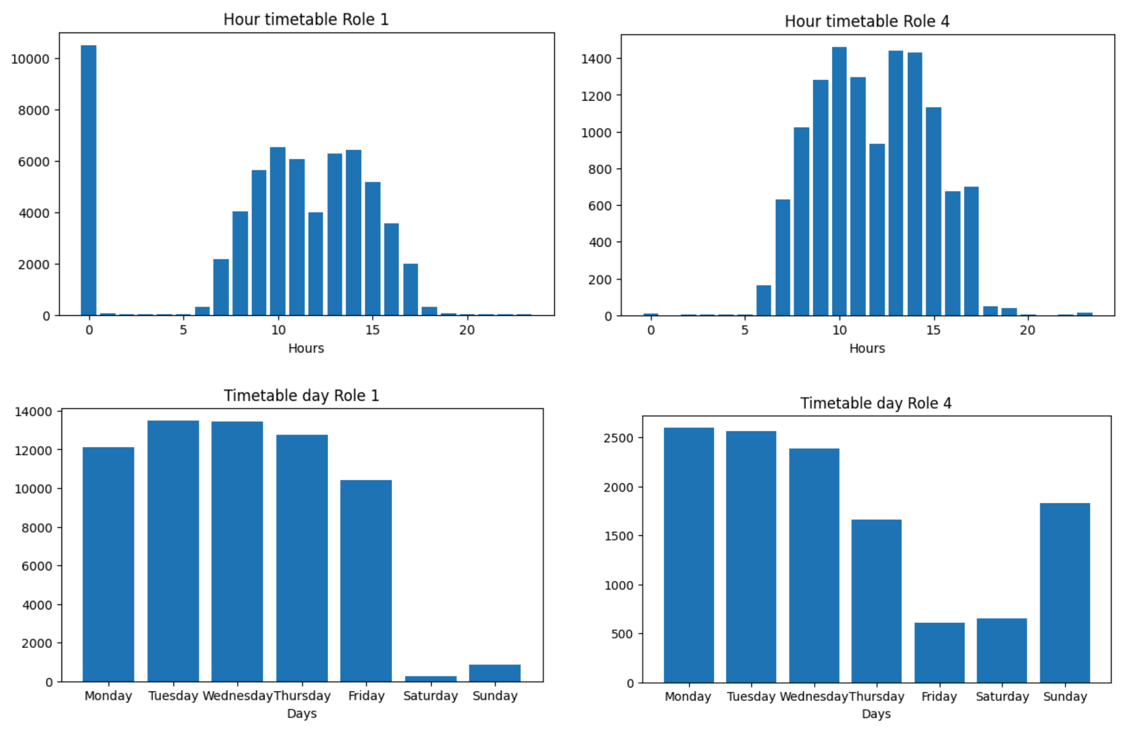


Figure 19. Histograms of working hours and time for Role 1 and Role 4

BRANCHING PROBABILITIES

In order to retrieve the branching probabilities, we need the Petri net that we are going to convert to a BPMN diagram to use in the BIMP simulator. The Petri net that we use is the one discovered through the action *Mine with Inductive Visual Miner* with the standard parameter from our Event log. In this way, we use a Petri net that has all the activities, whereas if we used one of the Petri nets discovered in the previous paragraph we would not have all the activities present. To obtain the branching probabilities we use the action *Multi-perspective Process Explorer,* we set the MPE MODE to *Show Performance Mode* and in the configuration panel we choose the 1st measure to be *Percentage (local).* The following is the Petri net above mentioned with its branching probabilities:

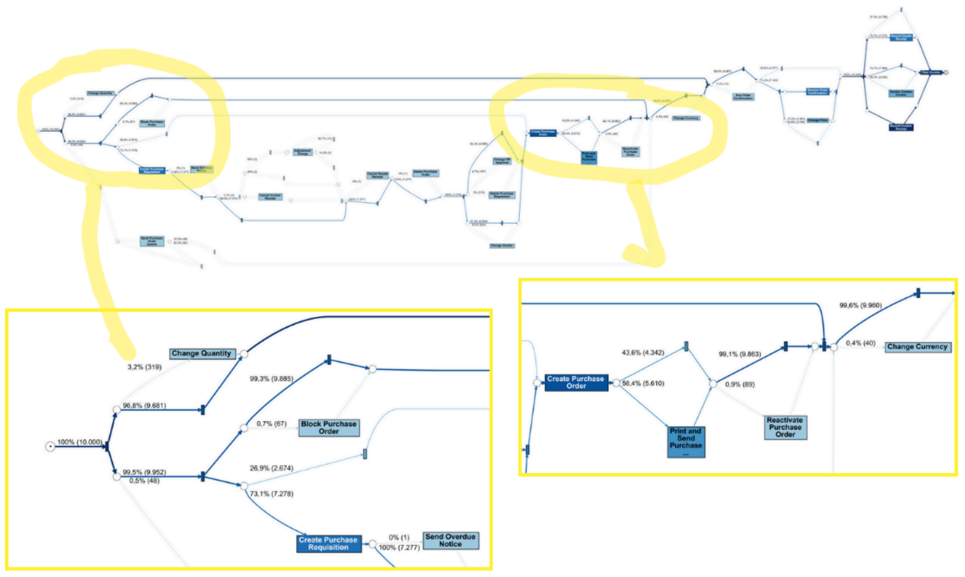


Figure 20. Branching probabilities with zoom for better visual

SIMULATION

To begin our simulation we need a BPMN model: starting from the Petri net discovered before (see figure 20) we use the Prom tool to convert it to a BPMN model using the action *Convert Petri net to BPMN diagram.* The resulting BPMN model that we are going to use to build our simulation is represented in figure 21.

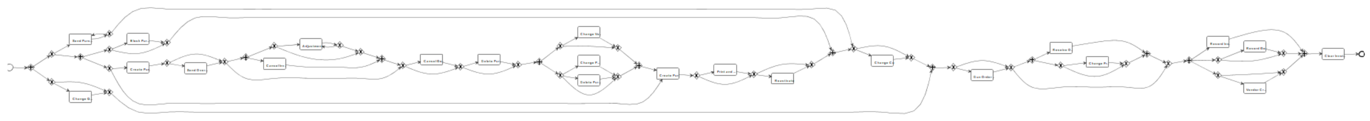
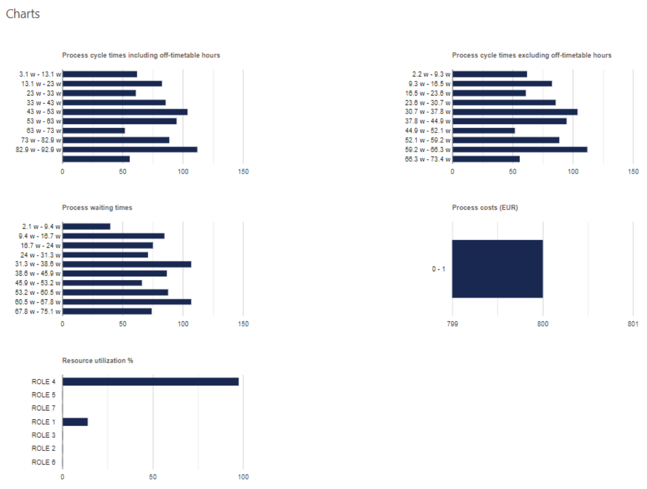
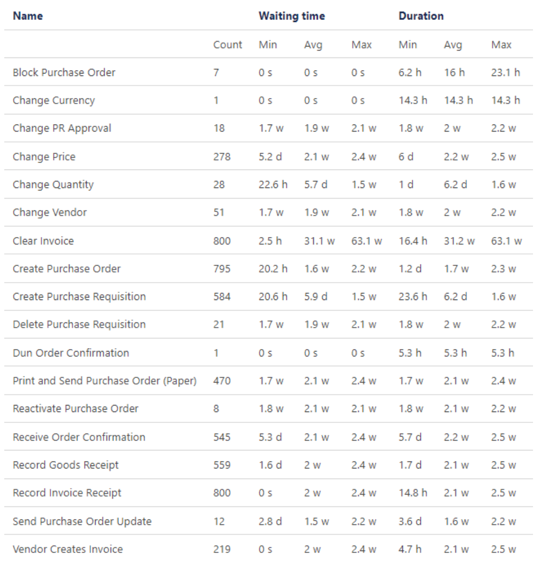


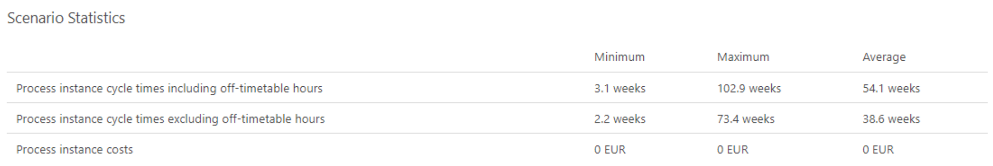
Figure 21. BPMN model converted from Petri net

Since we have retrieved all the different simulation parameters and we have obtained the BPMN model, we are now ready to build our simulation model.
We start by uploading into the BIMP simulator our BPMN file. We set the Inter arrival time to Exponential with an average of 102.06 seconds (see figure 15). We set the total number of process instances to 1000 with a warm-up period of 10% of the traces (both for the start and for the tail), just to start with the first simulation. Indeed, if we try already with 10000 instances with this BPMN model, the BIMP simulator returns the error *Maximum task queue size limit reached for a resource.*
We then set the roles with their number of resources as in figure 18. We assume that all resources in each role work all day (from 00.00 to 23.59) from Monday to Friday: this assumption is due to the case arrival distribution shown in figure 16.
We then fill the parameters asked for each task: to which role they belong and the distribution. We can find this information in figure 18 and figure 17.
The last parameters to add are the one related to the exclusive gateways and their probabilities, that we can find thanks to figure 20 that represents the branching probabilities.
Once the scenario is set up, we can start the simulation. We ran the same scenario at least 5 times to take care of the stochasticity of the simulation.
We are now going to show the results that we obtained with this first scenario in one of the simulations launched.





Name		Waiting time			Duration		
	Count	Min	Avg	Max	Min	Avg	Max
Block Purchase Order	7	0 s	0 s	0 s	6.2 h	16 h	23.1 h
Change Currency	1	0 s	0 s	0 s	14.3 h	14.3 h	14.3 h
Change PR Approval	18	1.7 w	1.9 w	2.1 w	1.8 w	2 w	2.2 w
Change Price	278	5.2 d	2.1 w	2.4 w	6 d	2.2 w	2.5 w
Change Quantity	28	22.6 h	5.7 d	1.5 w	1 d	6.2 d	1.6 w
Change Vendor	51	1.7 w	1.9 w	2.1 w	1.8 w	2 w	2.2 w
Clear Invoice	800	2.5 h	31.1 w	63.1 w	16.4 h	31.2 w	63.1 w
Create Purchase Order	795	20.2 h	1.6 w	2.2 w	1.2 d	1.7 w	2.3 w
Create Purchase Requisition	584	20.6 h	5.9 d	1.5 w	23.6 h	6.2 d	1.6 w
Delete Purchase Requisition	21	1.7 w	1.9 w	2.1 w	1.8 w	2 w	2.2 w
Dun Order Confirmation	1	0 s	0 s	0 s	5.3 h	5.3 h	5.3 h
Print and Send Purchase Order (Paper)	470	1.7 w	2.1 w	2.4 w	1.7 w	2.1 w	2.4 w
Reactivate Purchase Order	8	1.8 w	2.1 w	2.1 w	1.8 w	2.1 w	2.2 w
Receive Order Confirmation	545	5.3 d	2.1 w	2.4 w	5.7 d	2.2 w	2.5 w
Record Goods Receipt	559	1.6 d	2 w	2.4 w	1.7 d	2.1 w	2.5 w
Record Invoice Receipt	800	0 s	2 w	2.4 w	14.8 h	2.1 w	2.5 w
Send Purchase Order Update	12	2.8 d	1.5 w	2.2 w	3.6 d	1.6 w	2.2 w
Vendor Creates Invoice	219	0 s	2 w	2.4 w	4.7 h	2.1 w	2.5 w



Scenario Statistics

	Minimum	Maximum	Average
Process instance cycle times including off-timetable hours	3.1 weeks	102.9 weeks	54.1 weeks
Process instance cycle times excluding off-timetable hours	2.2 weeks	73.4 weeks	38.6 weeks
Process instance costs	0 EUR	0 EUR	0 EUR

Figure 22. Results of a simulation of the first scenario

As we can see from these results, we have some cases in which the process takes up to 73.4 weeks (in the process instance cycle times excluding off-timetable hours) and if we look at the table with all the activities we can see that the average waiting time for most of the activities is more than 1 week. Furthermore, looking at the resource utilization percentage chart, we can see that only the first and fourth roles are used, and ROLE 1 in particular is overloaded with work. To further investigate which are the bottlenecks of the process and which are the activities that take the most time to be completed, we upload on the Disco tool the MXML log that results from this simulation. We report in figure 23 the map of the process colored by mean duration and a screenshot of the animation of the process where we individuated bottlenecks.

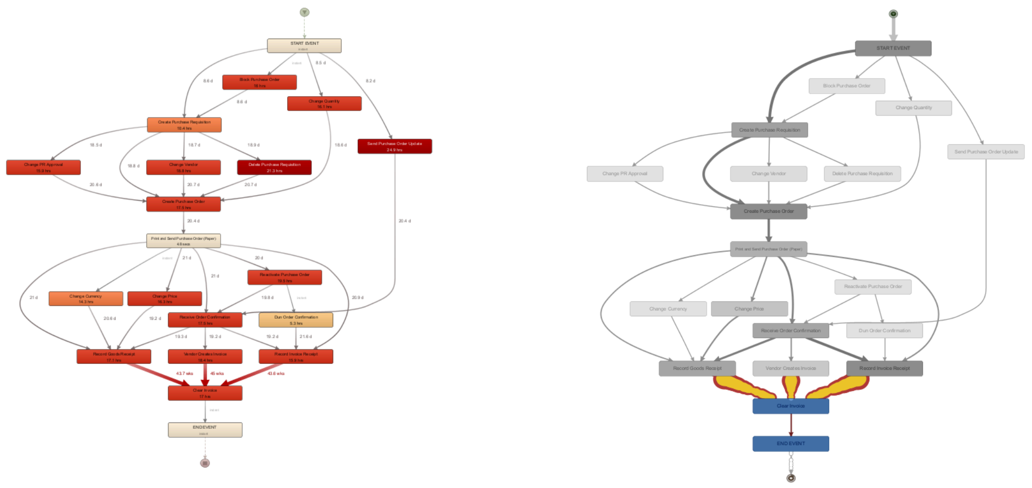


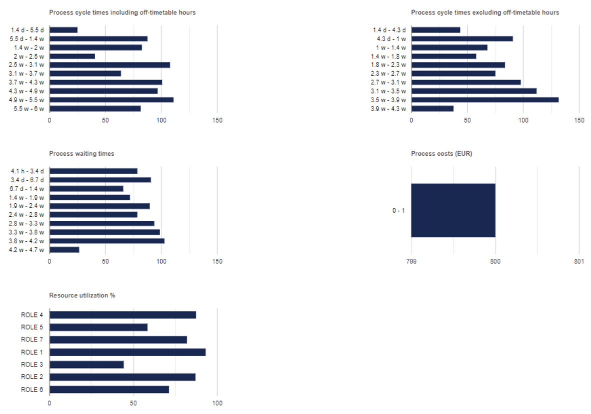
Figure 23. Map of the process with mean duration and bottlenecks

To improve our process, we try changing the way that the resources are attributed to each role and also the activities that are included in each role. Indeed, the ROLE 1 has too much work and therefore we have to reassign some of the activities that are performed by that role to other roles. In particular, from our analysis we have found out that a useful change to make is to assign the activity *Create Purchase Order* and the activity *Create Purchase Requisition* to a role that does only that activity. The parameters that we modified and that gave us the best results for the process so far are the ones shown in figure 27.

```
Role 1 ['Adjustment Charge', 'Cancel Goods Receipt', 'Cancel Invoice Receipt', 'Change Quantity', 'Delete Purchase Requisition',
        'Record Invoice Receipt', 'Vendor Creates Invoice'] with  22 resource(s)
Role 2 ['Block Purchase Order', 'Record Goods Receipt'] with  14 resource(s)
Role 3 ['Create Purchase Requisition'] with  16 resource(s)
Role 4 ['Change Price', 'Clear Invoice'] with  24 resource(s)
Role 5 ['Change PR Approval','Change Vendor', 'Delete Purchase Order', 'Print and Send Purchase Order (Paper)',
        'Reactivate Purchase Order', 'Send Purchase Order Update'] with 3 resource(s)
Role 6 ['Create Purchase Order'] with  22 resource(s)
Role 7 ['Change Currency','Dun Order Confirmation', 'Receive Order Confirmation',  'Send Overdue Notice'] with  14 resource(s)
```

Figure 24. Changed roles with the associated number of resources

With this configuration of the parameters, we report here the results of a simulation with 1000 instances, only to compare it with the first simulation of figure 22. We can see that now the maximum process instance cycle times excluding off-timetable hours is around 4.3 weeks, and no activity has a waiting time on average of more than a week. Moreover, the percentage of resource utilization is more balanced.

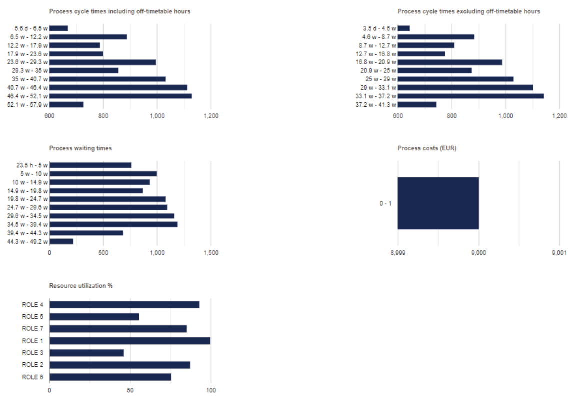


Name		Waiting time			Duration		
	Count	Min	Avg	Max	Min	Avg	Max
Block Purchase Order	4	0 s	0 s	0 s	3.2 h	7.5 h	11 h
Cancel Invoice Receipt	2	9.8 h	16.8 h	23.8 h	14.1 h	21.3 h	1.2 d
Change Currency	5	2.6 h	5.1 h	8.8 h	4.4 h	13.6 h	1 d
Change PR Approval	11	1.7 d	2.8 d	3.8 d	2.4 d	3.4 d	4.3 d
Change Price	265	0 s	10.4 h	1.5 d	12.9 m	22.1 h	2.4 d
Change Quantity	23	0 s	0 s	0 s	1.3 h	10.6 h	22.9 h
Change Vendor	57	1.4 d	2.9 d	3.9 d	1.9 d	3.3 d	4.7 d
Clear Invoice	800	0 s	12.3 h	1.6 d	3.5 m	1 d	2.6 d
Create Purchase Order	797	2.5 h	5.3 d	1.1 w	7.4 h	5.8 d	1.2 w
Create Purchase Requisition	588	1.2 d	6.4 d	1.6 w	1.2 d	6.7 d	1.7 w
Delete Purchase Requisition	14	16 m	21.4 h	1.5 d	11.9 h	1.4 d	2.2 d
Dun Order Confirmation	2	11.5 h	16 h	20.6 h	18.2 h	1.2 d	1.7 d
Print and Send Purchase Order (Paper)	453	0 s	2.1 d	4.3 d	4.8 s	2.1 d	4.3 d
Reactivate Purchase Order	7	0 s	2.4 d	3.8 d	22.8 h	3.1 d	4.6 d
Receive Order Confirmation	571	0 s	1.2 d	3.8 d	18.7 m	1.7 d	4.8 d
Record Goods Receipt	572	0 s	1.2 d	2.9 d	16.6 s	1.7 d	3.9 d
Record Invoice Receipt	800	0 s	1.8 d	3.7 d	17.3 m	2.3 d	4.5 d
Send Purchase Order Update	8	0 s	1.5 d	3.4 d	12.2 h	2.2 d	4.4 d
Vendor Creates Invoice	212	0 s	1.8 d	3.7 d	1.6 h	2.4 d	4.5 d

	Minimum	Maximum	Average
Process instance cycle times including off-timetable hours	1.4 days	6 weeks	3.4 weeks
Process instance cycle times excluding off-timetable hours	1.4 days	4.3 weeks	2.4 weeks
Process instance costs	0 EUR	0 EUR	0 EUR

Figure 25. Results of a simulation of the scenario with changed roles 1000 instances

After uploading the last MXML log on Disco, if we animate the process, no particular bottlenecks are found.
However, only 1000 instances for a simulation are not enough to really grasp the complexity of the process. Indeed, we need to try a simulation with 10000 instances, that is the maximum number that the BIMP simulator allows us to try. We set the parameter of the percentage of warm-up to 5% (both for the start and the tail).
The results of a simulation with 10000 instances with the parameters of figure 24 are shown in figure 26.



Name		Waiting time			Duration		
	Count	Min	Avg	Max	Min	Avg	Max
Adjustment Charge	9	1 w	1.7 w	3.1 w	1.1 w	1.8 w	3.2 w
Block Purchase Order	56	0 s	17 h	1.2 d	2.3 h	1.1 d	2 d
Cancel Invoice Receipt	5	6.8 d	1.4 w	2 w	1.1 w	1.5 w	2.1 w
Change Currency	38	0 s	1.7 w	4.1 w	2.9 h	1.8 w	4.2 w
Change PR Approval	176	6.3 d	2.8 w	3.2 w	1 w	2.9 w	3.3 w
Change Price	3129	0 s	2.5 d	1.1 w	18.7 m	3 d	1.2 w
Change Quantity	294	0 s	3.2 d	5.7 d	7.3 h	3.7 d	6.4 d
Change Vendor	543	6.1 d	2.8 w	3.2 w	6.2 d	2.8 w	3.4 w
Clear Invoice	9000	0 s	2.5 d	1.1 w	41.5 s	3 d	1.3 w
Create Purchase Order	8956	2.5 d	8.6 w	12 w	2.6 d	8.7 w	12.1 w
Create Purchase Requisition	6482	6.3 d	9 w	17.1 w	6.4 d	9.1 w	17.2 w
Delete Purchase Requisition	203	3.9 d	2 w	3.6 w	4.3 d	2.1 w	3.8 w
Dun Order Confirmation	6	1 d	2.8 w	4.4 w	1.1 d	2.9 w	4.5 w
Print and Send Purchase Order (Paper)	5037	0 s	1.9 w	3.3 w	4.8 s	1.9 w	3.3 w
Reactivate Purchase Order	69	0 s	1.8 w	3.2 w	8.4 m	1.8 w	3.3 w
Receive Order Confirmation	6466	0 s	1.7 w	4.8 w	6.7 s	1.8 w	5 w
Record Goods Receipt	6476	0 s	5.9 d	1.8 w	55.1 s	6.4 d	1.9 w
Record Invoice Receipt	9000	23.5 h	3.6 w	6.3 w	1.6 d	3.6 w	6.4 w
Send Purchase Order Update	108	0 s	1.4 w	3.2 w	8.7 h	1.5 w	3.2 w
Vendor Creates Invoice	2301	2.2 d	3.6 w	6.3 w	2.2 d	3.6 w	6.4 w

	Minimum	Maximum	Average
Process instance cycle times including off-timetable hours	5.6 days	57.9 weeks	30.7 weeks
Process instance cycle times excluding off-timetable hours	3.5 days	41.3 weeks	21.9 weeks
Process instance costs	0 EUR	0 EUR	0 EUR

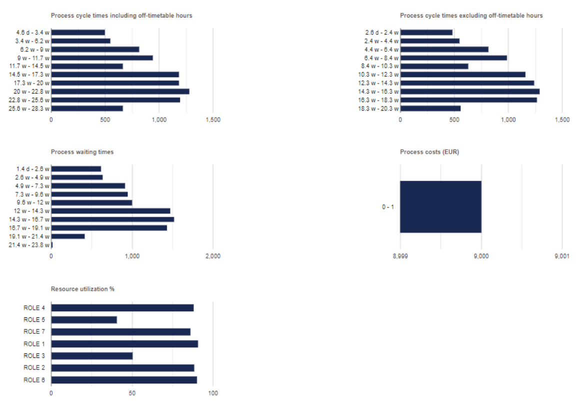
Figure 26. Results of a simulation of the scenario with changed roles 10000 instances

With 10000 instances, we have an average duration of 21.9 weeks. However, we can see that the percentage of role utilization of ROLE 1 is very close to 100% and this cannot happen in an optimized process. Since just rearranging the roles and the resources was not enough to have an optimized process, we are going to increase the number of employees in order to have all the roles with a better utilization percentage. We decide to hire new employees and reassign the resources to new roles as shown in figure 27. We could have decided to employ even more resources, but each resource is a new salary so we do not want to spend too much money.

```
Role 1 ['Adjustment Charge', 'Cancel Goods Receipt', 'Cancel Invoice Receipt',  'Delete Purchase Requisition',
        'Record Invoice Receipt', 'Vendor Creates Invoice'] with  47 resource(s)
Role 2 ['Record Goods Receipt'] with  27 resource(s)
Role 3 ['Create Purchase Requisition'] with  29 resource(s)
Role 4 ['Change Price','Change Vendor','Clear Invoice'] with  53 resource(s)
Role 5 ['Block Purchase Order','Change Currency','Change PR Approval','Change Quantity', 'Delete Purchase Order',
        'Dun Order Confirmation', 'Print and Send Purchase Order (Paper)',  'Reactivate Purchase Order',
        'Send Purchase Order Update'] with  7 resource(s)
Role 6 ['Create Purchase Order'] with  37 resource(s)
Role 7 ['Receive Order Confirmation', 'Send Overdue Notice'] with  27 resource(s)
```

Figure 27. Table of number of resources for each role

With this configuration of the parameters, we have the following results for a simulation of 10000 instances:



Name		Waiting time			Duration		
	Count	Min	Avg	Max	Min	Avg	Max
Adjustment Charge	24	0 s	5.2 d	6.7 d	2.7 h	5.7 d	1.1 w
Block Purchase Order	60	18.7 h	1.4 w	3 w	22.4 h	1.4 w	3 w
Cancel Invoice Receipt	10	0 s	3.6 d	5.8 d	7.3 h	4.1 d	6.1 d
Change Currency	49	0 s	4.6 d	3 w	1.5 h	5.1 d	3.1 w
Change PR Approval	181	0 s	6.6 d	3.1 w	5 m	1 w	3.1 w
Change Price	3080	0 s	1.8 d	3.4 d	5.4 m	2.3 d	4.4 d
Change Quantity	291	17.1 h	1.5 w	3 w	1.4 d	1.6 w	3.2 w
Change Vendor	552	0 s	1.4 d	3.4 d	6.6 m	1.9 d	4.3 d
Clear Invoice	9000	0 s	1.7 d	3.4 d	13.6 s	2.2 d	4.4 d
Create Purchase Order	8955	1.4 d	5.7 w	8.4 w	1.7 d	5.8 w	8.5 w
Create Purchase Requisition	6539	3 d	4.6 w	8.8 w	3.1 d	4.6 w	8.9 w
Delete Purchase Requisition	215	0 s	2.3 d	6 d	37.7 m	2.8 d	7 d
Dun Order Confirmation	8	0 s	20.7 h	4.8 d	4.3 h	1.4 d	5 d
Print and Send Purchase Order (Paper)	5145	0 s	3.6 d	3.1 w	4.8 s	3.6 d	3.1 w
Reactivate Purchase Order	75	0 s	2.9 d	2.7 w	8.3 m	3.4 d	2.8 w
Receive Order Confirmation	6442	0 s	1.1 w	1.7 w	13.7 s	1.2 w	1.8 w
Record Goods Receipt	6486	0 s	3.1 d	4.4 d	1.9 m	3.6 d	5.4 d
Record Invoice Receipt	9000	0 s	4.7 d	6.8 d	2.6 m	5.2 d	1.1 w
Send Purchase Order Update	124	0 s	1.4 w	3.1 w	3.7 h	1.5 w	3.2 w
Vendor Creates Invoice	2375	0 s	4.7 d	6.8 d	16.6 s	5.2 d	1.1 w

	Minimum	Maximum	Average
Process instance cycle times including off-timetable hours	4.6 days	28.3 weeks	15.9 weeks
Process instance cycle times excluding off-timetable hours	2.6 days	20.3 weeks	11.4 weeks
Process instance costs	0 EUR	0 EUR	0 EUR

Figure 28. Results of a simulation of the final scenario with 10000 instances

Even though there are still some long waiting times for some activities, this configuration is the best we have found to balance the percentage of utilization of each role (in average no role

surpasses the 90% of utilization) and the average duration of a process. We want to point out that we have arrived at an average duration of around 3 months, that is much less than the average duration of the first simulation that, by the way, had only 1000 instances.
To summarize, what we would advise the company to do is to better define the roles of their employees, dividing the activities in each role in a more performant way in order to save time and therefore money in the process and to hire more resources to have a faster process.