Homework 2

Miriam Wagner
373045

June 29, 2018

Contents

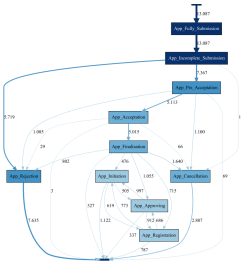
1 Process Models **1**
1.1 The model of the application's lifecycle 1
1.2 The model of the proposal's lifecycle 4
1.3 Combinded Model 5
1.3.1 Endstate APP_Approved 5
1.3.2 Endstate APP_Cancelled 5
1.3.3 Endstate APP_Rejected 6
1.3.4 All 3 Models 6
1.4 C-net of the proposal process 6
1.4.1 Analysis of the C-net 6
1.5 Own Petri net of the proposal process 7
1.6 Analysis of the performance of Application and work process 7

Introduction

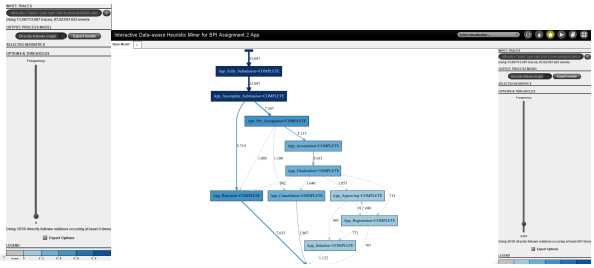
1 Process Models

1.1 The model of the application's lifecycle

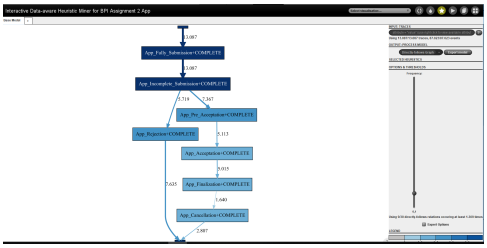
For modeling the application lifecycle first the data has been filtered. Just the events beginning with "App_..." are required. This data set is saved by the name "Filtered App".
The first try I choose the frequency 0.1 and had a look at the directly followed graph. I also checked 0.2 and 0.51. For comparsion in the end I checked also the original directly followed graph. My first choice was the directly followed graph with 0.1 as threhold for frequency, because it was a simple model that still tells us a lot about the main process. In figure 1 the 4 considered directly followed graphs can be seen. Obviously the original graph does not fullfill the criterium of simplicity and also the graph with frequency 0.051 still looks not as simpel as I would like.
In the next step I checked the conformance of the corresponding petri net, which I exported from the Interactive Data-aware Heuristic Miner, by combining the data and the petri net for the conformance checking with the Replay tool for Conformance checking. The results in 2 showed me, that the model with 0.2 also has the same conformance than 0.1, what is not surprising, because they have the same petri-net. Based on this and the fact, that the conformance of 0.1 filtered is still not so bad I considered 0.051 and 0.1 for the precision check.
Applying the Multi-perspective Process Explorer and choosing "show precision mode".
Checking the precision, 4, and combine it with the results before, I came to the conclusion, that 0.1 is not good enough as model and 0.051 is good enough, but too complicated. Starting by this I again tried different frequency filters outgoing by 0.075 to find a model, which has a similar simplicity than the 0.1 frequency model, but a better conformance and precision. And already the frequency filtering 0.076 gives me the wished result.
This model has a good simplicity, but still has a path fitness of 96.58% and precision of 94.8%, so it is still pretty good. Overall I so decided to choose the 0.076 frequency model for the application's lifecycle.



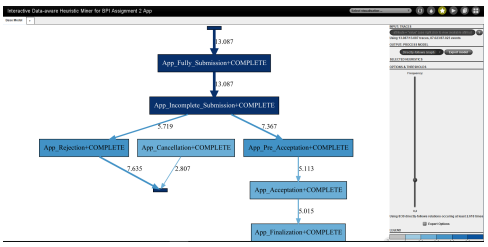
(a) Directly followed graph without frequency filtering



(b) Directly followed graph 0.051 frequency

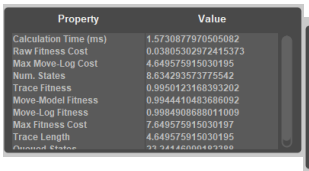


(c) Directly followed graph 0.1 frequency



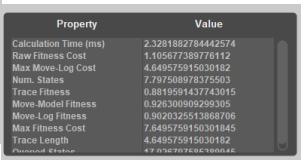
(d) Directly followed graph 0.2 frequency

Figure 1: Considered directly followed graphs



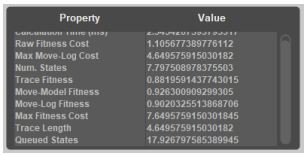
Property	Value
Calculation Time (ms)	1.5730877970505082
Raw Fitness Cost	0.038053029724153 73
Max Move-Log Cost	4.649575915030195
Num. States	8.634293573775542
Trace Fitness	0.9950123168393202
Move-Model Fitness	0.9944410483686092
Move-Log Fitness	0.9984908688011009
Max Fitness Cost	7.649575915030197
Trace Length	4.649575915030195

(a) Conformance for 0.051 frequency



Property	Value
Calculation Time (ms)	2.3281882784442574
Raw Fitness Cost	1.105677389776112
Max Move-Log Cost	4.649575915030182
Num. States	7.797508978375503
Trace Fitness	0.8819591437743015
Move-Model Fitness	0.926300909299305
Move-Log Fitness	0.9020325513868706
Max Fitness Cost	7.649575915030845
Trace Length	4.649575915030182

(b) Conformance 0.1



Property	Value
Raw Fitness Cost	1.105677389776112
Max Move-Log Cost	4.649575915030182
Num. States	7.797508978375503
Trace Fitness	0.8819591437743015
Move-Model Fitness	0.926300909299305
Move-Log Fitness	0.9020325513868706
Max Fitness Cost	7.649575915030845
Trace Length	4.649575915030182
Queued States	17.926797585389945

(c) Conformance for 0.2 frequency

Figure 2: Conformance checking

Avg activity precision	100%
# Moves Observed	144.789
# Moves Possible	144.789
Avg fitness	99,6%
% Violations	0,8%
% Event Violations	0,8%
% Data Violations	0%
# Correct Events	60.849
# Wrong Events	0
# Missing Events	498

(a) Precision for 0.051 as frequency

Avg activity precision	88,1%
# Moves Observed	97.074
# Moves Possible	110.161
Avg fitness	88,6%
% Violations	21,7%
% Event Violations	21,7%
% Data Violations	0%
# Correct Events	52.096
# Wrong Events	8.753
# Missing Events	5.717

(b) Precision for 0.1 as frequency

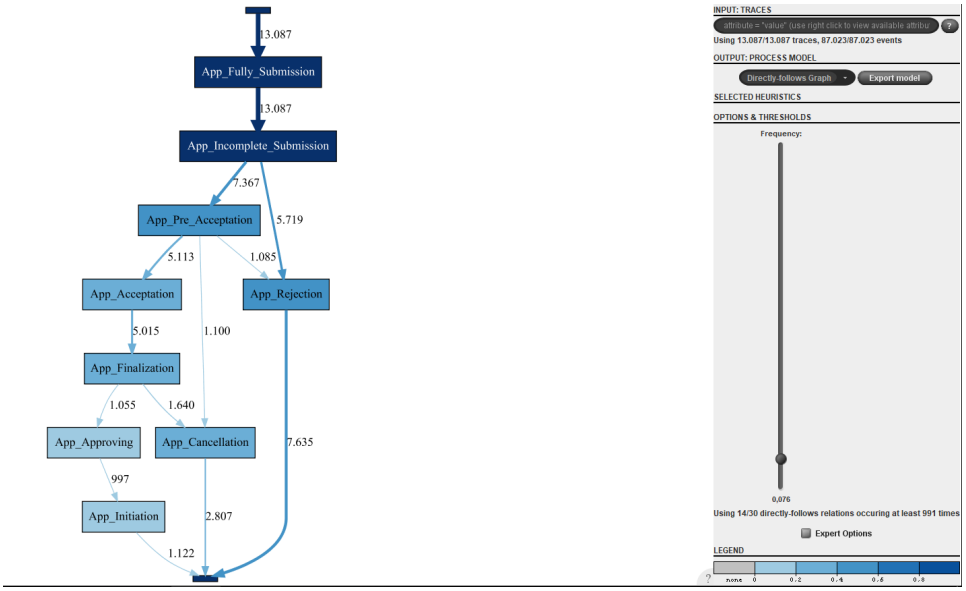
Figure 3: Precision checking

Property	Value
Calculation Time (ms)	2.3191716971039815
Raw Fitness Cost	0.3322380988767475 5
Max Move-Log Cost	4.64957591503021
Num. States	8.79514021548101
Trace Fitness	0.9657921716262099
Move-Model Fitness	0.9944410483686092
Move-Log Fitness	0.9566108861210915
Max Fitness Cost	7.649575915030202
Trace Length	4.64957591503021
Queued States	22.111409267746 71

(a) Conformance

Avg activity precision	94,8%
# Moves Observed	133.936
# Moves Possible	141.304
Avg fitness	96,6%
% Violations	7%
% Event Violations	7%
% Data Violations	0%
# Correct Events	57.772
# Wrong Events	3.077
# Missing Events	1.271

(b) Precision

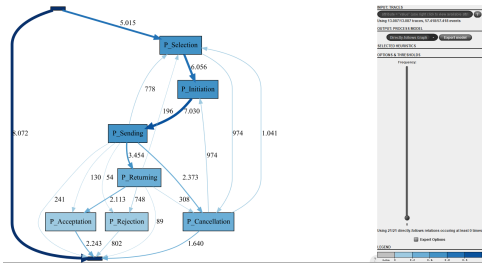


(c) Directly followed graph

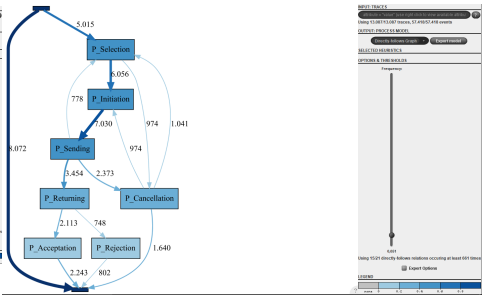
Figure 4: Frequency 0.076

1.2 The model of the proposal's lifecycle

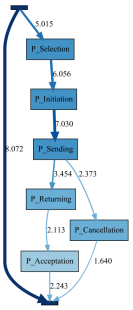
Applying the same steps on the proposal lifecycle gave me first 4 models to have a closer look at.



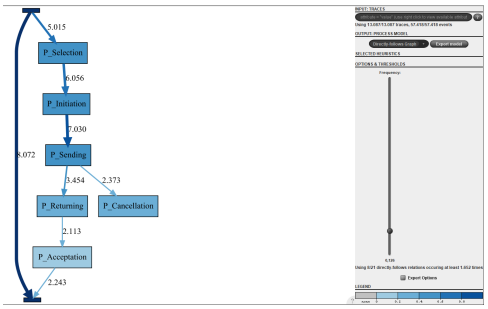
(a) Directly followed graph without frequency filtering



(b) Directly followed graph 0.051 frequency



(c) Directly followed graph 0.1 frequency



(d) Directly followed graph 0.126 frequency

Figure 5: Considered directly followed graphs

Because of simplicity I first checked the conformance and precision just for 0.051 and 0.1 frequency filtering.

Property	Value
Raw Fitness Cost	0.07778711698632224
Max Move-Log Cost	2.3874073508061433
Num. States	6.670130664017713
Trace Fitness	0.9882694969197188
Move-Model Fitness	0.9937748638349697
Move-Log Fitness	0.9959403350088936
Max Fitness Cost	2.3874073508061433
Trace Length	2.3874073508061433
Queued States	16.723542446702886

(a) Conformance for 0.051 frequency

Property	Value
Raw Fitness Cost	0.7597615954764291
Max Move-Log Cost	2.387407350806137
Num. States	6.684114006265772
Trace Fitness	0.9154572453183918
Move-Model Fitness	0.9867960571559562
Move-Log Fitness	0.9298519372518611
Max Fitness Cost	2.387407350806137
Trace Length	2.387407350806137
Queued States	12.33177962863913

(b) Conformance 0.1

Avg activity precision	83,6%
# Moves Observed	103.538
# Moves Possible	123.899
Avg fitness	37,4%
% Violations	3,2%
% Event Violations	3,2%
% Data Violations	0%
# Correct Events	30.740
# Wrong Events	504

(c) Precision for 0.051 as frequency

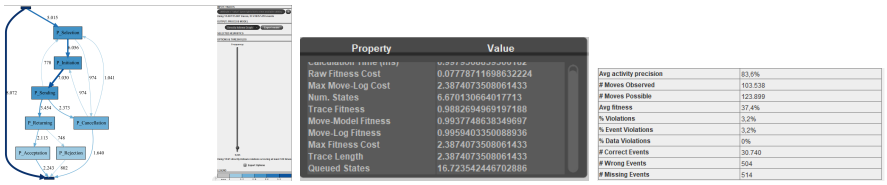
Avg activity precision	80%
# Moves Observed	72.228
# Moves Possible	90.330
Avg fitness	30,3%
% Violations	31%
% Event Violations	31%
% Data Violations	0%
# Correct Events	22.119

(d) Precision for 0.1 as frequency

Figure 6: Conformance and precision checking

Having a look at the different conformance and precision outcomes 6, I decided, that the 0.1 frequency model is not good enough, but wanted to check, if there is a model better or in simplicity or in performance for the 0.051 model. The models best for simplicity fitting had 0.08 frequency or 0.025.
After checking all results in 7 I had to choose. This was a hard decision, but I chose simplicity over the precision and picked the model with 0.025. The fitness is above 90% and precision is also okay. Lower frequency threshold just makes the model to complicated.

(a) 0.08 frequency



(b) 0.025 frequency

Figure 7: Directly followeg graphs, conformance and precision

1.3 Combinded Model

For combinded Models I first filtered the data to have a dataset with all proposal data combined with the application data. This data I filtered with Heuristic filter (all configurations to 100% and just deciding what the endstate is) with the outcomes/endstates "APP_rejected" or "APP_cancelled" or "APP_approved". I saved them under the names "Filtered P App with approv", "Filtered P App with canc" and "Filtered P App with rej".

1.3.1 Endstate APP_Approved

Like for approved and proposal I first checked different frequency filter setting to have a first idea, which models fullfill simplicity. Then for every chosen frequency in begin I checked conformance and precision.

	Frequency			
	0	0.1	0.2	0.3
Simplicity	-	-	+	++
Fitness	99.84	99.47	93.93	93.93
Precision	93.3	94.5	91.7	94.8

Figure 8: Results for approved as endstate

Based on the results, 8, I chose the model with 0.3 filtering. This one has a high simplicity, but still has surprisingly good results.

1.3.2 Endstate APP_Cancelled

	Frequency		
	0	0.049	0.1
Simplicity	-	+	++
Fitness	99.96	99.39	96.84
Precision	92.5	91.8	94.6

Figure 9: Results for cancelled as endstate

Like for APP_Approved I checked different configurations and based on the results, 9, I decided to pick 0.1 filtered frequency model. The fitness and precision is still higher than 90%, but it is also the most simpel model.

1.3.3 Endstate APP_Rejected

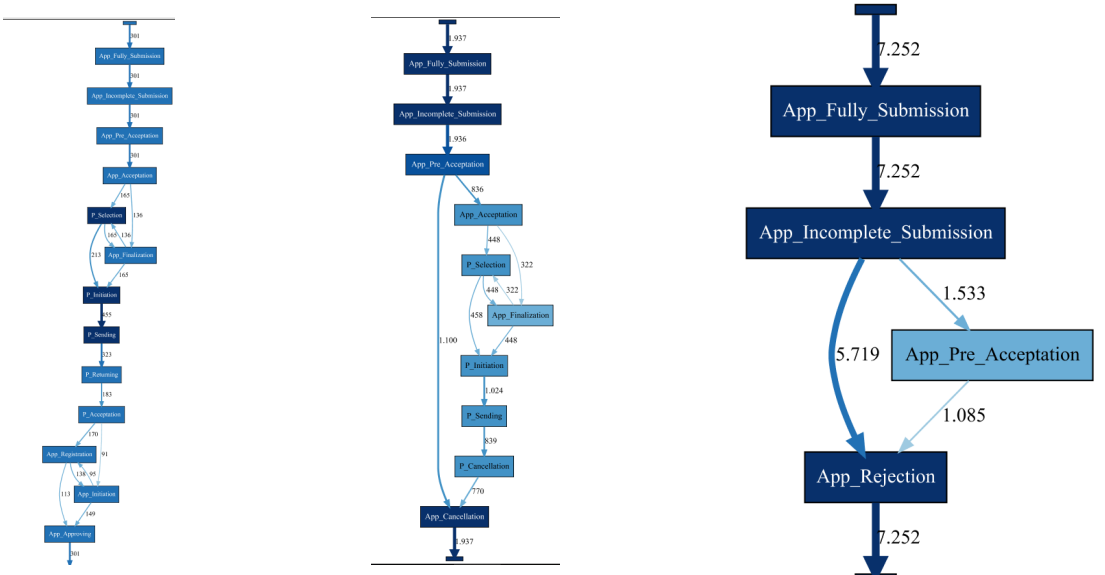
The last analysis is of the models ending in APP_Rejected. In the first step I checked different frequency filters and decided based on simplicity and traceability I had a closer look at 0, 0.025 and 0.1.

	Frequency		
	0	0.025	0.1
Simplicity	-	+	+++
Fitness	1.00	99.60	96.91
Precision	99.3	98.7	100

Figure 10: Results for cancelled as endstate

Based on 10 I chose 0.1 filtered frequency model as the best. It is really easy to follow and has a good fitness.

1.3.4 All 3 Models



(a) Endstate APP_Approved (b) Endstate APP_Cancelled (c) Endstate APP_Rejected

Figure 11: Models for the different endstates

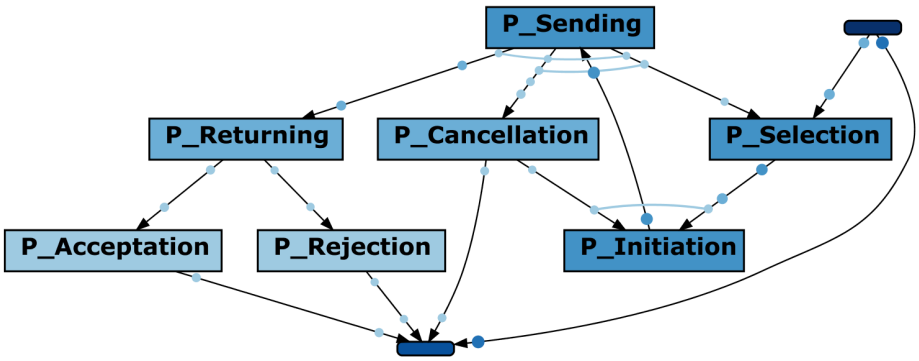
1.4 C-net of the proposal process

Based on the results of my analysis of the model before I chose the same frequency filtering for the C-net of the proposal process.
In 12 I show two C-nets of the proposal process for comparsion. 12a is the one I picked and I will discuss it later in detail. In 12b the original C-net of the process is to see and obviously it is much more complicated and not so intuitive than the filtered one.

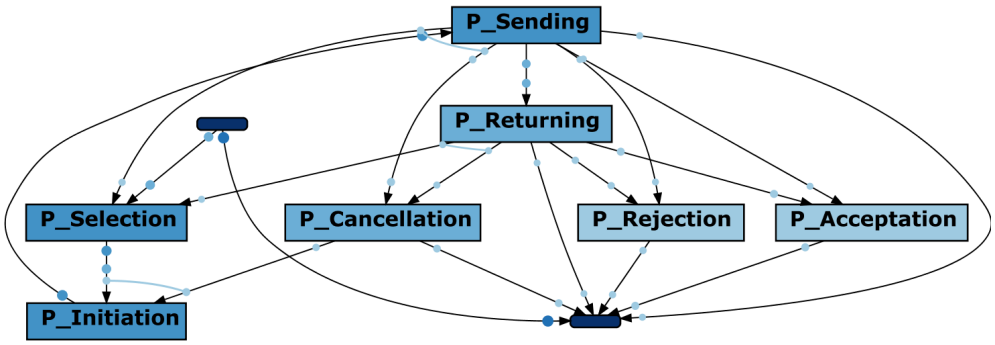
1.4.1 Analysis of the C-net

For simplicity reasons I will not write "P_" as prefix of every activity. The first thing I did is having a look at the maximal number of bindings. This is for sending with 6 possible bindings. The input is always from initiation, but there are 4 different outputs possible. Having a look at all possible traces and the corresponding procedures. Just counting the possible traces shows you, that there are infinite many possible traces, because of a loop between sending and initiation.

- Done



(a) C-net with 0.025 filtered frequency



(b) C-net without filtered frequency

Figure 12: C-Nets of the proposal process

- Selection →Initiation →Sending →
  - Returning →
    - Acceptation →Done
    - Rejection →Done
  - (Cancellation →Selection) or (Selection →Cancellation) →Initiation →Sending ...
  - Cancellation →Done

1.5 Own Petri net of the proposal process

1.6 Analysis of the performance of Application and work process

Questions

Conclusion