Analyzing Application Process for a Personal Loan or Overdraft of Dutch Financial Institute with Process Mining Techniques

Chang Jae Kang, Chul Kyu Shin, Eun Sang Lee, Ju Hui Kim, Min Ah An

School of Business Administration, Myongji University,
50-3, Namgajwa-dong, Seodaemoon-gu, Seoul 120-778, South Korea
{servor.official, chulgs15, les870408, rure1114, minah0910h}@gmail.com

Abstract. Process mining is suitable for the question which the managers of firms or institute are interested in. We tried to analyze the real-life log with various tools and techniques.

Keywords: process mining, BPIC, spaghetti process, ProM, Disco .

1 Introduction

Process mining is a useful technique for analyzing log data recording process execution. However, although we, typically, use a process mining technique, it is not trivial to analyze real processes because they are very complex. Analyzing the complex target process of BPI Challenge 2012 gives us pain and a good opportunity to testify the greatness of process mining simultaneously.

In this report, we tried to ferret out the facts of the process by taking the focus on what the data owner is interested in. We believe that estimators for the total cycle time and our answers to the following questions, are all valuable information: which resources generate the highest activation rate of applications, how does the process model look like, which decisions have great influence on the process flow and where are they.

We used various tools and techniques for the completeness of analysis. Details of tools and techniques are presented in this chapter. Incomplete cases were removed for the perfect analysis.

1.1 Used Tools

There are a lot of tools and techniques to help perform process mining. *DISCO* is one of the most useful process mining tools. The improved fuzzy algorithm adopted by *DISCO*, suits our purpose because it generates appropriate models for very complex process called *spaghetti process*. Most of our analysis has been performed by *DISCO*.

To perform social network analysis, we mainly used *ProM*(version 6.1). *ProM* helped us identify key persons of the target process.

Lastly, we used DBMS(Database Management System) for handling additional information related to the amount registered by customers. The log contains too many events. Thus, instead of using *MYSQL,* we decided to use *Oracle 11g*. Fortunately, our university has academic license of it. We also used scripts written in *Perl* to put the log data in database and to transform filtered data to XES format.

1.2 Removing Incomplete Cases

An *incomplete case* means unexpected case appearing because of extracting data from particular period of time. Since information system records events continuously, the log contains some cases which haven't finished yet or even haven't started. The scrutiny of the log data revealed that complete cases end with *A_ACTIVATED*, *A_CANCELLED*, or *A_DECLINED* activities. Thus the cases not including these final state activities were regarded as incomplete ones. Hence, we got rid of incomplete cases with *DISCO* or SQL(Structured Query Language) of *Oracle 11g*. This elimination resulted in 12688 complete cases.

2 Discovering Descriptive Process Models

There is a need to go into detail about process models before we talk about other topics. Here, *process model* means a descriptive one explaining real-life process. We want to lay the foundation stone of further analysis by having a general idea via descriptive process models. They can also be an answer to the question, '*how does the model look like?*', the data owner is interested in. *DISCO* helps us to figure out what we want with a variety of filtered log data.

2.1 Approach

To discover models, we used *DISCO* which draws process model with improved fuzzy algorithm. It also shows meaningful information such as frequency, duration, etc and provides powerful filtering features.

First of all, we scanned through the process derived from the whole data without including incomplete cases. It shows us the helicopter view of real-life process (See Figure 2-1). But since the log contains lots of cases and events, this helicopter view can drive us to make wrong results. Thus, we split the whole cases into 3 groups by final state activities(i.e. *A_ACTIVATED*, *A_CANCELLED*, *A_DECLINED*). As we discussed in the introduction, that is all possible scenarios which are able to happen in complete case. Also, we extracted process model based on the cases including a particular activity for understanding details of how work gets done. Finally, using the *AMOUNT_REQ* attribute, we investigated the differences of models between what the value is high and what is not.

2.2 Analysis

Incomplete cases were ruled out in order to discover appropriate and simple process model because they generates extra paths. The paths are far from true. Figure 2.1 shows a big picture that is based on facts. The numbers attached to arcs and activities refer to absolute frequency. According to the model, the marked activities have been executed frequently and repeatedly(*W_Completeren aanvraag*, *W_Nabellen offertes* and *W_Nabellen incomplete dossiers*). It can be a prime suspect of poor performance.

Though this helicopter view gives us useful information, some questions came to mind. First, *W_Valideren aanvraag* is important and complicated work in this process, but it seems insignificant to us. Second, the process ends with *A_DECLINED*. Third, there is no activity after *W_Beoordelen fraude* and *W_Nabellen incomplete dossiers* were performed. It seems that there are some cases that end aberrantly in spite of the fact that the log doesn't contain incomplete cases. We are trying to uncover the truth.

In order to confirm facts deeply, we split the whole traces into 3 groups according to final state activities (i.e. *A_ACTIVIATED*, *A_CANCELLED*, *A_DECLINED*). The group that ends with *A_ACTIVATED* has 2246 cases. The *A_CANCELLED* group has 2807 cases. The final one has 7635 cases. There is no case which has two or more final state activities.

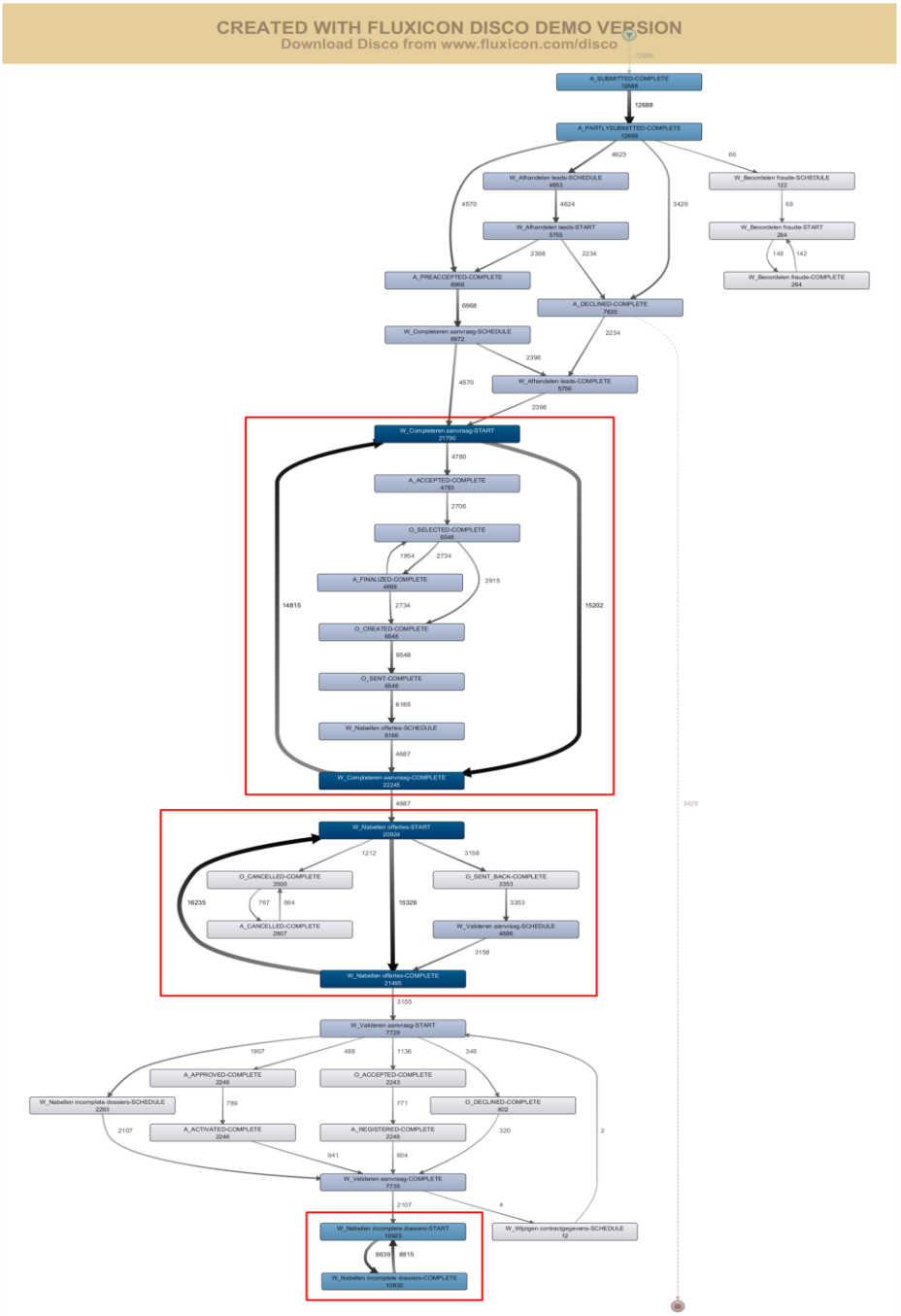


Fig. 2.1. The helicopter view of the process.

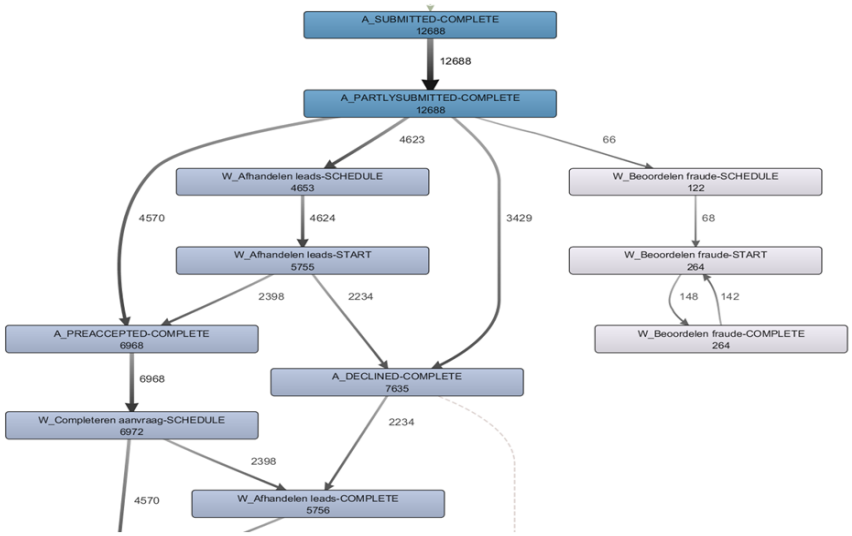


Fig. 2.2. The top part of the helicopter view process model.

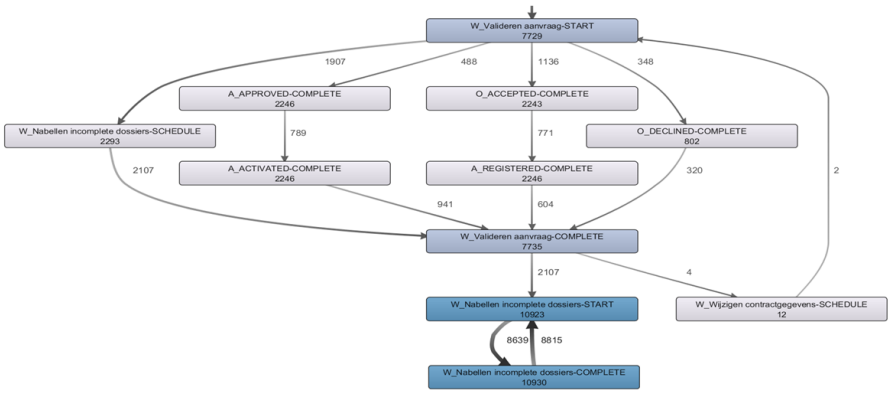


Fig. 2.3. The bottom part of the helicopter view process model.

Figure 2.4 shows the part of the model containing activity *A_ACTIVATED*. There are two remarkable differences as compared with Figure 2.1. First, we found a new loop of *W_Valideren aanvraag*. The other one is about final activity of traces. The process ends with *W_Valideren aanvraag+COMPLETE* in case of having final state of *A_ACTIVATED*. Figure 2.5 shows the part of what contains *A_CANCELLED*. As you can see, *W_Valideren aanvraag* flows in one direction. The model containing *A_DECLINED* is pretty similar to helicopter view model because of size effect.

There are too rare cases containing activity *W_Beoordelen fraude* or *W_Nabellen incomplete dossiers*. For that reason, it was hard to find out the flows when these activities have been occurred. We checked on the models after have extracted cases containing each activity. Figure 2.6 and Figure 2.7 show the part of them. We made sure of what happens when each activity has completed.

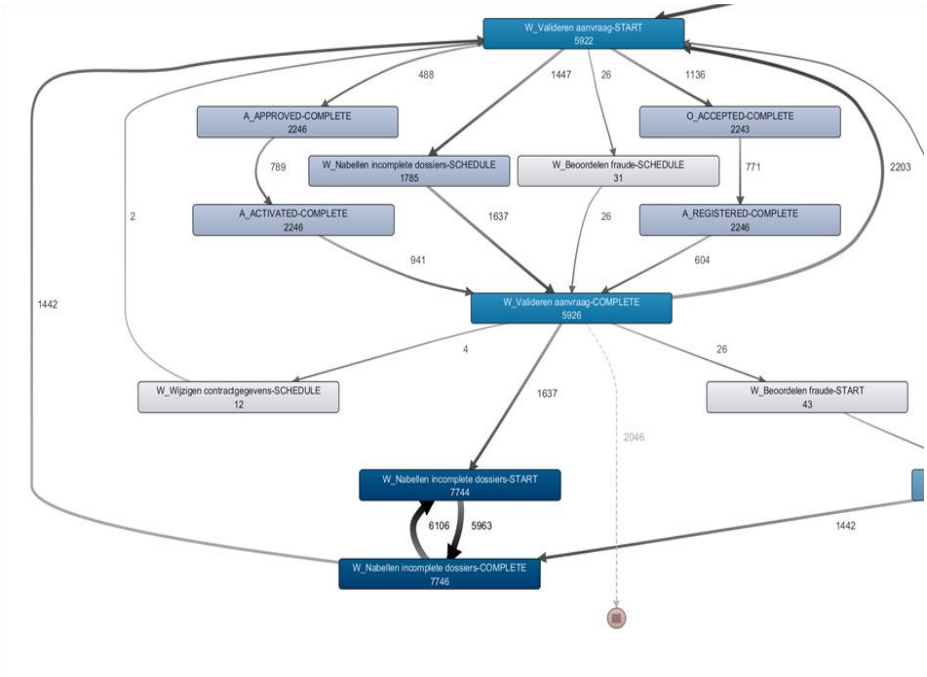


Fig. 2.4. The bottom part of the process model including *A_ACTIVATED*.

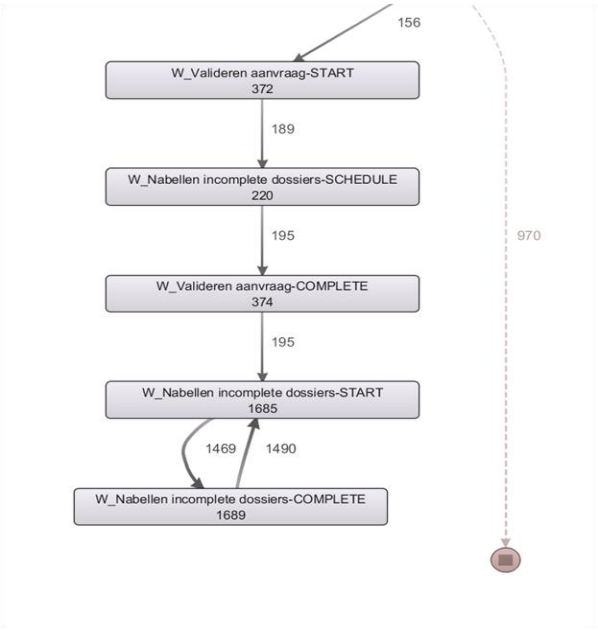


Fig. 2.5. The bottom part of the process model including *A_CANCELLED*.

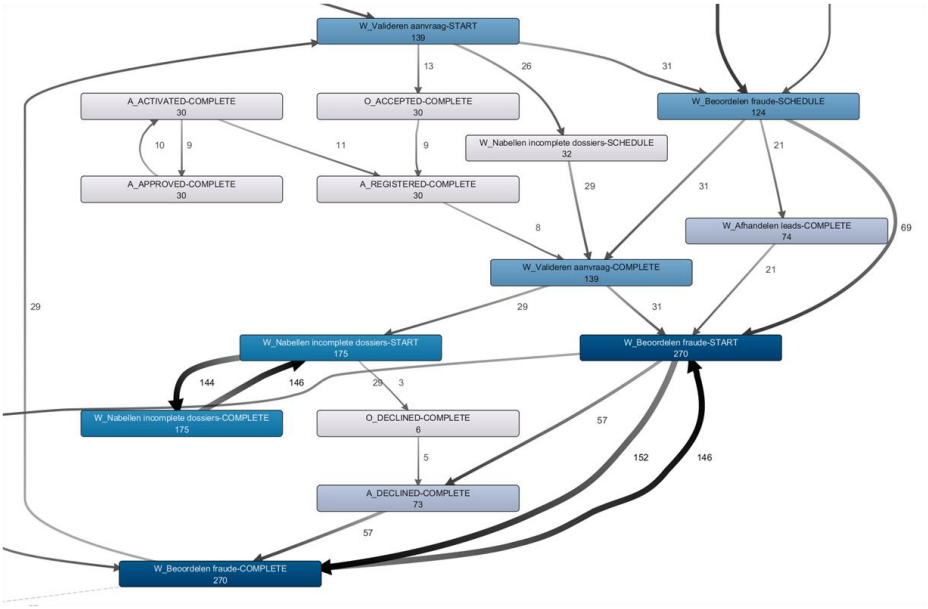


Fig. 2.6. The bottom part of the process model including *W_Beoordelen fraude*.

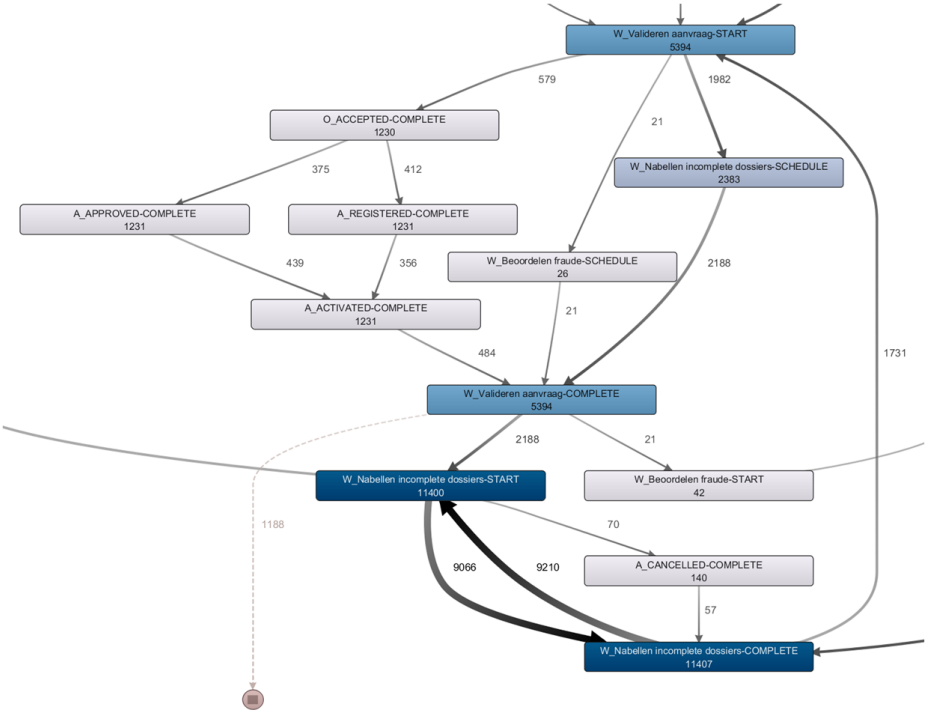


Fig. 2.7. The bottom part of the model including *W_Nabellen incomplete dossiers*.

AMOUNT_REQ attribute indicates the amount requested by customer. Naturally, it is predicted to affect the process. We extracted the cases that the value is over 30000 from database with own technique and the cases what is under 3000 were also extracted. Then, we were compared with each other. The model discovered from what is under 3000, tends that the activities located in front of process are executed frequently. There were more differences which need to deal with, but they have to be discussed with other important information, e.g. domain knowledge. All models that display whole picture are presented in appendix.

2.3 Conclusion

Discovering process model is basic technique of process mining. Process model can be used in further analysis so discovering reliable model is very important. That's the why the discovered model has to deal fully with other information, especially domain knowledge. We felt the lack of some reference models or other information about target institute in order to find accurate and meaningful results. Nevertheless, we think our investigation creates quite good models and information.

3 Description for the total cycle time

In this chapter, we'll deal with valuable information about the cycle time. Before generating the cycle time of the appropriate process flow, analyzing the appropriate process flow and defining the meaning of the cycle time will be preceded. And we'll generate the mean cycle time of the appropriate process flow with excluding incomplete cases. Then we'll analyze the main cause of increasing total cycle time based on appropriate process flow. Finally we'll specify the main cause of increasing total cycle time based on the originator and *Amount_REQ* attribute.

3.1 Approach

To generating the total cycle time, at first we will look into the appropriate process flow with excluding incomplete case. Incomplete case may include all of the activities. Therefore appropriate process flow with excluding incomplete case may include *A_Activated+COMPLETE, A_Declined+COMPLETE and A_Cancelled+COMPLETE*. Again, it is appropriate cases (not incomplete cases) that *A_Activated+COMPLETE, A_Declined+COMPLETE and A_Cancelled+COMPLETE* included. That means these three activities not included is excluded. Then we are able to find out an appropriate process flow.

3.2 Analysis

In order to look into an appropriate process flow, first we use a *DISCO* tool and select the attribute filter. Then, check the filter by 'Activity' and choose the filtering mode 'Mandatory'. 'Mandatory' means that this filter removes all cases that do not have at least one event with one of the selected values. And check three event values (*A_Activated+COMPLETE, A_Declined+COMPLETE and A_Cancelled+COMPLETE*). Next, apply these filter and we are able to look into an appropriate process flow. When we look into the process flow, we are able to select 'Frequency' and 'Performance' option. Among these options, we will select the 'Performance' option because we need to find out the duration between two activities. Then we select the 'Total duration' option than 'Mean duration' option because 'Total duration' option lets us see high impact areas for delays in the process flow by showing the cumulative times.

In Figure 3.1, we are able to see the appropriate process flow by 'Total duration' option with 5% paths slider position. That process flow applies to filtering option by choosing three activities *(A_Activated+COMPLETE, A_Declined+COMPLETE and A_Cancelled+COMPLETE*).

Then, we are able to generate the mean cycle time of the appropriate process flow. Before generating the total mean cycle time of the appropriate process flow, we need to define the meaning of cycle time. For each case, each case has first and last activity. The gap of time between first and last activity is 'Duration' that term used in the *DISCO*. 'Duration' meaning corresponds to the meaning of cycle time. Therefore, the

meaning of cycle time is applied to each case duration. To generate the total mean cycle time, every case duration value's sum and total case number are used. And every case duration value's sum divided by total case number equals to approximately 8.3 days. In other words, the total mean cycle time is about 8.3 days.

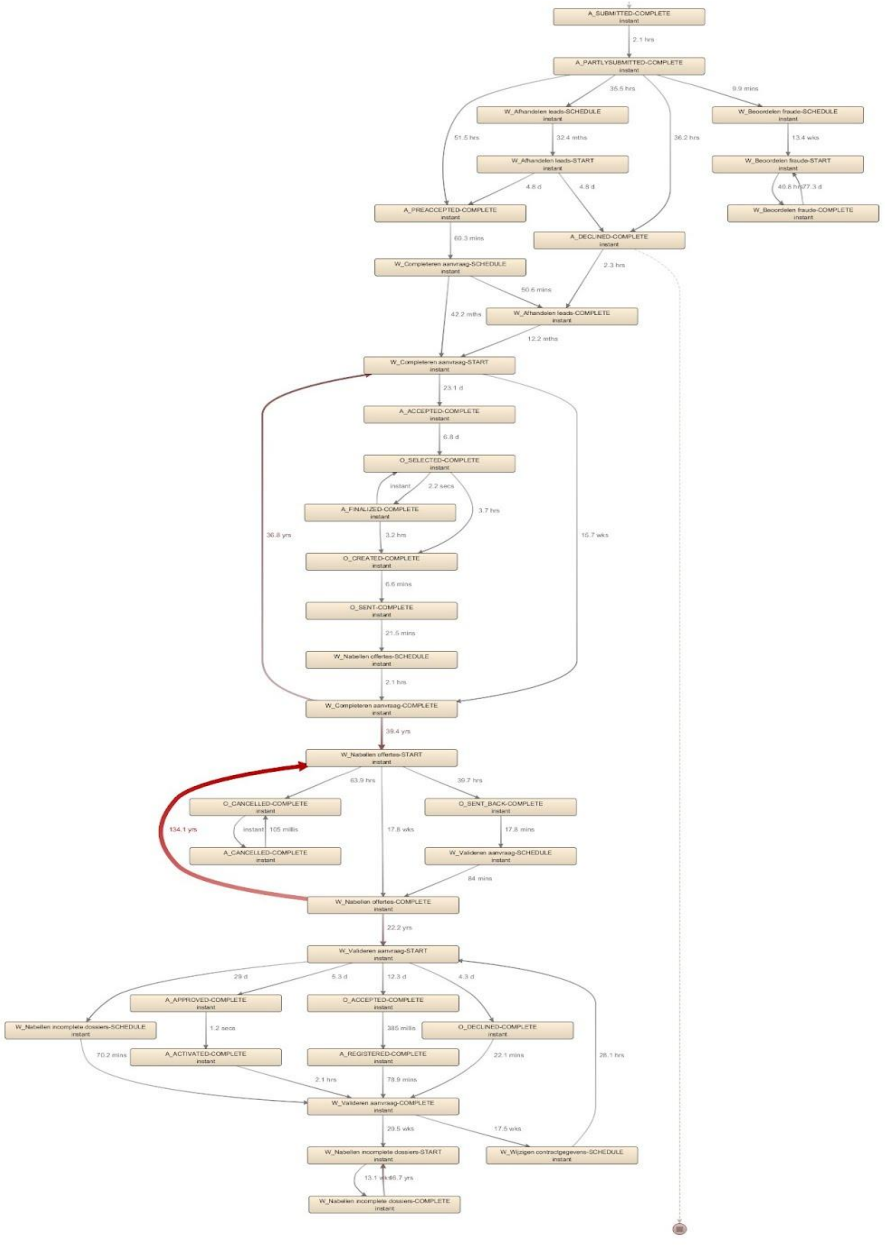


Fig. 3.1. Appropriate process flow by 'Total duration' option. (5% paths slider position)

After we find out the appropriate process flow with excluding incomplete case and the total mean cycle time of the appropriate process flow, we are able to analyze the main cause of increasing total cycle time. To analyze the main cause of increasing total cycle time, we need to examine Figure 3.1 again. When we look into the Figure 3.1, there are especially three points that delay the whole process flow. Firstly, the total duration from *W_Completeren aanvraag+COMPLETE* to *W_Completeren aanvr-aag+START* is 36.8 years. It is very high value. Because the frequencies of the activities and paths are included in cumulative view, too many executions of two activities may be done. We'll indicate this total duration as *'Point A'*. Secondly, the total duration from *W_Completeren aanvraag+COMPLETE* to *W_Nabellen offer-tes+START* is 39.4 years. We'll indicate this total duration as *'Point B'*. Lastly, the total duration from *W_Nabellen offertes+COMPLETE* to *W_Nabellen offertes-+START* is 134.1 years. It is the highest value out of the Figure 3.1. We'll indicate this total duration as *'Point C*'. Figure 3.2 shows the appropriate process flow that highlighted by *'Point A'*, *'Pont B'* and *'Point C'*.

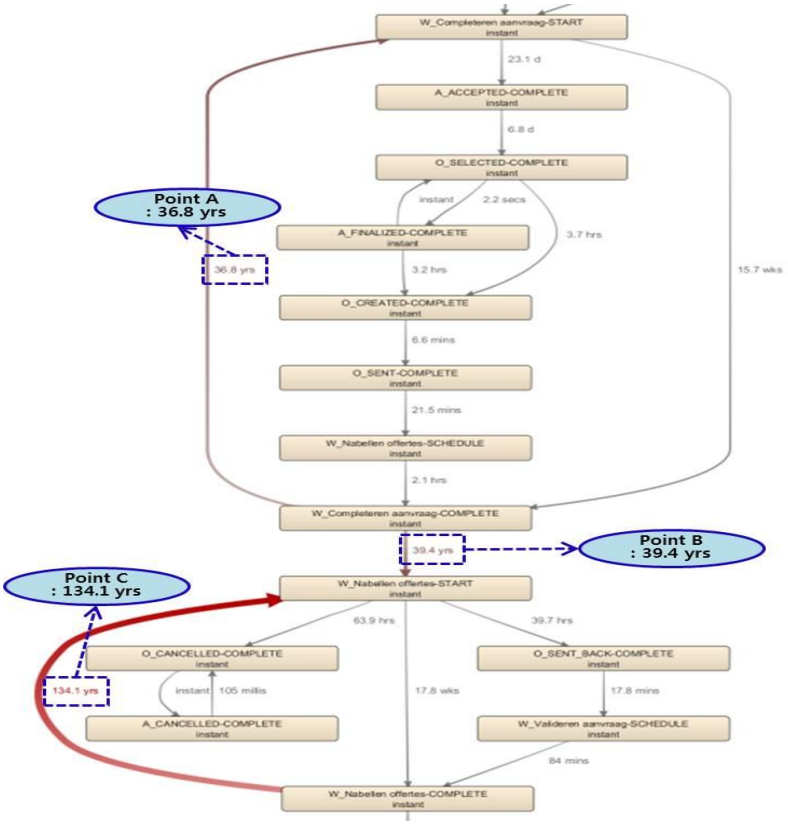


Fig. 3.2. Appropriate process flow that highlighted by *'Point A'*, *'Point B'* and *'Point C'*. (5% paths slider position)

In the prior section we discovered that causes of increasing cycle time are '*Point A'*, '*Point B', and 'Point C'*. So in this section we discovered what the cause of '*Point A', 'Point B'* and '*Point C'* is. To discover it, we estimated that causes are originator and *Amount_REQ*. At First we analyzed the originator. After grasp the originator who related with '*Point* A', '*Point* B' and '*Point* C' then we considered about relationship among originators.

Cause analysis about originator is as the following. First we divide each *Point* with start activity and end activity. Second we extract the two group of top 10 originator ranking by frequency that performed start activity or end activity. But we extract only 9 originators because top originator was always Null at every Point. Third we make up three table filled by handover frequency, total duration and mean duration.

The next figure shows how the table is made. Arrows in the first figure shows all of the handover that generated by originator 'A' and the value of total duration.

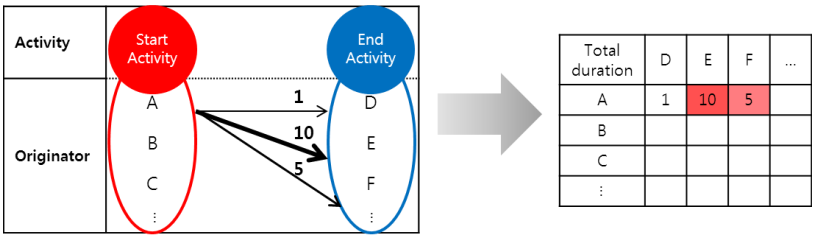


Fig. 3.3 Method of extract three tables at each *Point.*

The total duration above the arrows fills the table 'Total duration'. The first cell shows what this table shows, the first column shows the start originator in ranking order, and the first line shows the end originator in ranking order. And then we record the value that each handover has. For example so we can discover that the total duration during the handover to D from A is 1. If a certain value is close to max value of the table, its cell be colored red. If the certain value close to max value, its cell will be the red.

After extract the three tables for each *Point* by above method, we extracts problematic handovers based on the each value. Cause the criteria of the cycle time was the 'Total duration' for discovering the 'Point A', 'Point B' and 'Point C', so we analyze the handover based on same criteria. And we also analyze the handover which has abnormality value on the mean duration or frequency.

The next table 3.1. shows the handover total duration among the originator related with *'Point A'*. Start activity of *'Point A'* is *W_Nabellen offertes+COMPLETE* and end activity is *W_Nabellen offertes+START*. *'Point A'* is considered to loop because *' Point A'* consists of two activities that have different event type and same activity name.

The first problem of 'Point A' is the self-loop which was made by same originator. It is easy to find that the highest value of total duration is related with the self-loop handover. All of these handover has frequency over 150, so it means self-loop occurs very frequent. But the self-loop handover which is made by same originator means that originator doesn't work efficiently. So that originator is considered to need work improvement.

Table 3.1. The total duration table of '*Point A*'.

Total duration		*W_Nabellen offertes+START*								
		11180	11181	11119	10909	11203	10861	10913	11201	11189
W_ Nabellen offertes + START	11180	1515	380.21	416.7	431.9	730	353.5	422.8	1180	337.4
	11181	380.2	1161.9	221.9	332.5	447.13	657	232.4	245.7	151.9
	11119	620.5	227.5	946	211.4	410.6	471.5	144.20	205.1	140.7
	10909	453.2	489.7	165.2	891.21	480.6	260.4	298.90	265.3	291.2
	11203	276.5	228.2	312.2	471.5	1314.0	347.2	428.88	444.1	50.9
	10861	456.3	325.5	232.4	269.5	219.1	1049.4	232.40	221.9	477.5
	10913	544.5	229.6	99.4	293.3	277.2	158.2	702.63	182.7	109.9
	11201	166.6	492.8	310.1	115.5	489.7	130.9	198.10	742.2	119
	11189	200.2	392.4	219.8	235.9	219.8	249.2	148.40	191.8	906.4

The second problem of *'Point A'* is the handover from originator 11180 to originator 11201. Although this handover occurred only 59 times, its total duration was 1180 days because mean duration was 20 days. So if we could decrease its mean duration to similar with mean duration of *'Point A'*, we can save almost 997 days.

The final problem of *'Point A'* is that the average of top 9 originators' mean duration was 4.44 days. It is longer than 1 day when we compare average duration of the all originator who is related with *' Point A'* and 4.44 days. So we should decrease the frequency of *'Point A'* and mean time for decreasing the cycle time.

The next table 3.2. shows the handover total duration among the originator related with *'Point B'*. Start activity of *'Point B'* is *W_Completeren aanvraag+COMPLETE* and end activity is *W_Nabellen offertes+START*.

Table 3.2. The total duration table of *'Point B*'.

Total duration		*W_Nabellen offertes+START*								
		11180	11181	11119	10909	11203	11259	10861	11049	11201
W_ completeren aanvraag + COMPLETE	11181	71.9	274.4	164.5	133.7	112.7	22.3	301	31	90.4
	11189	80.1	336.7	112.7	73.3	78.4	48.1	68.1	53.7	100.8
	11169	76.3	160.3	96.6	85.8	85.1	98	85.7	76.9	58.8
	11201	92.4	312.2	128.8	68.6	158.2	41.5	88.7	91.7	319.9
	10861	116.2	158.9	172.2	66.7	58.1	49.8	141.4	32.9	211.4
	11203	144.2	122.5	137.2	157.5	324.8	70.1	100.1	61.7	238
	11119	169.4	64.7	177.8	100.1	154.7	49.6	205.1	53.7	33.6
	11180	192.5	162.4	145.6	105.7	146.3	41.1	70.4	17.6	172.9
	11179	103.6	41.4	144.9	33.7	93.8	6.8	41.5	79.6	62.2

Same with the prior *'Point A'*, there are some handover which has same originator. But it is hard to say there are problem at originator because *'Point B'* is not the loop and that handover mean duration isn't higher than other handover mean duration. But originator 11203's mean duration is high and frequency is low. So we can say that there are the problems with work efficiency at originator 11203.

Alike originator 11203, there are four handovers which have high mean duration. Originator 11201 spent 312.2 days to handover to 11181, originator 10861 spent 211.4 days to handover to 11201, originator 11203 spent 238 days to handover to 11201 and originator 11181 spent 301 days to handover to 10861. All of those four handovers mean duration is longer than 8.5 days. Although the handover total duration from originator 11189 to 11181 is 336.7 days, it doesn't be matter because its mean duration isn't high than others.

We could find a feature about mean duration. It is that if handover mean duration from A to B has high value; mean duration from B to A has high value too. The next table shows this phenomenon.

Table 3.3. The gap of mean duration between two resources.

Originator		Handover		Gap
		→	←	
11181	11201	8.2	9.5	1.3
11181	10861	9.7	7.6	2.1
11181	11203	8.1	6.8	1.3
11181	11119	0.031	3.4	3.369
11181	11180	4	6.8	2.8
11201	10861	11.1	10.1	1
11201	11203	6.9	8.5	1.6
11201	11180	7.1	7.5	0.4
10861	11203	2.771	6.3	3.529
10861	11119	6.2	7.3	1.1
10861	11180	5.8	5.4	0.4
11203	11119	9.8	8.6	1.2
11203	11180	9.6	8.1	1.5
11119	11180	7.4	5.2	2.2

There are two originators who did handover to each other in one line. The next column "Handover" shows the direction of handover and under that column shows the mean duration for each handover. And the next column shows gap between two mean durations. We can find the line which contains both two mean durations is high. So we can say the two originators that are contained such line are far from each other or they has problem for handover. In this *'Point B'* we found problematic handover, for example, handover from originator 11201 to 10861.

The next table 3.4. shows the handover total duration among the originators related with *'Point C'*. Start activity of *'Point C'* is *W_Completeren aanvraag+COMPLETE* and end activity is *W_Completeren aanvraag+START*. *'Point C'* is considered to loop because *'Point C'* consists of two activities that have different event type but same activity name.

Table 3.4. The total duration of *'Point C'*.

Total duration		W_Completeren aanvraag+START								
		11181	11201	10861	11203	11180	11169	11179	11119	11189
W_ completeren aanvraag + COMPLETE	11181	416.71	54	165.90	159.60	56.20	85.50	72.10	125.30	49.90
	11201	96.6	428.88	173.6	224	67.7	66.9	76.7	58.1	5.8
	11203	95.2	144.9	124.6	602.3	179.2	92.4	160.3	95.2	46.5
	10861	156.8	182.7	608.33	726.96	210	189.7	70.6	162.4	27.7
	11180	186.2	301.7	84.9	59.4	422.79	61.2	57.2	93.8	22.6
	11179	105.7	28.5	224	108.5	211.4	26.6	441.04	75.3	19.1
	11169	18.6	79.10	243.6	82.4	59.5	380.2	116.9	42.4	73.6
	11189	31.1	28.30	44.4	59.1	56.3	72.9	66.8	4.7	200.2
	11121	39.2	65.20	16.9	28.5	28	59.6	18.6	5.8	30.4

At *'Point C'* we found the self-loop again which had made by same originator. You can find that easily the most highest value of total duration is the handover which is made by self-loop. All of these handover has frequency over 170 and it is the higher value than others. So it means that this self-loop occurs very frequent. The self-loop which is made by same originator means there are originator who needs work improvement because it means that originator doesn't work efficiently.

Another problematic handover is the handover which spent 301 days from originator 11861 to 10861. This handover mean duration is 7.494 days and frequency is 97. So it is considered problematic handover that should decrease mean duration for decreasing cycle time.

In common problem with '*Point A'*, *'Point B'* and *'Point C'* is self-handover. Because '*Point A'* and '*Point C'* is the loop, the self-loop means that originator doesn't work efficiently. The real problem is that if the loops occur many times in 1 case it generates wasting time too much. And it occurs by many originators, it seems to not only originator's problem but also problem on this process.

Before analyzed the '*Point A'*, *'Point B'* and *'Point C'* we set total duration to criteria and estimated the cause to duration and frequency. At each point both problematic handover which has high frequency or high mean duration exists altogether. The high mean duration means there are some problem with handover and originator. And if mean duration is high, it always generates increasing cycle time. So to decrease the cycle time, first of all decreasing handover mean duration is essential. Secondly there are problematic handover with high frequency. In these case's cause is not the originator's working efficiency but the work distribution or heavy handover to certain originator. So it'll be solved when the originators work distribution is efficiently.

After looking into the appropriate process flow (Figure 3.2), we assume that the cause of high total duration about *'Point A'*, *'Point B'* and *'Point C'* may be associated with *Amount_REQ* attribute. The amount requested by the customer is indicated in the case attribute *Amount_REQ* and every case contains this attribute. Therefore, we analyze *Amount_REQ* attribute value. The whole case number about the appropriate process flow with excluding incomplete case is 12,688 whereas the whole case number of total event logs is 13,087. And we find out that the minimum

value of *Amount_REQ* is 0 and the maximum value of *Amount_REQ* is 99,999. In addition to this fact, we find out that the average value of *Amount_REQ* is about 15,511. Then, we divide the *Amount_REQ* into four parts from 0 to 99,999. we set the four sections on the basis that the whole case number (12,688) is comparatively distributed evenly. Then, we divide the four sections as *'Section 1'*, *'Section 2'*, *'Section 3'* and *'Section 4'*. The *'Section 1*' is under 5,000 *Amount_REQ*. The *'Section 2*' is over 5,001 and under 9,999 *Amount_REQ*. The *'Section 3'* is over 10,000 and under 15,000 *Amount_REQ*. The *'Section 4'* is over 15,001 *Amount_REQ*. In succession, we are able to show the four sections in a process flow with total duration option and 5% paths slider position. They are shown as Figure 3.4, Figure 3.5, Figure 3.6, and Figure 3.7.

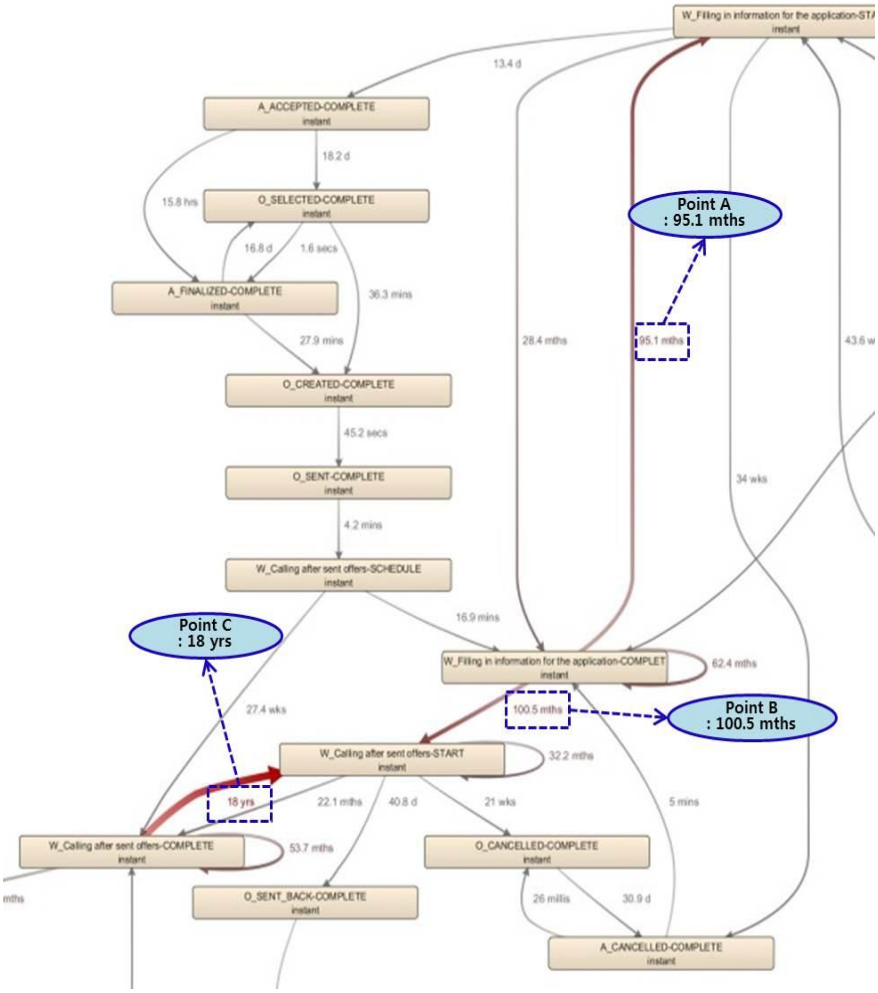


Fig. 3.4. The *'Section 1'* : under 5,000 *Amount_REQ.* (5% paths slider position)

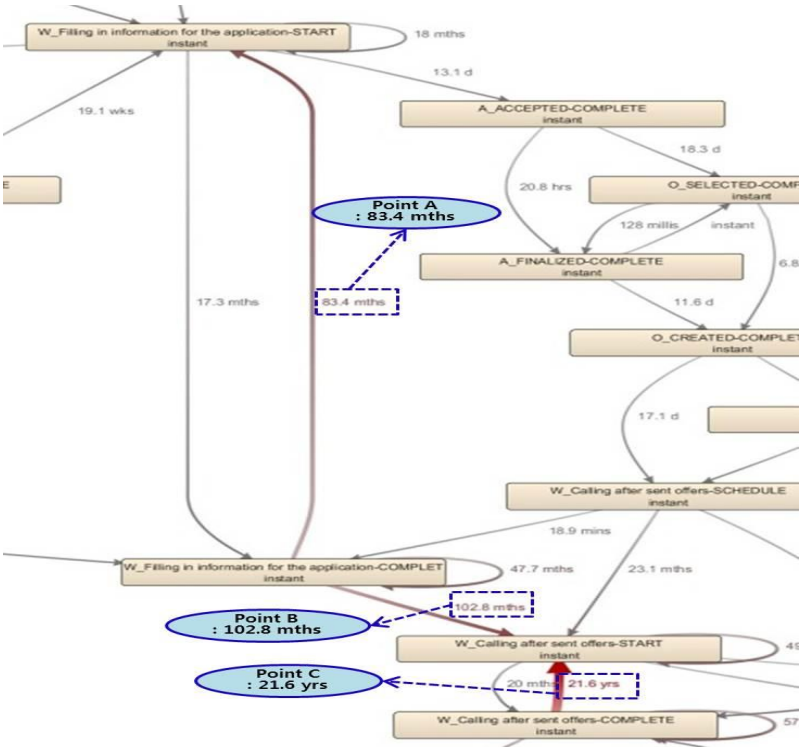


Fig. 3.5. The *'Section 2'* : over 5,001 and under 9,999 *Amount_REQ.* (5% paths slider position)

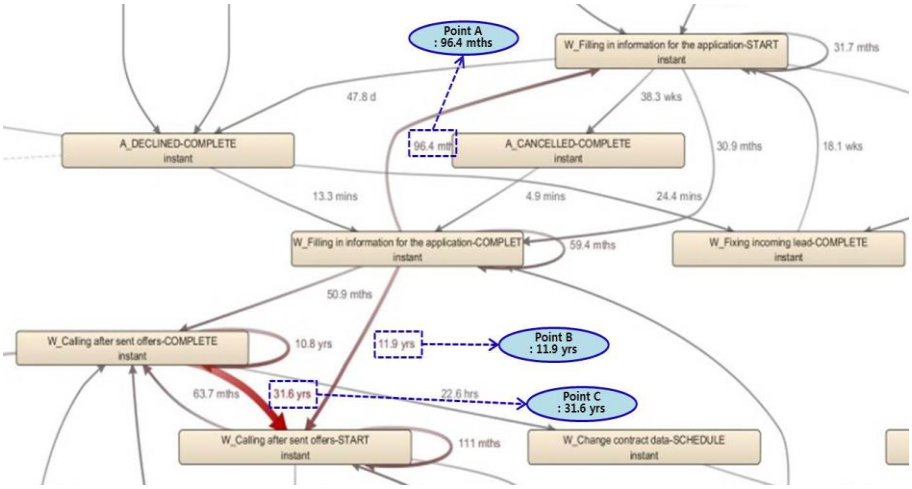


Fig. 3.6. The *'Section 3'* : over 10,000 and under 15,000 *Amount_REQ.*
(5% paths slider position)

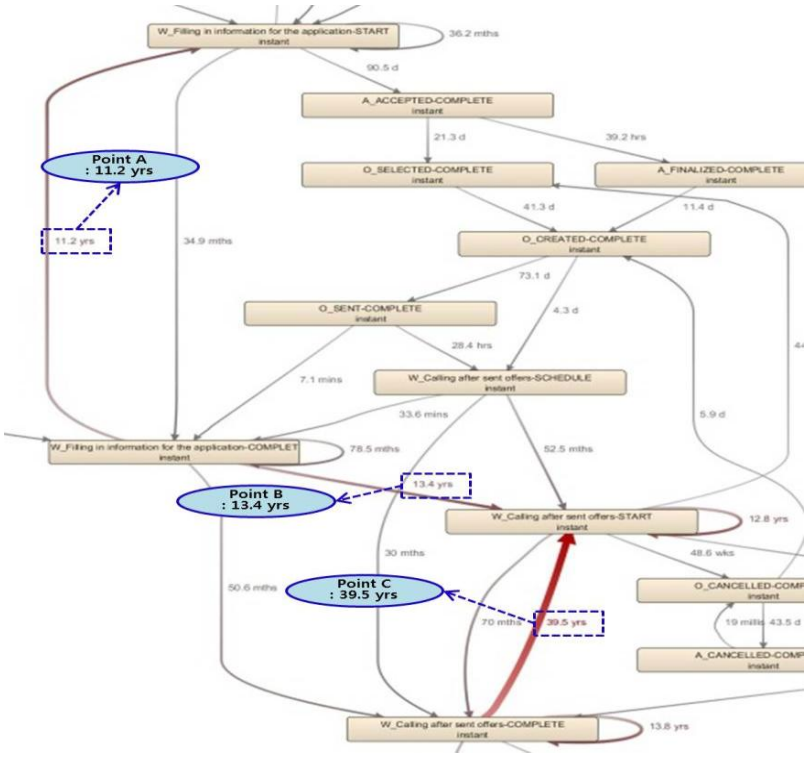


Fig. 3.7. The *'Section 4'* : over 15,001 *Amount_REQ.* (5% paths slider position)

And then, we come up with a table on the basis of Figure 3.4, Figure 3.5, Figure 3.6, and Figure 3.7. Depending on each four sections, *'Point A'*, *'Point B'* and *'Point C'* values of total duration are described below the table.

Table. 3.5., Every point's values of total duration depending on each four sections.

Point / ***Section***	***Point A***	***Point B***	***Point C***
Section 1 (under 5,000)	95.1 mths (7.9 yrs)	100.5 mths (8.4 yrs)	18 yrs
Section 2 (over 5,001 and under 9,999)	83.4 mths (7 yrs)	102.8 mths (8.6 yrs)	21.6 yrs
Section 3 (over 10,000 and under 15,000)	96.4 mths (8 yrs)	11.9 yrs	31.6 yrs
Section 4 (over 15,001)	11.2 yrs	13.4 yrs	39.5 yrs
Total Section (over 0 and under 99,999)	36.8 yrs	39.4 yrs	134.1 yrs

When we look into the table, we find out some facts. As each section moves from *'Section 1'* to *'Section 4'*, each point's total duration value is increased (except for *'Point A'* between *'Section 1'* and *'Section 2*'). Therefore the more *Amount_REQ* value is increased, the more total duration is increased. That is, *Amount_REQ* value influence the total duration. And it is apparent that increasing *Amount_REQ* value causes a bottleneck. Especially, *'Section 4'* is the highest total duration value among the four sections. In other words, *Amount_REQ* value over 15,001 may be the most important factor about delaying for each point.

3.3 Conclusion

In this chapter, we examined valuable information about the cycle time. And we generated the total mean cycle time of the appropriate process flow with excluding incomplete cases. Then, we analyzed the main cause of increasing total cycle time based on appropriate process flow with excluding incomplete cases. Finally, we specified the main cause of increasing total cycle time based on the originator and *Amount_REQ* attribute. It is obvious that *Amount_REQ* value influences the total duration. Unfortunately, there is a certain limit to domain knowledge that we do not exactly know. Besides using only the event logs, therefore, we'd better use the domain knowledge together. If we do so, the better analysis about the cause of delaying total duration is verified.

4. Finding the originators who make the highest activation

4.1 Approach

The question that we deal with in this chapter is '*which resources generate the highest activation rate of application?*'. Before finding an answer, we'd like to make clear definitions of two words, 'resources' and 'activation'. Resources are originators who perform activities and activation is approval for overdraft or personal loan. Therefore, we need to get data that contain activity *A_ACTIVATED* in cases. Based on this discussion, we changed the original question as follows:

"Who is the key originator when applications are activated?"

To answer this question, we used two process wining techniques**.** One is social network analysis, included in *ProM*(version6.1), the other is to analyze the activity, *W_Validerenaanvraag*.

4.2 Analysis

Social network analysis. We tried to find the key originators by using a social network analysis in *ProM*(version6.1). It helps identify how originators are connected based on the transfer work among them. Before starting, the analysis we divided the cases into two types, one is activated cases and the other is declined and cancelled cases. We performed social network analyses separately based on the two types of cases.

We found the model for activated cases(see Figure 4.1). The more one originator makes exchanges with others, the more he/she moves at the center of model. This means that he/she is more related to activation than the others. We identified 15 originators in decreasing order of betweenness(i.e. 10138, 112, 10609, null, 10809, 10972, 11049, 11122, 10629, 11169, 11119, 10899, 10932, 11180, 10861).

Also, we found the model for inactivated cases(see Figure 4.2).We identified 15 originators in decreasing order of betweenness;11121, 10982, 10138, 11203, 10880, 11180, 10912, 11169, NULL, 11003, 10909, 10861, 112, 11181, 10932.

Taking these results into consideration, even though there are important originators in the activated cases, they are not key originators when they are also important originators in the inactivated cases. The rationale for this assertion is that they largely contribute themselves to both activation and inactivation. According to Figure 4.3, it is key originators that do not participate in the inactivation but actively join in the activation; 10609, 10809,11045, 10920, 11259,11000, 10863.

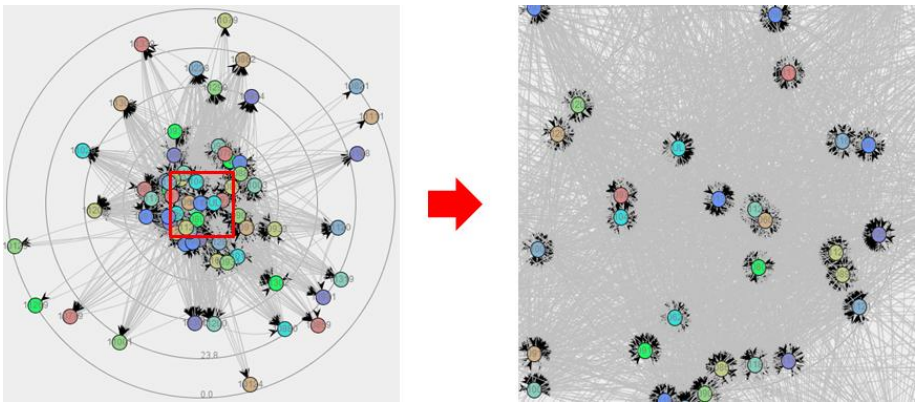


Fig. 4.1. The model of social network analysis for activated cases

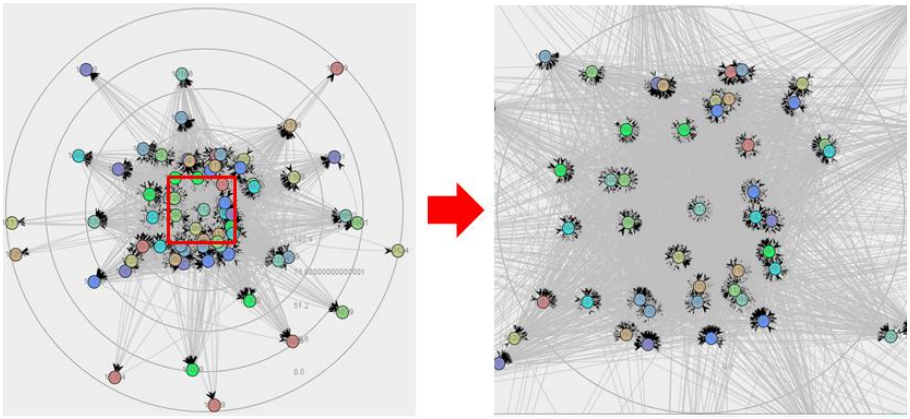


Fig. 4.2. The model of social network analysis for inactivated cases.

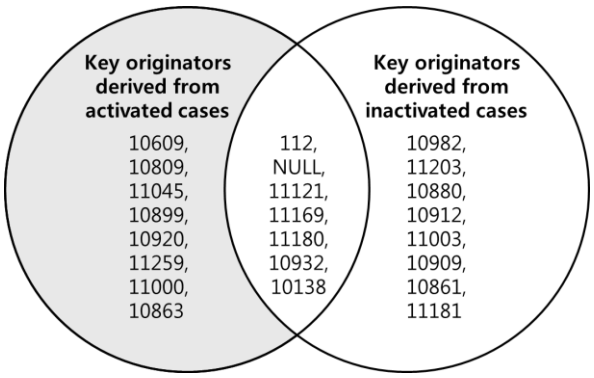


Fig. 4.3. Key originators that only actively join in the activation.

However, the weak point of the above analysis is only to consider betweenness. So, we could not know whether activities related in activation are significant or not. To overcome this weak point, we need to identify originators who performed a key activity for the activated cases.

Identifying originators based on the key activity for the activated cases. As noted above, the shortcoming of a social network analysis is to ignore the importance of activities. First, we discovered a process model for the activated cases to select a key activity. Second, we discovered the originators who perform the key activity Third, we compared them with originators who was discovered in social network analysis before.

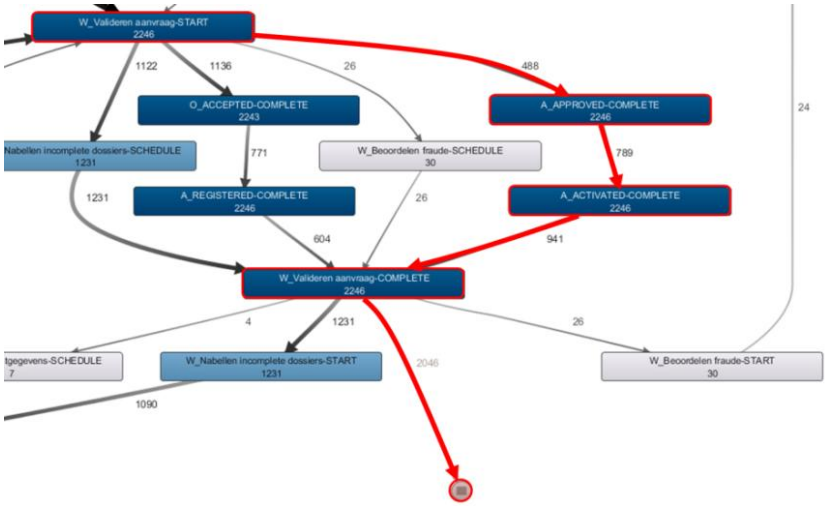


Fig. 4.4. The part of process model for activated cases.

To find a model, we used a *DISCO* to discover a model that must include a *A_ACTIVATED* in the cases. Figure 4.4 is part of a model. In detail, flow of is *W_Validerenaanvraag+START → A_Approved+COMPLETE → A_Activated+ COMPLETE → W_Validerenaanvraag+COMPLETE→ END'*. Since cases should get through *W_Validerenaanvraag* in behalf of activation, we chose it as an important activity. For this time, we do not involve *W_Validerenaanvraag+ SCHEDULE'*. Those originators are not real performers. However, they are automatically registered due to a business rule.

After finding an important one, we tried to figure out how originators implement *W_Validerenaanvraag* in activated cases of complete cases. So, we make a formula.

$$\frac{Eachoriginator\&W\_Validerenaanvraag(STARTorCOMPLETE)\&Activatedcases}{Totalcompletedcases} \quad (E1$$

Thanks to the fomula, we can compare the orginators who carry out activity, *W_Validerenaanvraag* more. For example, the originator, 112, accomplishes

W_Validerenaanvraag for 6174 times in activated cases. The counts are divided by total complete cases, 12688 for comparing each other.

Table 4.1. Theresult on how many times originators do *W_Validerenaanvraag + START&COMPLETE*.

Originator	Counts	Values (Count/ Total completecases)
112	6174	0.487
10138	5957	0.469
NULL	5458	0.430
10972	4984	0.393
10609	4410	0.348
10629	3604	0.284
10809	3182	0.251
11049	3069	0.242
11169	2713	0.214
11259	2363	0.186
11189	2278	0.180
10899	2275	0.179
10913	2270	0.179
11181	2118	0.167
10909	2100	0.166

Taking a close look, as obtaining higher values, originators play an important role in activation because they carry out more *W_Validerenaanvraag* than the others. On the other hand, the lower values mean that originators do not often take part in *W_Validerenaanvraag* in activated cases.

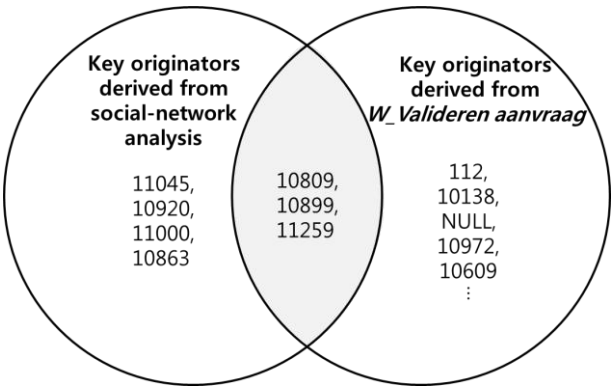


Fig. 4.5. Key originators that only actively join in the activation and carry out *W_Validerenaanvraag*

We could know who acts as important roles in activated cases and which originators largely perform an important activity through equation. Taking account of

these 2 results, we discovered the key originators. According to figure 4.5, the key originators are located in the center of diagrams; 10809, 10899, 11259.

4.3 Conclusion

To fulfill the owner's request, we defined the resources as key originators and then find them. We use social network analysis to find those who have the highest relationship and the formula to find those who participate in do *W_Validerenaan-vraag*. Then, comparing with both results, the key originators are found. However, choosing important activities based on event logs are incomplete because there are a variety of activities in real which have a great influence on process. Therefore, through interviewing with acting partner to gain more information what activities and originators are important, the results make more precise and useful.

5 Conclusion

In this report, we tried to analyze spaghetti process with various tools, techniques. Firstly, the descriptive process models are discovered. The helicopter view model gave us useful information but needs to go detail to investigate the truth. We made good use of *DISCO*. Performance analysis was also performed and we tried to find the causes in the aspect of resources and amounts registered by customer. Lastly, we looked for the resources who generate the highest activation rate of applications. Social network technique is used to answer the question but it has some problem to certain it is appropriate one. So further analysis was performed with own technique. Because there is important information, *AMOUNT_REQ* attribute, we used database to handle additional data.

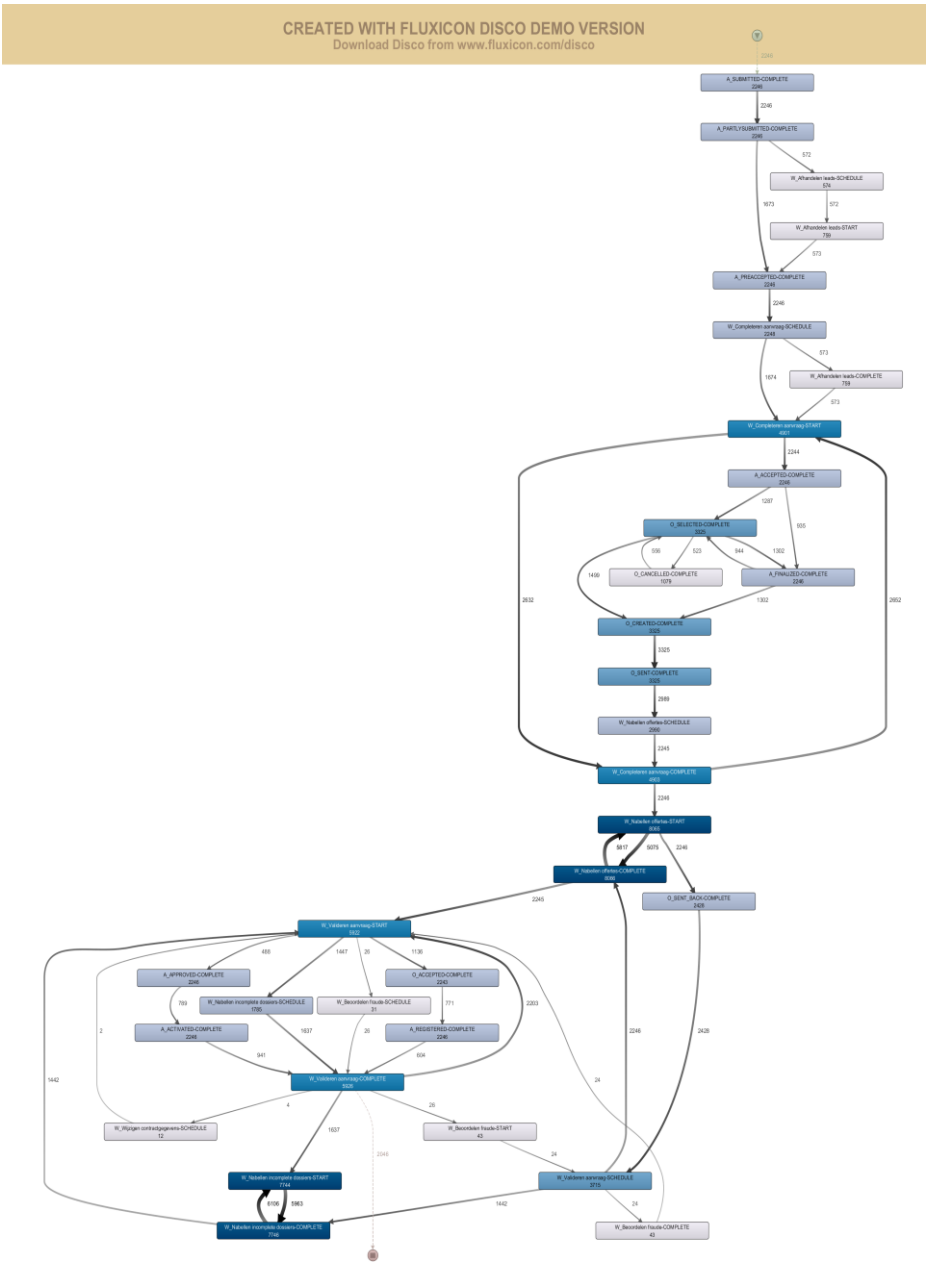
As we analyze, we reached the limit of information about target institute, time and even language. We learned that the domain knowledge or useful information from such interviews are very important for making a perfect result. There was another topic of causal analysis of problematical process flow, but we couldn't find the answer because of time. We are so sorry for that.

Also, We are sad for not using lots of tools and techniques which may appropriate for our analysis, e.g. dotted chart, sequence and pattern analysis or etc. It can probably be a more perfect report if we used all the possible ones.

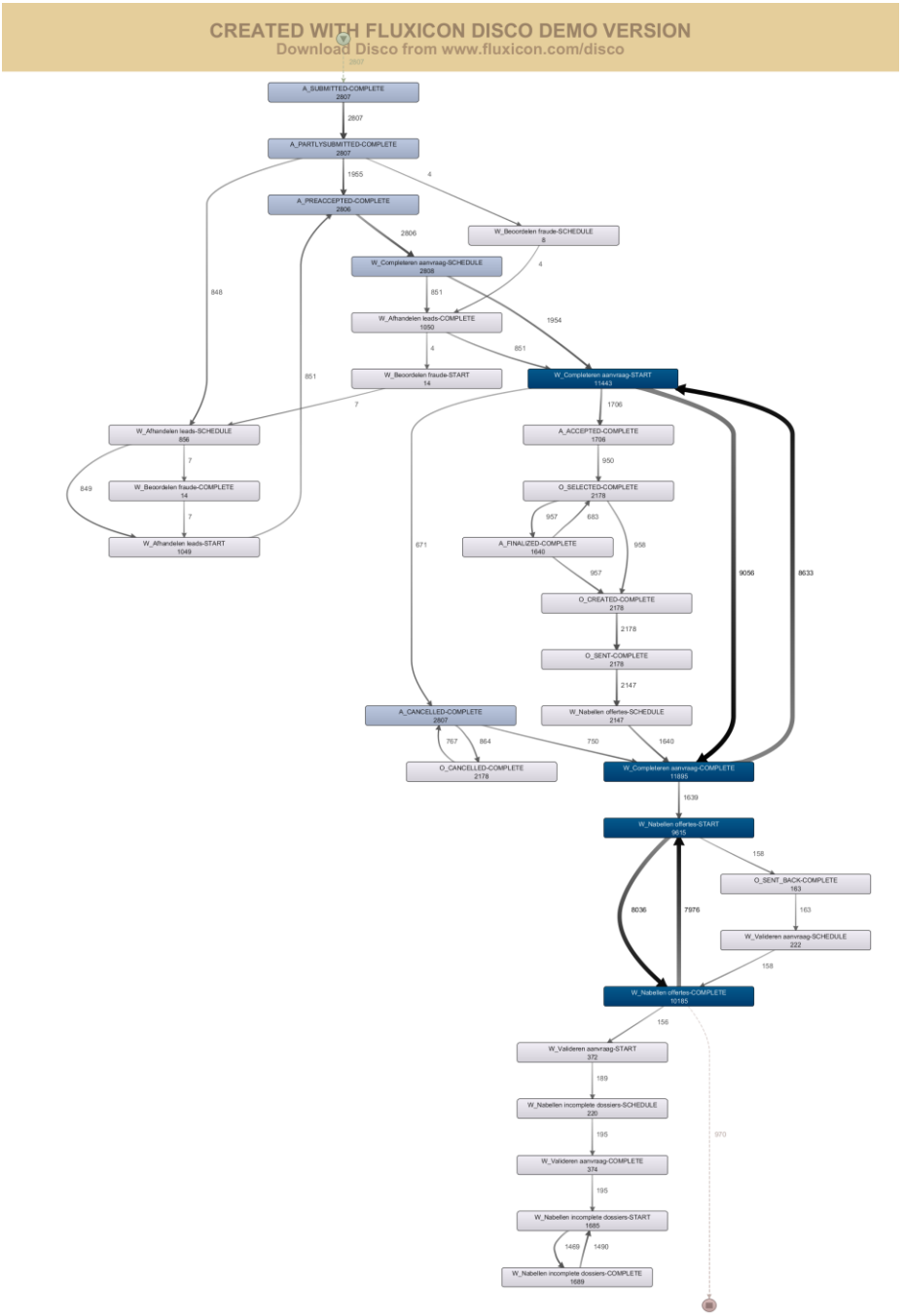
Finally, thank you for providing real-life log. It is one of good opportunity for testing our skills and knowledge. Officially, it is so hard to meet the chance at here, South Korea. We believe that the results are going to being used somewhere.

Appendix A

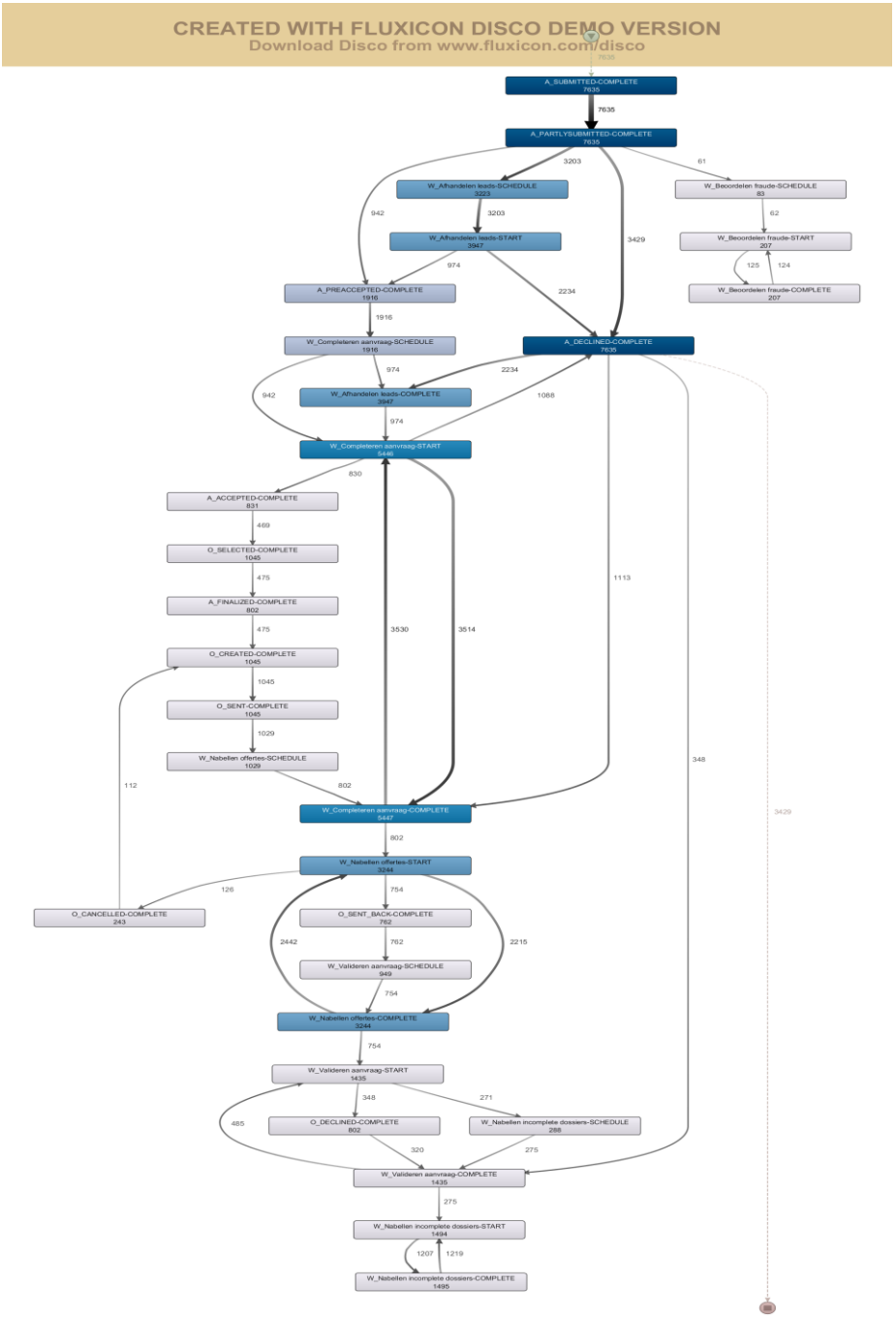
A.1 The process model including *A_ACTIVATED*



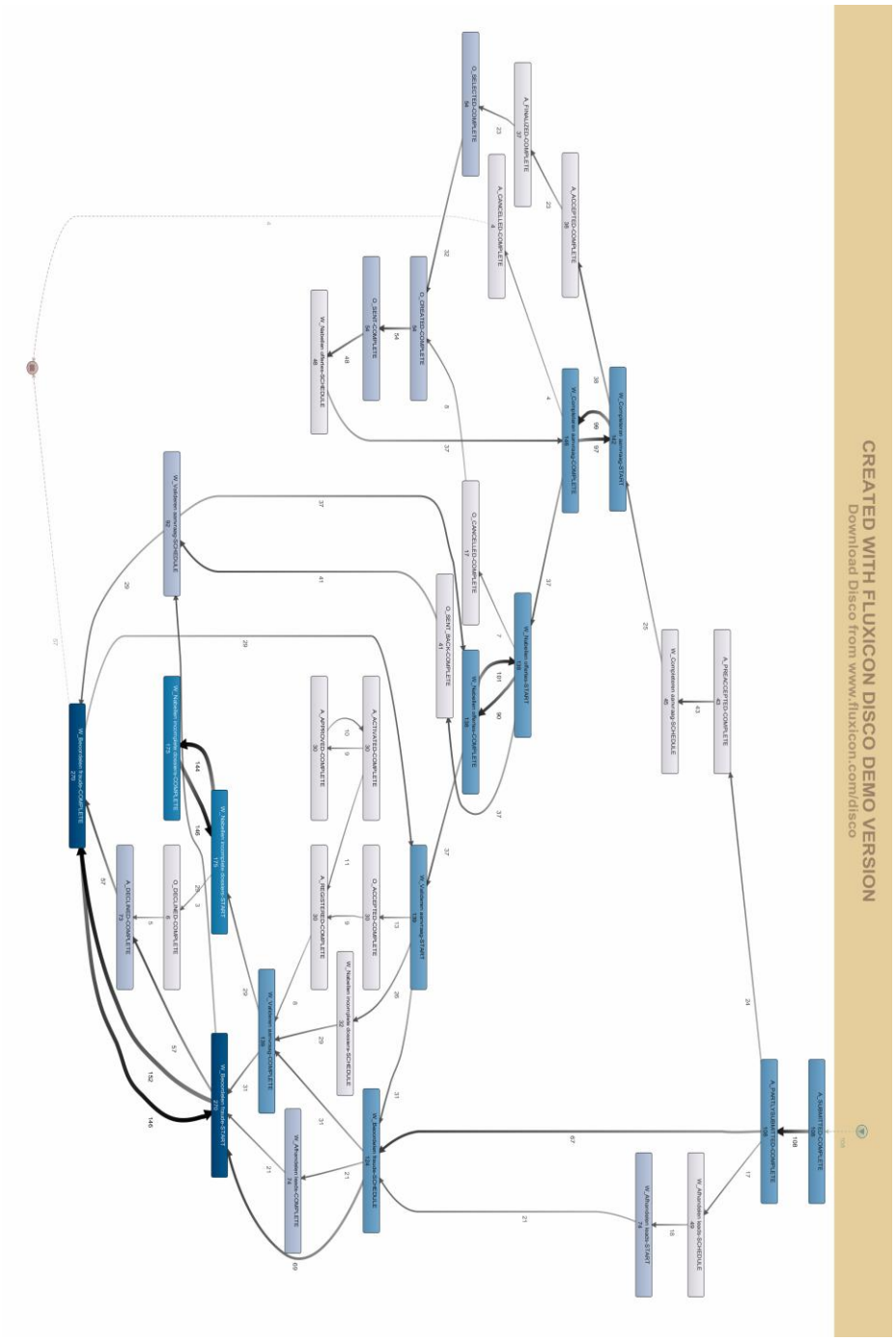
A.2 The process model including *A_CANCELLED*



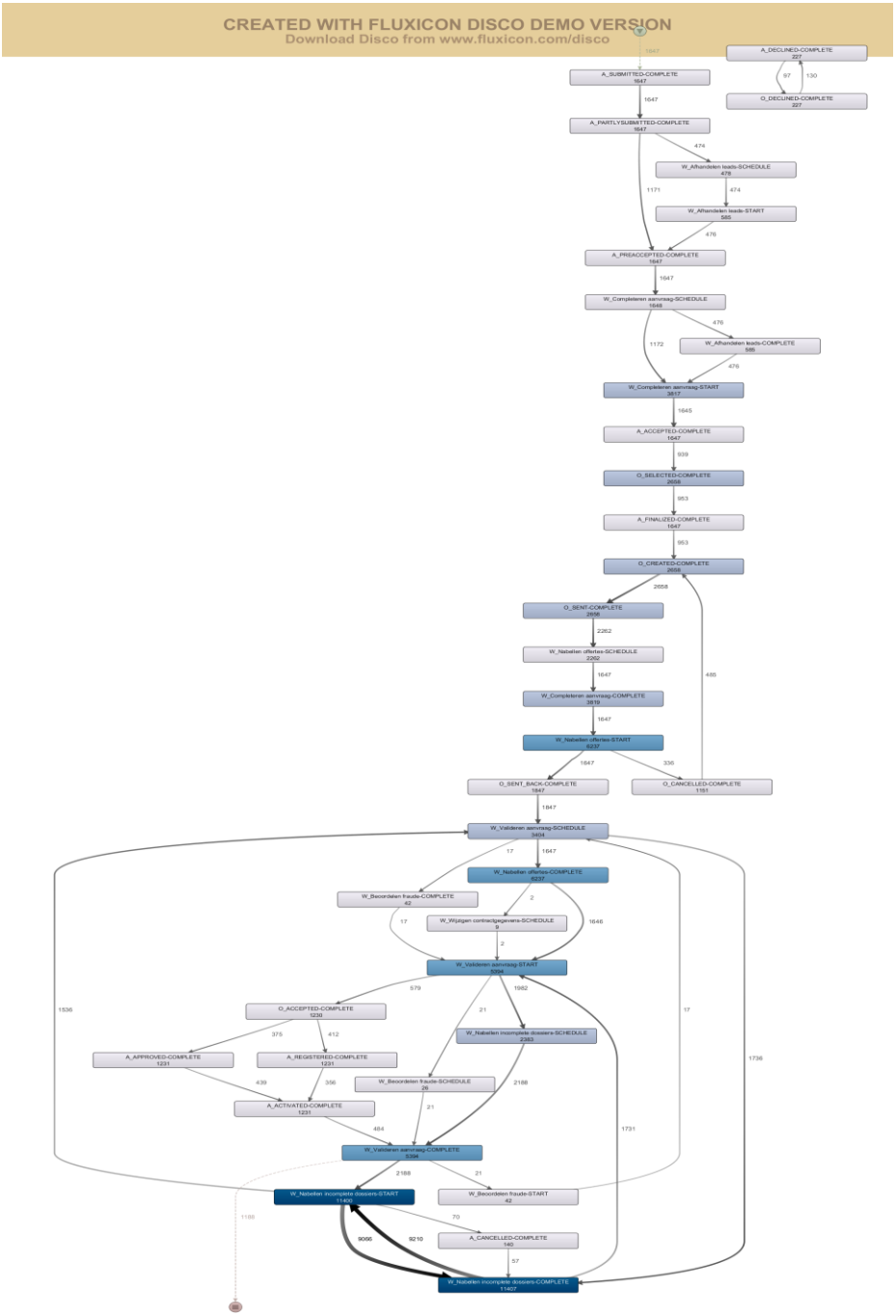
A.3 The process model including *A_DECLINED*



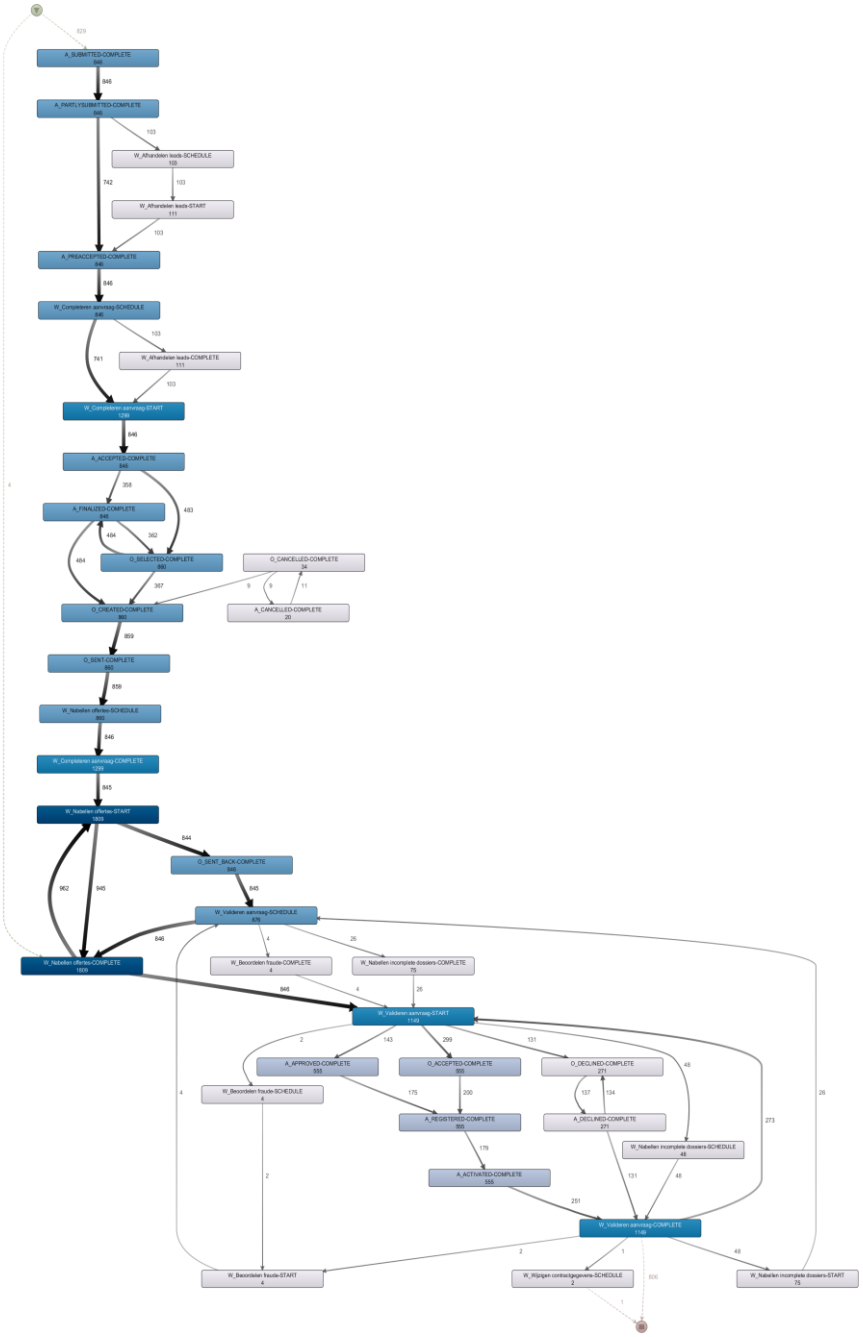
A.4 The process model including *W_Beoordelen fraude*



A.4 The process model including *W_Nabellen incomplete dossiers*



A.5 The process model including cases that the amount value is under 3000



A.5 The process model including cases that the amount value is over 30000