Analysis of Loan Application Process Using Process Mining

Raimundo Carmona, Rodrigo Cofré, Carolina Naranjo, Orlando Vásquez, Jonathan Lee, Juan P. Salazar Fernández, and Michael Arias

Computer Science Department, School of Engineering
Pontificia Universidad Católica de Chile, Santiago, Chile
[racarmona, rlcofre, cnnaranj, oevasque, wllee, jpsalaza, m.arias]@uc.cl

Abstract. The BPIC Challenge 2017 provides an event log based on a real financial entity that represents the loan approval process triggered by the clients' applications submission. The mentioned log consists of 31,509 cases, there are 4,047 variants and from the whole process are recognized 26 activities. The analysis was done using process mining tools and spreadsheet manipulation. The challenge proposed four questions that would allow us to explore certain subsets of the data and analyze them. The first question is about the throughput times of the process; the second question concerns about the conversion rate of multiple and single offer cases; the third question try to discover the system users influence in the final outcome of its clients application; and the fourth questions is about the behavior of clients whose offers are cancelled. We found that clients were slower than bank's workers and this happens in spite of the type of case. Also, the system user - client affects the final status of a given offer and that the value of this type of interaction can be quantified. Besides, the conversion rate is similar between single and multiple offers. We proposed an inactivity model to make predictions about the possible outcomes given certain characteristic of each case. Finally, we give some recommendations to handle these situations.

Keywords: process mining, process analysis, process prediction, probabilistic analysis.

1 Introduction

In the context of the Business Processing Intelligence Challenge (BPIC) 2017 [1], we took the task of analysing a real life event log. We used open source and commercial tools, and our own tools. The analysis of the process represented in the provided event log helps to get a better understanding of it and the role that process mining can take in the different industries.

The log provided by BPIC 2017 is about a Loan application process of a real financial institution. Through all the analysis done to the presented data, we aimed to understand in detail particular components, interactions and how they influence the process itself. To do so, we separated the analysis into sections

that consider the process' throughput time, what and who makes it slow, the overall behavior of each case during different stages of the process and how this behavior changes through the process itself.

To perform the analysis we mainly used process mining tools and techniques that could help us get a better insight about the process and be able to describe the whole process as clearly as possible. With this, we attempted to generate an analysis that is not only helpful for the bank, but also innovate in a sense that gives answers to problems that are not easily addressed using conventional analysis tools nor using conventional approaches, and it is on this issue that we believe that we make a contribution.

The paper is structured as follows: Section 2 describes the given process in the BPIC 2017, explains the log that represents this process, presents the questions proposed in the challenge, and the tools used to respond these questions. Section 3 presents the analysis of throughput times in the process. In Section 4, we present the relationship between system workers or employees and the outcome of the process. Section 5 analyze the conversion rate of single and multiple offers. Section 6 comes with an innovative approach to study cancelled offers. Finally, in Section Section 7 we conclude the paper, where a global conclusion of the general problematic that the bank faces is presented, along with recommendations on how to handle this situation.

2 Understanding the data and its process

2.1 The Process

The data that defines the bank's loan process was published in the context of the BPIC 2017 [2]. It comprises a main data log provided by a financial institution, which describes the lifecycle of a loan application, and a sub data log [3] that contains the information of the offers made in each of those applications. All information contained in the offer data log can also be found in the main application data log. In the Business Process Intelligence Challenge from 2012[1], a similar data set was given. The difference is that this time there are not only cases where a single offer is made for an application, but multiple offers can also be made for a single application. Both data sets were provided in *.xes* format.

In 2012, the BPIC organization described the ideal scenario of a successful application as follows:

> *"An application is submitted through a webpage. Then, some automatic checks are performed, after which the application is complemented with additional information. This information is obtained through contacting the customer by phone. If an applicant is eligible, an offer is sent to the client by mail. After this offer is received back, it is assessed. When it is incomplete, missing information is added by again contacting the customer. Then a final assessment is done, after which the application is approved and activated."*

[1] http://www.win.tue.nl/bpi/doku.php?id=2012:challenge

In the BPIC 2017 more offers can be sent after the first one. Official BPIC forum indicates that an application can be denied if it doesn't meet the acceptance criteria. Also, an application can be cancelled if the client doesn't send his documents or if he says that he doesn't need the loan anymore. After studying the official available information, it was observed that there are three main endpoints to the process: *Application Approved*, *Application Denied* and *Application Cancelled*. Also, there is only one starting point: *Application Created*.

2.2 The Log

A first and simple analysis of the data log using two process mining tools (Disco[2] and Celonis[3]) showed that it consists of 31,509 application cases. There are 4,047 variants of the process and some of them are incomplete cases. There are 26 activities that may be divided in sub activities. For example, *W_Call after offers* is one activity that can be at *schedule*, *start*, *suspend*, *resume*, *start* and *ate_abort* states. There are 3 families of activities: Application state activities (Activities that start with *"A_"*), Offer state activities (Activities that start with *"O_"*) and Workitem activities (Activities that start with *"W_"*). Notable activities and their description (found in BPIC 2017 forum) can be seen in Table 1.

In order to complete the analysis, additionally to Disco and Celonis, open source process mining tool ProM 5.2[4], and ProM 6.6[5] were used in order to process and overview the data logs. Also MS EXCEL[6] and Python[7] were used in order to further work with the resulting data sets and get useful statistical results. In particular, Python was used to make a probabilistic analysis to achieve process prediction.

The analysis was guided by the following topics:

1. Which are the throughput times in the process?
2. How do the system users influence the final outcome of its clients application?
3. What is the behavior of the Conversion Rate and multiple offers?
4. Is it possible to predict if an application is going to be cancelled due to inactivity?

These questions will be further explained in the next sections.

[2] https://fluxicon.com/disco/
[3] http://www.celonis.com/en/
[4] http://www.promtools.org/doku.php?id=prom52
[5] http://www.promtools.org/doku.php?id=prom66
[6] https://products.office.com/en-us/excele
[7] https://www.python.org/

Activity Name	Description
A_Create Application	An application is created by a user from the website or a new application was started by the bank.
A_Submitted	A customer has submitted a new application from the website.
A_Concept	The application is in the concept state: after its creation, a first assessment has been done to it automatically.
A_Accepted	After a call with the customer, the application is completed and assessed again. If there is a possibility to make an offer, the status is accepted. The employee now creates 1 or more offers.
A_Complete	The offer(s) has(ve) been sent to the customer and the bank waits for the customer to return a signed offer along with the rest of the documents (payslip, ID etc)
A_Validating	The offer and documents are received and are checked. During this phase the status is validating.
A_Incomplete	If documents are not correct or some documents are still missing, the status is set to incomplete, which means the customers needs to send in documents.
A_Pending	If all documents are received and the assessment is positive, the loan is final and the customer is payed.
A_Denied	If somewhere in the process the loan cannot be offered to the customer, because the application doesn't fit the acceptance criteria, the application is declined, which results in the status 'denied'.
A_Cancelled	If the customer never sends in his documents or calls to tell he doesn't need the loan, the application is cancelled.
O_Created	An offer was created by an employee.
O_Sent	An offer was sent to the customer by mail and/or via internet.
O_Accepted	An offer is accepted by and employee.
O_Returned	A sent offer is returned by a customer.
O_Refused	An offer is refused by an employee.
O_Cancelled	An offer is cancelled by an employee.
W_Call after offers	A worker calls a customer after an offer was made.
W_Validate application	A worker is validating documents and offer returned by a customer.
W_Asses potential fraud	A worker asses if a fraud is possible.
W_Call incomplete files	A worker calls a customer because of incomplete files.

Table 1: Table of Activities

3 Question 1. Analysing throughput times in different loan application scenarios

The question addressed in this section is about trying to recognize the slower throughput times in the process, and who is responsible of that: clients or bank's users. Throughput times of any process must be known to understand and fully comprehend it. That knowledge also allows an effective allocation of resources if the process needs to be optimized. In large processes, such as the one presented in the BPIC, bottlenecks and slow sub processes must be known to correctly distinguish between what needs to be tackled and what doesn't influences times that much. Further study is done by comparing key subprocesses throughput times between different loan goals cases.

3.1 Question 1. Preprocessing the Data

To tackle this issue, specific preprocessing, cleaning and simplification was done to the log. First, some data log columns were discarded, like offer attributes columns such as *Credit Score*, *Number of Terms* and *First Withdrawal Amount*, to simplify the log for its analysis. Those columns were deleted while importing the log file into Disco. Second, also using Disco, similar activities were merged in order to clean the log from noise and non contributing information. This activities usually were an instant follow up from other activities. Third, incomplete cases that may have hindered the data interpretation and data processing were removed from the data log. Disco was used in order to filter out cases that didn't finish in a valid endpoint. Finally, the data log was separated in four different logs depending on the final activity of the case and the corresponding percentage log coverage was calculated with Disco: a log that contained the cases where the application was completed successfully when only one offer was made (Cases where final activities was *A_Pending*, corresponding to 40% of the total); a log that contained the cases where the application was completed successfully when more than one offer was made (Cases where final activities was *O_Cancelled* after *A_Pending*, corresponding to 14%); a log with cases where the application was cancelled (32% of cases, where final activity was *O_Cancelled* after *A_Cancelled*); and a log with cases where the application was denied (11% of cases, where final activity was *O_Refused* after *A_Denied*).

Those 4 sub logs covered 97% of the total cases in the original log. This division was done in order to better analyze the whole process, to comprehend the throughput times and in order to make a comparison between the possible outcomes of the process.

3.2 Question 1. Analysis and Results

The analysis started by finding the mean duration time of each type of case. Each log was processed with Disco to obtain the following mean times: cases where the application was successful with a single offer took 16 days to finish, cases

where the application was successful with multiple offers took 23 days to finish, cases where the application was cancelled took 30 days to finish and cases where the application was cancelled took 16 days. Next, each one of the four logs were processed with Celonis to obtain a summary of the throughput times. Celonis gives a summary of all throughput times between each pair of followed activities presented in the log. Taking that into account and considering the type of case that represented each log, the slower and most important throughput times were taken from the program. Table 2 shows throughput times of single offer completed application cases; Table 3 shows multiple offer completed application cases; Table 4 presented cancelled application cases; meanwhile, Table 5 shows denied application cases.

Starting Activity	Ending Activity	Mean Time	Median Time	% of Cases
A_Complete	W_Validate application	8.6 days	7 days	97%
A_Complete	A_Validating	8.4 days	7 days	42%
A_Incomplete	O_Accepted	5.8 days	3.7 days	27%

Table 2: Single Offer Completed Applications Cases

Starting Activity	Ending Activity	Mean Time	Median Time	% of Cases
A_Complete	W_Validate application	9.3 days	7.8 days	49%
A_Complete	A_Validating	9.1 days	7.7 days	28%
A_Incomplete	O_Accepted	6.2 days	3.9 days	29%
O_Sent (mail and online)	A_Validating	7.8 days	6.7 days	25%
O_Sent (mail and online)	W_Validate application	7.5 days	6.6 days	49%
A_Complete	O_Create Offer	6.5 days	3.9 days	49%

Table 3: Multiple Offers Completed Applications Cases

Starting Activity	Ending Activity	Mean Time	Median Time	% of Cases
O_Sent (mail and online)	A_Cancelled	27.6 days	30.7 days	10%
A_Complete	A_Cancelled	27.4 days	30.7 days	78%
A_Complete	A_Create Offer	8.7 days	7.2 days	11%

Table 4: Cancelled Aplications Cases

Starting Activity	Ending Activity	Mean Time	Median Time	% of Cases
A_Complete	W_Validate application	9.3 days	7.7 days	82%
A_Complete	A_Validating	9.2 days	7.7 days	51%
O_Sent (mail and online)	W_Validate application	7.8 days	6.9 days	12%
A_Complete	O_Create Offer	7.1 days	5 days	13%
O_Returned	A_Denied	3 days	2.2 days	58%

Table 5: Denied Applications Cases

From results in Tables 2, 3 and 5, it can be seen that in successful application and denied application cases, waiting for the client to send the requested documents (*A_Complete* to *W_Validate application* or *A_Validating*) is the slower sub process. It takes near 9 days to receive the documents from the client, a little less when the application was a single offer one and successful. When the application was cancelled (Table 4), the sub process of waiting for the client to send the documents (*A_Complete* to *A_Cancelled*) takes around 27.5 days, with a median of 30.7 days, which is the maximum waiting time before an application is cancelled because of clients inactivity.

When a case had more than one offer created (Tables 3, 4 and 5), the subprocess of creating a new offer (*A_Complete* to *O_Create Offer*) took 6 days when the application was successful, 9 days when the application was cancelled and 7 days when the application was denied. Also, Table 5 shows that the subprocess of denying an application (*O_Returned* to *A_Denied*) took 3 days.

An interesting question would be if this tendencies are maintained when separating the cases by their loan goal. If different loan goals show different client behaviors, measures can be taken to hasten some processes from the beginning. Disco was used to get the distribution of cases separated by their loan goal (See Table 6). *Car* applications are the most common among all type of cases, then *Home Improvement* and *Existing Loan Takeover* cases. Column 'Others' gathers 11 other type of loan goals which cover near 28% of the cases. Only *Car*, *Home Improvement* and *Existing Loan Takeover* cases were taken in consideration to further study throughput times of the process. The other loan goals were not considered because of their small coverage of the log data.

Application Case Type	Car	Home Improvement	Existing Loan Takeover	Others
Completed Single Offer	27.7%	26.2%	17.7%	28.4%
Completed Multiple Offers	26.3%	24.7%	20.8%	28.2%
Cancelled	32.5%	22.3%	16.7%	28.5%
Denied	29.7%	22.5%	21.4%	26.4%

Table 6: Distribution of applications based on Loan Goal

Disco was used to filter and separate each of the four logs into three new sub logs, one for each loan goal. Table 7 shows throughput times (considering average times) of slower sub processes analyzed before. This results were obtained with Celonis after processing each new log. It can be seen that in spite of the type of case and sub process, cases where the loan goal is *Car* always take less time that cases with *Home Improvement* and *Existing Loan Takeover* goals. This shows that the loan goal may affect the duration of the application process and the behavior of the client in said process. Measures can be taken to address this situation and, for example, give priority to cases whose loan goal is different to *Car*. Finally, the sub process of waiting for the client kept taking more than 7 days to finish. Then, to reduce the time that takes an application case to finish, clients participation in the process, along with communication between the bank and clients be addressed in a way that manages to lower the time that takes them to send the documents.

Application Case Type	Starting Activity	Ending Activity	Car	Home Improvement	Existing Loan Takeover
Completed	A_Complete	W_Validate app.	7.9 days	9.4 days	9.5 days
Single Offer	A_Complete	A_Validating	7.9 days	9.2 days	9.3 days
Completed	A_Complete	W_Validate app.	8.3 days	10 days	9.8 days
Mult. Offers	A_Complete	A_Validating	8.5 days	9.7 days	9.9 days
Cancelled	A_Complete	A_Cancelled	26.9 days	28.3 days	27.9 days
	O_Sent (mail ...	A_Cancelled	27.1 days	27.1 days	28.6 days
	O_Returned	A_Denied	2.9 days	3.1 days	3 days
Denied	A_Complete	W_Validate app.	9 days	9.6 days	9.6 days
	A_Complete	A_Validating	9.2 days	9.7 days	9.5 days

Table 7: Throughput Times Based on Loan Goal

4 Question 2. Analysing the influence of applicant-system user interaction

In any real life process, there are different kind of interactions between its resources. In the case of the log presented for the BPI Challenge, the majority of the executed activities involve users of the observed system (i.e. the bank's employees) interacting with the system itself. But there are activities that needs information or documents from the applicant and therefore an applicant-system user interaction is achieved. In this section, we seek to answer the BPIC 2017 question of how the users influence or affect the final outcome of the offers done to its clients and also intend to quantify the value of this user-client interactions. The only activities that involve user-client or user-applicant interaction are *W_Call incomplete files* (i.e. request missing information from the applicant)

and *W_Call after offers* from now on referred as *Call for completion* and *Call after offers*, respectively. Finally, in order to get a better understanding of the executed procedures, each question has been separated and the obtained results are presented in the following subsections.

4.1 Question 2. Analysis and Results

We used Disco to extract from the given log the cases where at least one call for completion was executed. There is a total of 15,300 application where at least one call for completion was made, representing 48.56% of the total number of applications. Table 8 shows how the number of calls received by the applicants is distributed among the mentioned extracted data from the log. In this anal-

Number of calls						
1	2	3	4	5	6	7
60.9%	26.21%	8.53%	2.64%	1%	0.43%	0.34%

Table 8: Distribution of calls

ysis, we considered three final outcomes that complete any given trace. These are *O_refused*, *O_Cancelled* and *A_pending*. For these outcomes, we divided them into bad or good outcomes, considering the offer cancelled or refused as a bad outcome and the application pending as a good one. it's important to see that both classifications can be seen as bad or good for the applicant or the bank itself. From the applicant's perspective, not reaching the loan means that they cannot fulfill the plans for which the application was requested; From the bank's perspective, not completing the pending applications means that it loses a contract with a client for a determined amount of money (it loses a debt).
Considering the proposed classifications. In order to obtain information about the client's offer acceptance rate, we separated the information using MS EXCEL. We obtained a scatter plot shown in Figure 1.

In spite of the R^2 statistic with value 0.758 could be considered low, we can see that as the number of completion calls done increases the percentage of bad outcomes increases as well, but if we remove the point related to one *call for completion* then as shown in Figure 2 the R^2 statistic goes up to 0.81 and we can see that the relation between this two variables holds strongly. This relation could mean that probably the lack of information given by the applicants is simply due to the non-fitness of the applicants for the loan they applied to. Also, this result might reflect that there is a systematic misinformation from the applicants before they apply to a loan, therefore it is important to check if the information about the loans or specifically if the conditions that the applicants need to comply for different kinds of loans is clearly specified and informed. This

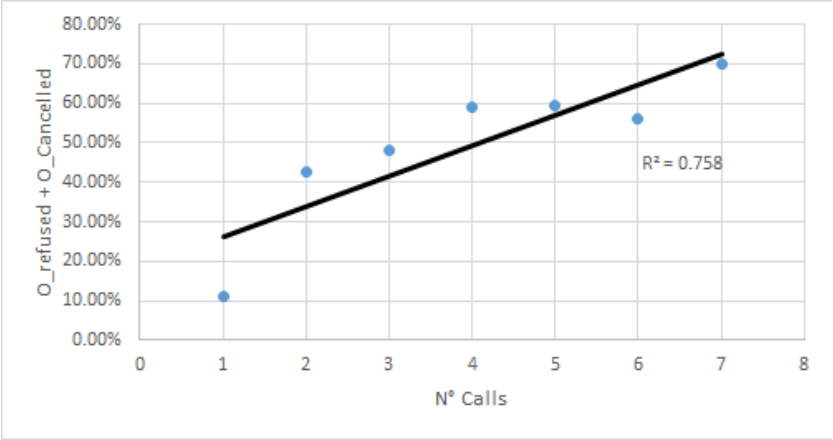

Fig. 1: percentage of bad outcomes by number of calls done

possibly misinformation produces a series of inefficiencies that traduces to a bad use of the employees' schedule.

Once taken into account one of the activities that involve applicant-system user interaction. We now consider the activity *W_Call after offers*. In this case we used the first activity analyzed (i.e. *Call for completion*) to separate the set of application that received a call after offers, and discover if there is any behavioral difference between these sets.

These calls are made by employees who are identified in the system as users. For this part of the analysis we considered three variables associated with each user, the *number of calls after offers* done($\#Calls$), the *number of application* where a call after offer was done and ended in *A_Pending*($\#A\_Pending$) and the total amount of money associated with its clients' applications(*Total amount*). Each one of them made a determined number of calls after offers with a certain percent of success (i.e. the final outcome of that trace is *A_Pending*) and has a total amount of money associated with the application of its clients. With this variables, we intended to see through information from the resources' behavior if there is any difference between clients that received a call for completion and those who did not. To do so, we created the following variable

$$User \quad value := \frac{\#A\_Pending}{\#Calls} \times TotalAmount \tag{1}$$

We used the *User value* to quantify the potential amount of money that the respective user could make its clients commits with the bank. Next, we calculated this variable for all the users who made a call after offers in each set obtained by the division made based on calls for completion and then plotted the results. Figure 3 and 4 are the plots of *User value* vs *number of calls done* in the set that did not received any *call for completion* and the one that received at least one call with this purpose, respectively.

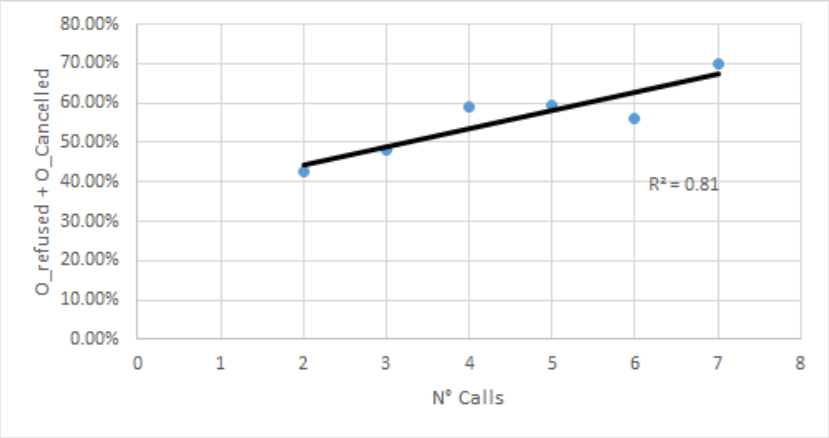

Fig. 2: percentage of bad outcomes by number of calls done

In Figure 3, we can see there is not a clear behavior of the user value as the calls increase. In fact, there are users that do a significant amount of calls (between 100-150) and the final outcome of this cases is not the desirable one, and example of this is the *user_100* who made 174 calls after offers and have an User value of 0. These particular cases (i.e. the ones that made calls after offers and have a user value equal to 0) should be studied to know exactly what is happening and evaluate if it is an avoidable problem. Meanwhile, in Figure 4, we see that the relation between user value and the number of calls done by a resource is stronger. This means that the inefficiencies that occur in the first set are not present in this one. We believe that the early contact with the client helps to gain information about its current situation during the application process and thus the employees can adjust its time accordingly to the clients' needs and potential value. It would be helpful to quantify how much potential debt the bank loses by not doing an early contact with its clients and to determine if the cost of doing this early contact is greater than the expected debt that could be acquired.

5 Question 3. Conversion Rate and multiple offers behavior

In this section we focused on the analysis of the factors that can influence a request of multiple offers by the client. We focused on preprocessing the data and analyze the characteristics that the offer has to see if there is a behavior pattern.

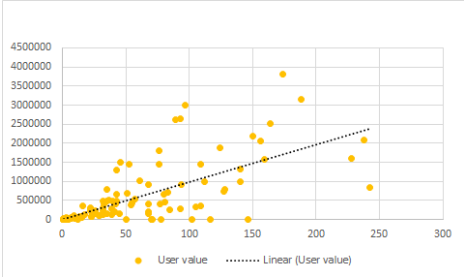

Fig. 3: User value vs N. Calls(without calls for completion)

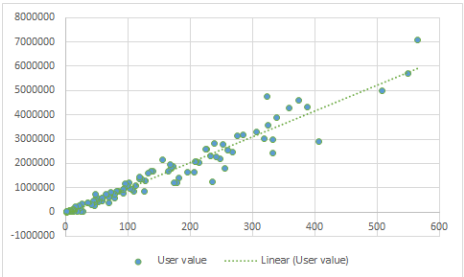

Fig. 4: User value vs N. Calls(with at least one call for completion)

5.1 Question 3. Preprocessing the Data

To answer the proposed questions, and perform an analysis in order to provide some recommendations. We used different tools that helped us to simplify and understand the log, find the answers that we were looking and an analysis that helped us to make some recommendations.We used Microsoft Excel for the preprocessing of the data, the clustering and cleaning of the activities of the log, and the creation of new, and more specific activities. Specifically, we created new activities to check for the answers to the proposed questions and find the way that the offers were made to the clients. The other tool that we used to answer this question was Disco. This tool was used to get a better understanding of the process. With this we filtered and got a more detailed information to see patrons in the model and the log traces.

To obtain an answer to the proposed question, it was important to preprocess the data in order to obtain more clear and representative results. For the analysis of the amount of offers made, we cleared of the *Application* (A) and *Workflow* (W) related activities. With this the log that was processed only contained *Offe*r (O) related activities. After this, we assigned a number that showed the position in which the offer was made and this helped to identify the offer that was accepted by the position in which it was proposed to the client.

After the preprocessing step, we were able of identify that the maximum number of offers that was given to the client, besides the distribution the amount of offers created. From that information we were able to find that the maximum number of offers given to the client is 10, but this case only occurs twice as is showed in the Table 9.

To get the amount of offers accepted in both cases, single and multiple offers, we preprocessed the log again to delete all the *Offer* activities that not leaded to an ending. In this case we only kept the activities related to *Create*, *Accept*, *Cancel* and *Refuse*.

5.2 Question 3. Analysis and Results

Figure 5 shows the new model. There are a maximum of ten offers, the number of single offers made is 22,950 and decrease fast to the 2 times that ten offers were made (see Table 9).

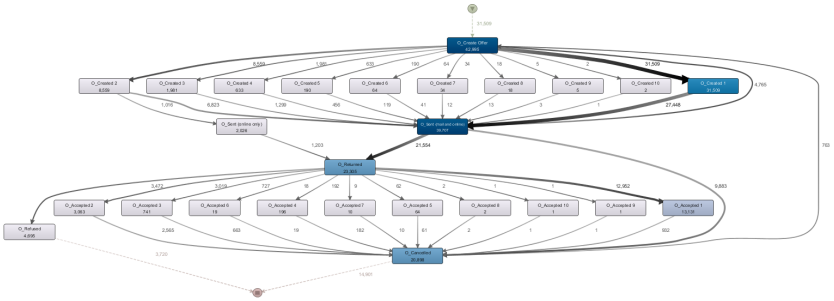

Fig. 5: Model with new activities and endpoints

	Offers Made										
	1	2	3	4	5	6	7	8	9	10	Total
N.of Offers	22,950	6,578	1,348	443	126	30	16	13	3	2	31,509
% of Offers	72.84%	20.88%	4.28%	1.40%	0.41%	0.10%	0.06%	0.03%	0.01%	0.01%	100%

Table 9: Number and percentages of applications by number of offers

The number of the times that a single offer was made correspond to the 72.84% of the cases, while the number of times multiple offers were made is 8,559 cases, which represents the 27.16% of the cases in the analyzed log as can be seen in Table 9. The amount of the single offers that were accepted was 12,178

offers (Table 10), while for the multiple offers the number was 5,049 this include cases in which multiple offers were made but the first was accepted. With this we can see that the conversion rate for a single offer is 53.12% and for multiple offers is 59.18%. We have to take into account that the number of times that the first offer was chosen when there were multiple offers was 940, that is, the 11% of the cases that have multiple offers. Table 11 shows how the accepted offers are distributed by the number in which were made.

		Offers Made										
		1	2	3	4	5	6	7	8	9	10	Total
	1	12,178	862	74	14	1	-	1	-	-	-	13,130
	2		2,913	135	13	1	1	-	-	-	-	3,063
	3			675	53	9	2	2	-	-	-	741
	4				183	11	2	-	-	-	-	196
	5					61	2	-	1	-	-	64
Offers accepted	6						16	3	-	-	-	19
	7							7	2	1	-	10
	8								2	-	-	2
	9									1	-	1
	10										1	1
	Total	12,178	3,755	884	263	83	23	13	5	2	1	12,227

Table 10: Offers accepted by offers made

We can see that the 76.25% of the accepted offers are the first offer made to the client, the second offer is accepted a 17.73% of the times, the third a 4.30% and the fourth a 1.14%. If we add all the offers left they represent the 0.59% of the accepted cases.

	Offers Accepted										
	1	2	3	4	5	6	7	8	9	10	Total
Offers accepted	13,130	3,063	741	196	64	19	10	2	1	1	17,227
% of Offers accepted	76.25%	17.73%	4.30%	1.14%	0.37%	0.11%	0.06%	0.03%	0.01%	0.01%	100%

Table 11: Number and percentages of the offers accepted

Since there were few cases in which there were more than four offers. We focused on of the factors that can influence the conversion rate and, specifically, we filtered by the Loan Goal. As we can see in the Table 6 we got that the types of loans that have the bigger amount of accepted offers are: Car (27.80%), Home

improvement (26.10%), Other (8.77%), Unknown (8.76%), Remaining debt home (3.16%), Extra spending limit (1.92%) and Caravan/Camper (1.22%). While the reasons with lower percentage of accepted offers are: Motorcycle (0.74%), Boat (0.67%), Tax Payments (0.42%) Business Goal (0.05%) and Debt Restructuring (0.00%).In the Figure 6 we can see that only 6 of the 14 different kinds of goals have offers accepted after the 5th. And from this only 4 of them have more than the 6th offer accepted.

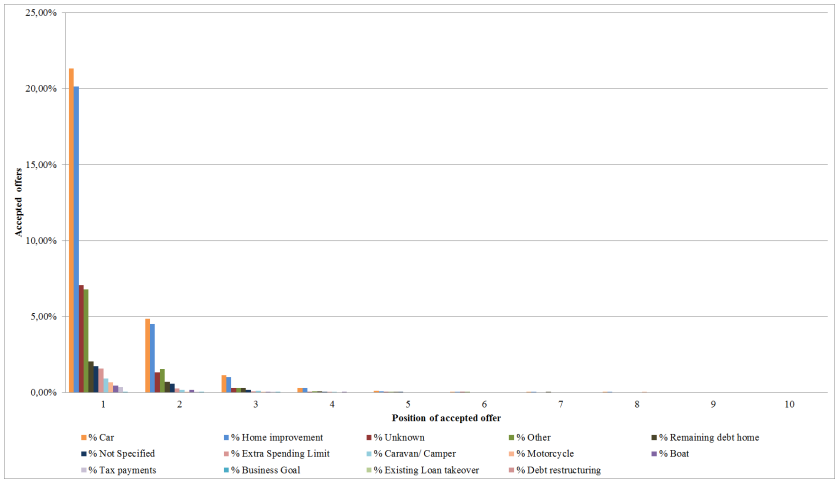

Fig. 6: Percentage of accepted offers vs Position of the offer for each loan goal

From the previous results it can be concluded that the number of offers accepted that were given to the user after the sixth offer are unimportant. To optimize the process they can be removed by a new business rule, this because they represent only the 0.61% of the offers made an the 0,59% of the offers accepted. Besides, there aren't successful cases for the *Debt Restructuring*, because of that we recommend stop offering that kind of loans. On the same side we recommend an analysis of the *Business Goal Loans*, because they only represent the 0.05% of the cases with a 26.67% convertion rate. Since removing this kind of loan means tha of 30 offers only 8 were accepted, the resources spent in this type of loans can be focused in a loan goal that has a bigger convertion rate like *Remaining debt home* or *House Improvement Loans* (see Table 12).

6 Question 4. Customer application rejection analysis

As part of the of the BPIC 2017, and in order to present a creative analysis, in this section we focused on the behaviour of the customers before their application were cancelled. We identified types of users and analyzed patterns of the customers that finished their application in the status *A_Cancelled*. Then,

	Loan Goals		
	Offers Made	Offers Accepted	Convertion Rate
Boat	201	115	57.21%
Business Goal	30	8	26.67%
Car	9,328	4,787	51.32%
Caravan/Camper	369	210	55.91%
Debt Restructuring	2	0	0.00%
Existing Loan takeover	5,601	3,074	54.88%
Extra Spending Limit	625	331	52.96%
Home improvement	7,669	4,495	58.61%
Motorcycle	275	128	46.55%
Not Specified	1,065	439	41.22%
Other	2,985	1,510	50.59%
Remaining debt home	842	544	64.61%
Tax Payments	152	72	47.37%
Unknown	2,365	1,509	63,81%
Total	31,509	17,227	-

Table 12: Number and percentages of the offers accepted

with this patterns we were able to construct a probabilistic model that will allow the bank to predict which applications are more likely to be cancelled and take action against this possibles outcomes.

In order to find the reasons that some customers' applications are rejected it is necessary to identify all the components (systems and users) involved in the process. For this reason, we evaluated the event log from an organizational point of view to discover which types of resources are involved in this process.

Using the BPIC 2017 log without performing any preprocessing, we ran the plugin *Similar-Task social network* (available in PROM 6.6) to find the groups of resources that are shown in Figure 7

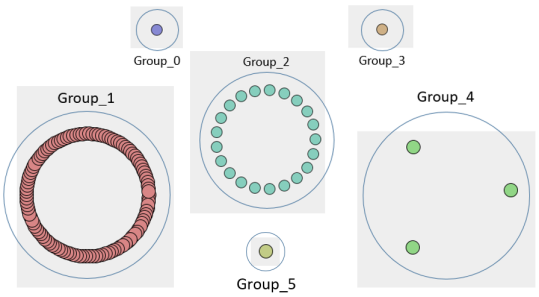

Fig. 7: Groups of resources found by *Similar-Task social network*

The obtained outcome presented in Table 7, only shows similar resources, but it doesn't explains their similarities. For this reason, we tried the *Organizational* plugin of PROM 5.2. Which is able to return a cluster assignment of the activities associated to each discussed group. Activity information was condensed in Figure 8 and Figure 9. Figure 8 shows the activities that distinguish between resources and Figure 9 summarizes statistical information of the resources and activities.

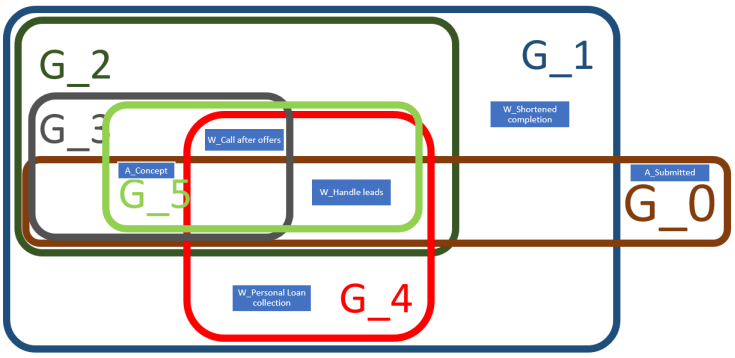

Fig. 8: Activity differences between groups

Looking at Figure 9, we found that exists 3 groups of user that only has 1 user instance each one. These users corresponds to company systems. In particular we noted that all activities of Group_0 belongs to *Application* category while activities of Group_3 and Group_5 go across all the other categories. In addition, looking at Figure 8 we can appreciate that the activity *A_Submit* is only done by group_0. Using this information we concluded that Group_0 corresponds to company web system where customers access to bank services.
The remaining 3 groups are formed by company employees who are in charge of *receive*, *validate* and *process* customers applications.
Once determined the user assignment for each group using Disco, we filtered the log to obtain the cases where the customer created an application and it was cancelled by an employee or a bank system. Using this traces we are able to study the most likely paths that an application goes through before getting cancelled and therefore is the first step into the probability model construction.

6.1 Question 4. Analysis and Results

After applying the aforementioned filters to the log, we found that the most important paths that arrive to *A_Cancelled* state come from:

- **A_Complete** : Customers didn't send the documents necessary to get the credit.
- **A_Incomplete** : Customers didn't fulfill the requirements to get the credit.

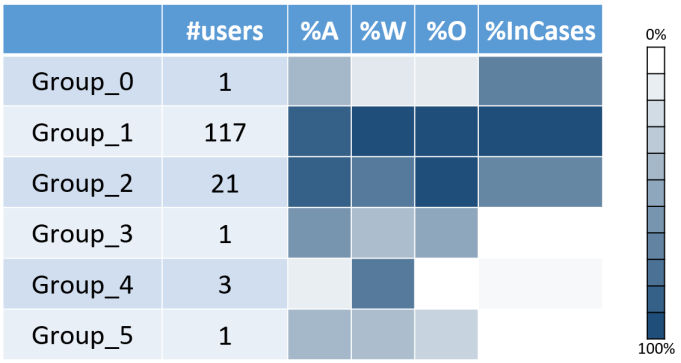

Fig. 9: Participation of each group in each activity type

– **O_Sent (mail and online)** : The company sent offers to a customer, but he never answered.

The frequencies of these paths are presented in Table 13.

Path to *A_Cancelled*	# cases
A_Complete → *A_Cancelled*	803
A_Incomplete → *A_Cancelled*	480
O_Sent (mail and online) → *A_Cancelled*	108

Table 13: Paths to *A_Cancelled* state

Using this traces as input, we use the plugin *Mine Petri net with Inductive Miner* (from Prom 6.6) in order to build an inactivity model (see Figure 10) for customers. This model can be useful to make predictions about the behaviour of new customers. With this objective we created the following formula:

$$P(A|T) = \frac{n}{n + dist(B, A, I)} * fitness(T, I, A) \tag{2}$$

Where the fitness is defined using the concept of *token replay*:

$$fitness(T, I, A) = 1 - \frac{m_A}{c_A} \tag{3}$$

The formula 2 computes the probability of arriving to state A given a sequence of n steps already performed $T = (A_1, A_2 \dots A_n)$. The right hand side is the product of two terms. the first one considers how far is the last known state of the sequence ($B \in T$) from the target state A (if B is far from A the probability of arrival will be smaller). The second measures the fitness of the trace with

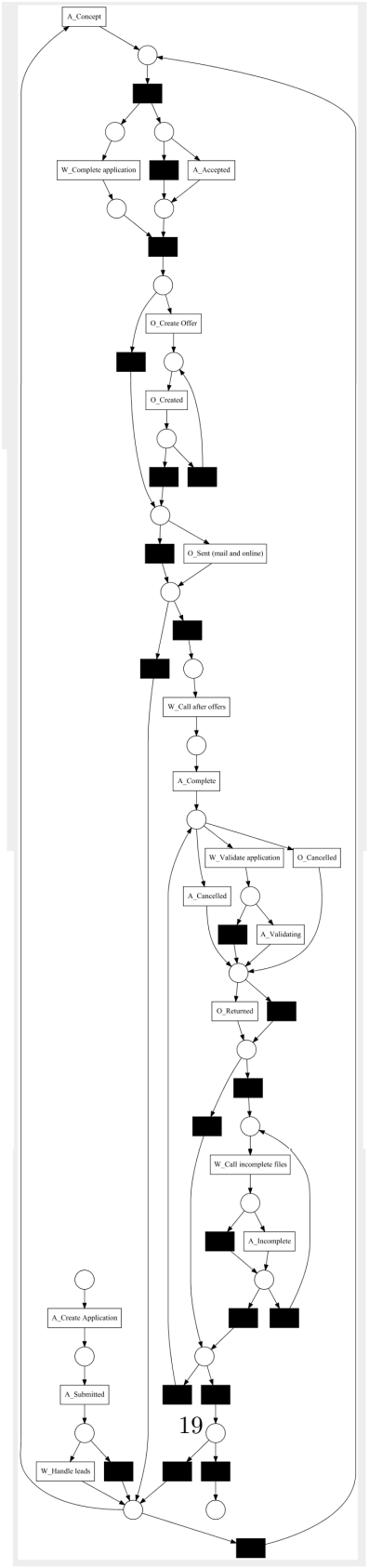

Fig. 10: Inactivity model

respect to the inactivity model (where m_A and c_A are the number of tokens missed and created respectively).

Using this formula we can compute the $P(A\_Cancelled|T)$, which means, the probability that an application would be *cancelled*. This would allow the bank's employees to avoid wasting time evaluating and waiting responses of clients that, with high probability, won't answer.

The use of a transition systems[4] may have been an alternative solution. A transition system might have been already constructed from a specific event log based on a priori information, and this transition system could be used to apply different techniques (e.g., decomposed into specific parts such a (sub)process models) in order to generate a model from the transition system. For simplicity, we prefer the use of Petri Nets approach to create the inactivity model. Therefore, the use of a transition systems are out of the scope of this paper. Additionally, a current limitation of our application rejection analysis has to do with the need to perform a further validation stage that would be useful to test our probabilistic approach.

7 Conclusions

After performing the analysis of the proposed questions in the BPIC 2017, we can conclude that there exist a generalized inefficient use of the employees' time. From each one of the questions analyzed can be obtained a possible action to be taken so the inefficiencies of the process are minimized.

As seen in the first question related to throughput times, the duration of the application depends of its loan goal, so the priority given to a certain application (i.e. when it is handled) could be given by its loan goal in a way that minimizes the employees' leisure time (e.g. application that would slow down the process could be handled later so the process doesn't queue up applications that would be quicker to process).

In the second question that took into account the user-applicant interactions we discovered that an early contact, in this case through the activity *W_Call After Offers*, with the client helps to increase the efficiency of how the applications are handled, then it could be studied whether making an early contact, in this case with the objective of get information about the current status of the client in relation with the application, is cheaper than the expected debt that could be acquire by doing this contact or whether in general this early contact causes a reduction of inefficiencies.

The third question concerns about the conversion rates and multiple offers behavior. Similar to the first one, we showed that depending on the loan goal of an application the chances of an offer being accepted by the client increases or decreases as the number of offers done increases. Then, a redistribution of the offer generator effort could achieve better results if it is focused to client groups that have a better response to a greater number of offers.

Finally, in the last question, we focused on the clients' behavior before cancelling its application. We proposed a probabilistic model that allow us to calculate the

probability of a given incomplete trace of eventually remain inactive and thus not being able to finish the process. Therefore, in order to increase the process efficiency, this model could be used to evaluate every application and set a probability threshold so each time an application passes this threshold an action could be taken against this case, this could mean, calling the client to know whether is going to continue with its application or drop it. Then, by using the probability model the bank could set up early warnings about certain application, take actions against this warnings and finally be able to redistribute the employee's' time into profitable activities (i.e. processing applications with low probability of eventually remain inactive) if needed.

References

1. Business Process Intelligence Challenge (BPIC). 2017. Retrieved from https://www.win.tue.nl/bpi/doku.php?id=2017:challenge
2. van Dongen, B., F.: BPI Challenge 2017. DOI:doi:10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b
3. van Dongen, B., F.: BPI Challenge 2017 - Offer log. DOI:10.4121/uuid:7e326e7e-8b93-4701-8860-71213edf0fbe
4. van der Aalst, W.M.P., Rubin, V., Verbeek, H.M.W., van Dongen, B.F., Kindler, E., Gunther, C.W.: Process Mining: A Two-Step Approach to Balance Between Underfitting and Overfitting. Software and Systems Modeling 9(1), 87–111 (2010)