Stairway to value: mining a loan application process

Ariane M. B. Rodrigues, Cassio F. P. Almeida, Daniel D. G. Saraiva, Felipe B. Moreira, Georges M. Spyrides, Guilherme Varela, Gustavo M. Krieger, Igor T. Peres, Leila F. Dantas, Mauricio Lana, Odair E. Alves, Rafael França, Ricardo A. Q. Neira, Sonia F. Gonzalez, William P. D. Fernandes
Advisors: Simone D. J. Barbosa, Marcus Poggi, and Hélio Lopes

Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, Brazil
arianebueno@gmail.com, calmeida@inf.puc-rio.br, saraiva92@gmail.com, fbaldino@gmail.com, georgesmss@gmail.com, guilhermevarela@hotmail.com, gkrieger@marlin.com.br, goriperes@gmail.com, leilaffdantas@gmail.com, mauriciolana@hotmail.com, oalves.rj@gmail.com, mec.gaseta@gmail.com, ricardo.quintano@gmail.com, s.fiol1987@gmail.com, williamducfer@gmail.com, simone@inf.puc-rio.br, poggi@inf.puc-rio.br, lopes@inf.puc-rio.br

Abstract. The aim of this work is to provide findings and insights from the analysis of the event log of a loan application process performed in a financial institution. In this work, we provide the identified executed process (AS-IS process), its performance and frequency indicators and some recommendations to improve the process. Our final AS-IS model obtained a fitness of 97.28% and a precision of 98.72%, maintaining its simplicity. Then, we present our findings related to the descriptive statistics about resources, cases, and overall productivity. In our diagnostic analysis we show that most bottlenecks are associated with a delay by the applicant to perform an action (*e.g.*, providing documents to the bank).

Keywords: Descriptive analytics, Diagnostic analytics, Process mining.

1 Introduction

Process mining allows discovering, analyzing, and improving business processes using event data. Process mining results are analogous to X-rays, in that they reveal what goes on inside processes and can be used to diagnose problems and suggest proper actions [1]. To apply process mining techniques, all data extracted from systems must be converted to an event log. An event log is composed of a list of events. Each event should have as minimal attributes: a case id (a unique number to represent the case), an activity name to indicate the performed action, and the date and time when the action was executed. This means that each event of the event log refers to an activity and it is related to a particular case [2].

This work aims to provide findings and insights from the analysis of a loan application process performed in a financial institution, through its event log.

In terms of contributions, we first provide the identified executed process (AS-IS process). We aimed to strike a balance between fitness, precision, and simplicity of the model. Our final AS-IS model obtained a fitness of 97.28% and a precision of 98.72%, while maintaining its simplicity. Then, we present our findings related to the descriptive statistics about resources, cases, and overall productivity. In our diagnostic analysis we show that most bottlenecks are associated with a delay by the applicant to perform an action (*e.g.*, providing documents to the bank).

The remainder of this paper is organized as follows. Section 2 explains the loan process data. Section 3 describes how we identified the current process executed, *as per* the event log (AS-IS process). Sections 4 shows a descriptive analysis of some variables of the loan application process, through its event log during the period of observation. Section 5 presents our process analysis. Section 6 answers the 2017 BPI challenge questions. Section 7 makes further recommendations. Finally, Section 8 concludes the work suggesting future work.

2 The Loan Process Data

The BPI Challenge 2017 provided an event log that contains all applications filed in 2016,. In addition, the dataset includes subsequent handling up until February 2^{nd}, 2017. In total, there are 31,509 loan applications. For these applications, a total of 42,995 offers were created.

There are 26 types of activities, which are divided in three types. The first is related to Application (A) state changes. The second type is related to Offer (O) state changes. Finally, the third is related to Workflow (W) events and 149 resources (employees or systems). For all applications, the following data are available:

- Requested load amount (in Euro);
- The application type;
- The reason the loan was applied for (LoanGoal); and
- An application ID.

For all offers, the following data are available:

- An offer ID;
- The offered amount;
- The initial withdrawal amount;
- The number of payback terms agreed to;
- The monthly costs;
- The credit score of the customer;
- The employee who created the offer;
- Whether the offer was selected; and
- Whether the customer accepted the offer.

For each (uniquely identifiable) event, the employee (*resource*) who triggered the event is recorded, as well as the event timestamp and lifecycle information.

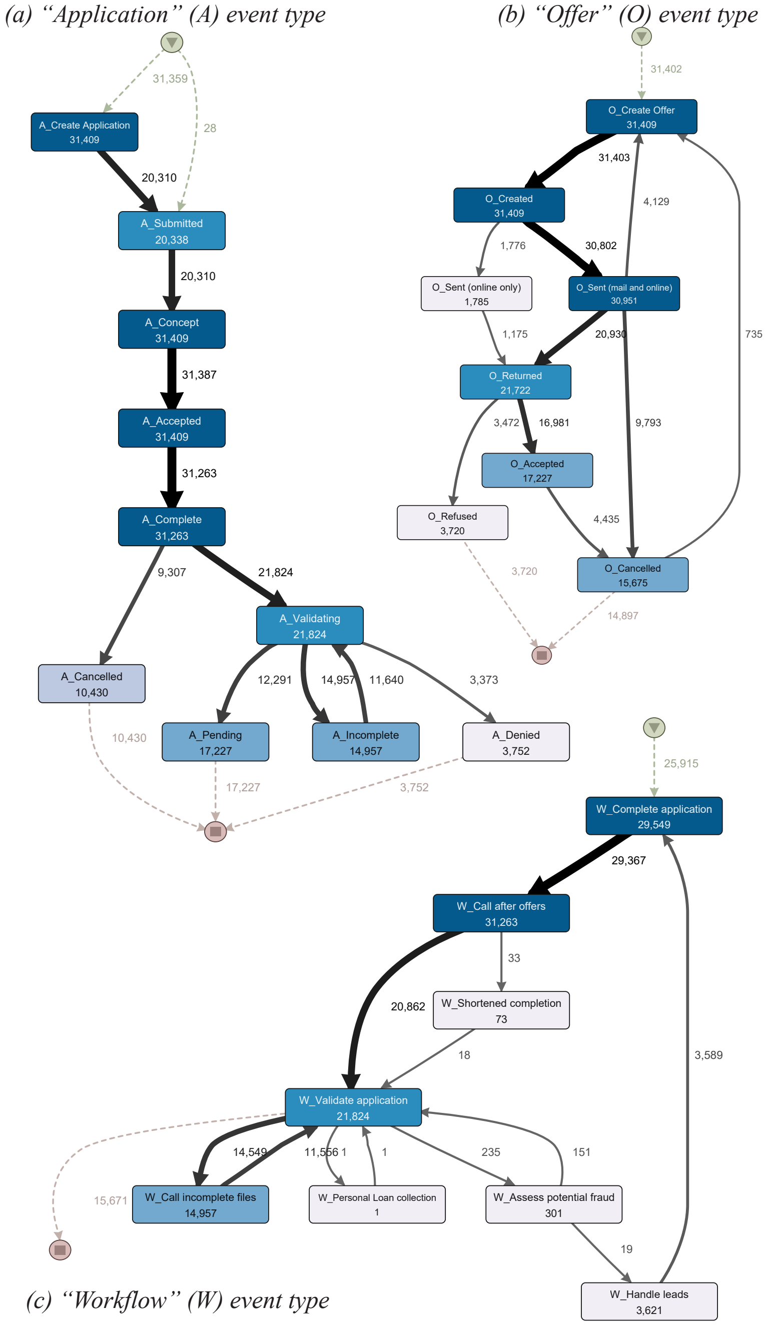
To have a better understanding of the process, we filtered the event log to include only complete cases. Systems normally record data continuously, so the provided event log may contain some cases which had not yet come to a conclusion by the last logged date. As a first step, we used Fluxicon's Disco [3] to remove cases without any of the following endpoint activities: A_Cancelled, A_Denied, A_Pending, O_Cancelled, or O_Refused. Then, we removed cases that had not received a final decision provided either by the bank (approved or rejected) or by the applicant (canceled). In this way, our analysis considered cases which have at least one of the following activities: A_Pending, A_Cancelled and A_Denied. This filtering procedure eliminated 100 of the original 31,509 cases.

3 AS-IS process model

We first identified the process that the institution has followed for providing credit to applicants (the so-called AS-IS process), as represented in the event log. Inspired by the methods presented by Adriansyah and Buijs [4], we separated the filtered event logs in three parts, each one containing a single type of event: "Application" (A), "Workflow" (W) and "Offer" (O). Using the Disco tool with each filtered log, we generated the process model of each type (Figure 1).

We printed the three models, put them together, and manually connected their activities using the four representative process variants in terms of number of cases and outcome (approved, denied, or cancelled). Table 1 presents the process variants we considered for understanding the process. In Variant 1 (12.28% of the cases) the loan applications are canceled due to lack of customer response. In both Variant 2 (6.71% of the cases) and Variant 3 (5.69% of the cases), the loan applications are approved by the bank and accepted by the applicant. The main difference between them is that Variant 2 includes specific activities to deal with incomplete files (W_Call incomplete files, A_Incomplete). Finally, in Variant 9 (1.82% of the cases), the loan applications are denied by the bank.

We then created a model in Disco using only the selected variants (Figure 2). The manual exercise of connecting the activities of models A, W, and O, together with the analysis of Figure 2, helped us to understand the process.



(a) “Application” (A) event type

(b) “Offer” (O) event type

(c) “Workflow” (W) event type

Fig. 1: Discovered process models of each type of event: “Application” (A), “Offer” (O), “Workflow” (W), considering all activities and only the most representative connections (100% activities and 0% paths in Disco).

Table 1: Process variants considered for understanding the AS-IS process. Each variant should be read from top (first activity) to bottom (last activity).

Variant 1 (12.28%)	**Variant 2** (6.71%)	**Variant 3** (5.69%)	**Variant 9** (1.82%)
A_Create Appl.	A_Create Appl.	A_Create Appl.	A_Create Appl.
A_Submitted	A_Submitted	A_Submitted	A_Submitted
A_Concept	A_Concept	A_Concept	A_Concept
W_Complete appl.	W_Complete appl.	W_Complete appl.	W_Complete appl.
A_Accepted	A_Accepted	A_Accepted	A_Accepted
O_Create Offer	O_Create Offer	O_Create Offer	O_Create Offer
O_Created	O_Created	O_Created	O_Created
O_Sent (m&o)	O_Sent (m&o)	O_Sent (m&o)	O_Sent (m&o)
W_Call after offers	W_Call after offers	W_Call after offers	W_Call after offers
A_Complete	A_Complete	A_Complete	A_Complete
A_Cancelled	W_Validate appl.	W_Validate appl.	W_Validate appl.
O_Cancelled	A_Validating	A_Validating	A_Validating
	O_Returned	O_Returned	O_Returned
	W_Call incomp. files	O_Accepted	A_Denied
	A_Incomplete	A_Pending	O_Refused
	W_Validate appl.		
	A_Validating		
	O_Accepted		
	A_Pending		

Analyzing the process in Figure 2 we observe that the time elapsed between some activities was so short that it would make sense to group them together to simplify the process. We therefore grouped together activities which took place less than 2 minutes apart, and which were usually performed by the same user (Table 2).

Table 2: Activities included in each *clustered* activity.

Clustered activity	**Activities of the cluster**
A_Create Application	{A_Create Application; A_Submitted; A_Concept}
W_Complete application	{W_Complete application; A_Accepted; O_Create offer; O_Created; O_Sent; W_Call after offers; A_Complete}
W_Call incomplete files	{W_Call incomplete files; A_Incomplete}
W_Validate application	{W_Validate application; A_Validating; O_Returned}
A_Denied	{A_Denied; O_Refused}
A_Cancelled	{A_Cancelled; O_Cancelled}
A_Pending	{O_Accepted; A_Pending}

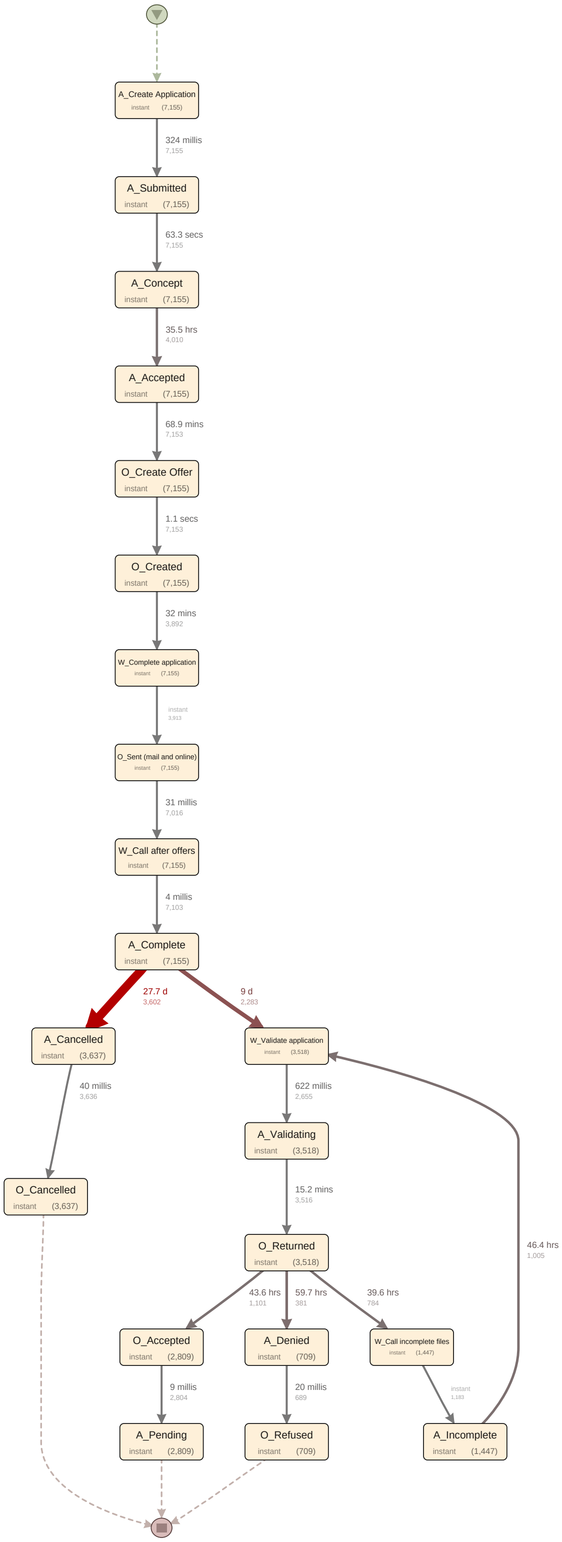


Fig. 2: Process model considering “Variant 1”, “Variant 2”, “Variant 3”, and “Variant 9” (100% activities and 0% paths in Disco).

We then created a model in Disco using the clustered activities, shown in Figure 3, which also helped us in creating the AS-IS process.

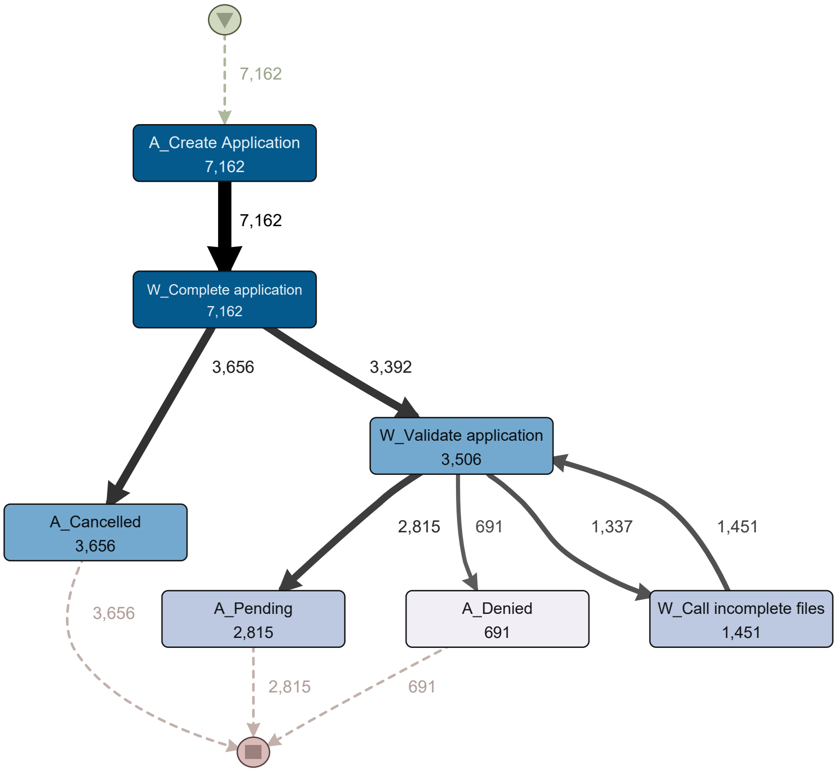


Fig. 3: Process model discovered using the clustered activities (100% activities and 0% paths in Disco).

Based on our understanding of the process, we created a first version of the model using a Petri net (Figure 4). To test how well our model represented the actual process (*as per* the event log), we executed the plug-in "Replay a Log on Petri Net for Conformance Analysis" in ProM [5] to calculate the fitness, and the plug-in "Check Precision based on Align-ETConformance" to calculate the balanced precision. For both plug-ins, we used the default parameters. We obtained a fitness of 95.72% and a precision of 100%.

In order to find a model that better reflected "reality", with high precision and fitness, we analyzed the deviations presented by the "Replay a Log on Petri Net for Conformance Analysis" plug-in and manually updated the process model. We added paths and activities to our Petri net model, until we arrived at a final model, which considers most activities with high precision and fitness values.

We also took into account the model simplicity. Table 3 presents the fitness and precision for each Petri net model we developed.

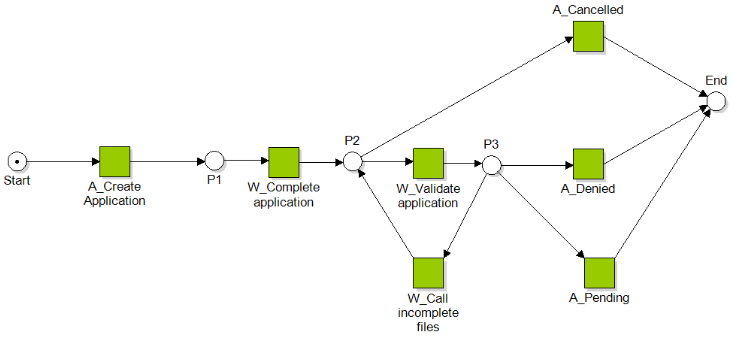


Fig. 4: Petri net of the first process model discovered.

Table 3: Fitness and precision of each Petri net.

Petri net	Fitness	Precision
1	95.72%	100.00%
2	96.26%	99.83%
3	96.93%	99.14%
4	98.76%	95.40%
5	98.89%	88.90%
6	86.50%	95.25%
7	92.57%	97.26%
8	97.56%	96.47%
9	99.17%	93.61%
10	99.16%	91.99%
11	98.20%	93.95%
12	96.59%	98.03%
13	94.98%	99.25%
14	95.85%	98.92%
15	97.25%	98.31%
16	97.25%	98.19%
17	97.65%	96.87%
18	97.68%	97.36%
19	97.28%	98.72%

The Petri nets in Table 3 were generated as follows. From Petri net 1 to 4 we aimed to improve the fitness of our first model by adding new routes. From Petri net 5 to 11 we added more activities (O_Create Offer; A_Complete; W_Handle leads; and A_Concept), maintaining high fitness and precision; and from Petri nets 12 to 19 we aimed to improve the fitness of the last model (model 11), by adding new routes in order to reduce deviations, thus increasing the precision and keeping the simplicity.

Figure 5 shows the discovered AS-IS model after all those improvements.

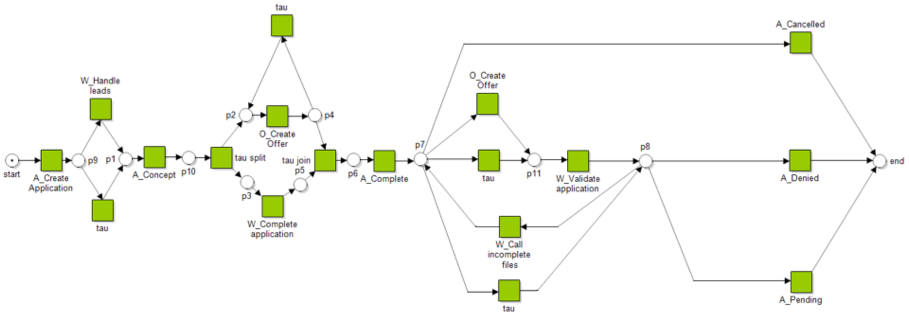


Fig. 5: Final Petri net of the process model discovered.

4 Descriptive Analytics

This section presents our first insights about what happened in the loan application process as recorded in its event log during the period of observation. To do so, we analyzed some relevant variables.

4.1 Application Frequency Analysis

We compared how many applications were created at each day of the week in the total period of the database. Figure 6 shows the distribution of applications created by weekday and month on the study period. We could note in this figure that the volume of new applications is higher in the beginning of the week, as well as in the period corresponding to the summer holidays.

4.2 Process end analysis

For each application the bank may create several offers for the customer to choose from. If the customer selects an offer and the bank validates his documents, the case ends in a state called "Pending" and is considered a success. There are many cases in which the customer does not answer the bank offers, and which

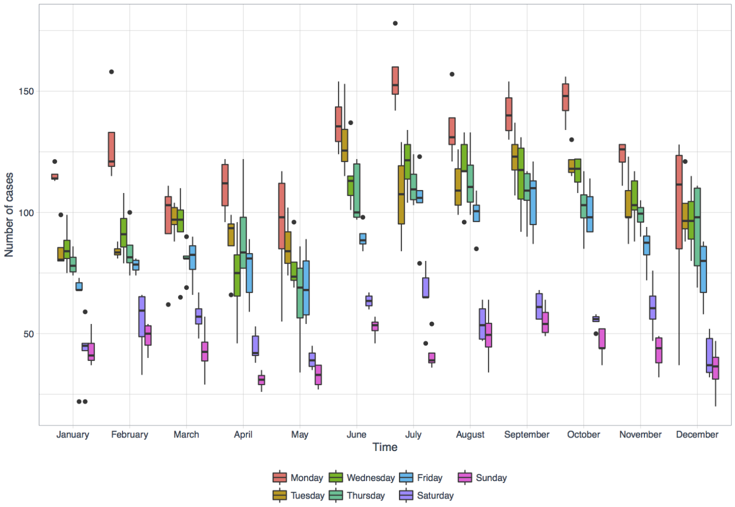


Fig. 6: Distribution of application by weekday on the study period.

end in a state called "Cancelled". Finally there are many other cases in which the bank does not approve the case after the customer sends his/her documents, which end in a "Denied" state.

We have observed that 55% of the applications made are "Pending". The "Cancelled" states appear in 33% of the cases. Perhaps it would be interesting to make some experimental changes (*e.g.*, lowering the monthly cost and increasing the standard 120 number of terms to lower the rate of cancelled cases). The "Denied" state occurs in 12% of the cases. We cannot assess whether it is high or low, since we do not have any default information or history to verify.

4.3 Application Type Analysis

Customers can inform in their application whether they want a new credit (89% of the cases) or a limit raise in an existing application (11% of the cases). From the total of 28,022 new credit applications, 14,742 (52.6%) were successful and 9,741 (34.8%) were cancelled. And from the total of 3,387 limit raise applications, 2,485 (73.4%) were successful and only a few applications were either cancelled or denied. Generally, the bank approves a higher fraction of limit raises than new credit applications. This is expected, since the bank had already approved a new credit for that customer earlier. Figure 7 summarizes this analysis.

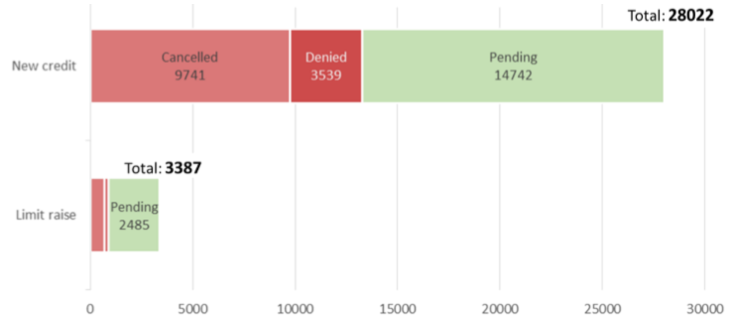


Fig. 7: Application type analysis.

4.4 Loan Goal Analysis

The application also has a field for customers to inform their goals with the loan. Most loan applications aim to invest in car, home improvement, and existing loan takeovers from which 50% to 60% were successful, obtaining final status “Pending”. Another 6,396 applications were not specified, but also had a success rate of approximately 55%. It is worth noticing that the percentages of each final status are similar, regardless of the loan goal. Figure 8 summarizes this analysis.

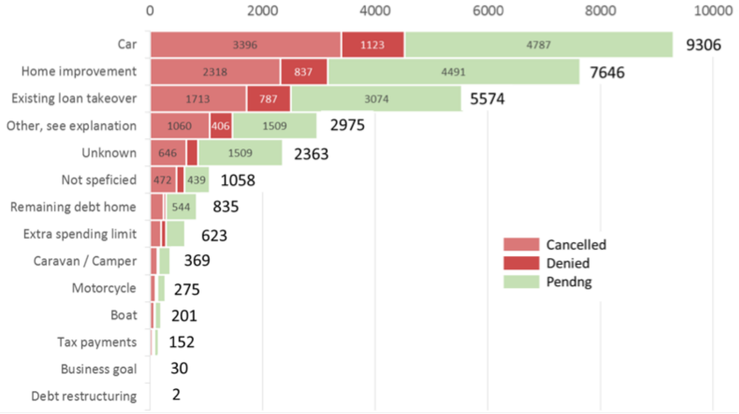


Fig. 8: Loan goal analysis

4.5 Requested Amount Analysis

This bank usually works with a range of loans between thousands and tens of thousands monetary units. Most loan applications are in the range of 10k to 100k

(20,997, or 66.9%), with an acceptance rate of 57%. The second most frequent group correspond to applications in the range of 1k to 10k, in which there are 8,299 (26.4%), with an acceptance rate of 45%. There are no requests for above 450k.

An interesting fact that we observed is that there are many applications in which the client does not fill the requested amount field, and yet those applications are pending at the end of the process.

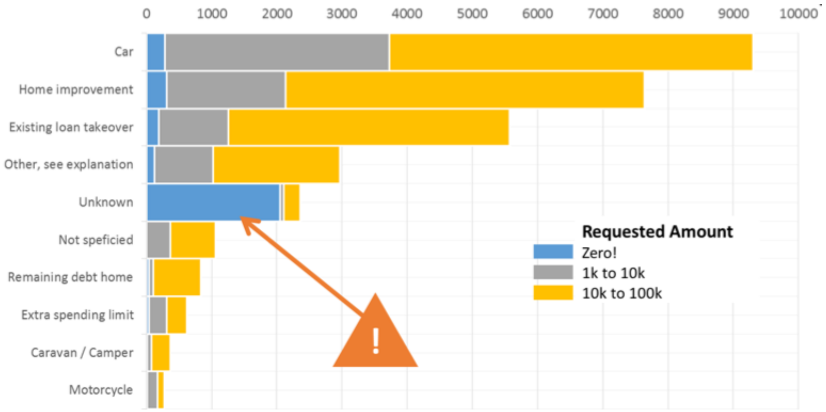


Fig. 9: Application count by Requested Amount and Loan Goal.

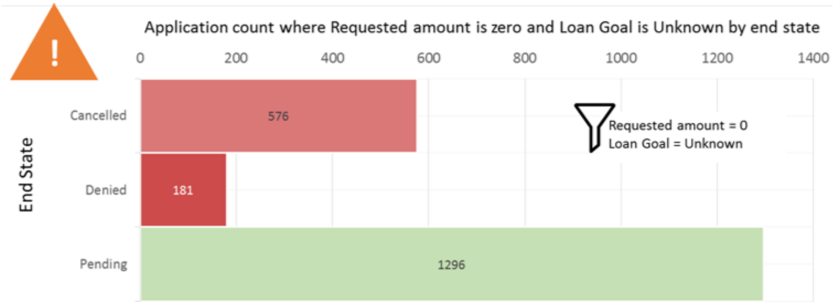


Fig. 10: Application count where Requested Amount is zero and Loan Goal is unknown by end state.

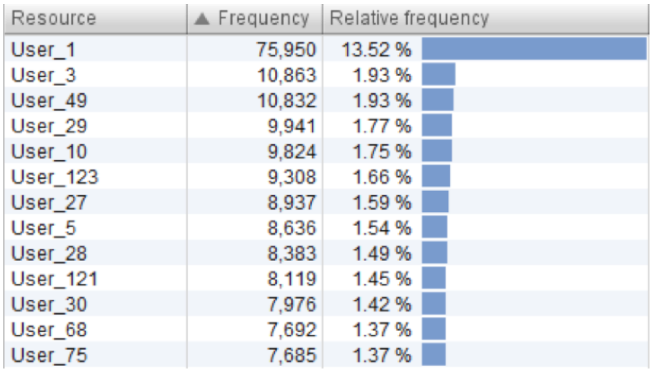
Figure 9 shows the number of applications by loan goal, but this time grouped by order of magnitude of requested amount. Again, we observe that there are many loan goals with requested amount equal to zero. There is a particular concentration of those cases with "Unknown" loan goal.

This fact calls our attention. We could not find any mention of it in the ProM forum, and it is hard to understand why this occurs and what it means. Most disturbingly, out of these applications, 1,296 were successful (63%) and only 37% were cancelled or denied (see Figure 10).

4.6 Resource analysis

In the event log, we find 145 resources, whose names are: User_1,. . . , User_145. There are several parameters to be analyzed for each resource, such as event frequency (the number of times they are responsible for an event), case frequency (on how many cases they appear), the time spent on an event, and so forth.

Our first efforts targeted identifying the automated and the human resources. An initial analysis of the process and the information given leads to the conclusion that User_1 is the only automated resource in the log. We arrived at this conclusion based primarily on 2 factors: First, the frequency of events worked on by User_1 dwarfs its peers and stands alone on his own while the others stand close together (see Figure 11); Second, User_1 is the only resource which works on the A_Submitted event (see Figure 12), which is an automatic event generated when a client submits his application in the site.



Resource	▲ Frequency	Relative frequency
User_1	75,950	13.52 %
User_3	10,863	1.93 %
User_49	10,832	1.93 %
User_29	9,941	1.77 %
User_10	9,824	1.75 %
User_123	9,308	1.66 %
User_27	8,937	1.59 %
User_5	8,636	1.54 %
User_28	8,383	1.49 %
User_121	8,119	1.45 %
User_30	7,976	1.42 %
User_68	7,692	1.37 %
User_75	7,685	1.37 %

Fig. 11: Frequency of events worked by the 13 resources who worked the most.

Figure 12 shows a macrograph of which resource is responsible for which event. It focuses on the events and does not provide a fine-grained view, but it gives a good overview of a resource's activity. It helps identify areas of interest in the process, like the aforementioned fact that only User_1 is linked to A_Submitted; events associated to few resources, such as W_Assess potential fraud and W_Shortened completion; and other useful information.

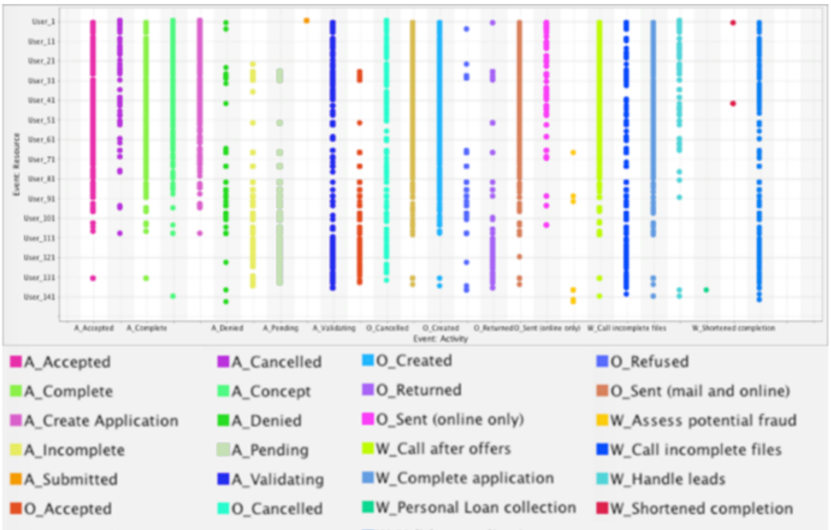


Fig. 12: Graph relating resources to their corresponding events.

A brief analysis shows that resources 128, 124, 106, 8, 108, 92, 50, 138, 6, 144, 136, 64, 89, 143, 135, 140, 110, 103, 139, 141, 111, 82, 142 and 145 are responsible for 1% of events and work on 4% of the cases. Those resources may be considered as adding low value to the process, so their work could be passed on to another resource or concentrated in only one of them.[1] The only exception is resource 144, but his/her role will be discussed later.

It is possible to see several instances of events in which a handful of resources are responsible for most of the frequency, such as W_Call incomplete files and W_Validate application. This distribution may be an example of bad planning, as it seems to show that the resources are not clear on what events they should or should not work in most cases. A better division of which events are specialized and should be performed only by certain resources, and which events can be performed by anyone, should clarify each resource's tasks and optimize their performance.

Looking at the rows of Figure 12, it is possible to see patterns between the resources who fill two major roles. The first role we called Initiators, as they are responsible for the creation and establishment of both applications and offers. The second role we called Solvers, as they collect missing information, validate, and finish the application. Each role is apparent in each resource, but there some cross overs that need further investigation and perhaps elimination.

[1] Resources 6, 89, 92, 108, and 110 seem to have left the company (or the loan applications process) by the second month of the period recorded in the event log, so one may choose to exclude them from this analysis. Likewise, resources, 106, 135, and 140 appeared in the log in Oct/Nov 2016 for the first time.

The first role, Initiators, is where most of the resources are assigned. This has some logic to it, since the sum of their events is larger than the sum of Solvers' events. However, the amount of time dedicated to the events assigned to the Solvers is much larger. When we observe the bottlenecks and the longest events, they all fall under the Solvers purview, to be discussed next.

Resources 27, 29, 30, 68, 75, 83, 87, 90, 95, 99, 100, 109, 112, 113, 116, 117, 118, 119, 120, 121, 122, 123, 125, 126, 131, 133 and 134 are Solvers. They are all responsible for most of the work invested on W_Validate application, and there are several patterns shared by them, which can further divide them in two groups. The first group consists of 29, 30, 68, 75, 83, 87, 95, 99, 100 and 109, and they dedicate most of their time to A_Incomplete, A_Pending, and W_Call incomplete files (excluding W_Validate application). The second group consists of 27, 112, 113, 116, 117, 118, 119, 120, 121, 122, 123, 125, 126, 131, 133 and 134, and they dedicate most of their time to O_Returned and A_Validating (excluding W_Validate application).

The difference between those groups leads us to believe that there is some unofficial division of tasks based on the current state of cases. We suppose that the second group is composed of some type of senior staff and/or managers, who receive a case from the first group when a problem occurs. They take over the case, solve the problem, and then give it back to the first group to finalize it. This handing back of cases to the first group is indicated by the fact that the second group has barely any instance of A_Denied and O_Refused, while the first group has the majority of them.

Finally, resources 138, 143 and 144 deserve special analysis. If observed in the whole process, their presence is minimal, working on 343, 177 and 245 events each, respectively. A deeper analysis of those events shows that they work mostly on the W_Assess potential fraud event, and most (85.63%) of the W_Assess potential fraud events are handled by them. This leads us to believe that they are managers or special operators who work specifically on fraud detection. As it seems this activity requires specialized knowledge, the other resources who only eventually work on fraud detection should be assigned to other events.

A complete descriptive analysis can be found at http://www.ideias.inf.puc-rio.br/ProcessMining/BPI2017.

5 Process analysis

In order to better understand the process, we broke our reconstructed model down into smaller parts. We grouped the events with high correlation and modeled a process using BPMN to demonstrate these variants in terms of decisions and loops (Figure 13). BPMN allowed us to express findings and some basic understanding directly into the process.

Through this analysis, we perceived three main blocks that define a process, namely: receiving applications, negotiating offers, and validating documents.

Receiving applications. The bank tries to receive an application and assure that this application has the minimum information necessary to proceed.

Applications are created by the customer on the website or by the bank in the presence of the client. When an application is received through the website, it passes through an automatic validation procedure and may or may not require a call to the customer, probably to require additional information and rule out any doubts.

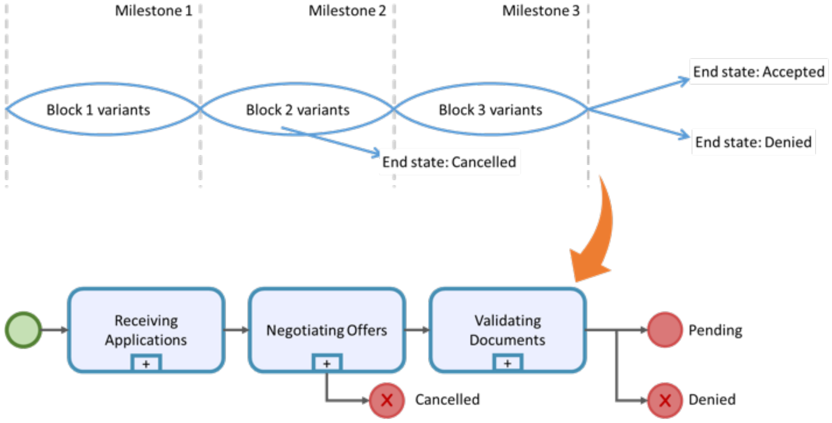


Fig. 13: BPMN model of the process in terms of decisions and loops.

Negotiating offers. From the moment the application is validated, the bank designs *offers*, which are proposals for the client to evaluate. Offers vary according to the payment conditions, namely: offered amount (not necessarily the same as requested amount), monthly cost, first withdrawal amount, and number of terms. This is a way for the bank to offer flexibility and for the customer to choose the most adequate cash flow to meet his/her financial situation and needs.

Validating documents. This step is usually unnecessary, but sometimes there is a missing document or inconsistent information in the customer's first answer, so the bank is forced to call and validate the documents again.

The next subsections detail these three blocks.

5.1 First milestone: Receiving application

We quickly explore differences between applications received online and directly at the bank. Surprisingly, it is possible to verify that, the closer the contact with the customer, the better are the chances of a successful (Pending) application. Figure 14 shows the BPMN model for this first milestone in the process.

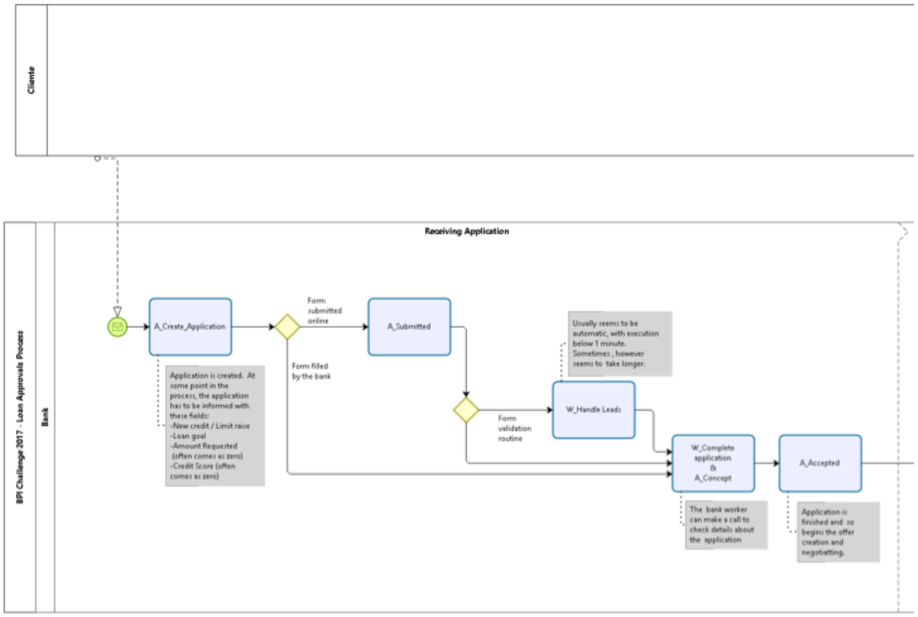

Cliente
BPI Challenge 2017 - Loan Approvals Process
Bank
Receiving Application
A_Create_Application
Form submitted online
A_Submitted
Form filled by the bank
Application is created. At some point in the process, the application has to be informed with these fields:
-New credit / Limit raise
-Loan goal
-Amount Requested (often comes as zero)
-Credit Score (often comes as zero)
Usually seems to be automatic, with execution below 1 minute. Sometimes , however seems to take longer.
W_Handle Leads
Form validation routine
W_Complete application & A_Concept
A_Accepted
The bank worker can make a call to check details about the application
Application is finished and so begins the offer creation and negotiatting.

Fig. 14: First milestone of the process: Receiving Application.

Approximately 2/3 of the applications are received via an online form. When the bank creates the application, there is a great rate of applications that end up *Pending*. After the application is created, it passes through the A_Submited state if it was submitted online, usually by the client itself. As we can see, the client sends most of the applications (20,338 applications, or 64.8%), with a success rate of 50%. The bank submitted 11,071 (35.2%) applications, of which 65% were successful.

Surprisingly, the online applications which passed through W_Handle Leads have a lower percentage of success compared to the others. We raise two hypotheses here for further investigation: either the bank chooses well where to invest human hours to handle leads that may harm the bank by defaulting, or there is something related to this call that is strongly correlated with a refusal by the end of the process.

If an application is submitted on the website, the first workitem created is *handle leads*. The application is then assessed for the first time, automatically. If the assessment cannot be completed because of technical problems, the workitem remains in *handle leads* and a new assessment can be done manually. We note that 52% of applications that did not pass through the assessment of *handle leads* were accepted, while only 37% of applications that passed through that assessment were accepted (see Figure 15).

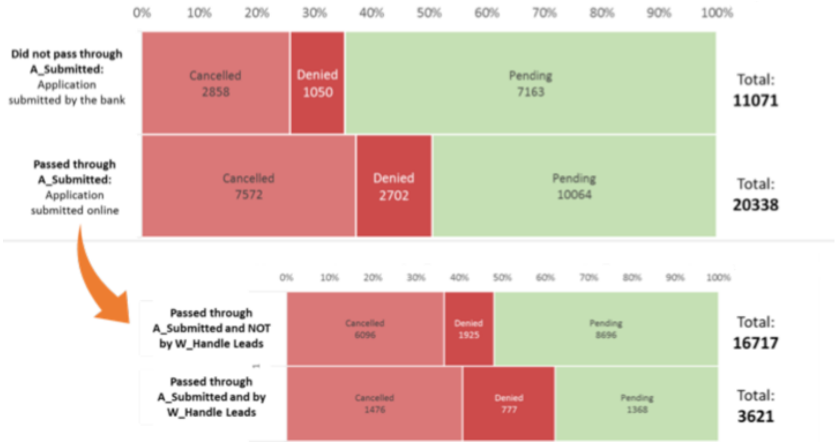


Fig. 15: Applications that passed or not by A_Submitted against applications that passed trough A_Submitted and passed or not by W_Handle leads, by end state.

5.2 Second milestone: Negotiating application

We now analyze the influence of an offer on the application outcome and on subsequent offers. We discuss the customer's decision and how the negotiation parameters, such as the number of offers sent, influence the final outcome of the process. Figure 16 shows the BPMN model for this milestone in the process.

An application can have more than one offer, but only one offer can be accepted (both by the client and by the bank) in each application. As we can see in Figure 17, the vast majority of applications receive only one offer. And a significant number of applications have between 2 and 3 offers. The horizontal axis is in logarithmic scale in order to properly show that there is a near exponential decay on the number of offers proposed by the bank.

The credit score is a numerical value that represents the creditworthiness of the client. Despite the large number of offers with zero credit scores, we understand that this item, despite being used for credit consultation, is not filled in for all applications, as one can observe in Figure 18.

The loan amount can be paid back in a certain number of months (called *terms*). As Figure 19 shows, most offers are created for a payback period of 1.5 to 10 years, with peaks in 10 and 5 years (120 and 60 months, respectively).

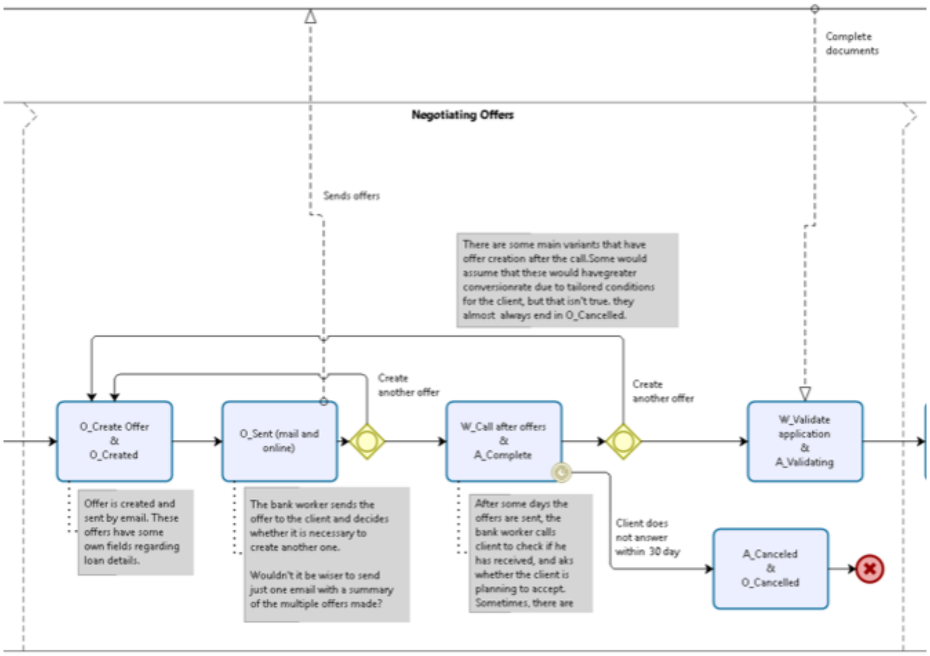

Complete documents
Negotiating Offers
Sends offers
There are some main variants that have offer creation after the call.Some would assume that these would havegreater conversionrate due to tailored conditions for the client, but that isn't true. they almost always end in O_Cancelled.
Create another offer
Create another offer
O_Create Offer & O_Created
O_Sent (mail and online)
W_Call after offers & A_Complete
W_Validate application & A_Validating
Offer is created and sent by email. These offers have some own fields regarding loan details.
The bank worker sends the offer to the client and decides whether it is necessary to create another one.
Wouldn't it be wiser to send just one email with a summary of the multiple offers made?
After some days the offers are sent, the bank worker calls client to check if he has received, and aks whether the client is planning to accept. Sometimes, there are
Client does not answer within 30 day
A_Canceled & O_Cancelled

Fig. 16: Second milestone of the process: Negotiating application.

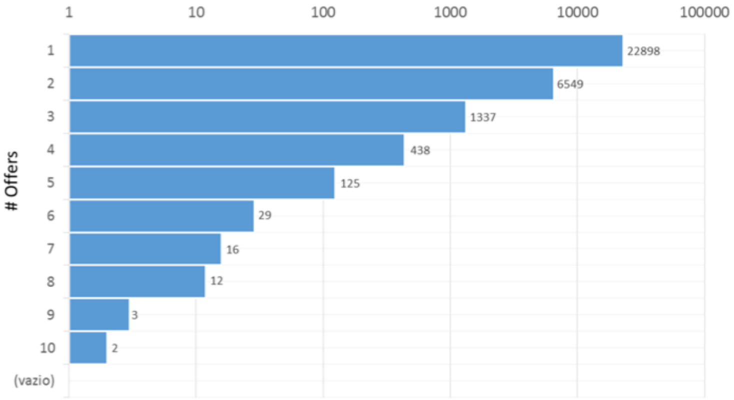


Fig. 17: Applications (log scale) by Number of offers.

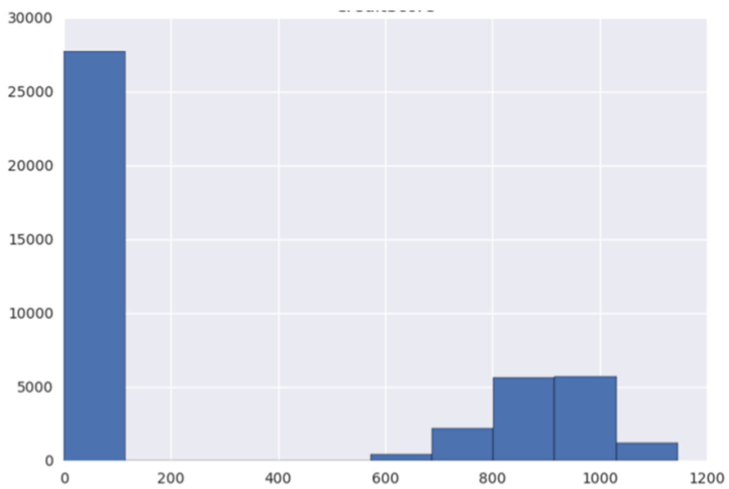


Fig. 18: Histogram of credit scores (of offers).

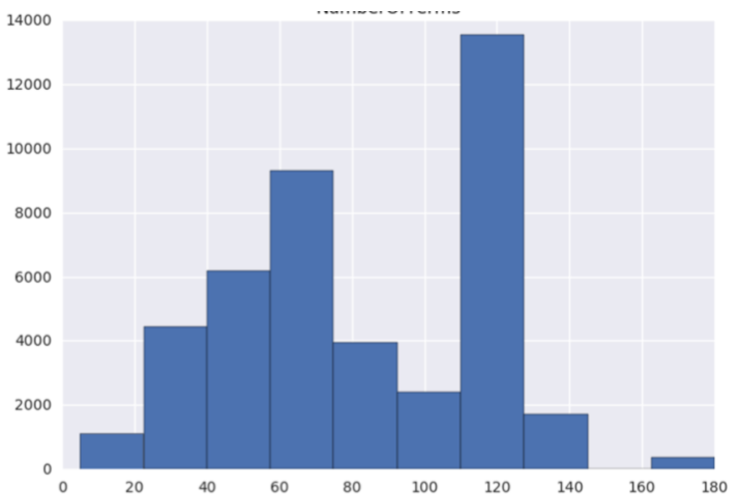


Fig. 19: Histogram of number of terms (of offers).

An application with only one offer is less likely to be successful. As the number of offers increases, so do the chances of success (see Figure 20). It seems therefore useful to make multiple offers. This raises a question about when to make those offers. As Figure 21 shows, only about 12% of all the successful (*Pending*) applications were from offers made 5+ days after the creation of the first offer, whereas the great majority of successful applications came from offers created closer to the first one. Therefore, the negotiation and better understanding of how to design an offer better suited for the client is not negligible.

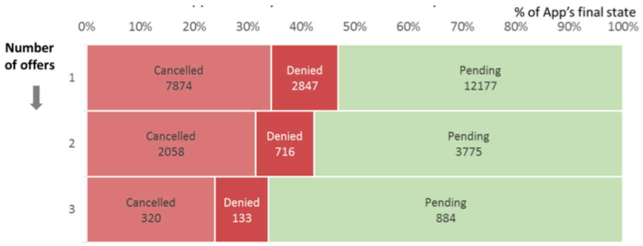


Fig. 20: Applications by number of offers and end state.

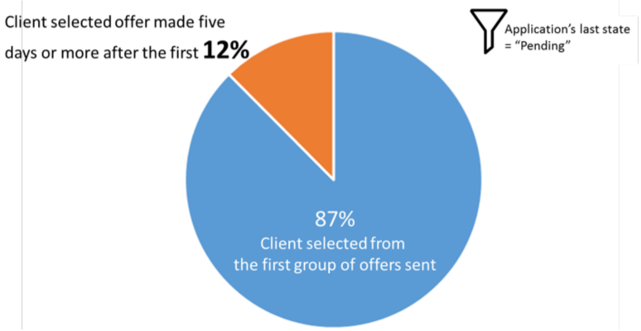


Fig. 21: Pending applications which the selected offer was created 5 days after the first.

5.3 Third milestone: Validating documents

We now briefly explore the last block of the process (see Figure 22). If a customer's documents are incomplete, the bank calls the customer, asks for additional documents, and validates them. The bank then decides whether additional documents are necessary. If not, the offer moves on to a final decision, either Pending or Denied.

Some questions posed in the challenge related to this milestone were: How does the number of calls influence the final outcome? Is there a chance to jeopardize the deal if there are many calls?

Figure 23 depicts the proportions of end states of the applications that passed through at least one validation activity. It shows that the proportion of acceptances does not alter significantly when there are multiple calls asking for incomplete files. This shows that calling the customers does not pose a major problem. What is interesting is that the calls change the proportions between Denied and Cancelled.

Our main hypothesis to explain this is that in one of these calls the customer may perceive he/she does not have the appropriate profile and then no longer responds, thus causing the application to be canceled.

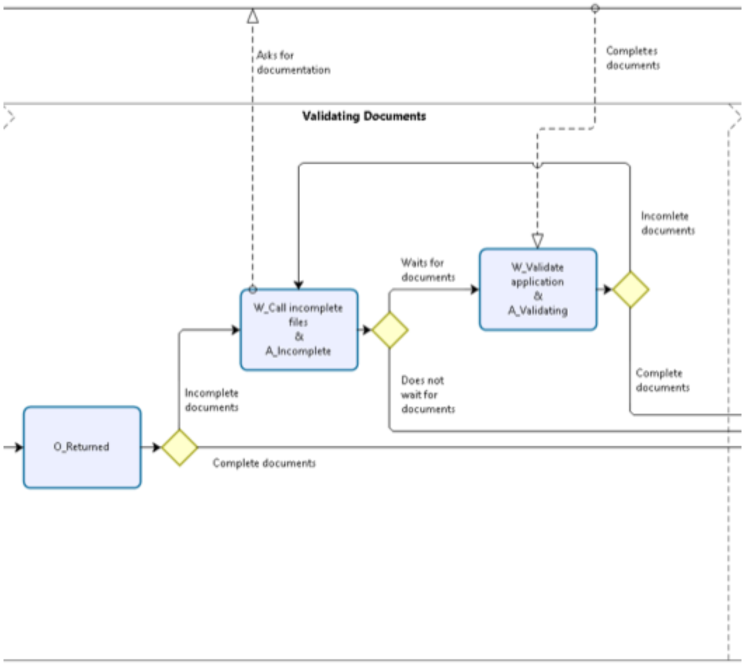


Fig. 22: Third milestone of the process: Validating documents.

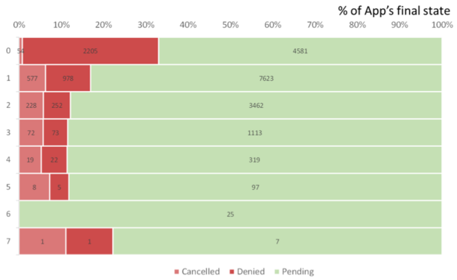


Fig. 23: Application that passed through first validation by number of "Incomplete File Calls", by end state.

6 Diagnostic Analytics

This section presents our insights to answer the three BPI Challenge questions. To do so, we first conducted a performance analysis.

6.1 Performance analysis

In order to do the performance analysis, we incremented the AS-IS model. The main updates in the model were:

1. We removed the first loop of "O_Create offer". We quadruplicated the activity and put them in sequence. This provided us the time and frequency between the offers creation;
2. We moved the "W_Complete application" from the parallel branch since in most cases this activity happened after an "O_Create offer";
3. We broke the "W_Validate application"–"W_Call incomplete files" loop in two levels, to provide us the performance and frequency for each validation step in case the application has more than one validation;
4. We repeated the "A_Cancelled" activity five times to capture from which exact part of the process the applicant decided to cancel the application. We also duplicated all other outcome activities of the process (A_Denied and A_Pending) and "O_Create offer" activities after "A_Complete";

Figure 24 presents the incremental model.

To present the performance results, we added a number to the name of each duplicated activity to indicate their relative order. For example, O_Create Offer1 and O_Create Offer2. In this example, we clearly know that O_Create Offer1 is located before O_Create Offer2. With the new model, we ran the plug-in "'Multi-perspective Process Explorer" with the default parameters to collect the performance and frequency metrics. To obtain global statistics of the process, we used Disco to filter the data according to the outcome.

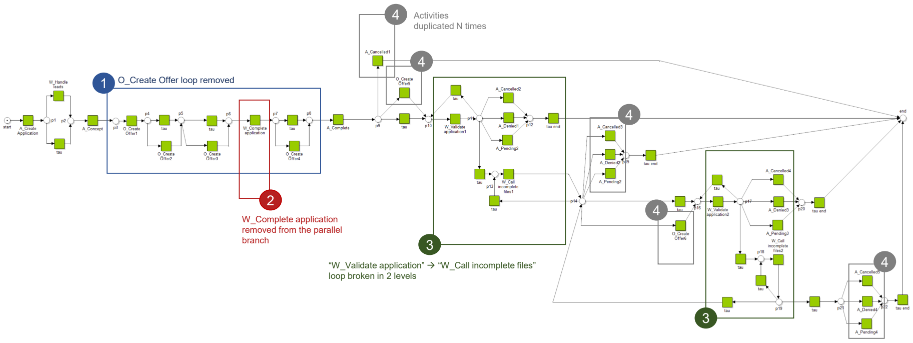


Fig. 24: AS-IS model used for performance checking analyses. In this model, we added in the name of some activities a number to indicate their relative order, to help the reader understand the results. When applying the process mining techniques, we removed these numbers.

For this study, we employed a combination of three process mining tools: Disco 1.9.9 [3], ProM 6.6 [5], and Yasper 2.1 [6]. We used Disco to generate event logs filtered by attributes and to create models; Yasper to create all Petri Net models; and ProM to perform conformance checking (plug-in ”Replay a Log on Petri Net for Conformance Analysis”), to calculate the balanced precision (plug-in “Check Precision based on Align-ETConformance”) and for performance checking (plug-in “Multi-perspective Process Explorer”).

The mean case duration from the event log was 21.8 days and the median was 19.1 days. The mean case duration according the outcome was:

1. Case approved (A_Pending): 18.1 days (median 14.8 days) - 17,227 cases
2. Case denied (A_Denied): 16.7 days (median 14.1 days) - 3,752 cases
3. Case cancelled (A_Cancelled): 29.9 days (median 31.6 days) - 10,430 cases

Table 5 presents the performance and frequency metrics of the process. Column A presents the name of the source activity. Column B presents the name of the target activity. Column C presents the waiting time between the source activity (A) and the target activity (B). Column D indicates whether the start of the target activity depends on an input from the applicant (*e.g.*, to send documentation to the bank). Column E indicates whether the start of the target activity does not depend on an input from the applicant, which means that the waiting time is associated exclusively to an internal process. Column F presents the frequency of transitions from the source activity to the target activity. Finally, column G indicates the percentage of cases that executed the path from the source activity to the target activity. We filled columns D and E based on the following assumptions:

1. An application will be canceled if the applicant does not demonstrate to the bank an interest in any offer. In this case, we believe that the bank is waiting

for a reply from the applicant. If the applicant does not provide an answer to the bank after 26 days, the application is automatically canceled;

2. Sometimes the applicant calls the bank to negotiate the application (*e.g.*, to ask for a reduction in the interest rate). When this happens, the bank creates a new offer. In this case, we believe that offers created after the application is complete (A_Complete) depend on the applicant. The first offers, which are created sequentially before the execution of an A_Complete, usually are created without such a dependency;
3. The validation of the application will happen only after the bank receives from the applicant the requested documents, *i.e.*, it depends on the applicant;
4. If the application has a direct outcome decision (Cancelled, Denied, or Approved) after a "W_Call incomplete files" activity, we believe that it depends on an input from the applicant (*e.g.*, sending additional documents).

Table 4 presents the number of offers per case.

Table 4: Number of offers per case.

number of offers	number of cases
1	22898
2	6549
3	1337
4	438
5	125
6	29
7	16
8	12
9	3
10	2

With respect to the number of offers per application and outcome, we observed that, as more offers are created in an application, the greater are the chances that the applicant continue the process and not cancel the application. With this analysis, we cannot conclude that the bank should create more offers, since the applicant him/herself may ask for more offers, meaning that he/she has a great interest (or need) to get the financing.

We noted that the waiting times to start the "W_Validate application1" activity (9.8 and 8.7 days) were higher than the times to start the "W_Validate application2" activity (4.7 and 2.5 days), indicating that the application probably has fewer pending documents and/or the applicant is really interested in obtaining the loan quickly.

Table 5: Performance analysis of the transition of two activities: (A) Source Activity; (B) Target Activity; (C) Mean time of the activity in days; (D) Question 1: Does the activity depend on applicant input to start?; (E) Question 2: Does the activity depend on internal action?; (F) Activity Frequency; (G) % of cases

Source	**Target**	**Mean time**	**Q1?**	**Q2?**	**Freq.**	**% of Cases**
(A)	**(B)**	**(C)**	**(D)**	**(E)**	**(F)**	**(G)**
A_Complete	A_Cancelled1	26.86	yes	no	8,322	26.5
A_Complete	O_Create Offer5	6.96	yes	no	4,355	13.9
A_Complete	W_Validate app1	9.80	yes	no	18,732	59.6
A_Concept	O_Create Offer1	1.08	no	yes	31,409	100.0
A_Create_app	A_Concept	<0.01	no	yes	27,791	88.5
A_Create_app	W_Handle_leads	0.26	no	yes	3,618	11.5
O_Create Offer1	O_Create Offer2	0.06	yes	yes	2,840	9.0
O_Create Offer1	W_Complete app	0.02	no	yes	28,569	91
O_Create Offer2	O_Create Offer3	0.20	yes	yes	330	1.1
O_Create Offer2	W_Complete app	0.01	no	yes	2510	8.0
O_Create Offer4	A_Complete	0.02	no	yes	7,990	25.4
O_Create Offer5	W_Validate app1	8.70	yes	no	4,355	13.9
O_Create Offer6	W_Validate app2	4.70	yes	no	1,610	5.1
W_Call inc. files1	A_Cancelled3	2.44	no	yes	632	2.0
W_Call inc. files1	A_Denied2	1.85	yes	no	152	0.5
W_Call inc. files1	A_Pending2	5.84	yes	no	2,576	8.2
W_Call inc. files1	O_Create Offer6	3.07	yes	no	1,610	5.1
W_Call inc. files1	W_Validate app2	2.5	yes	no	14,676	46.7
W_Complete app	A_Complete	0	no	yes	23,419	74.6
W_Complete app	O_Create Offer4	1.76	yes	no	7990	25.4
W_Handle_leads	A_Concept	<0.01	no	yes	3,618	11.5
W_Validate app	A_Cancelled2	25.83	yes	no	1,173	3.7
W_Validate app	A_Denied1	2.11	no	yes	2,398	7.6
W_Validate app	A_Pending1	1.54	no	yes	4,581	14.6
W_Validate app	W_Call inc files1	2.21	no	yes	14,935	47.6
W_Validate app	A_Cancelled4	21.31	yes	no	82	0.03
W_Validate app	A_Denied3	1.45	no	yes	1,135	3.6
W_Validate app	A_Pending3	0.68	no	yes	7,749	24.7
W_Validate app	W_Call inc files2	1.44	no	yes	7,320	23.3
W_Validate app	W_Validate app3	1.46	no	yes	1,615	5.1

6.2 Answering the BPI questions

Regarding the first question of the BPI Challenge, "What are the throughput times per part of the process, in particular the difference between the time spent in the company's systems waiting for processing by a user and the time spent waiting on input from the applicant as this is currently unclear?", we provided a detailed description of the waiting times in Table 5. We concluded (based on our previously detailed assumptions) that the activities which depend on the applicants are:

1. W_Validate application: the bank waits for the applicant to send the requested documents before starting the validation step. Among the activities of the applicants who accepted the offer, this is the activity that has a longer waiting time, with an average of 9.8 days for the 59.6% cases that accepted the offer shortly after the first offers (A_Complete). This is possibly due to the applicant's delay in sending the acceptance and files;
2. O_Create offers normally after A_Complete: the applicant calls the bank to negotiate an offer (e.g. to reduce the interest rate);
3. A_Cancelled: the applicant can cancel the application by calling the bank to clearly inform them that he/she will no longer continue the process or, by not doing anything regarding the application (and, as a consequence, the application will be canceled automatically).

The other activities of the process wait for a bank employee. We can see in Table 5 that the activity W_Call incomplete files1 is the one with the longest waiting time, which is performed after the W_Validate application1. This waiting time happens to 47.6% of the cases and takes an average of 2.21 days.

With respect to the second question of the challenge, "What is the influence on the frequency of incompleteness to the final outcome? The hypothesis here is that if applicants are confronted with more requests for completion, they are more likely to not accept the final offer", we noted that the application cancellation rate decreased significantly as the process flows: 26.5% of cases canceled in A_Cancelled1, 3.7% of cases canceled in A_Cancelled2, 2% of cases canceled in A_Cancelled3 and 0.03% of cases canceled in A_Cancelled4. A_Cancelled3 and A_Cancelled4 occur after "W_Call incomplete files1" and A_Cancelled5 occurs after "W_Call incomplete files2". **We conclude that, the more an applicant interacts with the bank, the more likely it is for him/her to accept the offer. However, acceptance by the customer does not necessarily result in successful applications. In fact, as Figure 25 shows, the more calls made, the lower percentage of A_Pending end states.**

Customer calls can occur at three specific points in the process: W_Complete Application, where the bank worker can make a call to check details about the application; W_Call After Offers, where the bank worker calls, after the offer is sent, to check whether the customer has received it, and asks whether the client is planning to accept it; and W_Call Incomplete files, where the bank worker calls to ask for documents.

The activity that defines the phone call to the client has no duration attribute so we cannot identify the duration of these calls. However, we can calculate how many W_Call Incomplete files activities appear in each of the applications. Figure 25 shows that up to 7 calls per application were made and that most applications have at most one call. It also shows the proportion between those applications that were successful.

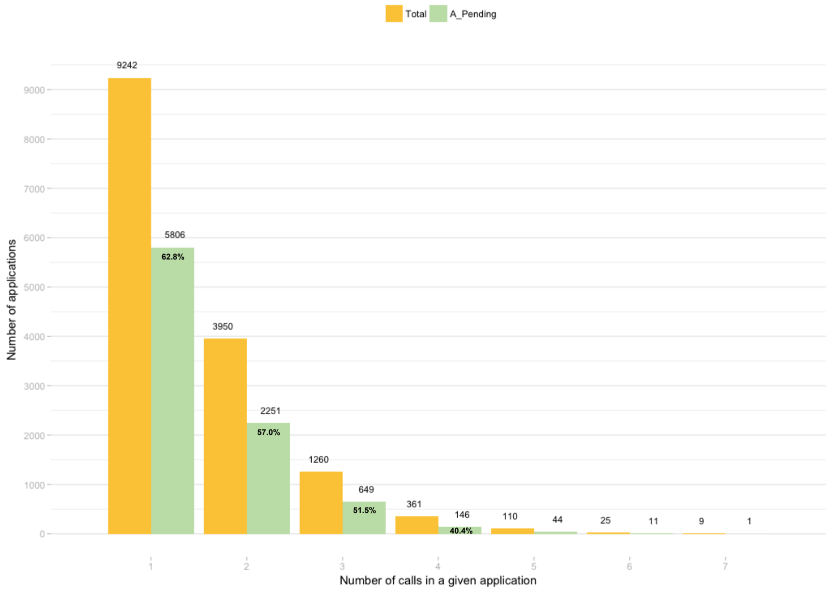


Fig. 25: Frequency of the number of W_Call Incomplete files per application.

Since the number of cases with incomplete documentation is high, we would like to recommend the bank to modify the loan input forms by making a checklist for the documents that are either mandatory or desirable.

Regarding the third BPI question, “How many customers ask for more than one offer (where it matters if these offers are asked for in a single conversation or in multiple conversations)? How does the conversion compare between applicants for whom a single offer is made and applicants for whom multiple offers are made?”, analyzing Table 4 we can conclude that 22,898 cases had only one offer and that 8,511 have more than 2 offers. We can notice that the more offers the bank creates to an applicant, the greater are the chances that the applicant continue the process and not cancel the application. As we said previously, with this analysis, we cannot conclude that the bank should create more offers since the applicant can, by him/herself, request additional more offers, meaning that he/she has a great interest (or need) to get the loan.

7 Further Recommendations

Application handling. We observed that those applications that performed the “W_Handle Leads” activity took longer to move from “A_Concept” to “A_Create Offer” (53.2 hours) than applications that did not perform the “W_Handle Leads” activity (20.9 hours). This suggests that applications which perform

"W_Handle Leads" can follow a different internal process than the default (most common) process, and that these applications may be somehow "forgotten" or undervalued by the bank employees. If our hypothesis is right, we strongly recommend the bank to update the "Handle Leads" activity to treat their applications in the same way as the default (normal) process.

Loan process deviation. We identified that 25% of cases only created an offer after "W_Complete application". This makes us believe that this is a deviation in the process. We suggest the bank identify the reasons for this deviant behavior. If this is an undesired deviation, we suggest the bank explore some actions to reduce its occurrence.

Consistency in the negotiation. We identified that the cancellation rate is very high (33% of cases) and, in most of cases, it is because the applicants did not give an answer to the bank. We suggest that the bank create a new internal process to call applicants that are taking longer to give a reply, propose them new offers, and encourage them to continue the process. This new approach would help the bank to understand the reasons for cancellation and would give the applicants a feeling of personal service. In general, offers created at the same time (within seconds) present different options for the applicant (number of terms, offered amount, and/or monthly cost). We found a few cases that, from the first to the second offer, the only difference was the monthly cost, and the second offer proposed a higher monthly cost than the first offer (*e.g.*, application 1526223693 with an increase of 5 cents; application 1549000634 with an increase of 11 euros). Every offer created is automatically sent by e-mail to the applicant. We recommend the bank to update the system or review the process to avoid increasing the monthly cost in a second offer since the applicant can get offended or angry about this type of seemingly inconsistent or ill intended behavior. Alternatively, the applicant can think that the bank does not perform a high-level (professional) work and, as consequence, he/she can cancel the application and start the process with another bank.

Effort to close the deal. We assumed that applicants tend to call the bank to negotiate offers, trying to reduce the interest rate. We found cases with this behavior and we observed that the difference in the monthly cost had a very small reduction (close to 1 Euro). In our opinion, reducing the monthly cost by only 1 Euro is not interesting to any applicant; moreover, it can be considered insulting and make the customer look for another bank. Thus, our suggestion is to ensure that subsequent offers are more interesting to applicants, even if it means increasing the number of terms to keep the monthly costs down.

Table 6: Some examples of cases with only a small reduction in the monthly cost. These cases present offers with the same number of terms and offered amount.

Application ID	**Monthly cost First Offer**	**Monthly cost Second Offer**	**Difference in Euros**	**Variation**
449790608	133.86	133.00	0.86	0.60%
1231200933	199.56	198.54	1.02	0.50%
1452087651	119.02	118.15	0.87	0.70%
479369766	143.91	141.45	2.46	1.70%
2110565193	505.58	504.20	1.38	0.30%

8 Conclusions

In this work we presented an analysis of a loan process event log, in order to give value to the data. We answered the three questions of the 2017 BPI Challenge by performing an extensive descriptive and diagnostic analysis. To do so, we first reconstructed the identified executed process (AS-IS process) with a fitness of 97.28% and a precision of 98.72%, while maintaining its simplicity. We used this model to perform a deep analysis of "what happened?" and "why did it happen?" on the process. Finally, we made some recommendations to improve the business process.

We plan to continue our work by making some further predictive and prescriptive analysis. By the use of a Play-Out system we intend to stress the model by scaling the cases input frequency and see the impact of this stress on the bank answering time. We are also working on a predictive model to identify at an earlier stage which applications deserve more attention from the bank.

References

1. Van Der Aalst, W.: Process mining. Communications of the ACM **55**(8) (2012) 76–83
2. Van Der Aalst, W.: Process mining: Overview and opportunities. ACM Transactions on Management Information Systems (TMIS) **3**(2) (2012) 7
3. Fluxicon: Disco 1.9.9. https://fluxicon.com/disco/ (2017)
4. Adriansyah, A., Buijs, J.C.: Mining process performance from event logs. In: International Conference on Business Process Management, Springer (2012) 217–218
5. ProM: 6.6. http://www.promtools.org/doku.php (2017)
6. Yasper: 2.1. http://www.yasper.org (2017)