Process mining in the Common Agricultural Policy (CAP) process

René Acuña, Juan Cumsille, Nicolás Godoy, Jaime González, Martín Navarrete, and Michael Arias

Computer Science Department, School of Engineering
Pontificia Universidad Católica de Chile, Santiago, Chile
[rtacuna,jfcumsille,nlgodoy,jgonzalez1,menavarrete,m.arias]@uc.cl

Abstract. In the context of the International Business Process Challenge (BPIC, 2018) which is centered this year on the European Agricultural Guarantee Fund process, this essay present the results of the study of several questions made by the authors of this document. The first inquiry was to determine the differences between the cases in which the Payment Application was approved in the first attempt, versus the cases with a rejected Payment Application document, showing a loop path for rejected cases and an analysis with suggestions to improve the main process. The second question is related to the differences that can be perceived between the longest cases and the shortest, and it was found a "trigger" that can be used to determine whether or not a particular case will last longer than usual. The third question was asked in order to determine if there are patterns in the processes that could help predict which cases will be reopened. After they are reopened, the research group also study whether they behave similarly to the rest of the cases, where they found that this does not happen in a lot of cases. In the fourth question were determined the differences between the processes of the rejected and the approved cases, where it was found that perhaps it is possible to determine sooner whether a case is going to be rejected or not.

Keywords: BPIC · Business Process Intelligence Challenge · Process Mining Contest.

1 Introduction

The Common Agricultural Policy (CAP) is an agricultural policy of the European Union that implements agricultural subsidies and other programs in order to support young and old farmers to improve agricultural productivity. Nowadays, this policy represents the 38% of the EU budget.

The purpose of this paper is to both state and answer four questions related to the 2018 Business Process Challenge, related to the CAP policy. The objective is to gather unique insights into the process captured by the event log.

This year's challenge is based on data provided by the German company Data Experts, located in Neubrandenburg. It was extracted from their Java Enterprise system profil c/s. This system supports different kinds of administrative processes in federal ministries of agriculture and local departments, but for this challenge only the yearly allocation of direct CAP payments are considered. The log consists of nine documents which have states that allow different actions. These actions can be executed either manually or automatically.

The usage of all tools, techniques and methods available to us is strongly encouraged by the organizers of the event. We will be using business process software such as Disco[3] and Celonis[4], as well as different algorithms implemented in those platforms for discovery and conformance.

The *case id* defined by the authors will generally be the *case id* of the application when we are interested in what happens from the point of view of a person asking for a direct payment. This ID number is the same for all documents of a single application, regardless of the type or subprocess involved.

It is wort mentioning that given the guidelines of the BPIC 2018, it was decided to analyze and study the proposed business questions, which were created by the authors, without prejudice to the questions proposed by the challenge.

The paper is structured as follows: Section 2 presents then link between business questions and technical analysis. Between Section 3 and Section 6, the proposed questions are analyzed, including for each question: (1) Question description, (2) selected cases and activities, (3) data management, (4) observed results and (5) question conclusions. Finally, general conclusions are presented in Section 7.

2 Link between business questions and technical analysis

Table 1. Link between questions and analyses

Report Question	Business Question	Analysis
First Question	Created by the authors	Differences between rejected and non-rejected Payment Document
Second Question	Created by the authors	Differences between longest and shortest cases
Third Question	Created by the authors	Differences between regular and reopened cases
Fourth Question	Created by the authors	Differences between rejected and approved cases

Table 1 details the link between report question, business question, and the analyses included in this report:

3 Differences between rejected and non-rejected Payment Document

3.1 First Question

What are the differences between the cases in which the Payment Application was approved in the first attempt, and the rejected cases in this document? We would also like to study and characterize the cycle begin editing-calculate-finish editing in Payment Application.

Answering this question allows a better handling of each case, since it permits the formulation of suggestions for the applicant. This would improve both the experience of the applicant as well as the efficiency of the process as a whole.

3.2 Case and activities

For this question we only use the documents that had a type of *Payment Application*. We works specifically with the section where the value of the payment was calculated. The main structure consists of two important parts. One where the price is calculated (*beginning, calculation* and *end*) and another where the user can decide whether to accept or reject the calculated price in order to know whether to continue with the process and recalculate the price. This main structure studied can be seen in Figure 1. The *case id* is the *application id* and the activities are the ones originally named as such in the document.

3.3 Data management

We filtered the data by removing the cases with the path that goes from *Payment Application Finish editing* to *Payment Application Calculate* again (as shown in Figure 1). With this filter, we could visualize the cases that were included in the *happy path* or *Process map* -most common path- without having to recalculate the value of their payment.

We also analyze the path between *Revoke Decision* and *Application Calculate*. This because it is the main way to return in the cycle and it is important to see their behaviour and calculate how many cases do this.

3.4 Observed results

We can notice in the Figure 2 that approximately 73000 cases exist in the logs which enter in the calculate loop. At the same time exist over 19,272 cases exist which go back in the cycle between *revoke decision* and *calculate*. This is very important in the Payment application because it makes a difference in the way

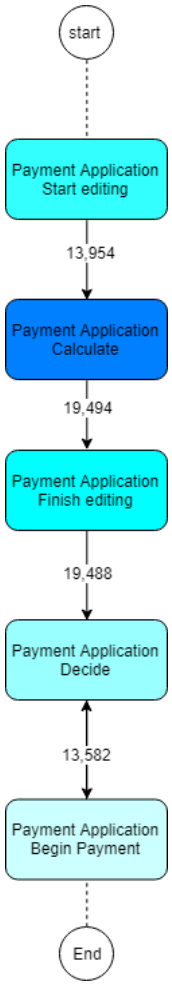

Fig. 1. Power point diagram shows the common path in *Payment Application Document* without the Editing loop. Made with Disco and Draw.io[5]

we see the price calculated. The natural question here is how many cases follow the *common path*. For the authors, this is represented by all the cases that do not have the path in the calculate cycle.

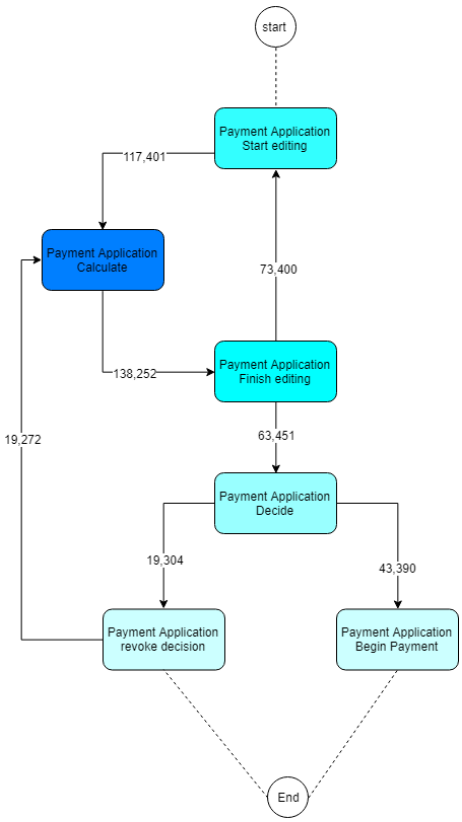

Fig. 2. Power point diagram shows the common path in *Payment Application Document*. There is a main loop between *Payment Application editing* actions. Made with Disco and Draw.io

Notice that 44% of the cases do the revoke decision in order to re-calcuate the payment value and start the payment in some later part of the process.

Figure 3, which contemplates all event logs, shows an iterative loop between *Payment Application editing* process and *Payment Application Calculate amount* process. This figure also shows the performance of the cases during the process (indicated by mean hours) through the Disco software.

As we can see in Figure 4, the common path for all cases where there was no path for re-calculate the *Payment Application amount*. This diagram shows

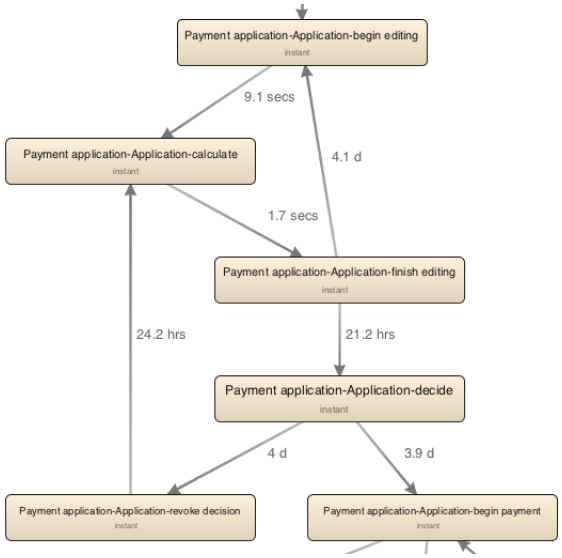

Fig. 3. Disco diagram shows the common path in *Payment Application Document*

that cases without *Payment Application Calculate* loop do not have the revoke options in this document.

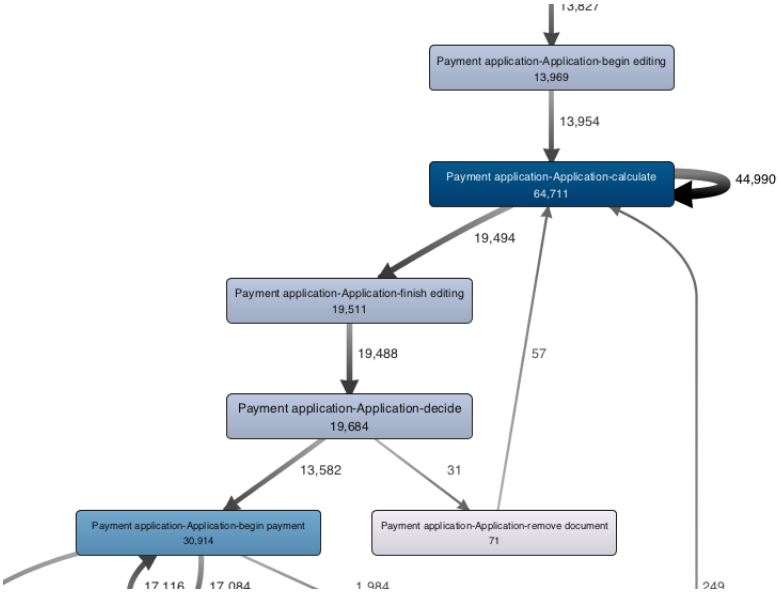

Fig. 4. Disco diagram shows the common path in *Payment Application Document*

In Figure 4, we notice some important facts. After the decide in the payment, the revoke decision doesn't appear. This means that all cases that made a good calculation of the payment value do not have to do the cycle so they are faster and can be completed before. The total duration in *application decide - revoke decision - calculate - finish editing* and *begin editing* is 9 days approximately assuming that the logs only back up once (showed in Figure 3).

There are more than 13,582 cases of this type and they represent approximately the 31% of all the cases that start the begin payment in some point.

3.5 Conclusions

It takes over 9 days minimum of delay to back up, assuming that logs only return once per cycle. It's relevant to make some changes in the order of the activities to make more efficient the process of re-calculating the payment value. Our next step is to verify how often a log does the cycle and to establish how much time this takes more precisely.

4 Differences between longest and shortest cases

4.1 Second Question

What differences can be perceived between the longest cases and the shortest, in term of the sequence of it's activities? Is there a trigger that can be used to determine whether or not the case will last longer than usual?

The importance of this question is linked to the fact that a bigger case length can increase the administrative cost for the European Agricultural Guarantee Fund. Furthermore, an analysis of the process followed by the longest cases can be useful for future applicants.

4.2 Case and activities

For this question, we separated the cases which had a long duration time (over a year). For this we selected the cases that were uncommonly longer than average, which include 76% of all cases, as shown in Figure 5

Subsequently, we also selected the cases that were not covered by the previous filter, so we could obtain two paths that could we could use to determine differences between activities that explained differences in case duration.

4.3 Observed results

Despite the reason for the penalties that appears in the log are not explained, we decided to use those penalties as our analysis focus for this question.

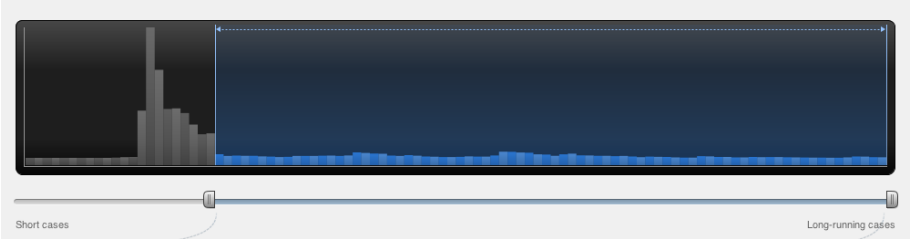

Fig. 5. Disco range of cases by cases performance

In Figure 6 we represent the part of the process in which cases have penalties. The main difference between these cases and regular cases (those without penalties and not rejected) is the time that the whole process of *Payment Application* takes to be finished with success. For a case with penalties, the time that it takes to get back to the *Control Summary Main Save* is about half of a year. This subprocess takes a lot of time and perhaps, it may be more convenient for a farmer in this situation to do the whole application again.

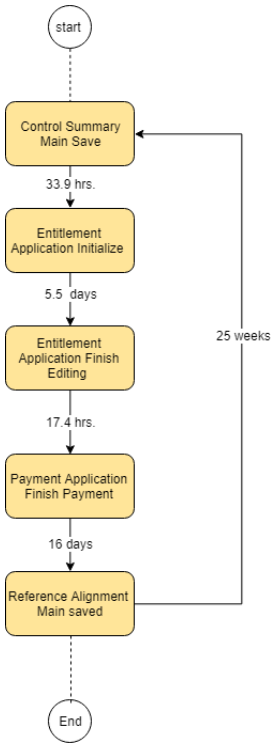

Fig. 6. Disco diagram shows the *Process map* in Payment Application Document

4.4 Conclusions

As in the previous section we establish that penalties cases where considerable longer than regular cases. One important aspect of this situation is that maybe there are some errors generated when the institution makes the records every year. However, we establish that not filtering these cases from the log will affect directly in the metrics used by the European Union.

5 Differences between regular and reopened cases

5.1 Third Question

Are there patterns in the processes that help predict which cases will be reopened? After they are reopened, do they behave similarly to the rest of the cases?

The importance of this question is based on the idea that it is important to understand the reopened cases, in order to determine if they should be treated as the ones that are just beginning.

5.2 Case and activities

To answer this question, we use every application as a different case. Then, we grouped the different activities of the process in seven main activities as shown in Figure 7: *Mail income*, *Pre check*, *Department control*, *Alignment*, *calculate payment*, *Decide* and *Finish payment*.

5.3 Data management

We use Disco's filters to separate the cases that were reopened using the subprocess attribute to differentiate them. If the case had at least one event with the subprocess attribute equals to *change* or *objection*, it means that the case was reopened. After that, we filtered by endpoints permitting only *mail income* or *mail valid* as starting activities and only *finish payment* as ending activity for all cases (as should it be in the real process). It is important to mention that the presence of the *finish payment* activity does not mean the non rejection of the application, it only means that the process was finished.

5.4 Observed results

When we filtered the results by endpoints, we observed that the process consisted of basically the same paths for both the cases that were reopened as well as for those that were not. An exception to this is the loop that exists for the cases that were reopened as shown in Figure 8. It is interesting to note that after filtering Disco showed that a 67% of cases reopened did not fulfill this restriction,

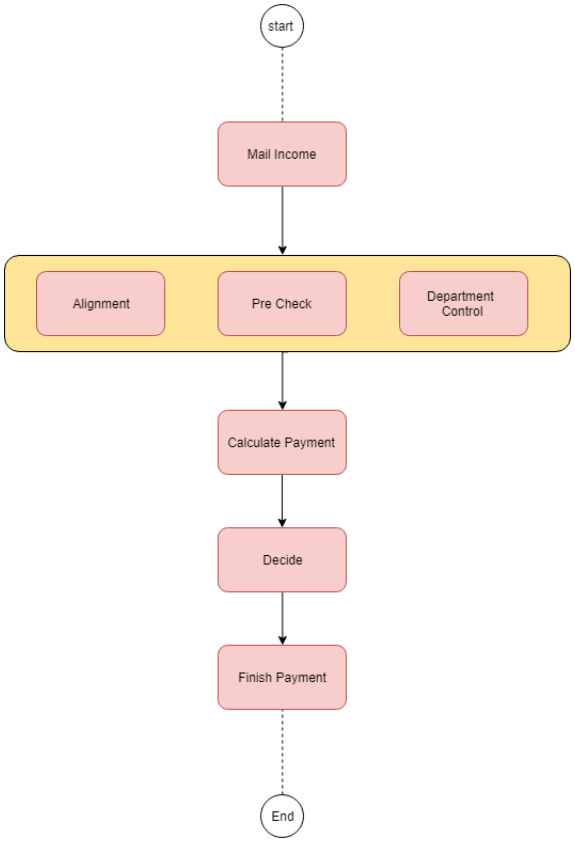

Fig. 7. Power point diagram shows the whole process grouped in seven activities.

versus a 25% in the rest of the cases.

For the reopened cases, a 66% does not have *finish payment* as final activity. That is to say, the process is not being properly completed. As an example of this, the most common variant finishes in recalculate.

It is worth mentioning that after applying a filter by attribute in Disco, it was found that only a 1.9% of the 66% were rejected cases.

5.5 Conclusions

We can conclude that before the cases are reopened, they behave similarly to the cases that are not reopened, therefore, there is no indication that the way in which the process was conducted caused the case to be reopened. Nevertheless, there was a 66% of the reopened cases in which the process do not end properly, what can be interpreted as a lack of prolixity on those cases that were analyzed

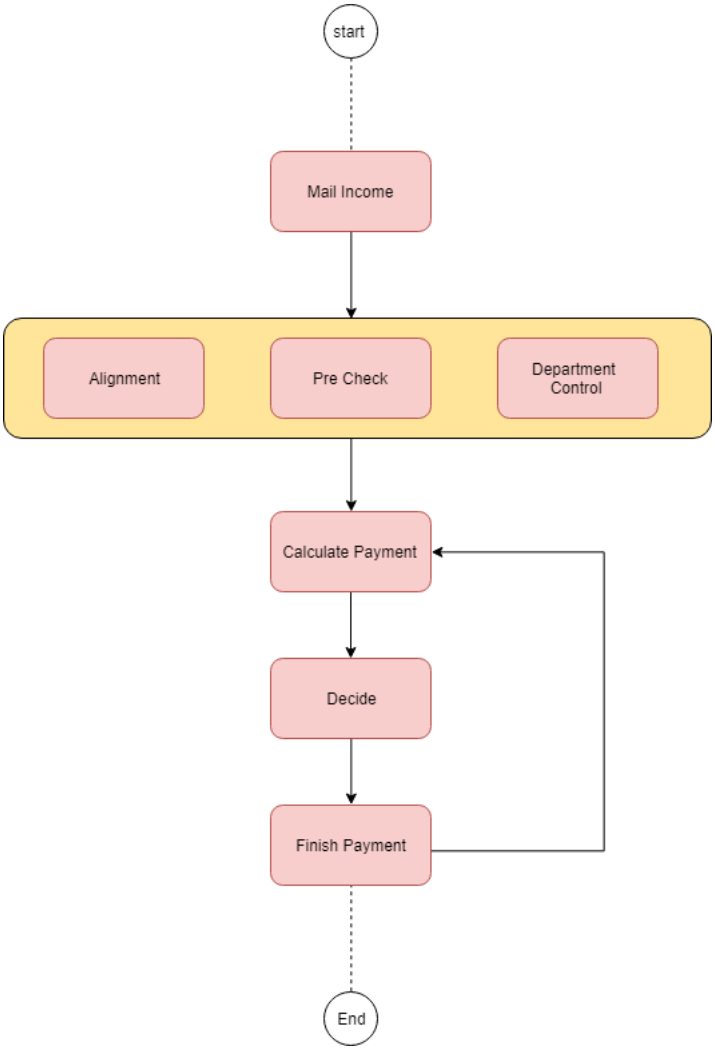

Fig. 8. Power point diagram shows the process for reopened cases.

before, which could have different consequences such as not notifying the rejection of the application (for the 1.9% of cases that were rejected), among others. We would expect for a reopened case, that the whole subprocess of analysis and calculation is performed.

6 Differences between rejected and approved cases

6.1 Fourth Question

What are the differences between the processes of the rejected and the approved cases? In which aspect do the two types of processes differ the most?

Understanding the path followed by the rejected cases allows for better recommendations and feedback to applicants, so that they can avoid having their own case rejected.

6.2 Case and activities

In order to answer this fourth question, each application was considered as a different case. Therefore, our *case id* was the one of the application. Regarding the activities, if two or more activities were consecutive and had the same doctype and subprocess, they were grouped into one major activity, what makes it easier to see which subprocess are involved in each type of application. This was done to simplify our understanding of the model, as we did not need the level of detail of the original log, which included an attribute named "*activity*". This attribute was not necessary to answer this question as we prefer to look at subprocesses, since they offer a sufficient level of granularity. The creation of the new activities was done with Python's csv package.

6.3 Data management

For this question, it is important to consider that a rejected case is one which has a value of "True" in the "(case) rejected" attribute for all of its activities. We first selected the rejected cases that started after March 1st 2016 and end before March 1st 2017 (which correspond to a full year), and then the same amount of cases (63) that were not rejected in the same period of time. Even though there is far more accepted than rejected cases (99.6% vs 0.04%, respectively, found with Disco's filters), we decided to take the same number of both types of cases, since the added complexity and long loading times associated with working with thousands of cases offered very little benefit in terms of accuracy. The 63 non-rejected cases where taken randomly from the initial log with Python's csv and random packages.

6.4 Observed results

By analyzing the rejected cases and the accepted cases separately and analyzing diagrams on Celonis Cloud for each one (Figure 9 and Figure 10 respectively), it was observed that the most frequent path between them was considerably different. In order to compare the number of cases approved and rejected with the two different type of process found we apply conformance technique (Van der Aalst, 2016). In that sense, using the most frequent path from the non rejected cases as a model, Celonis showed that only an 8% of the rejected cases follow this path, and also that in average the rejected cases has more steps and throughput time than the non rejected cases.

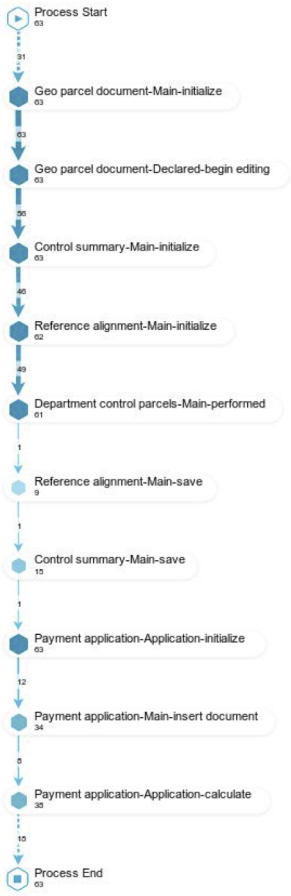

Fig. 9. Rejected cases *happy path* in Celonis

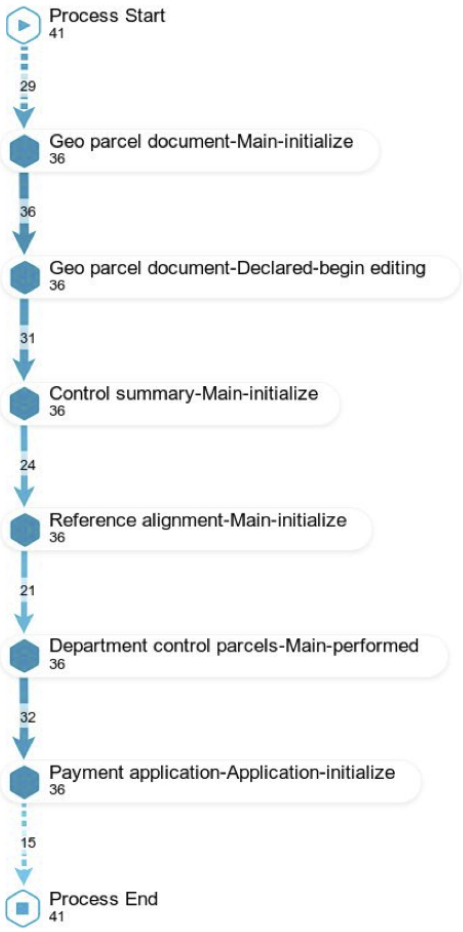

Fig. 10. Non-rejected cases *happy path* in Celonis

It could also be observed that, in most cases, the misalignment between the processes begins after the activity corresponding to the Main subprocess of the Department control parcels document, which contain the results of the checks regarding the validity of parcels of a single applicant.

6.5 Conclusions

The fact that the processes diverge when they do, indicates that the main difference between the rejected and non rejected cases is whether or not the parcels

were deemed to be valid or not, as should be expected. However, efforts could be made to determine the validity of the parcels sooner and reduce the duration time difference between the rejected and the non rejected cases.

7 General Conclusions

Among the main conclusions for the first question is the fact that the minimum time the loop takes to finish, with only one cycle, is 9 days, which is detrimental for the applicants because of the delay it introduces. That is why we recommend generating the required actions to give each applicants sufficient information, so that he or she can avoid committing mistakes that introduce a delay in the payment.

Regarding the second question, it is evident that the cases in which there is some type of penalty tend to suffer an important delay, which affects considerably the metrics of the logs. On the other hand, it must be considered that there is a final step that lasts for half a year approximately, and even though it aggregates smaller steps, it involves a significant delay that affects both the metrics as well as the analysis of the process.

In the third question, a 33% of reopened cases start and finish in the expected way. Therefore two thirds of the cases were not initiated properly or do not end in the way they should, which could impact different aspects of the application, such as proper notification.

For the fourth and last question, the processes of the rejected and non-rejected cases diverge in a point that indicates that the rejection is a consequence of invalid parcels. We believe, based on this data, that efforts should be made to determine this validity sooner, therefore reducing the length of the rejected cases.

8 References

[1] Van der Aalst, W. M. (2016). Process mining: data science in action. Springer.
[2] Business Process Intelligence Challenge 2018 (BPIC). From `https://www.win.tue.nl/bpi/doku.php?id=2018%3Achallenge`.
[3] Disco Software. Download from: `https://fluxicon.com/disco/`
[4] Celonis Software. Download from: `https://www.celonis.com/`
[5] Draw.io. WebApp Available on: `https://www.draw.io/`