Process Control-Flow Discovery

Hugo Martins Dias Simões Monteiro

Instituto Superior Técnico, Lisboa, Portugal

October 2021

Abstract

Process mining focuses on the discovery and analysis of processes based on event logs that are recorded by the information systems in an organization. With more and more event data being recorded by such systems and with processes representing the core value of an organization, process mining becomes a fundamental e crucial task.

In this document first we start by present some of the tools used in Process Mining and how they use logs as input to apply discover and produce the respective process models. With focus on what transformations can be made to information.

We propose a solution using process mining to discover a specific process based on a ticketing system. For the implementation phase we used Splunk Business Flow a tool from Splunk big data platform capable of producing a process model and respective metrics from logs previously indexed. We used this tool since the project in case was implemented based on Splunk and we knew in this tool it would be possible to apply transformations to data used.

Keywords: Process, Process Mining, Process Discovery, Process Model, Event Log

1. Introduction

Nowadays all organizations have their value services structured around processes with focus on different areas as for example manufacture products, purchase goods, handle applications and manage systems. This represents the core functionality of an organization and on this note organizations are getting more interested on studying and understanding the inner processes on which they rely for their business to get the most value out of it.

In this challenge, the main focus was on processes known as sequence of steps that have an outcome and Process Mining which is a field of Computer Science that allows analysis like conformance checking (namely monitoring deviations between observed behavior and modeled behavior), discovery (learning a process model) and Extension of an organization's processes generated by existing information systems. Process Mining aims to help organizations navigate the business. That is, understand all possible routes, recommend the best path, calculate each journey time and other Key performance indicator metrics. All this lead to continuous process improvement, provide insights, identify bottlenecks, anticipate problems, with direct adjusts on a service level meaning more productivity and less costs at the end. Process Mining has been successfully applied in a variety of domains like healthcare, electronic business to high-tech systems and auditing.

The motivation was a project in a specific global organization with focus on measure and control Service Level Agreements in the context of an Fibre to the Home project between two telecommunications organizations based on a ticketing system. Since this process is about a network installation, sometimes errors can occur related to the equipment and to solve this problem a fault ticket needs to be opened in order for the fault to be repaired. This faults can be of different types, which originates different paths from start to finish that we want to discover. Since the project was already implemented in Splunk Enterprise [1] the proposed solution is based on Business Flow tool from Splunk.

The main goal is to use process mining over real event logs by applying process discovery in order to get all different paths a ticket can follow. Once the respective process is known with all its steps, this information can then be used as initial configuration for the engine and be updated automatically every time a change occurs. As off now this configuration is set and updated always by human intervention which is the main obstacle to overcome.

The remaining part of the document is organized as follows: in Chapter 2 some of the Process Mining main concepts and tools will be described; in Chapter 3 we present the FTTH_SHARE project with details on different types of tickets and the dataset used as input for our work; in Chapter 4 we present

[1] https://www.splunk.com/

our solution and how it was implemented; in Chapter 5 we present our results and experiments with data from FTTH_SHARE project by using Splunk with Business Flow tool; finally, in Chapter 6 we include conclusions and a discussion about future work to do based on current limitations of the system used in our experiments.

2. Related Work

This section presents the main concepts about process mining and an overview of different process mining tools and how they transform event logs and normalize the data to apply process discovery algorithms. This tools were studied so that our proposed solution can address problems encountered in them and provide additional features.

2.1. Process Mining: Main concepts and techniques

2.1.1 Event Log

The starting point for process mining is an event log, without data it is not possible to analyze processes and make improvements. Logs are made of events that represent activities, a well-defined step in some process as result of critical business actions by a certain resource. However this data comes in different formats as it depends on the system where it is produced from a database to a web service. The first step is to transform this machine data that can come in any format to a format that each tool is able to read and to analyze. For this to occur the event log as to follow a specific format containing the following fields: Case ID, Activity and Timestamp[1].

2.1.2 Business Process Management System and Business Process

Enterprises can take even more benefits from this information if they use software systems for coordinating the activities involved in business processes, called business process management systems [2]. This type of system is driven by explicit process representations to coordinate the enactment of business processes to take advantage of process improvements including reducing costs with lower execution times resulting in lower error rates, but also gaining competitive advantage through innovation as described in [3]. A business process consists of a set of activities that are performed in coordination in an organizational and technical environment.

The desired behaviour for a process[4] can be further represented in a so called process model that illustrates the work done step by step by representing the flow from a starting point all the way to the end. To represent a process model there are very different techniques some of the most used are BPMN and Petri Net.

2.1.3 Process Mining Purposes

As the number of events recorded increase day by day the need to analyse this critical information also increases in order to provide detailed insights about the most important processes. To improve and support processes in rapidly changing and competitive environments, process mining raises as a technique focused on extracting information from event logs and is the bridge between process analysis and data analysis. There are three fundamental types of process mining: Discovery, Conformance and Enhancement.

Process mining is based on facts and so behavior recorded in event logs and the respective models. Using this information is possible to apply different types of analysis depending on what the goal is. This way Process Mining can be related to four different perspectives: Control-Flow, Organizational[5], Case[6] and Performance.

2.1.4 Process Mining for Process Discovery

Process discovery is one of the most common process mining ways of producing a model from the information captured from different applications. Due to the extremely high volume of data that is available for each process, this activity is way more than process modelling. Using this information to identify bottlenecks, the basic causes of problems, deviations from the process, and the distribution of process events. In other words it captures what happens in real life and provides a meaningful simulation to enable process optimization. So the produced model is as accurate as the data linked to each event, containing in the best case starting time, ending time, resource information and data.

When it comes to process discovery different methods can be distinguished: evidence-based discovery, interview-based discovery and workshop-based discovery.

2.2. Overview of Process Mining Tools

2.2.1 Open-Source: ProM

ProM is an open-source framework that allows the interaction between several plug-ins. Each plug-in representing an implementation of an algorithm that can be added separately with no need to modify the framework by adding only the corresponding file[7]. This tool can read files in different formats like XML(Extensible Markup Language) through the log filter component that deals with large data sets and sorts the events within a case on their timestamps before processing the data. After this the mining plug-ins are applied and the result is stored in memory. This framework also allows to use multiple plug-ins one after the other, so the output of a plug-in can be used as input to another

plug-in. In the pre-processing phase is possible to apply filters to the processes[2].

2.2.2 Commercial: ARIS

ARIS is a product of Software AG[3] that focus mainly on monitoring performance through process instance analysis using different visualizations in dashboards. This tool gives the possibility to see a potential model of each executed process instance, all done automatically and afterwards the option to save it in a repository.

Model analysis is possible through "slicing and dicing" at the process level[8]. This tool is capable of dynamically generating an aggregated process view for each query. The structure can also be visualized as a Gantt-chart to comprehend easily the sequence and overlap of the activities in the process. This is specially suited for the detection of waiting times within a process. ARIS supports the import of data in arbitrary order, as it keeps track of all events imported at any time[9].

2.2.3 Commercial: Disco

Disco is a tool from Fluxicon[4] made to speed up process mining by making data import really easy through automatically detecting timestamps, remembering configuration settings, and fast loading data sets. Supports control-flow with the ability to filter by task or path absolute frequency and supports performance perspectives and has limited support for organizational perspective.

Disco starts by guessing what each column might mean, but it is possible to adjust configuration before the import proceeds by using the preview mode. For each column is possible to ignore, set as the caseID, set as activity name, set as timestamp, set as resource or as an additional attribute[5].

2.2.4 Commercial: Minit

Minit[6] is a process discovery tool that works with data from different information systems like ERP or CRM.Allows import formats as CSV, XEX, MXML, SQL Server, Excel, Access and ODBC[7]. Also, if a process spans across several different systems, the respective logs are combined to create a complete picture of a process.

Minit since version 4.5 allows to choose data type for each column, detects and offers suggestions but is possible to change it. The available types are string, float, integer, duration and date[8].

When importing a log file is possible to preview the data. Next step is to pick the attributes caseID, activity and timestamp necessary for process mining algorithms to work. After this step, is possible to review the import settings with a new column called "attribute level" that shows if the corresponding field is event or case level.

2.2.5 Commercial: Celonis

Celonis[9] is one of the marketing leaders in Process Mining and has a process cloud based solution called "The Intelligence Business Cloud" and has released "Celonis Snap" the first free cloud process mining platform[10]. With this tool is possible to connect with new sources of data like Servicenow or Google Sheets platform or use the traditional formats.

This tool works with Data models as source of data, which are collections of tables used for the same purpose of analysis[11].

2.2.6 Commercial: bupaR

Another Process Mining tool is bupaR[12], an open-source integrated suite of R-packages for the handling and analysis of process data. Here we focus on the main package "bupaR" for creating event log objects. Data transformations are allowed, as sometimes data will not come at the desired format[13]. The common issues are: lack of transitional lifecycle, lack of resources, activity log.

3. Problem Analysis

The project in case is supporting a fault based ticketing system, based on full manual process configuration of its steps and corresponding transactions. The project and functionality will now be described.

3.1. Overview of Project: FTTH_SHARE

The goal of the project was to measure and control service-level agreement in the context of the FTTH_SHARE project, which is a joint Fibre to the Home project between two telecommunications organizations. All types of interactions in the context of FTTH_SHARE process are captured by logging all service executions, including input and output parameters at the Middleware platform level to be collected in Splunk for further analyze.

[2] https://www.promtools.org/doku.php
[3] https://www.softwareag.com/en_corporate.html
[4] https://fluxicon.com/disco/
[5] https://fluxicon.com/book/read/import/#importing-data-sets/
[6] https://www.minit.io
[7] https://www.minit.io/blog/how-to-prepare-your-data-for-a-process-mining-project
[8] https://medium.com/minit-process-mining/minit-4-5-is-out-heres-a-rundown-of-what-s-new-5333314627fa
[9] https://www.celonis.com
[10] https://www.youtube.com/watch?v=-EQcqxeYUMg
[11] https://support.celonis.de/display/CPM4E/Datamodels
[12] https://www.bupar.net
[13] https://www.bupar.net/creating_eventlogs.html

This interactions when related enable the analyzes of the time it takes to resolve a technical failure. This calculations can become very hard when dependencies happen, like process steps that are not relevant for the final calculations or repetition of steps that represent process internal cycles.

This type of agreements between the two parts have to be respected because exceeding limits result in very expensive penalties per ticket. This solution will give a quick overview over data to gain vision on given penalties.

In this case, the project was already built in Splunk which is a big-data platform used mainly for processing logs, that enables data integration from different sources in real time and build fast searches to be used in reports,dashboards and generate threat detection alerts. In order to facilitate writing queries it has its own processing language so that a higher level of abstraction is provided to users. Splunk is composed by three main components: forwarder, which forwards the data to the remote instances; indexer, which stores the data and answers the searches; search head, which is the front-end usually accessed through a web interface. In this project the different knowledge objects (lookups, dashboards, saved searches) exist in the search head grouped under an app that includes an engine that categorizes all collected events previous to visualization. The main goal with this engine is to separate data processing from visualization.

The different dashboards available in the application allow for an analysis on points such as:

- The evolution of average service-level agreement time.
- Evaluate most common types of technical failures.
- Understand most common resolution types.
- Search for open tickets per fiber provider.

3.2. Ticketing System

During the process life-cycle, when network sharing happens between two organizations problems can emerge and this cases are reported using tickets.

The implemented fault processes were:

- **"Avaria em Acesso de Cliente"**: This fault process refers to an unique optical distribution point or in point to point solutions to an unique fibre. Should be initialized by the beneficiary operator when a problem is detected in client's access, after performing remote testing. The beneficiary, when opening the ticket should provide information for the necessary controls to determine the nature of the problem.

  The process starts with a create ticket operation by the beneficiary, with the necessary information as client access id, type and source of fault, urgency level and observation notes if existent. If accepted then ticket is pending for acceptance otherwise the response is failure and ticket is rejected.

  After notifyTicket is called and the beneficiary is informed about the resolution type, if success then the ticket can be marked as resolved and can reopen in the next few hours depending if it is urgent or not. The next Update and Notify interactions represent possible joint interactions as a result of a reopened ticket.

- **"Avaria em Serviço Cliente Detectado em Provisão"**: This process should be used only when the beneficiary operator is doing his own client installation and can not be concluded because of a fault in one of the optical distribution point ports.This type of fault can be created by the beneficiary when related to a problem found while client is in Provisioning. In this case, the urgency is higher and the priority is set as input parameter to "M_URG_2H". After creation, the provider has 15 minutes to accept or reject this request, sent through notifyTicket. NotifyTicket can be sent with different parameters as "ACEITA_AVR_PROV" in this case the fault stays with the same priority and the respective service-level agreement, when set to "NAO_ACEITA_AVR_PROV" here the respective priority and service-level agreement go lower and the ticket life cycle continues with this new values. In case the 20 minutes are exceeded and the beneficiary gets no information about the state of the fault and gets immediately rejected.

- **"Avaria Comum: Detectada pelo Beneficiário"**: The beneficiary notifies the owner by sending a generic create failure request with information about the ticket, ports, respective optical distribution point and extra information. In this type of fault, the parameter type identifies who detected the fault, in this case the fault type is "AV_COMUM_BENEF". After this, the owner sends the identifier of the fault in question through createFailureResponse method and then both parts can add comments. After the intervention, the owner sends a resolution notification with the root cause and an information about all client fault's affected by that. When closed, means all other individual faults are closed too.

- **"Avaria Comum: Detectada pelo Proprietário"**: The provider detects a fault in his own network and sends a create generic fault to the beneficiary. In this request, the owner is responsible for sending information about the ticket, ports, respective optical distribution point and the fault type will be "AV_COMUM_PROP". Like in the generic fault, both parts can do comments during the process. After resolution, the owner sends the notification with the root cause and the information about all client faults affected. After closure, all other unique faults are closed too.

4. Proposed Solution

Considering that the current process configuration with all possible steps and paths used as input for the FTTH_SHARE engine, was created and updated manually and having in mind that the main goal was to find a way to discover and turn this process fully automatic. Different process mining tools like ProM and ARIS were analyzed as shown in section 3, to understand how they use logs as input to further apply the different algorithms in order to discover processes. Considering that the logs in case were already in Splunk and the developed engine too, was made the decision to explore Splunk applications that could discover processes and enable the possibility to visualize them. Splunk Business Flow was chosen since is a process mining tool that besides the process models produces statistics and enables different interactions by applying all kind of filters like time and resource based and data transformations.

4.1. Implementation

After studying how Splunk Business Flow uses logs and initial configuration, work was divided in two phases where each one uses different indexed data as input, to first discover the process model with the already known configuration steps and transitions produced by the engine and then discover the process from the original logs.

4.1.1 Phase 1: High-level process discovery

For this phase, the implementation was based on data from esi_process_final splunk index that contains steps and transitions produced by FTTH_SHARE project engine. This way, the search (listing 1) was written having in consideration only tickets with esi_process_sequence value "OK", meaning for each ticket every transition was checked for the all sequence to be considered valid. A transition is valid if the respective steps inside a ticket are in the same order as they occur in time. Since the events are collected in Splunk not as they occur in real time, their time column needs to change to the corresponding time in which they occur as in esi_process_step_time field.

In order to explore the corresponding flow model, the input parameters used by Splunk Business Flow need to be configured. The "Sample Size" was defined as 10 000 events and the "Max Duration" parameter as 365 days to cover tickets that have a duration of one year. The other input fields used to correlate events, were selected based on the definitions already explained before, for "Correlation ID" the selected field "process_id" since it is an unique identifier of each ticket, for "Step" the selected field "esi_step_label" that represents each step name and different "Attributes" were selected for further analysis as actionSubType, actionType and circuitID.

4.1.2 Phase 2: Process discovery from raw data

In this second phase the goal was based on indexed raw data present in esi_payload as a result of each API call, discover the initial process configuration. In this case, was considered only data extracted from December 2018 since it was previously indexed in Splunk. In this search (listing 2), all calculations are defined under a macro component called ticket_process_discovery which is a knowledge object from Splunk that works as a function in a way that can be called with different parameter values.

This macro has three input parameters **steps**, all functions considered for a ticket,**filters** that represent splunk field names for this functions and an **ignore_filter** with all the field values to exclude from the search. Using a macro for this implementation makes it easier for future changes in case of process adjustments, with an easier way of adding new steps. This macro (listing 3) first applies the ignore_filter to the already existing search string and using eval function creates a search field for each input parameter (step and filters). Since both fields have multiple values separated by commas, split function is applied to create a Splunk multi value followed by an eval to create a new field called "step_custom_name" that is further used by Splunk Business Flow tool to build the corresponding process model. This custom field is made of the function name applied to the ticket and the corresponding parameters used, for example CreateTicket:URG,AVR_AC. In this first solution of phase 2 the parameter names are still coded without any translations being applied.

For this phase the input parameters defined in Splunk Business Flow were similar to phase 1 with the exception of "Step" parameter that was defined to use field "custom_step_name" as a result of the search previously described.

In this second phase since data used for input in Splunk was a result of each API call, the corresponding values were coded in a sense that were not understandable by anyone outside of the team. To solve this problem was produced one different macro called ticket_process_discovery_decoded with the same parameters as the previous one but with the difference that translates each field value called "Motivo Resolução" to the corresponding text that anyone can look at the process and understand it.

Listing 1: Phase 1 - Splunk search used as input in Splunk Business Flow

```
index=esi_process_final esi_process_sequence="OK"
| blueeval _time=esi_process_step_time
```

Listing 2: Phase 2 - Splunk search used as input in Splunk Business Flow

```
index=esi_payload source="
      esi_payload_ftth_share_20181201_20181231.txt"
`ticket_process_discovery("CreateTicket,NotifyTicket,
      UpdateTicket,TicketCreateResponse","dt.in.
      CreateTicket.Ticket.type,dt.in.CreateTicket.Ticket.
      priority,dt.in.NotifyTicket.Ticket.actionType,dt.in.
      NotifyTicket.Ticket.actionSubType,dt.in.
      UpdateTicket.Ticket.actionType","(NOT dt.out.
      ResultsStructure.StatusMessage=*) (NOT dt.in.
      NotifyTicket.Ticket.actionType="COMENTARIO")
      (NOT dt.in.UpdateTicket.Ticket.actionType="
      COMENTARIO")")`
```

Listing 3: Phase 2 - Splunk macro used in search

```
$ignore_filter$
| blueeval step="$steps$",
        filters="$filters$",
        step_multi_field = purplesplit(step,","),
        filters_multi_field = purplesplit(filters,","),
        step_custom_name= purplemvmap(
              step_multi_field, purpleif(purplelike(_raw
              ,"%".step_multi_field."%"),
              step_multi_field, purplenull())).": "
| blueforeach *
    [blueeval step_custom_name = step_custom_name.
          purpleif(purpleisnotnull(purplemvmap(
          filters_multi_field, purpleif("<<MATCHSTR
          >>" = filters_multi_field orangeAND '<<
          FIELD>>'!="", '<<FIELD>>', purplenull()))
          ,purplemvmap(filters_multi_field, purpleif("<<
          MATCHSTR>>" = filters_multi_field
          orangeAND '<<FIELD>>'!="", '<<FIELD
          >>'.", ", purplenull())),"")]
| blueeval step_custom_name =purplesubstr(
      step_custom_name,1,purplelen(
      step_custom_name)-2)
```

4.2. Results

During the implementation phase not everything went smoothly as expected. The process model produced and the corresponding graphs created by different filters, could not be saved as images and exported from the application, this is a known issue from this tool. Every time Splunk Business Flow was opened the corresponding flow search had to run to produce the graphs and enable further analyzes. In this time period, the license ended and after trying different approaches to solve this, new license could not be added. With this limitation over application usage, the only access possible was flow's initial page with the search and filters. All images used in section 5 were created before the license expired with focus on showing process models for the different ticket types and analyzes over the most frequent process cases, with focus on the most frequent case showing step details and count. In this analyzes, is shown an overview over the attributes where we can see the most frequent values for the most important parameters used in each API call and also Splunk Business Flow has a section "Metrics" that has statistics for the overall cases, with average duration and count.

5. Demonstration

In order to conduct our main experiments, we had to install Splunk Business Flow tool on a server where the logs were previously copied to. This data was exported from a month period corresponding to every API call as a result of every interaction between the two organizations. In preliminary experiments along with this information, the corresponding processed data from the engine was also collected as a starting point to our solution. As mentioned before, during this phase not everything went as expected and the software license ended earlier, restricting the results we could export. After trying different approaches to solve this, a new license could not be added and with this limitation the interaction with the tool got very restricted to only access flow's definition page with the corresponding search and filters.

Since this tool does not enable the option to export models as images we had to do it in the traditional way with prints each time a result was produced and this limitation is another reason why we could not proceed with a formal evaluation phase and all process images in this work are a result of it. All models used in section 5 are a result of applying process mining with focus on showing the corresponding processes for the different ticket types and analyzes over the most frequent process cases. In this analyzes, is shown an overview over the attributes where we can see the most frequent values for the most important parameters used in each interaction and also Splunk Business Flow has a section "Metrics" that has statistics for the overall cases, with values as average duration and count.

5.1. Process Model Produced

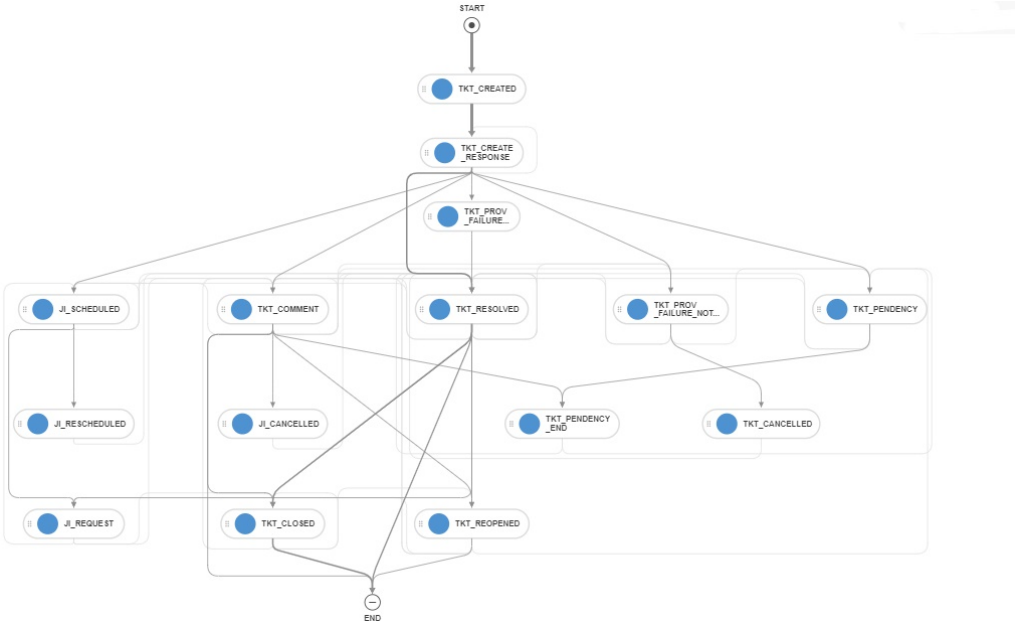

Figure 1: Phase 1 - Process model

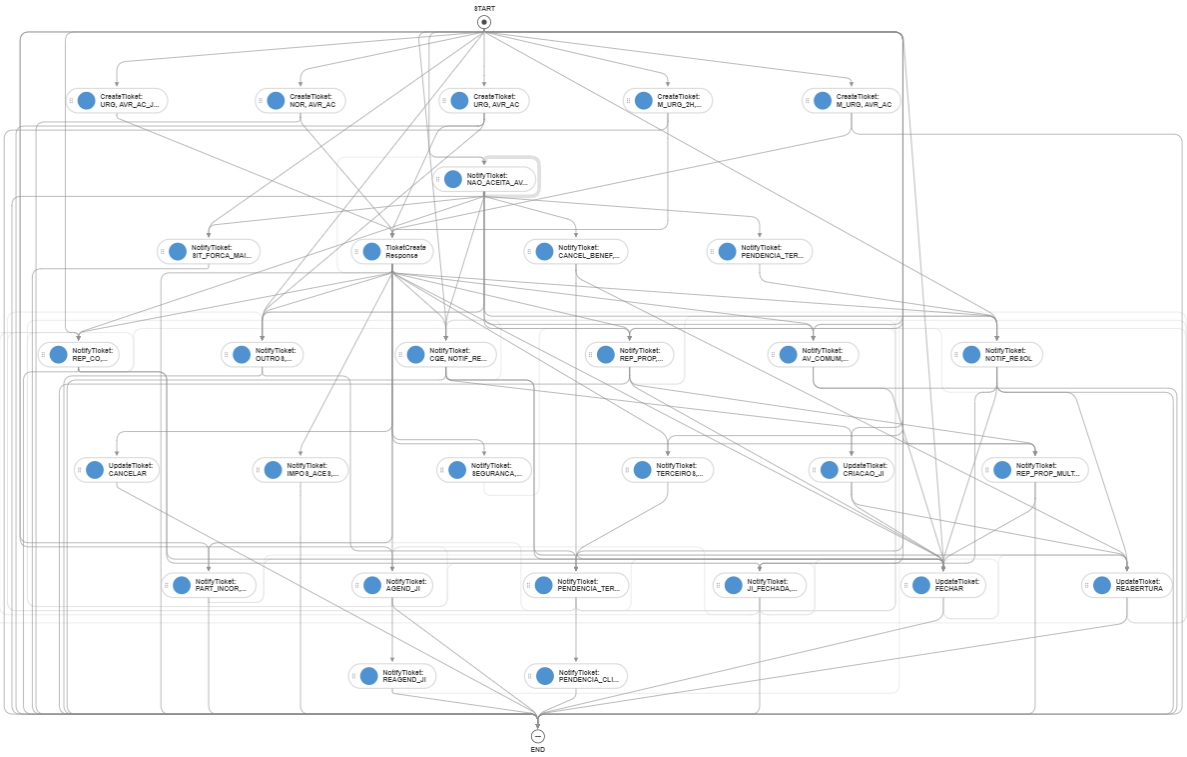

Figure 2: Phase 2 - Process model

As a starting point to our solution we used logs with the already processed step names, as a result of the engine in the context of the FTTH_SHARE project. The goal was to use this logs to apply process mining using Business Flow tool in order to reach the configuration that was already known. This configuration is a large table with values for all possible steps and transitions for each ticket type and so it defines the overall process.

The first step was to find the model that could better describe the process by searching for only valid tickets where it could be possible to see from start to finish the different fault types related to each ticket since the moment they were created until they closed. All model diagrams shown in this section were produced in Business Flow, as shown in Fig.1 for a ticket to be valid it has to start with create operation from one of the organizations followed by the corresponding response saying success or failure in creation. Following these operations is a sequence of steps and the process only finishes when we have a call over close or resolved operation, only these cases were considered as valid for mining the corresponding process.

In this phase, the valid sequences were already computed following these rules so we could apply mining directly from the log without the need to apply any transformations. After the process model was produced we could filter the results by the correlation_id attribute which was the field that was responsible for joining related events depending on each ticket type.

Once we got the process model for each ticket type, the next step to our solution was to produce this models but with the additional information present in the logs as a result of each interaction between the two parts. This second phase worked as a "drilldown" from the one presented, since here(Fig.2) we have access to the parameters that were used in each call resulting in much more steps and overall complexity of the process model. In this case since the information available did not receive any treatment and was directly from production log, we had to do an additional processing step since some cases reflected situations corresponding to processes that did not finish during that month period or to an API call that failed resulting in invalid processes. This processing step was were most of the time was spent during this phase because for each different correlation_ID we had to filter out every sequence that did not ended in one of the supposed steps (resolved or closed).

To separate the process model in each ticket sequence of steps, we used the parameter "type" that is used in the "CreateTicket" call. With this information is possible to divide the original process in one for each type of ticket. From the process model shown in Fig.2 we can see that the first step that was previously called "TKT_CREATED", now corresponds to three different steps when referring to "Avaria em Acesso de Cliente" depending on the parameters used in each call for example priority parameter can assume between the following values: URG, M_URG and NOR. With this results we were looking to find similar results as the state diagram presented when it comes to tickets of "Avaria em acesso de cliente". This representation is used as a starting point for the project and represents all interactions between the owner of the service and the beneficiary. Here we translate this interactions to a sequence of actions that build a "conversation" between the two organizations.

5.2. Attributes and Metrics Overview

In this section we present the main metrics and the analyzes produced in Splunk Business Flow, over the process models presented in Section 5.1 for each separate phase. In "List" view is possible to see all process cases listed for the results of phase 1 and also for phase 2 with the corresponding start and end times as well as the total duration and step count. This view enables a drilldown by clicking on a "process_id" value to see the sequence of steps for that particular case. Also, on this tab is possible through the filters side bar to see the most frequent case occurrences grouped by the respectively complexity.

In Conversion tab we can see a diagram (Fig.3) showing the distribution of process instances considering the four main stages. From the total of 357 process instances used for analyzes, all went through the TKT_CREATED operation and all got resolved but only 56.02% of this tickets could be closed. We can also see that one ticket got reopened, probably because the problem was not resolved completely on the first attempt.

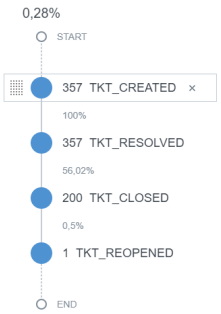

Figure 3: Phase 1 - Conversion path

In the Attributes page we can see the analyzes for the first phase showing the distribution of values for each defined attribute. The actionType and actionSubType are the two parameters used on notify and update actions. For the actionType we can see that all tickets have a NOTIF_RESOL meaning that all have a reason for their resolution as we have seen previously all tickets got resolved. For this resolution step the reason behind it is shown on actionSubType parameter that can have some of the following values OUTROS, CQE and REP_PROP_MULTIPLO. From this charts we can see that the actionType has the value "FECHAR" 200 times, which reflects the analyzes made in Fig.3 that shows 200 process instances got closed.

In the metrics page is shown overall statistics for the process, with focus on average duration for a process instant around seven days meaning that a fault ticket on average gets a week to be resolved which is too much since it affects clients, restricting their access to equipment and limiting their life's. Also is show the total of process instances and their distribution over that period. We can also see an analyzes over the conversation rate showing that only 0.28% of the processes end. This is not a good result, since every organization wants their service to be delivered as fast as possible with the best quality, specially if it has negative impact on peoples life's.

As shown previously the process model produced by Splunk Business Flow can be filtered by their attributes values and also on FlowChart view is possible to add the option to show step and path count to be shown on the produced mode. From this analysis we can see that the most common path for a ticket is their creation, then waiting for the other organization to respond to the request and after this the ticket is resolved and then closed. From this analyzes is also possible to see that the most executed steps are ticket created since it is needed every time a fault occurs and the corresponding resolution to the problem even if sometimes the resolution applied does not solve the problem 100%. Also it is possible to make an analyzes based on the average duration and count for each transition showing that a ticket cancellation is the operation that takes more time related to the overall decision to be made. When a comment is made from one of the organizations usually is an operation that takes 113 hours to be done again, since is the time for the other part to answer on it.

Based on the results we got for phase 2 we can conclude that they were similar to phase 1 with the addiction of distribution of values shown for actionType and actionSubType parameters split by NotifyTicket and UpdateTicket operations.

In this phase the metrics produced for average duration and journey count were different as expected because the data that was used as input was different. In this case, we chose to exclude all NotifyTicket operations that had actionType parameter with value "COMENTARIO" since is not a relevant step to understand the process and how ticket interactions work resulting in a significant reduction on the overall average time for a process instance since it was the slower step to complete as explained before. In this phase we have more journeys because we were less restricted on the criteria to chose what a valid sequence of steps was, considering all process instances that ended with no other limitations.

For this phase we also made an analyzes over the process model, showing that CreateTicket that we have seen before as the operation that was always executed as a starting point for the process is now divided in different steps with "Avaria em acesso de cliente" with priority "Urgent" as the most common type of fault. The rest of the steps have similar distribution with focus on NotifyTicket being called 58 times to report situations where fault requests are not accepted. Relative to the average time for the possible paths, we can see that the most time spent is on NotifyTicket steps since most of the time reflects a decision made by one of the organizations based on the received ticket information.

In this chapter, we presented the results of applying Process Mining in the context of the FTTH_SHARE project to produce a process model using Business Flow Tool. The different ticket processes were presented alongside some of the most important metrics. After showing and analyzing the results of the process model produced, we compared metrics for both phases of the solution related to the overall count and duration of process instances. In the end, we presented a more detailed model with information about each step and path count as well as the respective duration.

6. Conclusions and Future Work

6.1. Conclusions

This document addressed a problem related to process discovery between two well known organizations related to Fibre to the Home share service. The process in case was about a fault management system based on tickets and the current solution implemented in Splunk was based on a configuration table that described the process, updated always by human intervention. We used Splunk Business Flow tool from Splunk which is a an application used for Process Mining that enables using logs previously indexed to run searches to apply transformations on data before producing the respective process models and metrics over it. Since the logs were already in Splunk we used this information with the necessary

transformations to produce the corresponding process model that better described the fault process and its different possibilities.

We proposed a solution for three different types of tickets "Avaria em Acesso de Cliente", "Intervenção Conjunta" and "Serviço Cliente Dectado em Provisão", by correlating events based on their process instance. The solution was divided in two phases, first using already processed logs as a result of an engine programmed on Splunk that categorizes the events and correlates them resulting on only valid processes. Then we opted for a more complex solution using the original logs that had all interactions between the two organizations and with this information we produced custom step names composed of the operation name plus the parameters used on each API call. In the end, we produced some analyzes over the process model with focus on average duration and count for the corresponding process instances and steps. During this period where we had to use Splunk Business Flow not everything went as planned, the first steps before we could actually start using the tool and see some results were the most difficult and where the most time was spent because we had to find the correct period of data to use and apply the necessary adjustments to the corresponding events. Another obstacle during the implementation was that the software license ended before we could export the models with better quality or even go deeper on the analyze over the produced models and metrics. Since we lost access to Splunk Business Flow tool right after the implementation phase we could not proceed with the evaluation, where the plan was to compare the produced models with the configuration used for each ticket process.

6.2. Future Work

we believe the results we got would serve as a base for future work after going through evaluation phase for discovering the process with more detail. In this work, we focused only on three types of tickets and discovering the respective processes but with what was produced this solution could be extended to other fault types, meaning we could have multiple combinations only by adding the corresponding operation names and their field names to the macro parameters used in the search.

One feature that we see as a starting point for future work is to use the process models produced and transform it on the corresponding process definition that could be imported and used as initial configuration to the process this would reduce human intervention to almost zero, preventing possible errors when updating it and also would enable constant process adjustments. This could be done by investigating how the tool produces the model and if it saves the corresponding steps and transitions on an output file.

Since we had to use Splunk because the logs and the corresponding engine to process the events was already implemented there, it would be interesting to try one of the tools presented in specific Disco since it enables the possiblity to export in XML format and that would facilitate the transition from the process model to a definition that could be imported to the table that serves as initial configuration to the process.

References

[1] Wil Van der Aalst. Process mining: Overview and opportunities. *ACM Transactions on Management Information Systems*, 3:7.1–7.17, 07 2012.

[2] Mathias Weske. *Business Process Management Concepts,Languages,Architectures*. Springer, 2007. 978-3-642-28616-2.

[3] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A. Reijers. *Fundamentals of Business Process Management*. Springer, 2018. 978-3-662-56508-7.

[4] Sérgio Guerreiro. Introduction to Business Process Automation. 2017.

[5] Wil Van der Aalst, H. A. Reijers, and M. Song. Discovering social networks from event logs. *Information Systems*, 14(6):549–593, 2005.

[6] Wil Van der Aalst, H.A. Reijers, A. W., and Others. Business process mining: An industrial application. *Information Systems*, 32(5):713–732, 2007.

[7] B.F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, and Wil Van der Aalst. The ProM Framework:A New Era in Process Mining Tool Support.

[8] Wil Van der Aalst. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011. 978-3-642-19344-6.

[9] Dr. Tobias Blickle and Dr. Helge Hess. Automatic Process Discovery with ARIS Process Performance Manager (ARIS PPM) “It’s about behavior!”. October 2010.