

# Process Mining Techniques in Internal Auditing: A Stepwise Case Study

Flavia Maria Santoro<sup>1</sup>, Kate Cerqueira Revoredo<sup>2</sup>, Rosa M.M. Costa<sup>1</sup>, Thais Mester Barboza<sup>3</sup>

<sup>1</sup>IME/Universidade do Estado do Rio de Janeiro (UERJ)  
Rio de Janeiro, Brazil

<sup>2</sup>Wirtschaftsuniversität Wien (WU)  
Viena - Austria

<sup>3</sup>BSI/Universidade Federal do Estado do Rio de Janeiro (UNIRIO)  
Rio de Janeiro, Brazil

flavia@ime.uerj.br, kate.revoredo@wu.ac.at, rcosta@ime.uerj.br,  
thaismester@gmail.com

**Abstract:** *A business process is a sequence of activities organized in a logical way in order to produce a service or a product that is valued for a particular group of customers. Process auditing in corporate environment aims to assess the degree of compliance of processes and their controls. Due to the volume of information that needs to be analyzed in an audit job, auditing's cost can be very high. We argue that process mining techniques have the potential to improve this activity, allowing the auditor to meet the short deadlines, as well as bringing greater value to the senior management and reliability in the service provided by the audit. The goal of this paper is to discuss, through a case study, how process mining techniques can optimize and bring agility to the verification of process model compliance against the process actually performed. With this approach, it will be possible to detect errors and/or failures in activities or controls of a running process. The main contribution of this paper is to describe a simple set of steps that could be applied by auditors and experts in order to get introduced and to obtain the first insights in the process mining area.*

**Keywords:** *Process mining, Process Auditing, Conformance Checking*

## 1. Introduction

Recently, corporate fraud scandals have drawn the attention of the media. Especially after the emblematic case of the company Enron, which after problems related to accounting came

to close its doors. One consequence of this event was the establishment of the Sarbanes-Oxley (SOX) law<sup>1</sup>. This law was created in the United States to ensure compliance and to prevent the investor from withdrawing investments caused by a loss of confidence in the accounting deeds and corporate governance principles. It requires companies to restructure their processes in order to increase controls, security, and transparency in conducting business, financial management, bookkeeping, as well as information management and disclosure. SOX forces companies to adopt mechanisms that enable senior management to assess and monitor the compliance, adequacy, effectiveness, and efficiency of their business processes. With this requirement, the business process audit area has been gaining prominence in large companies.

A business process is defined as a sequence of activities ordered in a logical way that aims to produce a service or product that is valued for a particular group of customers (Dumas *et al.*, 2013). A relevant issue in operational process management is ensuring that processes are aligned with the organization's strategic objectives and follow previously defined standards and policies. Process auditing in the corporate environment intends to assess the degree of compliance of processes and their controls. This activity helps to correct failures and errors in the process chain, increasing its quality, and consequently the value of the audited company (Dias, 2006).

According to Roubtsova and Wiersma (2019), in practice, process auditing is responsible for obtaining evidence whether a process complies or not with predefined rules. Auditing generally does not require the existence of a specific business process model, since in some scenarios the process requires adaptability and flexibility. The main purpose of the audit is to discover evidence of predefined rule violations and possibly find the reasons for the violations. Audit evidence is often obtained manually by conducting interviews with users about the process they are following or by taking samples of cases performed on the system. This evidence should be compared with documented procedures. However, a complete business process is usually not specified in documented audit procedures. The auditor then focuses his or her effort on formulating audit instructions in the areas where most risks are perceived. Risk is calculated as a function, depending on the number of cases (process instances) that do not comply with the instructions, such as the financial loss associated with all non-compliant cases.

Cunha *et al.* (2010) state that, in order to achieve its objective, an auditor would need to analyze all the cases (instances) of a process. Therefore, depending on the size of the company, it is necessary to collect a large volume of information. According to the authors, this analysis is based on a technique called substantive testing, which consists of manually analyzing one hundred percent of the process transaction logs to verify their compliance both with internal and the market standards. For this reason, the cost of auditing can be very high. The auditor also performs analyses using sampling techniques. However, because these techniques do not consider the total amount of process data under analysis, non-compliant activities may go unnoticed.

---

<sup>1</sup> <https://portaldeauditoria.com.br/introducao-lei-sarbanes-oxley-sox/>

On the other hand, since Information Systems play a major role in supporting processes, a large amount of data about the execution history of these processes is stored in the form of logs. Moreover, according to Aalst *et al.* (2007), process mining is a set of techniques that aim to analyze information system event logs in order to discover a company's process models. The so-called process mining techniques also offer automated solutions for process compliance analysis. Process mining is described in the literature as a tool to support the discovering and monitoring phases in the life cycle of Business Process Management (BPM) area (Dumas *et al.*, 2013).

Recent literature highlights that many experiences have been made with process mining in order to perform a better analysis of the process flow based on historical data besides auditing processes on-the-fly and to support the work of the auditor (e.g., Corradini *et al.*, 2019; Chiu and Jans, 2019; Kogan *et al.*, 2019; Jans, 2019). Although many papers related to this topic have been published, to our best knowledge, none of them was concerned to present a detailed description of how the process mining tools could be used to support the auditor. In this paper, we argue that process mining has the potential to improve the performance of process and control analyzes, enabling the auditor to meet the short deadlines imposed by the client and to bring greater value to senior management and audit service reliability.

The goal of this paper is to discuss, through a case study, how process mining techniques can optimize and bring agility to the verification of process model compliance against the process actually performed. With this approach, it will be possible to detect errors and/or failures in activities or controls of a running process. The main contribution is to describe a simple set of steps that could be applied by auditors and experts in order to get introduced and obtain the first insights in the area.

This paper extends the results of Barboza *et al.* (2019). In the original paper, we presented the first insights on how to analyze the results from the application of those techniques for auditing. Then, we could find out the more adequate techniques and define a proper research question to support the conduction of the case study. In the current paper, we extended the review of related works; included what are the steps to be followed in order to perceive benefit from process mining techniques in the audit field; and provided an additional discussion of the contributions observed with the execution of the case study, such as (i) the application of traditional rules or guidelines of the auditing process could be automated and extended to the entire population of process cases, (ii) a process execution may not provide all insights into materiality, (iii) the importance of the process under consideration to be entirely supported by an information system.

The paper is structured in the following sections: Section 2 presents the theoretical foundation and basic concepts of the research; Section 3 describes the application of Process Mining to Internal Process Auditing; Section 4 discusses the history of related work in the area; Section 5 describes a Real-Case Study applying the suggested techniques; Section 6 discusses the results; and Section 7 concludes the paper and presents future research perspectives.

## **2. Auditing of Internal Process**

Auditing is an independent assessment and advisory activity, focused on the examination and evaluation of the adequacy, efficiency and effectiveness of control systems, as well as on the performance of a certain area of a company in relation to the attribution of responsibility and plans, goals and policies defined for it (Franco and Marra, 2000). In internal process auditing, the principle is the same, but focused on the company's processes.

### **2.1. Organization areas for auditing**

According to Neto and Solonca (2007), an internal auditing is divided into six areas within a company: government programs, administrative, accounting, financial, operational and information technology.

Government audit programs are responsible for monitoring, reviewing and evaluating the execution of government programs and projects. The administrative audit encompasses the organization's plan, procedures, guidelines, and decision support documents. The accounting audit is related to the reliability of the institution accounts. It is intended to provide assurance that operations and access to assets are carried out in accordance with appropriate authorizations. The financial audit, or audit of the accounts, analyzes the accounts, the financial condition, the legality and regularity of the operations and accounting, financial, budgetary and equity aspects. Thus, it must verify if all transactions were correctly authorized, settled, ordered, paid and recorded. Operational auditing consists of analyzing all management levels at the programming, implementation and supervision stages from the perspective of economy, efficiency, and effectiveness. In addition, the operational audit also analyzes the execution of existing decisions and verifies the extent to which the intended results have been achieved. Information technology auditing specifically checks internal controls, the environment, systems and information security, thus identifying their strengths and weaknesses.

The operational and information technology audits focus on the analysis of the business processes of the audited organization. These two areas of auditing are the focus of this paper.

### **2.2. Internal Controls**

According to Attie (2007), internal controls are divided into two aspects: accounting and administrative. Accounting controls include the organization plan and all related methods and procedures, particularly with the safeguarding of assets and the reliability of accounting records. They typically include the following controls: authorization and approval systems; separation of accounting and reporting functions from those related to operations or custody of securities; and physical controls over these values. Administrative controls, on the other hand, include the organizational plan and all methods and procedures that concern operational efficiency and the political decision made by management. They often cover statistical analyzes, time and motion studies, performance reports, and quality controls.

The research presented in this paper focuses on internal administrative controls, which are the central concern of the internal process auditing.

### 2.3. Techniques for Internal Auditing

With the growing use of information systems to support business processes, more data is stored in the transaction logs from the interaction between activities. To understand and analyze the dynamics of these process logs, Gomes (2000) explains that internal auditors use computerized investigation tools such as Audit Command Language (ACL)<sup>2</sup>, Interactive Data Exploration and Analysis (IDEA)<sup>3</sup> and Statistical Analysis System (SAS)<sup>4</sup>. These types of software require from the auditor a high degree of knowledge about the concepts involving database, audit trail and correlation of information (Gomes, 2000).

According to Oliveira (1989), auditing tests are the fundamental process by which the auditor gathers evidence. These tests can be applied to all transactions or to a suitable representative sample. Sampling is the most widely used technique in auditing, but according to Oliveira (1989), the problem faced by the auditor is how to determine the nature and extent of the verification required, i.e. how far to go, how much to investigate, and what are the variables to consider. It is important to assure that the conditions tested include the problematic ones, and thus, they can provide the correct results.

The tests are classified into compliance tests and substantive tests (Oliveira, 1989). Compliance tests determine whether certain internal control procedures established by the company's system are correctly performed; they aim to approve the credibility of such controls and not the correct recording of transactions. Substantive tests are intended to provide enough and convincing evidence about transactions, balances, and disclosures in the financial statements that provide reasonable grounds for issuing the report.

Before the beginning of the tests, the auditor should elaborate a sampling plan. Based on this document, the auditor will follow the sampling process. This plan is usually developed in thirteen steps, such as follows:

1. Define the objectives of the test.
2. Determine acceptable risk levels for the test.
3. Specify the population.
4. Calculate the amplitude of the sample.
5. Specify the sampling unit (dozen, cent, etc.).
6. Determine the sampling method to be used.
7. Select the sample.
8. Apply audit procedures to the sample.
9. Apply alternative procedures where audit procedures cannot be applied.
10. Form conclusions about each sample item.
11. Evaluate the sample results (inference).

---

<sup>2</sup> <https://techsupply.com.br/>

<sup>3</sup> <http://www.audtrack.com/>

<sup>4</sup> [https://support.sas.com/rnd/emi/APM\\_main/index.html](https://support.sas.com/rnd/emi/APM_main/index.html)

12. Review the test work.
13. Provide opinion about the final findings.

Furthermore, Oliveira (1989) explains that sampling could use two techniques, spontaneous and intentional sampling. Spontaneous sampling aims to analyze a sample without bias by the auditor. This is not a trivial technique since the auditor may end up choosing items according to their easy location or verification. Therefore she/he needs to pay attention to the following points: to form an opinion only of the populations pertinent to the samples taken; to let every item in the population have an equal or known chance of selection; to make sure that no standard population model will affect random choice of sample; and to prevent personal bias from affecting the selection of sample items.

In some cases, it is not possible to choose an unbiased or haphazard sample, due to the nature and quality of the population, or the sample which is statistically supported may not be convenient. Under these circumstances, the auditor is forced to use the intentional sampling technique, in which a subjective selection of items is made, or a restricted number of items examined. Although intentional sampling is a recognized and acceptable auditing technique, the auditor should exercise special care in delivering a projected opinion to the population to approximate the degree of mathematical confidence in the Statistical Sampling. Oliveira (1989) points out that with the use of this technique, it is very difficult to affirm that the sample collected from a given population represents exactly the population. This is because it is a sampling process and not a complete census (examination of all items in the population).

Based on the arguments presented, we claim that process mining techniques can be used to assist the audit testing step in order to minimize the complexity of the analysis in each test and increase the accuracy of these analyzes, for the reason that the tests are performed on the complete database.

### **3. Process Mining for Internal Auditing**

According to Santos (2014), business process modeling consists of the technique to represent how the process occurs in practice in an abstract way, considering intrinsic characteristics, such as resources, controls, roles and responsibilities. Process modeling is interesting for the internal process auditor as it assists in the process mapping phase of a company. These models can be categorized as descriptive and prescriptive. The descriptive model attempts to capture existing processes without being normative. According to Lopes (2015), an example of this model would be a hospital process whose goal is to react to an urgent situation; however, the flexibility to act differently from the normal flow of actions is also of utmost importance. A prescriptive model describes how the process can be performed; it is used to reinforce a particular path of activities supported by information systems (Lopes, 2015).

The process discovery is the process cycle phase of the collection of information about an existing process and its representation in a model. In practice, gathering information is often time consuming and complicated (Dumas, *et al.*, 2013) because business processes are dynamic and complex, requiring more sophisticated techniques than simply modeling. Thus, the goal of process mining is to discover, monitor and improve real processes to extract

information and knowledge from event logs of a company's information systems. Process mining techniques can discover process models from an event log; verifies compliance between a process model and its day-to-day execution; monitors execution deviations of a process; repairs models; makes predictions and recommendations based on the history of events (van der Aalst, 2016). Examples of tools that enable the application of process mining solutions are DISCO and PROM. Disco<sup>5</sup> is a process mining solution that enables automatic discovery of process models directly from raw data logs. ProM<sup>6</sup> is a framework that supports a wide variety of process mining techniques available by plug-ins.

A standard has been proposed in order to make process mining tools work with these logs. The eXtensible Event Stream (XES)<sup>7</sup> standard defines a grammar for a tag-based language. Its goal is to provide information system designers with a unified and extensible format for capturing system behaviors through event logs and event streams defined in the XES standard. In addition, it includes a basic collection of so-called XES extension prototypes that provide semantics for certain attributes normally recorded in the event log or flow.

Jans *et al.* (2012) state that logs are potentially valuable for auditing not only because it provides the auditor with more data to analyze, but also because additional data can be logged regardless of the action of a person whose behavior is the subject of the audit. By accessing an event log, the auditor also has access to “metadata” about the circumstances under which the users made these inputs. This metadata encompasses much more than simple transaction timestamps, because using tracking data, logging allows the auditor to reproduce the history of any transaction.

Thus, the auditor is able to trace the relationship between a particular input and its author for all recorded transactions, in addition to the paths in which processes are being executed in practice (Jans *et al.*, 2012). For example, through process mining, the auditor has the ability to compare whether processes, such as “purchases payable” were actually conducted correctly, or to determine how the dismissal of a key employee impacted the segregation of taxes controls. Such business process visibility is very complicated to analyze from an isolated transaction. However, it becomes viable when transactional data is supplemented by the metadata and history contained in the event logs and made visible by mining techniques.

### 3.1. Conformance Check

The conformance checking technique in process mining refers to the detection of inconsistencies between a process model and the corresponding actual execution log, as well as quantifying and qualifying the level of compliance by statistical metrics such as fitness (Rozinat, 2010). The class of process mining algorithms responsible for process compliance analysis is called compliance checking techniques, as these algorithms check for conformity between any existing model and what really runs daily (Rozinat, 2010).

<sup>5</sup> <https://fluxicon.com/disco/> último acesso: 10/06/2018

<sup>6</sup> <http://www.promtools.org/doku.php> último acesso: 11/06/2018

<sup>7</sup> <http://www.xes-standard.org/start> último acesso: 24/06/2018

Fitness is a metric that quantifies how well the behavior observed in the log fits into the discovered or modeled process model (Lopes, 2015). Perfect fitness indicates that all test cases were executed with decision paths identical to the model process. Therefore, if fitness is not perfect, we can conclude that path deviations occurred in the execution of one or more test cases (Accorsi and Stocker, 2012).

Fitness can be calculated from a compliance check technique called replay, which uses the event log and process model to execute it. Replay forces the test log to be run through the process model, even if the model and test log are not compatible, so activities that are missing or left over during execution are accounted for and entered into the fitness calculation to obtain a compliance index (Lopes, 2015).

### 3.2. Petri Nets

Petri net is a directed bipartite graph, in which the nodes represent transitions (i.e. events that may occur, represented by bars) and places (i.e. conditions, represented by circles). The directed arcs describe which places are pre- and/or postconditions for which transitions (signified by arrows). In graphical notation, the place is represented by a circle, and the transition is represented by a square.

Like standards such as UML activity diagrams, Business Process Model and Notation and event-driven process chains, Petri nets offer a graphical notation for stepwise processes that include choice, iteration, and concurrent execution. The concatenation between places and transitions is done by a structure called arc. An arc links only one position in a transaction and vice versa. Therefore, an arc between two positions or two transitions is impossible (van der Aalst and Stahl, 2011).

Different from the standards mentioned, Petri nets have an exact mathematical definition of their execution semantics, with a well-developed mathematical theory for process analysis. Some process mining algorithms generate outputs in the form of Petri nets as well as some of them uses Petri nets as input. Thus, the process models shown in this paper are Petri nets.

## 4. Related Works

Aalst *et al.* (2012) proposed an audit framework that employs process discovery and compliance verification and discussed some challenges of applying process mining to auditing. Some researchers investigate this approach in different domain areas.

Jans *et al.* (2012) emphasized that the use of process mining techniques in Internal Fraud Risk (IFR2) software can add value by reducing the risk of internal fraud in companies. To verify if a process corresponds to a reference designed model, Jans *et al.* (2012) employ process discovery techniques to reconstruct a process model that shows the actual behavior of the process. The authors also analyzed the relationships between process activities and the people involved, i.e. the compliance of segregation of work restrictions. More recently, Jans (2019) discusses the insights how event logs are currently structured and the consequences of this structure for the analytical procedure in the context of auditing. The author argues that

different preparation steps could lead to varying analytical procedures and as a result to other audit evidence. Furthermore, Jans and Hosseinpour (2019) proposes a framework that aims at combining data mining and process mining adding the auditor as a human expert to deal with the typical alarm flood. Our paper aims to describe the practical steps for using an option of such log structure.

Accorsi and Stocker (2012) analyzed a process in the area of bank credit. A synthetic log was generated through a simulation of the process, parameterized by business rules such as loan acceptance rate, number of individuals involved in the process, and cases in which the process needs to be aborted. As an example of a deviation case in the synthetic log, we may cite the delays in more than 7 days for loan approval. The log generated was used for the process compliance verification analysis in ProM with the replay technique. Also, in the financial domain, in a similar domain, Werner (2019) presents an approach to visualize process mining results specifically for financial audits in an aggregate manner as a materiality map.

Riz *et al.* (2016) propose the adaptation of process mining compliance analysis techniques to healthcare to assist in the discovery and improvement of the flow of activities. The authors applied the Conformance Checker algorithm to obtain results on a real scenario.

The works from Accorsi and Stocker (2012), Werner (2019) and Riz *et al.* (2016) analyze issues related to specific domains. Our paper, although contextualized in case study, describes steps that could be followed in any domain.

Nevertheless, the existing frameworks for process mining auditing assume that the business process model is somehow built for auditing, and furthermore they assume that proper audit instructions are available. For example, the proposal presented by Sadiq *et al.* (2007) suggests finding a business process model in the log and verifying the compliance of the recorded process with the audit rules. However, the direct application of this framework is problematic because real-world process instances in the event log may not contain the activities and other type of data mentioned in the audit rules.

Barnawi *et al.* (2013) designed a framework for mining and comparing *de jure* and *de facto* business process models. The BPMaaS (BPM as a Service) framework for runtime compliance verification assumes that business process management practice begins with the expert defining and modeling business process requirements using BPMN. This template is used to filter only cases of the audited logs process. Another role is the compliance specialist who is responsible for formulating audit rules using compliance standards. This proposal would require every company to have a compliance specialist in its staff. The traditional role of an auditor, who formulates audit rules as principles, and may not be familiar with compliance standards and business process details, is not present in the BP-MaaS framework.

Bukhsh and Weigand (2017) explored a remittance monitoring scenario to evaluate the Smart Auditing Framework, which integrates process mining techniques and business process ontologies. The authors used simulated data to identify the potential audit / monitoring capabilities of ProM plug-ins. The initial assessment revealed that rule-based

auditing is successful and concluded that there is a limitation to automatic rule conversion in the LTL-checker plug-in.

Roubtsova and Wiersma (2018) argue that the current process mining frameworks are not oriented on real auditors; instead they tend to replace the auditors with compliance experts. Thus, they do not address the steps of refinement of audit statements and do not provide support to the analysis of the process instances on correspondence to audit goals and policies. The authors proposed an extension of the audit frameworks with process mining to include a participatory workshop. Our work described a set of practical steps illustrated within a case to use process mining in auditing. The work of Roubtsova and Wiersma (2018) could be a complement to those steps.

In the scenario of using blockchain transactions in auditing, Corradini *et al.* (2019) propose adopting process mining techniques to evaluate smart contracts and to support the work of the auditor. The authors argue that the models obtained can be used to analyze if the solution works as required. This paper deals with a specific context of application. The systematization that we describe in our paper is generic, but it could be integrated in Corradini *et al.* (2019) proposal.

Chiu and Jans (2019) affirm that process mining can produce a new type of audit evidence. The authors concluded from the results of a case study that process mining could help auditors in identifying audit-relevant issues such as non-standard variants, weekend activities, and personnel who are involved in multiple violations. This paper does not intend to systematize the activities of process mining applied to auditing like ours; however, the results could be used to support further research.

Our research is aligned with the work of Jans *et al.* (2012) and Riz *et al.* (2016) in which existing techniques are applied in different domains, showing their benefits and limitations. Different from those authors, we present a step-by-step procedure in order to make process mining techniques clear and usable for a first understanding of an auditor.

## **5. Case Study: an application scenario**

The goal of this research was to analyze the benefits of process mining in auditing in a real application scenario. Thus, we choose the Case Study methodological approach. The research question formulated to guide the research is: “What are the steps to be followed in order to perceive benefits from applying process mining techniques in the audit field?”. We aimed at understanding and analyzing in depth what would be the procedure of using a log of events to find inconsistencies and possible non-conformities. Therefore, we selected one unit of analysis, a well-known real case available in the literature. The choice for this case was based on a number of requirements: a real case in which issues related to conformance would be critical to the business (subject to assessment from the government); the log with all the necessary attributes is available and it has been used in diverse research contexts, which attests its credibility; and, the log is not ready for using, i.e., all the steps of preprocessing should be done.

The scenario chosen (the case) was the event log of a Business Process Intelligence Challenge from the information system of the participating company (BPIC<sup>8</sup> 2018) (van Dongen and Borchert (2018)). The BPIC 2018 Challenge was set in the context of the European Union (EU) agricultural budget dynamics. The European Union spends a large fraction of its budget on the common agricultural policy (CAP). Among these expenditures there are the direct payments, which are primarily intended to provide a basic income for farmers decoupled from production. The rest of CAP's budget is spent on market and rural development related expenses. The processes governing the distribution of these funds are subject to complex regulations captured in EU and national law. For this reason, member states are required to operate an integrated administration and control system (IACS), which includes information systems to support grant distribution processes.

The process analyzed covers the handling of EU direct payment claims to German farmers from the European Agricultural Guarantee Fund. The process repeats each year with minor changes due to changes in EU regulations. About 10% of cases are subject to more stringent on-site inspection. A log of this process, called the payment application, was made available with pre-processing of the data, so no action was needed to identify incorrect data that could impact the results of the analysis performed.

However, the only type of log provided by BPIC was the company day-to-day log related to its bill payment process. Thus, to verify compliance between the model and the actual execution of the process, it was necessary to create a model from the available log to serve as the basis for this compliance analysis, which will be in fact the reality of an auditor in real cases. So, the following steps have been performed: identify log relevant attributes; discover log activities; analyze the dynamics between activities; idealize the reference model and segregate log; apply the compliance check plug-in. These steps are detailed in the next sections.

### **5.1. Identifying log relevant attributes**

The first step performs the analysis of the log preparing it for the process mining task. The attributes used were: Case ID, attribute referring to the unique identifier of the case; Activity, attribute identifying the activity executed. Its name is given by the log name, followed by the name of the sub process and finally the name of the activity; Resource, attribute indicating who executed the activity, i.e., whether it is manual, automatic or from a coded variable for privacy reasons; Timestamp, attribute indicating when the activity was executed. The other fields contained data irrelevant to this case study, so they were excluded from the log.

### **5.2. Discovering log activities**

In order to analyze the data from the bill payment process log, it was necessary to import the log into the software Disco to perform the process model discovery procedure. Disco uses the Disco miner (Günther and Rozinat, 2012) as the process model discovery algorithm. This algorithm is an improved version of the Fuzzy miner (Günther and van der Aalst, 2007), which was the first mining algorithm to introduce the “map metaphor” to process mining.

<sup>8</sup> Business Process Intelligence Challenge (BPIC): <http://www.win.tue.nl/bpi/doku.php?id=2018:challenge>  
 iSys: Revista Brasileira de Sistemas de Informação (iSys: Brazilian Journal of Information Systems)  
<https://sol.sbc.org.br/journals/index.php/isys>

The fuzzy miner is a process discovery algorithm that discovers process graphs (and not Petri nets, as the traditional ones). Process graphs are especially useful to explore and get initial insights in the dataset. This version includes advanced features such as process simplification and highlighting of frequent activities and paths.

Most process mining techniques follow an interpretive approach, i.e., they try to map the behavior found in the log to process design patterns (for example, if a split node has AND or XOR semantics). The Fuzzy approach focuses on mapping the behavior found in the high-level log. Thus, creating a preliminary process model (not simplified) is simple: all classes of events found in the log are converted into nodes (activities), whose relevance is expressed by unary significance. For each observed precedence relation between event classes, a corresponding directed edge is included in the process model. This edge is described by its binary significance and by the correlation of the ordering relationship it represents. Subsequently, three transformation methods are applied to the process model, which will successively simplify specific aspects of it. The first two phases, conflict resolution and edge filtering, remove edges (that is, precedence relations) between nodes (activities), while the final phase of aggregation and abstraction removes and / or groups less significant nodes. Removing the edges of the model first is important due to the less structured nature of real-life processes and the measurement of long-term relationships. The initial model contains sorting relationships that may not correspond to valid behavior and need to be discarded (Günther and van der Aalst, 2007).

This approach is comparable to looking at a map of a country where all cities and towns are represented by identical nodes and all roads are represented in the same way. The resulting map is correct, but not suitable for a user. Therefore, the concept of a script is used as a metaphor to visualize the resulting models. Based on an analysis of the log, the importance of activities and relationships between activities is taken into account. Activities and their relationships can be grouped or removed depending on their role in the process. The names of the activities as well as the relationships between them come from the logs. One of the log attributes is the activity's name (the dataset metadata). The Fuzzy approach allows grouping activities based on the frequency of such activities in the log. Thus, certain aspects can be emphasized graphically, just as a roadmap emphasizes highways and large cities over dirt roads and small towns. Mining Fuzzy's flexible approach adaptively simplifies the mining process models. Thus, we chose DISCO because, according to Günther and Rozinat, (2012), it performs better than, for example ProM.

In DISCO, a template with all paths and activities is automatically generated when the event log is imported. Due to the complexity of this log, the first model generated did not offer much information about the bill payment process. However, it was possible to manipulate the generated model in order to study its activities and their correlations. Thus, reducing the number of alternative paths and the maximum number of event log activities to a maximum resulted in the model shown in Figure 1. We explain all the steps in Section 5.2.

From this model, we highlight the main activities performed in this process. The activities are presented with different colors which represent their frequency of execution: activities in gray are the infrequent activities, and activities in dark blue are the more frequent

activities. In addition to the colors, we notice the presence of weights in each process path, which are indicated as arrows connecting activities. The weight refers to the time spent transitioning each activity.

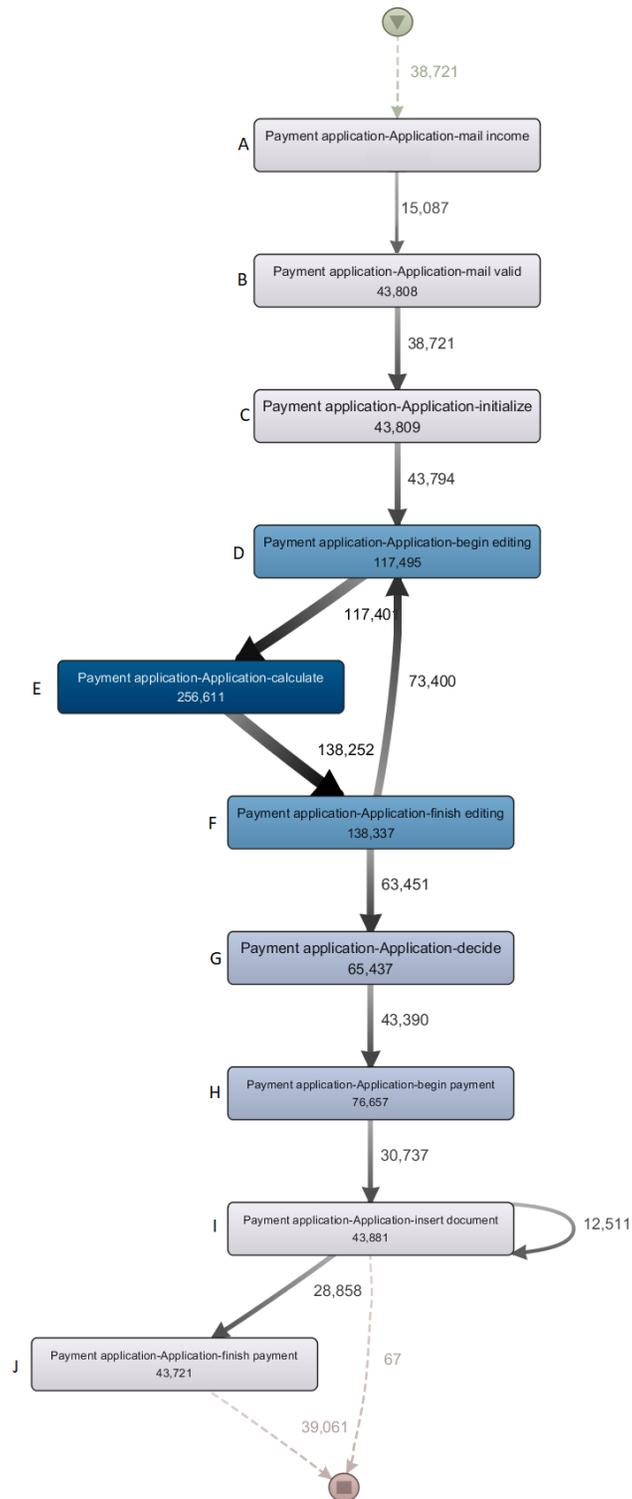
The log payment application consists of the Application, Change, Main, and Objection sub processes. The distribution of the percentages of activities present in the log among the four sub processes is presented in Table 1. The activities in this model refer to the Application subprocess, which represents 94.26% of cases, thus being the most relevant subprocess of the log.

**Table 1 – Relative frequency of the sub-processes**

Sub-process	Relative Frequency of the Activities
Application	94,26%
Change	2,93%
Main	1,54%
Objection	1,28%

### 5.3. Analyzing the dynamics of the activities

After identifying the activities contained in the bill payment process using the process model discovery algorithm Disco miner (Günther and Rozinat, 2012), the next step was to analyze the dynamics of interactions between these activities so that it was possible to better understand the role of each activity and to know what the inputs and results of each of them are. The main flow of the bill payment process (Figure 1) can be summarized as follows: A. mail income (email payment request); B. mail valid (email validation); C. initialize (payment analysis initialization); D. begin editing; E. calculate; F. finish editing; G. decide (decision); H. begin payment; I. insert document; J. finish payment.



**Figure 1 – A simplification of the process model**

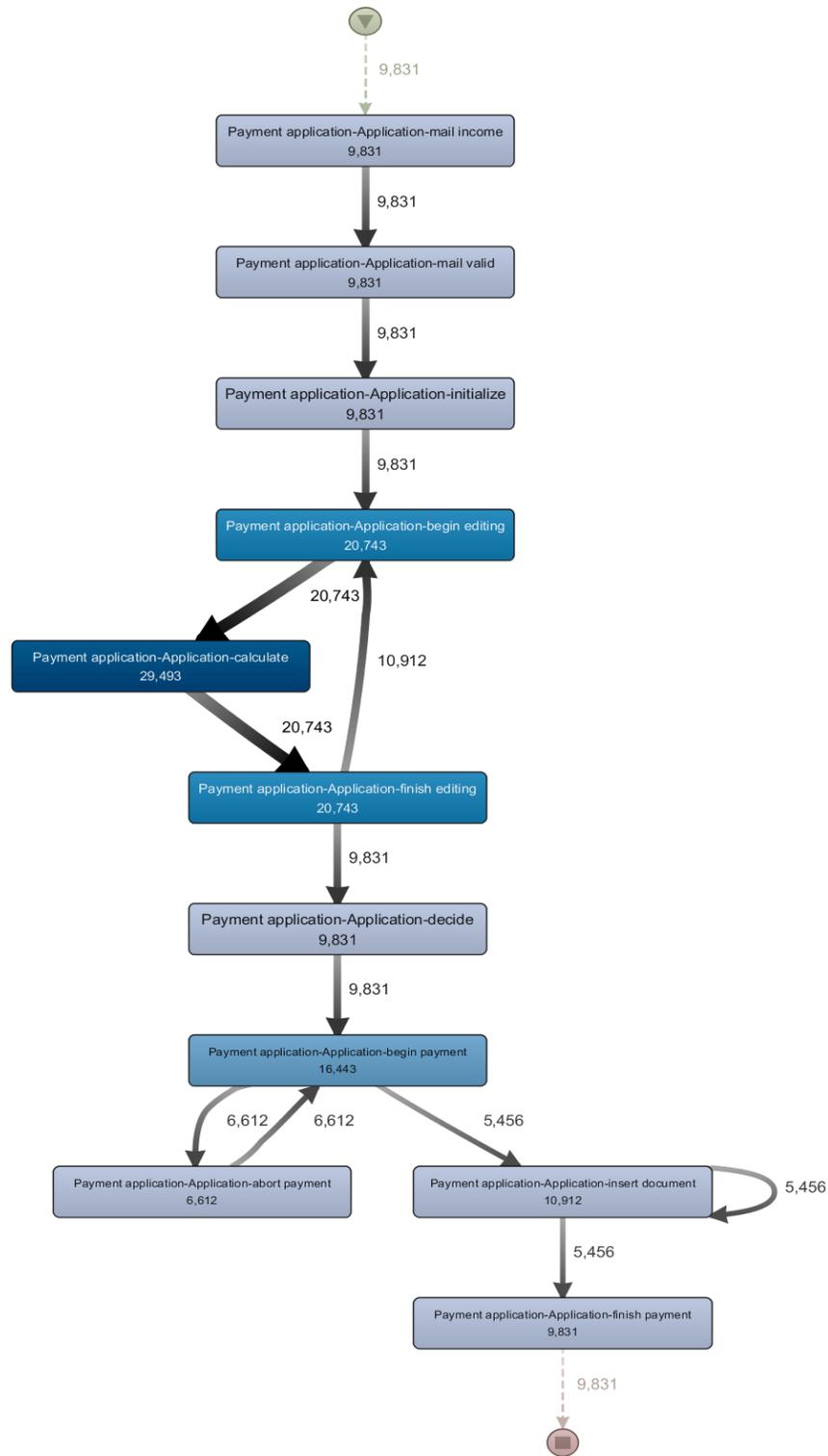
The steps that comprise this process are explained as follows. To start the process, it is necessary to receive a payment request by email, so the first activity is receiving this request, called ‘mail income’ (A). The next step is the confirmation of this request: ‘mail valid’ (B). When the order is confirmed, the ‘initialize’ activity is performed (C), when the payment process actually starts. Then, the ‘begin editing’ activity (D) is carried out. Its purpose is to start editing the input data for formatting that will allow the ‘calculate’ activity (E) to be sequenced, which automatically calculates the value to be received by the German farmer.

As the editing step is performed manually, typing errors may occur and consequently the calculation will be incorrect. To avoid this, after the payment calculation activity a data confirmation is performed by the ‘finish editing’ activity (F). If something is inconsistent at this stage, it will be necessary to eliminate its failures by redirecting the sequence of activities back to the begin editing activity (D). If the calculation corresponds to the predicted value, the next activity will be the ‘decide’ (G). This activity is responsible for deciding how the amount due will be paid, for example the number of installments. Immediately, the payment process is started by the ‘begin payment’ activity (H) which is followed by the ‘insert document’ activity (I).

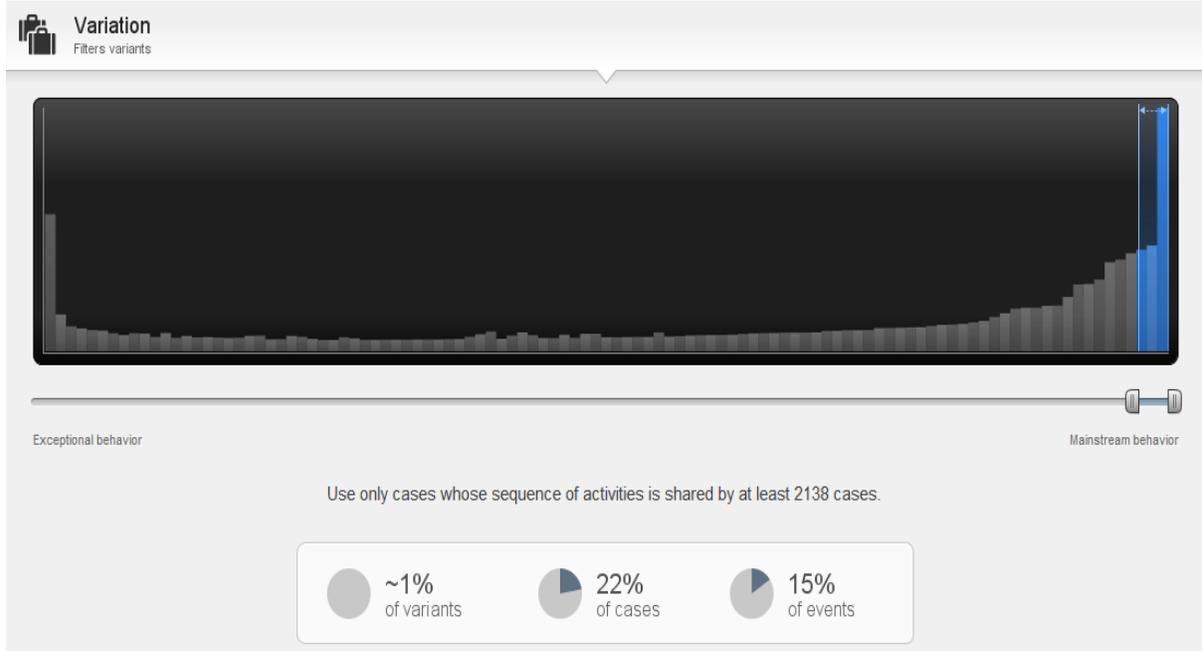
Nevertheless, there are annual changes to the EU regulation, and this can also lead to changes in the pattern of documents to be attached. In this sense, the insertion of documents can occur many times either by the motivation described or by another human error. However, after such insertions there are two paths that can be followed: the termination of this process by some external interference, which is a less likely path, or the execution of the last ‘finish payment’ activity (J), which is responsible for properly completing the payment process. before the process ends.

#### **5.4. Idealizing the reference model and segregating log**

After understanding the dynamics between the activities of the bill payment process and the removal of irrelevant data from the log, we proceed to the idealization stage of the log model to later segregate the log into the two logs needed to perform the experiment: log related to the bill payment process template and the actual process execution log. In the idealization of the log model, the main activities illustrated in Figure 1 were used, but with one modification: an alternative path has been included after the ‘begin payment’ activity has been performed. Thus, after the start of bill payment analysis, in addition to the paths mentioned above, the process can be aborted in the ‘abort payment’ activity as shown in Figure 2, which is the final model used as a reference. With the reference process model already devised, the next step was the segregation of the original log. For this, we used the “Variation” filter of DISCO only in those cases that characterized the idealized model. This filter, depicted in Figure 3, removes parts of a log. In this case, about 20% of cases were removed from the model log and exported to XES format. After removing the model log, the remaining cases from the original log, corresponding to the rest of the cases added to the 80% of the cases referring to the model, were exported to XES format, representing the real execution log of the company, which we will call the test log.



**Figure 2 – Reference Model**



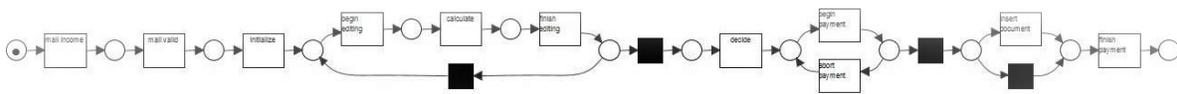
**Figure 3 – “Variation” Filter considering 20% of the cases in the model log**

### 5.5. Applying conformance checking plug-in

After preparing the model log and the test log, we applied a Conformance Checking algorithm. The model log and the test log are used as inputs to run the ProM conformance check plug-in. The first step was to import the model and test logs into ProM in the format “ProM log files (XESLite - MapDB - slow random access).”

The plug-in used for conformance checking was the “Replay a Log on Petri Net for Conformance Analysis”. This plug-in is based on the replay technique which aims to compare a test log with a model. A Petri Net is generated with various information about how a process is running against the reference model, such as: fitness between log and model, quantification and evidence of alternative paths to the model, resource overload points and individual performance analysis of the modeled activities.

The “Replay a Log on Petri Net for Conformance Analysis” plug-in receives as input the log to be tested in XES format and the model to serve as the basis for comparison analysis in Petri Net format. To generate the Petri Net model, the “Mine Petri net with Inductive Miner” plug-in was used, with the model log as input. When running the above plug-in, a settings box is displayed allowing the user to adjust some parameters in order to avoid noise. As we needed perfect fitness, the noise was set to zero to generate the Petri Net. Figure 4 shows the resulting Petri Net representation of the process.



**Figure 4 – Petri net of bill payment process – Available at :  
<https://tinyurl.com/iSysFigure4>**

The Petri Net in Figure 4 is represented slightly different from the original notation. In this figure, Rectangles represent process activities, circles represent transitions between activities, and black rectangles represent more than one possibility of transition between activities. For example, a loop or a path decision to make. With the Petri Net model generated, we can run the conformance check plug-in by inputting the test log in XES and the previously produced Petri Net (based on the model log) as input.

The initial configuration of the conformance check plug-in shows all activities of the reference process and the corresponding test log activities. If the test log has activities that have not been mapped to the model, these are listed, giving the auditor the option to return to the previous step to map them according to the model. This means that the test log in question has additional activities that are not present in the model. These activities represent activities that do not conform to the model, so it was not necessary to map any additional activities.

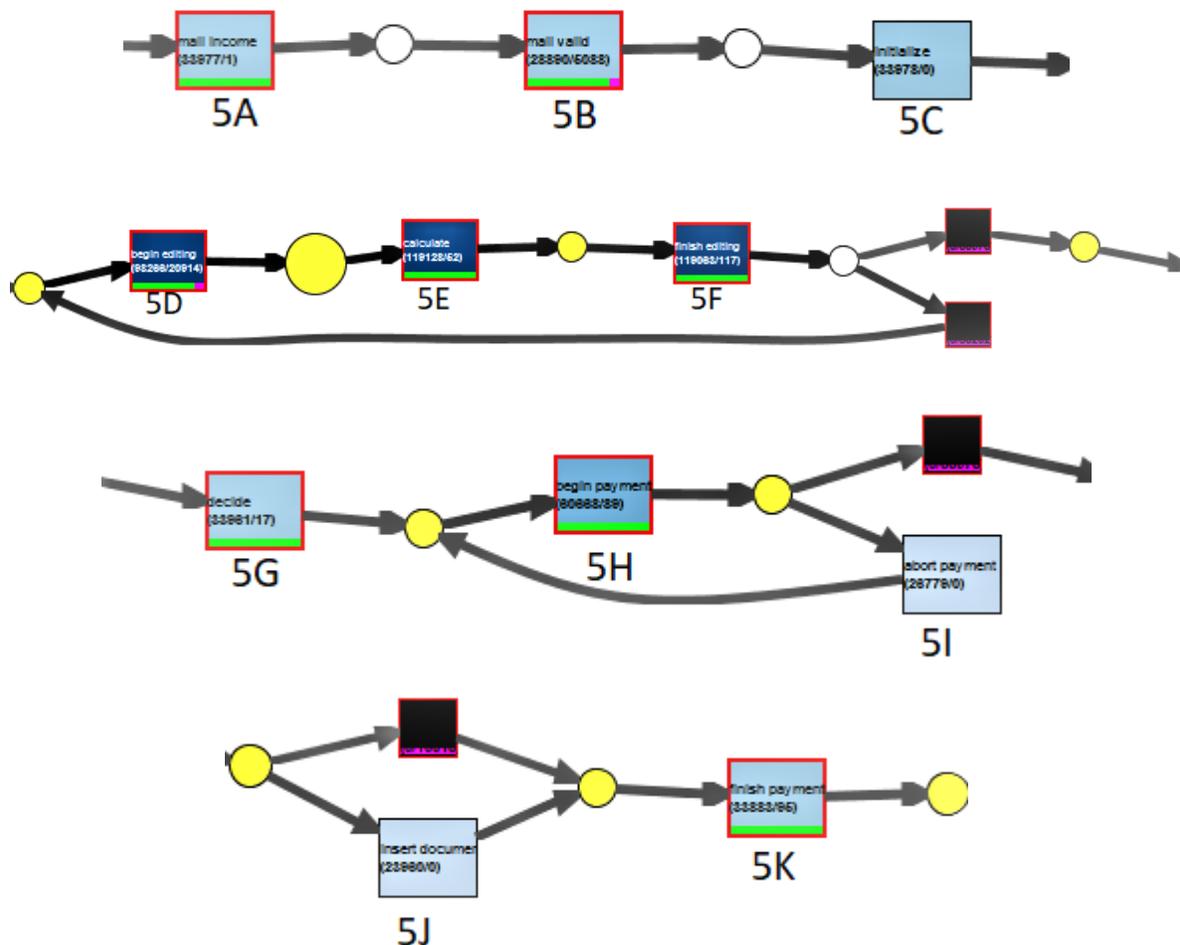
In the plug-in, the option “Measuring fitness” was selected and the option regarding the penalty of improperly completed routes was selected. After these settings, ProM allows the auditor to adjust the cost of each movement between activities. In this initial analysis, no different costs were added, as the purpose was to evaluate the pure fitness between the test log and the model.

The output of the plug-in execution is a Petri Net with additional information, such as specific colors and shapes. The resulting Petri Net is shown in Figure 5, which was divided into four parts for better viewing. The darkest activities are those performed most often. The clearest activities, on the other hand, are activities performed a smaller number of times. Thus, we observe in Figure 5 that activities related to payment editing occur more often than others. This may indicate a problem receiving payment information, or rework for failure to edit a payment.

The green bar at the bottom of the activities indicates the frequency of cases when the log performed the task in question in synchronism with the model (the darker the more frequent). From Figure 5, we can conclude that the frequency of cases conforming to the model is quite high. The lilac bar at the bottom of the activities indicates the frequency of cases with divergent executions compared to the model.

In Figure 5, the number of cases in which divergences occur is so small that the lilac bar is not seen. Activities with a red border (5A, 5B, 5D, 5E, 5F, 5G, 5H, 5K) signal that the test log did not perform them synchronously or correctly with the reference model.

Another way to highlight this information is by looking inside the activities. We can see that in more than 98,000 cases, this activity is aligned with the execution of the model. By contrast, in more than 20,000 cases, this activity has not even been performed. That is, in more than 20,000 cases the editing activity was not performed.



**Figure 5 – Output of plug-in “Replay a Log on Petri Net for Conformance Analysis” – available at: <https://tinyurl.com/iSysFigure5>**

White circles indicate paths that were followed in accordance with the model. While yellow circles indicate occurrences of movements outside the model, i.e., nonconforming. In Figure 6, after the execution of activity *decide*, the reference model allows only a transition to the *begin payment* activity. However, there are cases in the test log where after execution of the activity *decide*, a transition to other activities that are not in the model have occurred. These trajectory changes and their respective frequencies can be seen in the compliance plug-in's auxiliary window, as shown in Figure 7.

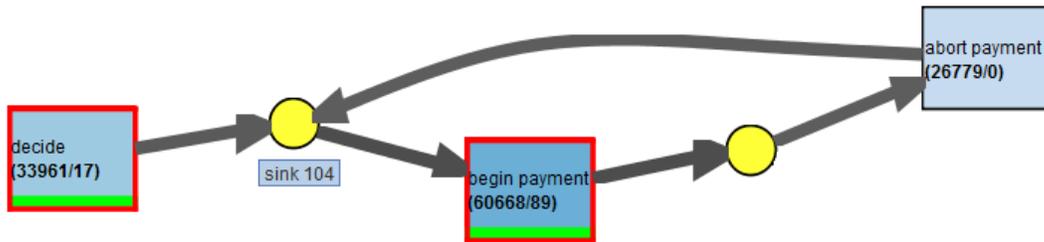


Figure 6–Example of a process instance executing activity *decide* and *begin payment*

Elements Statistics

Selected elements\* sink 104

Marking	Move on Log	# (Freq)	#Traces
[sink 104]	approve+complete	106	106
	begin admissibility check+complete	185	120
	begin editing from refused+complete	24	18
	begin editing+complete	241	231
	begin payment+complete	240	239
	calculate+complete	695	281
	check admissibility+complete	183	120
	decide+complete	402	280

Figure 7 – Flows after activity *decide* has been executed

Larger circles indicate more frequent alternate movements. In Figure 8, we can see the presence of a larger circle than the others after the *begin editing* activity. This circle indicates that, in relation to the average of all alternative paths executed in the log, there was a higher frequency of alternative paths exactly after the *begin editing* activity, as evidenced also in Figure 9.

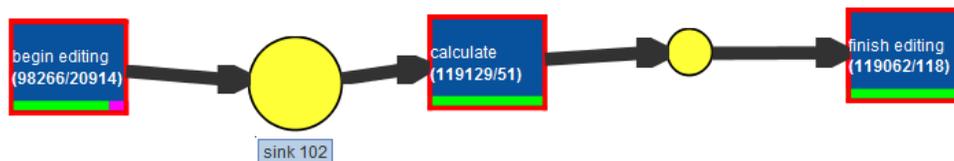


Figure 8 – Flow after activity *begin editing*. Big yellow circle indicates high number of instances passing through this path.

▼ Elements Statistics			
Selected elements* sink 102			
Marking	Move on Log	# (Freq)	#Traces
	begin payment+co...	56	34
	calculate+complete	110597	30103
	check admissibilit...	43	23
	decide+complete	20891	20564
	finish editing+com...	77	63
	finish payment+co...	25	25
	initialize+complete	28	28
	insert document+...	350	212
	refuse+complete	18	18
	remove document...	152	130
	revoke decision+c...	19839	19143

Figure 9 – Statistics after execution of activity begin editing

Black rectangles indicate path decision points to follow, after an activity has been executed, which can be a loop in one or more activities, a parallel activity execution, or as shown in Figure 10, a XOR decision.

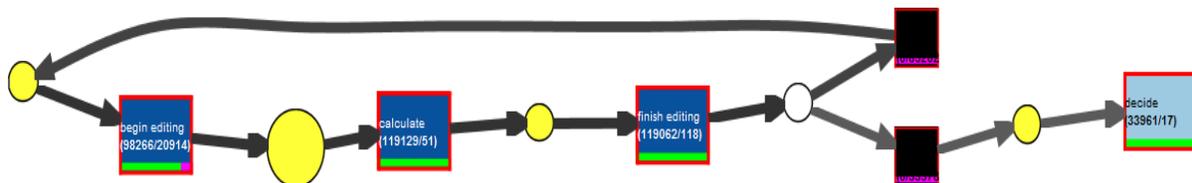


Figure 10 –Decisions are represented by black rectangles

## 5.6. Comparison of Results

In the next step of this case study, the executions of the “Replay a Log on Petri Net for Conformance Analysis” plug-in were compared considering variations on the cost adjustment of the activity. The unfiltered execution of the plug-in was taken as a reference and the metrics were compared based on changes in the cost of the activities in the “Move on Model Cost” and “Move on Log Cost” parameters.

### A. “Move on Model Cost” Filtering

The replay technique aims to analyze the log activities execution flow in relation to the model sequentially, based on the activities queuing position. The reading of the activities, which is called “movement”, occurs based on the position of the activities. For example, in a model there are activities A, B, C, D and E, performed in the following sequence: ABCDE, but when analyzing the test log, we note that the following sequence was performed: ACDE. This means that a movement occurred only in the model because activity B was not

performed in the test log, so it was not possible to read, i.e. move, the activity related to the test log.

The parameter “Move on Model Cost” represents a filter that allows the auditor to establish a cost when there is movement only in one activity in the model, consequently crediting a penalty to the test log. In other words, a penalty is applied when in the eventual absence of a particular log activity in relation to the analyzed model (van der Aalst. *et al.*, 2012). Therefore, we applied the “Move on Model Cost” filter. In this example, the log would be penalized in relation to activity B. The weight of this penalty, in turn, would influence the fitness result of the log.

Regarding the analysis of the bill payment process, a greater weight was parameterized for activities that were considered important to be performed. Based on this information, the begin payment, calculate, and finish payment activities were weighted 5 on the “Move on Model Cost” filter, while the others were weighted 1. By running the plug-in, the results shown in Table 2 were obtained. Log fitness (Trace Fitness) increased over unfiltered performance. This occurred because the begin payment and calculate activities, whose importance was defined with higher weight, have a higher frequency of execution than the others. Thus, the penalty for the lack of these activities was not significant.

**Table 2 – Comparative of the execution with and without the “Move on Model Cost”**

Property	Value (without filter)	Value (with filter)
Raw Fitness Cost	7.19	7.21
Num. States	46.46	45.3
Mov-log Fitness	0.75	0.75
Trace Fitness	0.79	0.84
Max Fitness Cost	33.44	45.44
Max Move-Log Cost	24.44	24.44
Move-Model Fitness	0.95	0.98

The “Max Fitness Cost” variable represents the worst-case scenario where only log or model movement occurs, never both at the same time. This variable equals the total cost of movement by traversing the entire log without ever moving in conjunction with the model. In Table 2, we can observe that the maximum cost increased after the weight was added to the “Move on Model Cost” filter. This means that if the log performed only non-model activities, the total cost would be higher than the cost of running without model filters. Moreover, the rest of measures (such as Raw Fitness Cost and Move-Model Fitness) did not vary significantly as expected.

The results in Table 2 confirm that auditors could use the filters provided by this tool in order to understand better the behavior of the process regarding one (or more) specific activity. Therefore, she could decide on the observation of some auditing statement more precisely or even make it more flexible if possible.

## B. “Move on Log Cost” Filtering

The “Move on Log Cost” field is a filter that allows the auditor to establish a cost for a move that occurs only in the test log. In this filter, this movement is subject to penalty. Regarding the bill payment process, we have set a cost of 5 for all activities that are not part of running the model. Our goal was to penalize the execution of activities of the log that are not present in the model. The results obtained are shown in Table 3.

**Table 3 – Comparative of the execution with and without “Move on Log Cost”**

Property	Value (without filter)	Value (with filter)
Raw Fitness Cost	7.19	15.91
Num. States	46.46	43.05
Mov-log Fitness	0.75	0.64
Trace Fitness	0.79	0.63
Max Fitness Cost	33.44	40.37
Max Move-Log Cost	24.44	28.37
Move-Model Fitness	0.95	0.79

According to Table 3, the log fitness represented by the “Trace Fitness” decreased when compared to the unfiltered running fitness. Based on (van der Aalst *et al.*, 2012), we can infer that the log is performing a significant amount of non-model activity. Thus, the penalty for performing these activities weighed heavily on the calculation of fitness.

The “Max Fitness Cost” variable represents the worst-case scenario where only log or model movement occurs, never both at the same time. This variable equals the total cost of movement by traversing the entire log without ever moving in conjunction with the model. In Table 3, we can observe that the maximum cost increased after adding the weight on the “Move on Model Cost” filter. This means that if the log performed only activities that are not part of the model, the total cost would be higher than the cost of running without the model filters.

## 6. Discussion

The approach currently used in an auditor job can be greatly enhanced by the use of process mining techniques. The application of rules or guidelines could be: (1) automated and (2) extended to the entire population of process cases, rather than a sample, resulting in a transparent process overview. However, to replace part of the manual auditing method with process mining, some current limitations need to be addressed.

A process execution alone may not provide all insights into materiality. Therefore, additional information needs to be added to the model, such as how many transactions followed a certain path, how much value was created following this path, how many people were involved, does the path cover more than one financial reserve period? As such, questions that need to be answered in the search for a process translation into materiality are among others: “When does a certain divergent process execution require further examination to exclude a material misstatement?”; “How can a process deviation be quantified in terms of risk?”; “Is there a certain threshold of cases that follow a particular execution of the process

to consider it material or a threshold of the amount affected?” The materiality is an important audit issue and, given its delicate assessment, this is unlikely to be answered very easily. Probably, this problem, in a process mining approach, as in the current approach described in this case study, requires auditor knowledge and cannot be completely replaced by algorithms.

Moreover, to replace the full set of manual guidelines, it is important that the process under consideration is fully incorporated into an information system. The initial transaction, all subsequent transactions, and the final financial report transaction must all be captured by the information system. Otherwise, automated process mining cannot mine the entire process, but only the part that is supported by an information system. This restriction requires a certain level of maturity of the organization before process mining can be applied in an audit context. If only part of a process can be extracted, no assurance can be given about the process compliance, and its reporting results. The consequence of this limitation is that full integration cannot be achieved for all audits. However, as the digitization of the world is growing faster, more organizations and processes will be suitable for process mining in the future. If this trend continues, as assumed, the possibilities for application of process mining will increase day by day.

This paper advances the literature on process mining applied to the domain of auditing by distinguishing the technical tasks that should be understood and performed by an auditor illustrated in a case study. The rigor of a case study method was able to show how to choose the tools, reach and interpret the results.

## 7. Conclusion

With the increased storage of large amounts of events, possibilities open up to apply process mining in various contexts. Therefore, it is essential to establish a well-defined focus on an application scenario, such as auditing. The independent auditor provides 'confidence' to all shareholders related to the audited organization.

This paper discusses how process mining techniques can improve the quality of process analysis and controls performed by the process auditor, bringing accurate results from a relatively simple analysis with graphical features. It has been shown how to graphically manipulate event logs from an information system in order to discover process models and compare them with the actual execution log of these processes, with analysis such as fitness, quantification and evidence of alternative model paths, resource overload points, and individual performance analysis of modeled activities.

We identified some threats to validity in this research. First, the threat to conclusion validity (or limitations of the case study) is that only some statistical information has been discussed. Other metrics could also be analyzed, allowing conclusions to be drawn, which are not only noted with the graphical view of the resulting Petri Net. The “simplicity”, “precision” and “generalization” metrics could help to identify the exact log activities and/or paths that may cause some kind of problem. Thus, it would be possible to explain why the variation of fitness by applying different weights to the filters that influence the generation of Petri Net enriched with information from the replay technique.

Second, the threat to generation validity is that only one case study was used, and it was exemplified with one set of tools. Other cases would have the potential to generalize the conduction of the steps proposed. There is a great number of process mining tools and frameworks that could also be explored in those steps followed.

Future work is to deepen a study of the steps of the data selection phase to be used in the elaboration of a process reference model to avoid creating a model with activities and / or paths that may represent errors in the execution of this process. This will have no impact on the results of the conformance check analysis. This concern occurs, therefore, when using the replay technique having as input a test log that has these same activities and / or erroneous paths, it would not be possible to signal such errors, since they are part of the reference model. Another interesting point to be examined in future work is the relation of process mining to data science, which is the object of van der Aalst (2016), but in the specific context of auditing. We intend also to explore the process optimization in the cycle of auditing analysis.

The main contribution of this research was to discuss how process mining tools and techniques can be used in the process audit task. From the study with a real process, not only the results already presented in the literature in similar cases were extended, but also the practical side of the application of such techniques.

## Acknowledgments

The second author received a grant from Österreichische Akademie der Wissenschaften.

## References

- Barboza, T.M., Santoro, F.M., Revoredo, K.C., Costa, R.M.E.M. (2019) Um Estudo de Caso da Mineração de Processos em Auditoria. SBSI'19: Proceedings of the XV Brazilian Symposium on Information Systems, article no. 32, pp. 1–8.
- van der Aalst, W.M.P., van Hee, K.M., van der Werf, J.M., Verdonk, M. (2010) Auditing 2.0: using process mining to support tomorrow's auditor. *Computer* 43(3), 90–93 (2010).
- van der Aalst, W.M.P, Adriansyah, A., van Dongen, B.F. (2012) Replaying history on process models for conformance checking and performance analysis. *WIREs Data Mining and Knowledge Discovery*, 2(2), 182-192.
- van der Aalst. (2016) *Process Mining: Data Science in Action*. Springer-Verlag Berlin Heidelberg.
- van der Aalst., W.M.P, Reijers, H.A., Weijters, A.J.M.M., van Dongen, B.F., Medeiros, A.K.A.M., Verbeek, H.M.W. (2007) Business process mining: An industrial application. *Information Systems*, 32 (5), pp. 713-732.
- van Dongen, B.F., Borchert, F. (2018) BPI Challenge 2018. Eindhoven University of Technology. Dataset: <https://doi.org/10.4121/uuid:3301445f-95e8-4ff0-98a4-901f1f204972>.
- Stahl, C., van der Aalst, W.M.P. (2011) *Modeling Business Processes: A Petri Net-Oriented Approach*. MIT Press (MA).

- Accorsi, R., Stocker, T. (2012) On the Exploitation of Process Mining for Security Audits: The Conformance Checking Case. *ACM Symposium on Applied Computing*, Pages 1709-1716.
- Attie, W. (2007) *Auditoria*. 2. ed. São Paulo: Atlas.
- Barnawi, A., Awad, A., Elgammal, A., Elshawi, R., Almalaise, A., Sakr, S. (2016) An antipattern-based runtime business process compliance monitoring framework. *Framework* 7(2), 551–572.
- Bukhsh, F.A., Weigand, H., (2017) Compliance Checking of Shipment Request by Utilizing Process Mining Concepts: An Evaluation of Smart Auditing Framework, *2017 International Conference on Frontiers of Information Technology (FIT)*, Islamabad, pp. 235-240.
- Corradini F., Marcantoni F., Morichetta A., Polini A., Re B., Sampaolo M. (2019) Enabling Auditing of Smart Contracts Through Process Mining. In: ter Beek M., Fantechi A., Semini L. (eds) *From Software Engineering to Formal Methods and Tools, and Back*. Lecture Notes in Computer Science, vol 11865. Springer, Cham.
- Chiu, T, Jans, M. (2019) Process Mining of Event Logs: A Case Study Evaluating Internal Control Effectiveness. *Accounting Horizons*: September 2019, Vol. 33, No. 3, pp. 141-156.
- Cunha, P.R., Klann, R.C., Rengel, S., Scarpin, J. (2010) Procedimentos de auditoria aplicados pelas empresas de auditoria independente de Santa Catarina em entidades do Terceiro Setor. *Revista de Contabilidade e Organizações*, 4(10), 65-85.
- Dias, S.V.S. (2006) *Auditoria de processos organizacionais*. 2. ed. Rio de Janeiro: Atlas, v. 3000. 146p.
- Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A. (2013) *Fundamentals of Business Process Management*. Berlin: Springer Heidelberg.
- Franco, H., Marra, E. (2000) *Auditoria contábil*. 3. ed. São Paulo: Atlas.
- Gomes, M.A.C. (2000) *Uma Contribuição à prevenção de fraudes contra as empresas*. Tese de Doutorado em Contabilidade. FEA/USP, São Paulo.
- Günther C.W., van der Aalst W.M.P. (2007) Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics. In: Alonso G., Dadam P., Rosemann M. (eds) *Business Process Management. BPM 2007*. Lecture Notes in Computer Science, vol 4714. Springer, Berlin, Heidelberg.
- Günther, C. W., & Rozinat, A. (2012) Disco: discover your processes. In N. Lohmann, & S. Moser (Eds.), *Proceedings of the Demonstration Track of the 10th International Conference on Business Process Management (BPM 2012)* (pp. 40-44). (CEUR Workshop Proceedings; Vol. 940). CEUR-WS.org.

- Jans, M., Alles, M.G., Vasarhelyi, M.A. (2012) The case for process mining in auditing: Sources of value added and areas of application. *International Journal of Accounting Information Systems* 14 (1), 1-20.
- Jans, M., Alles, M.G. Vasarhelyi, M.A. (2014) A Field Study on the Use of Process Mining of Event Logs as an Analytical Procedure in Auditing. *The Accounting Review*: September 2014, Vol. 89, No. 5, pp. 1751-1773.
- Jans, M. (2019) Auditor Choices during Event Log Building for Process Mining. *Journal of Emerging Technologies in Accounting*: Fall 2019, Vol. 16, No. 2, pp. 59-67.
- Jans, M., Hosseinpour, M. (2019) How active learning and process mining can act as Continuous Auditing catalyst. *International Journal of Accounting Information Systems*, Vol. 32, pp. 44-58.
- Kogan, A., Mayhew, B.W., Vasarhelyi, M.A (2019) Audit Data Analytics Research—An Application of Design Science Methodology. *Accounting Horizons*: September 2019, Vol. 33, No. 3, pp. 69-73.
- Lopes, G.N. (2015) Verificação e classificação de não conformidades de processos de negócios na indústria farmacêutica através de técnicas de mineração de processos, Dissertação de mestrado, Rio de Janeiro: UNIRIO.
- Net, A.B., Solonca, D. (2007) Auditoria de sistemas informatizados. Palhoça: Unisul Virtual.
- Oliveira, F.N. (1989) Técnicas de amostragem utilizadas pelos serviços de auditoria interna de empresas no Brasil – Um estudo de casos. Rio de Janeiro: Fundação Getúlio Vargas.
- Riz, G., Portela, E.A., Loures, E.F.R. (2016) Análise de Conformidade na Área de Saúde com o Suporte da Mineração de Processos, Simpósio Brasileiro de Sistemas de Informação, pp. 52-59, Florianópolis, Santa Catarina, Brazil.
- Roubtsova E., Wiersma N. (2019) Involvement of Business Roles in Auditing with Process Mining. In: Damiani E., Spanoudakis G., Maciaszek L. (eds) *Evaluation of Novel Approaches to Software Engineering. ENASE 2018. Communications in Computer and Information Science*, vol 1023. Springer, Cham.
- Roubtsova, E. and Wiersma, N. (2018) A Practical Extension of Frameworks for Auditing with Process Mining. In *Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering*. Vol.1, pp. 406-415.
- Rozinat, A., van der Aalst, W.M.P. (2008). Conformance checking of processes based on monitoring real behavior. *Information Systems*, 33(1), 64-95.
- Sadiq, S., Governatori, G., Namiri, K. (2007) Modeling control objectives for business process compliance. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007. LNCS*, vol. 4714, pp. 149–164.
- Santos, R.C. (2014) Um modelo de apoio à tomada de decisão para análise de similaridade e unificação de processos de negócio, Rio de Janeiro. COPPE - UFRJ.

Werner, M. (2019) Materiality Maps – Process Mining Data Visualization for Financial Audits. Proceedings of the 52nd Hawaii International Conference on System Sciences. Manoa, USA.