MASTER

A Process Mining-based Approach to Accounts Payable Recovery Audit

van Asseldonk, Maarten D.A.

Award date:
2020

Department of Industrial Engineering & Innovation Sciences
Information Systems Group

A Process Mining-based Approach to Accounts Payable Recovery Audit

Master Thesis

Maarten van Asseldonk

Supervisors:
Dr. Anna Wilbik (IE&IS, IS Group)
Dr. Dirk Fahland (M&CS, AIS Group)
Frits van de Water (Company Supervisor)

Final Version

Eindhoven, April 2020

Abstract

The goal of an Accounts Payable Recovery Audit (APRA) is to recover liquidity leaks that arise from unjustified payments, i.e. to recover funds that have been paid superfluous. In executing such an audit, it is important to have as complete an overview of the accounts payable process at an auditee as possible. This study proposes to extend this overview by implementing a technique that infers dependency from the clearing relation between journal entries. This dependency provides a structure of how and in which order the accounts payable process at an auditee was executed.

We implemented a model, based on the model by Werner and Gehrke (2015), which we applied for the purpose of the APRA. Next, we came up with a way of simplifying the graphs that result from this model to make them easier to interpret for the auditor. Furthermore, we came up with a series of checks that make use of the inferred structure to find anomalies which are of interest to the APRA. Finally, we have evaluated the graphs mined from these dependencies and the results from the checks. In an evaluation of the accurateness, both proved accurate and in an evaluation of the business applicability both proved applicable.

The main contribution of this study for the business perspective is the addition of a different point of view for the APRA. This point of view is both useful in the current APRA as well as in identifying additional risks, as it provides context to the journal entries that are under review but also finds different risks and anomalies. The main contribution from an academic viewpoint is the adaption and implementation of an existing algorithm for a different purpose.

Preface

This thesis marks the end of the great time I have had as a student at Eindhoven University of Technology. It marks the end of a period in my life and kicks off a new one. I have learnt a lot about working in a company, working with people of different professions, as well as working on a big project all by myself. I have also learned (a lot) more about the financial domain than I thought, and maybe even more than I hoped.

First of all, I would like to thank Frits for the practical supervision of the project and for helping me find my way in the company. It was always very valuable for me to get your feedback and opinion, about my thesis but also about all other kinds of things. Thank you for the effort you made to help me succeed.

Second, I would like to thank Anna for being my university supervisor and mentor. From the moment we first met over Skype, you have been very supportive in my plans and efforts. Even though the subject for my thesis was, in the end, not completely in your area of expertise, I really valued your input. Both content-wise, but at least as much structure-wise.

Furthermore, I would like to thank Dirk for taking the time to help me in the beginning of my project and providing feedback afterwards as well as for the help and guidance during the rest of my master's.

Also, I would like to thank all of my colleagues and friends at the company I conducted this thesis in, especially my office mates Niek and Koert. I do not think I would come as far as I have without you, and it would also have been much more boring. Next to this, I would also like to thank Evelien for helping me relate to the business problems I try to solve, and David for helping me get the data I need.

Finally, I would like to thank my family, parents and other friends for supporting and helping me over the last year. I would like to thank my parents for supporting me and helping to make it possible to write my thesis part time, so I was also able to focus on starting up a company. I would like to thank Lotte for the ever present help, support and fun, both as a peer and a colleague but above all as a friend.

Writing this thesis has been a long, windy, but also very fun and interesting road for me. I have tried lots of different paths, only to find reasons that they did not work out and new opportunities at the end of them. In the end, I was able to find a path that I have followed to where we are now.

Enjoy reading my thesis!
Maarten

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Process mining techniques are increasingly used in an audit context. The focus of most studies regarding process mining in audit is on either internal or external audit or fraud detection. However, the context of applying process mining techniques in an Accounts Payable Recovery Audit (APRA) has not been studied yet. This context has a combination of the characteristics of all three of the areas mentioned before. However, it should be approached slightly differently. In this chapter, we will introduce the problem and motivation of this study. First, in Section 1.1, we introduce the research problem and formulate research questions. Next, in Section 1.2 we will motivate our approach. In Section 1.3 we describe the research methodology we follow and in Section 1.4 the contributions we make to the literature. Finally, in Section 1.5 we provide an outline for the rest of this thesis.

1.1 Research Problem

In this study, we aim to improve the process of detecting anomalies from the accounts payable of a company on which funds can be recovered. To this end, we study the effect of adding and analyzing data about the accounts payable process a payment has gone through. We consider this process from the first invoice receipt to the final payment in the analyzed data set. We have conducted this study at a Dutch audit firm that specializes in this Accounts Payable Recovery Audit, with data from one of their customers. Throughout this study we will refer to this company as the Auditing Company (AC).

In the current approach, the auditors only analyze the invoice documents to find anomalies. Most of these anomalies are found by automated versions of more traditional audit checks. However, by only looking at the final document, a lot of information about the accounts payable process this document followed within a company is lost. Furthermore, the current approach leads to a lot of false positives, meaning that the results of the risk assessment still have to be sampled to make the number of bookings that need examining workable.

We aim to include the data available in modern-day document-based ERP systems in this analysis to provide a complete view of the handling of an invoice at a company. To do this, we will try to exploit the characteristics of the document-based ERP system and the logic of double-entry bookkeeping in accounting to obtain process graphs that can be analyzed in a process mining fashion. However, modern ERP systems store even more information about the process of handling an invoice other then these documents contain. They also store, for instance, changes which the documents have gone through between their creation and finalization. These changes may provide valuable information for an auditor, in that they are stored by the system and therefore not tampered with by human interaction. In our work, we attempt to enhance our process graphs with information from data sources like these changelogs. We do this to get as complete an overview as possible of the handling of an invoice in the accounts payable process within a company. The research question is, therefore, formulated as follows:

How can accounting logic-based process mining be used to enhance the quality and quantity of the risks found in an APRA by identifying possible risks?

To answer this question, we formulate a set of sub research questions:

1. *How can process graphs be mined using the accounting structure to infer dependency, instead of temporal logic?*
2. *What data is required for this mining process and where to find it in different systems?*
3. *How can these process graphs be used to develop checks that improve the quantity and quality of the risks found in an Accounts Payable Recovery Audit?*

1.2 Motivation

In this study, we propose to tackle the process analysis of the entries in a financial system in a way that differs from the conventional way of process mining (Section 2.5). There are two reasons for this. First, the financial data that we would like to study suffers to a large extent from the challenges of convergence and divergence (Section 3.2.1). Second, the company we conducted this research in has a long record of analyzing financial data from invoice to payment. To align as much as possible with their current processes, we decide to dive deeper into the process from invoice to payment. Instead of combining it with data from purchase orders, goods receipt. and other steps further upwards in the process, like has been done by many of the studies referred to in Section 2.5.

1.3 Research Methodology

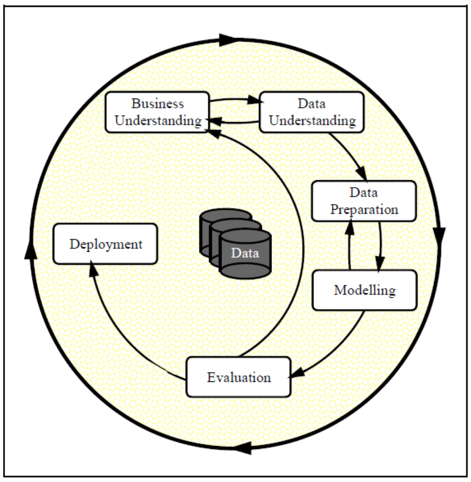


Figure 1.1: Five Phases of the CRISP-DM Model (Wirth, 1995)

For this study, we combine the two research techniques of Method Engineering (Brinkkemper, 1996) and the Cross Industry Standard Process for Data Mining (CRISP-DM) framework (Wirth, 1995). Users consider the CRISP-DM framework to be the most complete, popular standard in data mining research (Azevedo and Santos, 2008). Since we interpret data analysis as the primary goal of our study, CRISP-DM is the most appropriate research methodology. However, since the data analysis is not the primary goal of this research in itself, we will loosely follow the model, placing extra emphasis on the Modeling phase. The CRISP-DM model consists of five phases, shown in Figure 1.1.

In the first phase, business understanding, we put focus on understanding the business objectives and translating these into a data mining problem definition. In this phase, we also define account controls from a business perspective that the model will later be evaluated on. The second phase is data understanding. In this phase the initial collection of data takes place, followed by getting familiar with the data and identifying data quality problems. There is a strong link between the first and second phases, as some data understanding is required to define the business objectives accurately. The next phase is the data preparation phase, in which the available data is transformed in such a way that it can be used for the modeling phase.

For the modeling phase, we apply Method Engineering, as in this phase of CRISP-DM the development of the model takes place. Because there is a particular emphasis on this modeling phase in this study, we approach it with a separate research methodology. In Method Engineering, parts of different researches (method fragments) are combined to engineer new methods. If method fragments are missing for the method to be engineered, these fragments are developed. In this study, we combine the methods of the accounting data structure dependent control flow inference (Werner, 2017) with the usage of the changelogs and other data sources to come up with a method to conduct an APRA of a company as effective and efficient as possible. In the modeling stage, we will also identify controls that can be executed on the process models generated by the model to be evaluated later. In the evaluation phase, we will assess the ability of the model to identify anomalies, thus be a helpful analytical procedure for the auditors. For the final, deployment, phase we will provide the algorithm in pseudo-code to the developers of the detection algorithm at the audit company.

1.4 Contributions

The main contribution that we make in this study is the application of accounting-logic based process mining in an APRA setting. The main difference between this setting and a regular audit setting lies in the focus of this kind of audit. In an APRA focus is not as much on checking whether the bookings were made correctly, but rather on whether the contents of these bookings are also correct. This study focuses purely on activities related to the accounts payable process, while most other studies, especially the studies focused on internal audit, take the broader scope of the whole purchase-to-pay process. As not all companies make use of purchase orders or use separate systems for purchase orders and invoices, the approach of this study provides a more generally applicable solution. Furthermore, we use approaches that are common use in the temporal logic based process mining, like exploiting changelogs, and apply them for the accounting-logic based process mining. We combine the ability to deal with convergence/divergence the accounting-logic based process mining brings, with the level of detail that results from incorporating the changelogs.

1.5 Outline

The remainder of this work is structured based on the CRISP-DM model. In this section, the link between the chapters and steps in the CRISP-DM model will be made. Chapter 2 and Chapter 3 make up the business understanding steps of the CRISP-DM model, as well as touching the data understanding step. In these chapters, we explore the concepts we will be dealing with, as well as work that has been done by others on similar subjects. In Chapter 4 we execute the rest of the data understanding step and go through data preparation. Chapter 5 makes up the modelling step of the CRISP-DM model. This chapter elaborates on how the modelling was done, which decisions were made and how to reproduce it. Chapter 6 and Chapter 7 make up the evaluation step in the model. In Chapter 6 the results for the implemented checks are described, while in Chapter 7 the model and its resulted are evaluated in a factual and a questionnaire-based manner. Together with Chapter 5, Chapter 7 also provides the necessary input for the deployment step of CRISP-DM. Finally, Chapter 8 provides a concluding summary as well as recommendations and future work.

Chapter 2

Preliminaries

This chapter introduces the preliminaries that will be used throughout this thesis. We will provide a brief introduction to these preliminary topics. Starting with a general description of auditing in Section 2.1 and the different approach that is taken in an APRA in Section 2.2. Subsequently, we introduce the type of data source we are interested in, Enterprise Resource Planning (ERP), in Section 2.3. Next, we present the structure of accounting entries in these ERP systems in Section 2.4. Finally, we introduce the concept of process mining in general in Section 2.5.

2.1 Audit

An audit is defined as a methodical examination and review, or in a financial context as a formal examination of an organization's or individual's accounts or financial situation (Merriam Webster, 2019). In a financial audit, an opinion is formed whether either the financial statements of the audited company are following specified criteria, often international accounting standards. Alternatively, it is tested whether internal controls deliver the results they are expected to (Arens et al., 2012).

A distinction exists between internal audit and external audit. The main difference between these types of audit is who conducts it. An internal audit is conducted by employees of the company that is being audited. An external audit, in contrast, is executed by employees of an external Certified Public Accounting (CPA) firm. An external financial audit focuses mainly on the compliance of the financial statements of a company with accounting standards, while an internal audit is often broader in scope (Arens et al., 2012). In this study, the part of the auditing process we will focus on is on the analytical procedures (phase 3 in **??**). The Statement of Accounting Standard AU Section 329 states that analytical procedures are an essential part of the audit process. These analytical procedures study possible relationships between both financial and non-financial data with as their primary goal to aid the auditor conducting the audit in determining where to focus and time their audit (PCAOB, 2017).

2.2 Accounts Payable Recovery Audit (APRA)

We define an Accounts Payable Recovery Audit (APRA) as the process of going through the financial records of a company in order to find any liquidity leaks. These leaks could be unjustifiably paid amounts or unclaimed credits. The main difference between this kind of audit and a typical audit by a CPA firm is that in an APRA the focus is not on the compliance with accounting standards. Instead, the focus is on the business correctness of the financial statements, meaning whether they contain anomalies that led to liquidity losses. An APRA is, therefore, mainly focused on the recovery of funds that a company has paid unnecessarily. A more detailed description of this process at the company in question in this study is included in **??**.

2.3 Enterprise Resource Planning Systems (ERP)

Enterprise Resource Planning (ERP) systems are software packages that aim to support the execution of business processes. These systems consist of a collection of modules, including sales, human resources, and of particular interest for this study, finance and controlling. ERP systems provide cross-organization integration and support of data through embedding business processes. These systems can then be catered to the needs of an organization. (Esteves and Pastor, 2001).

The most well known and prominent ERP systems (SAP, Oracle, Microsoft Dynamics) are based on business documents (also known as artifacts) (Lu et al., 2015). Each transaction that is made within these systems either creates a document or changes one. These documents can be interpreted as an indication of the state of the process. A booked invoice can, for instance, be interpreted as a document, which could later trigger a payment document. These documents exist as separate entities in the ERP system. Another type of ERP system is a process-based ERP system, in which the process is central, and saved data is attached to the execution of these processes.

2.3.1 Changelogs

Changelogs, or redo logs, in ERP systems are logs that record all changes made to documents. They provide the administrator of an ERP system with the possibility to see which changes have led the document to evolve to its current state. Therefore, one could also see them as undo-logs(GNU, 2019). Most modern-day ERP systems have functionalities similar to this, to accommodate for users making mistakes that can be recovered. An example changelog for a financial module in an ERP system is shown in Table 2.1. In most systems these are activated by default.

ChangeID	Table	DocumentID	Timestamp	Field	Old Value	New Value
1	VendorMaster	100001	23-05-2018 14:55:12	LastChangedDate	12-02-2018	23-05-2018
2	VendorMaster	100001	23-05-2018 14:55:12	VendorName	Company A	Company B
3	Invoices	100005	28-05-2018 13:12:10	ExternalInvoiceId	12345678	987654321

Table 2.1: Example Changelog for a Financial Module in an ERP System

2.4 Accounting Logic

Modern-day accounting is based on the concept of double-entry bookkeeping (Carruthers and Espeland, 1991). Double-entry bookkeeping is a technique that dates back as far as the Medieval Italy (Bryer, 1993). The main principle of double-entry bookkeeping is that each journal entry posts at least two items, one as a debit into one account and one as credit into another corresponding negative entry into another account. This way, in theory, it is always possible to check whether the books are correct since all entries on all accounts should equate to zero. Modern-day accounting systems have varying ways of storing this data; in a document-based ERP system (Section 2.3), each journal entry is recorded as a separate document. However, from an accounting perspective, it does not matter which system is being used. One could say that no matter which accounting system is used, the heart of accounting is always the same.

These journal entries are created by the execution of process activities, as shown in Figure 2.1. Here we can see double-entry bookkeeping in action, each accounting activity creating at least two journal entry items where the credit and debit amounts are equal to each other. There are two types of journal entry items, ones that post an open item, and ones that clear items. If an open item is posted, this means that it still requires to be cleared from another activity. Clearing can, therefore, be seen as closing an open item. This structure is formalized in an ERD in Figure 2.2. Here we can see that each journal entry (each activity) contains at least 2 journal entry items and clears 0 to N journal entry items. This is a recursive data structure as one instance refers to multiple instances of the same data type.

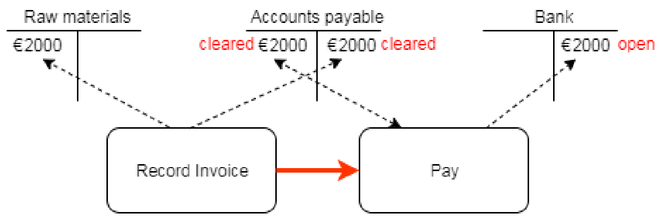


Figure 2.1: Activities with their corresponding journal entry items (Werner, 2017)

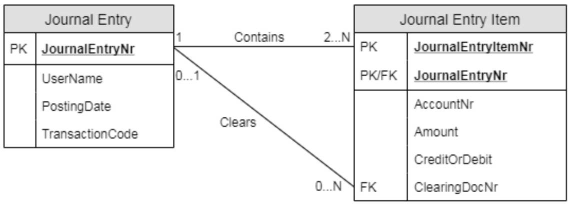


Figure 2.2: ERD Model of Journal Entries and Journal Entry Items (Werner, 2017)

2.5 Process Mining

Process mining techniques are able to extract knowledge from event logs, commonly available in information systems nowadays (Aalst et al., 2012). These techniques provide ways to discover, monitor, and improve processes in a variety of business domains. Process mining techniques take event logs, as shown in Table 2.2, as their input. These event logs contain at least columns with a case ID, activity name, and timestamp. Other attributes, like resource name or invoice details, are optional. The process mining algorithms then take this input and transform it into the output, as shown in Figure 2.3.

Case ID	**Activity**	**Timestamp**
1001	Receive Invoice	22-02-2020 14:31:43
1001	Receive Goods	27-02-2020 15:30:53
1002	Receive Invoice	27-02-2020 16:29:48
1001	Pay	02-03-2020 15:17:31
1002	Pay	02-03-2020 15:17:31

Table 2.2: Example Event Log from an Accounting Information System (AIS)

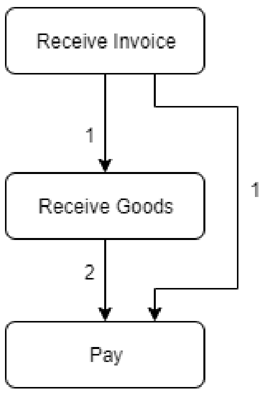


Figure 2.3: Example Process Model From the Event Log in Table 2.2

There are different ways of looking at business processes. Process mining tackles these differences by taking different perspectives from which to approach these processes. Aalst (2011) identifies several of these perspectives, among which the most important perspectives being:

- *Control-Flow Perspective (how?)* focuses on the order of events. This perspective aims to provide a good characterization of all possible paths in a model.
- *Organizational Perspective (who?)* focuses on information about resources in the event log. Who executes which tasks and who work together in what way.
- *Case/Data Perspective (what?)* focuses on the other, non-process, properties of cases. Which other attributes are important for the way the process is executed.

In this study, we will focus on the control-flow perspective. We will not go in depth about all different process mining algorithms that are out there.

2.6 Summary

In this chapter, we briefly discussed the preliminaries of this study. We first introduced the concept of financial auditing in general, and subsequently explained the differences with an accounts payable recovery audit: an APRA focuses on the business value of the financial statements, instead of its accordance with international standards. Next, we explained the accounting logic that can be used to infer control flow using process mining. Process mining, in general, is the last topic we briefly introduced.

Chapter 3

Related Work

In this chapter, we will conduct a short literature review to present the related work to this study. We will keep a broad focus of data mining related approaches that have been attempted in order to improve auditing in general. Especially in the field of internal auditing, a lot of research has been done. In Section 3.1 we will focus on other data mining applications in audit. Second, in Section 3.2 we will focus on applications of process mining in audit as well as different approaches that have been applied for this. Finally, in Section 3.3 we provide a summary.

3.1 Data Mining in Audit

The idea of applying data mining techniques on auditing problems is an established idea. Ngai et al. (2011) have provided an overview the body of literature on the application of these techniques, specifically on the identification of fraud. Even though this meta-study is dated itself, it provides an overview of the extent of the field up to 2011.

Amani and Fadlalla (2017) provide a broader and more recent overview of data mining techniques that have been applied to audit. In their work, Amani and Fadlalla (2017) categorize their findings into goals, applications and tasks. They first make a distinction into three different kinds of data mining goals: description, prediction, and prescription. Next, they identify three application tasks: application of data mining in financial accounting and accounting information systems (AIS), in managerial accounting and in assurance and compliance. Finally, a distinction is made into separate data mining tasks.

When looking at the goals in this research field, it becomes clear that emphasis is placed on prediction. For the applications in this research field, emphasis is placed on the assurance and compliance aspect. For the tasks, the research has been heavily focused on classification (67%), while only 2.5% has been on exploration (Figure 3.1.) Classification tasks can, for instance, be used in detecting fraudulent behavior by detecting outliers but could, for example, also be used to classify a company as a future bankruptcy. One of the significant developments in applying data mining as an analytical procedure in accounting is the use of text mining. Using text mining, auditors attempt to extend data being audited from merely quantitative to including qualitative data (Li, 2010).

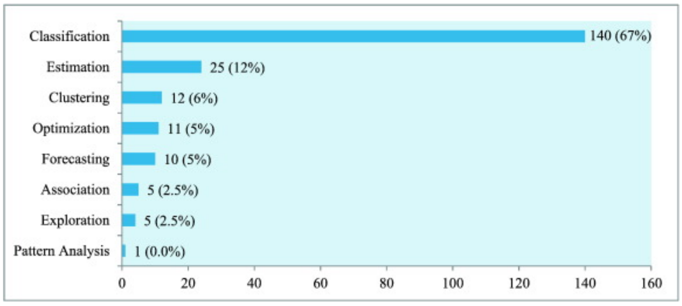


Figure 3.1: Applications of Data Mining in Accounting by Data Mining Tasks in the Literature (Amani and Fadlalla, 2017)

3.2 Process Mining in Audit

Multiple scholars propose process mining as a technique to assist auditors in their work (Aalst et al., 2010, 2011; Gehrke and Müller-Wickop, 2010; Jans et al., 2011, 2013, 2014, 2019; Jans and Soffer, 2018; Hosseinpour and Jans, 2016; Werner, 2013, 2017; Werner and Gehrke, 2015; Müller-Wickop and Nüttgens, 2014). In these works the following arguments are made for the use of process mining in supporting an auditor:

- Through event logs, process mining provides a chronological view over the entire process that is being audited, including all intermediate steps. This differs from the What-You-See-Is-What-You-Get (WYSIWYG) view that other audit techniques give, in which only the end result is visible.
- Process mining provides different perspectives for the same processes, leading to new ways to discover patterns.
- Results from process mining are more tangible for business users, making them easier to interpret then results from other data mining techniques.
- Process mining provides the opportunity to dive deeper into the root causes of found anomalies because it enables more extensive review.
- Process mining not only examines data recorded by the auditee, but also metadata saved by the system. This holds that it is likely to be more accurate, as it is not tampered with by humans.
- Process mining examines the population, not just a sample like regular auditing techniques do.

In the coming sections, we will discuss three approaches to process mining in audit, but also in general, that have been proposed previously. First, in Section 3.2.1, we look at the most common approach of using the temporal, timestamp-based logic. Next, in Section 3.2.2, we look at a more novel artifact-centric approach, focusing on so-called data-centric or document-based processes. Finally, in Section 3.2.3, we look at a specialized technique for process mining in accounting processes, following accounting logic.

3.2.1 Process mining in Audit, Temporal Approach

As explained in Section 2.5, process mining relies on timestamps to order events in a temporal manner. Jans et al. (2014) are one of the first to outline the opportunities process mining of event logs brings to towards the end of adding value as an analytical procedure in auditing. In their work, they put emphasis on the necessity of process mining analysis to meet the cost/benefit criteria of analytical procedures in auditing. Even more emphasis is put by them on a statement of Brewster

(2011): "When performing analytical procedures, auditors should treat *any* discrepancies between pre-developed expectations and management representations as indicators of heightened misstatement risk." Therefore they come up with a protocol, aiming at a systemic focus on transactions that require further audit investigation:

1. *Identifying the designed process*: Determine the events to be included in the event log.
2. *Creating the event log*: Creating the event log from the source system (often ERP), as well as documenting the assumptions made in this transformation.
3. *Process Discovery*: Show how processes actually run in a company to get a general overview.
4. *Role Analysis*: Analyze which users execute which events and compare this to internal regulations, like the Seggregation of Duties (SOD).
5. *Attribute Analysis*: Look for more specific attributes that are present in cases that deviate from the set processes.
6. *Social Network Analysis*: After finding anomalous transactions, a social network analysis can be conducted to find not just transactions that need review, but also employees that might need review.

The entity that is being followed through the process is, in most of the work by Jans et al. (2011) and related authors, a Purchase Order (PO). This focus is an important decision as it has a significant impact on the way the rest of the study should be interpreted. Focusing on the PO reduces the dimensionality of the examined process, and therefore makes it more difficult to interpret. By focusing on one entity the process models in multidimensional processes are not always self explanatory.

This problem of handling multidimensional processes instead of single-dimensional processes is an established problem in process mining research; the issue of convergence and divergence resulting from document-based source systems. If we want to include more complex auditing analytics, this problem needs to be tackled.

Convergence and Divergence

The problem of multidimensionality in process data, thus the occurrence of convergence and divergence, was first mentioned by Segers (2007). Process mining assumes that a single entity follows the process, but in document-based ERP systems this often is not the case. It is possible that one invoice is paid in multiple installments, leading to divergence. But, it is also possible that multiple invoices are paid in a single payment, resulting in convergence. An easy way to understand convergence is to look at it as combining or batching and at divergence as splitting. This problem of convergence and divergence is a problem many commercial software vendors still struggle with (Aalst, 2019).

To illustrate this, in Figure 3.2, we give an example of a typical accounting process. The red arrows in this figure show where multiple orders are combined on one payment, thus are converging. The blue arrows show one payment leading to two postings with clearing, thus diverging. In Figure 3.3, we show how a naive process mining algorithm would visualize this process. It is easy to see why this is undesirable. When comparing with Figure 3.2, the model in Figure 3.3 does not show an accurate representation of the real process. Several scholars, among whom Aalst et al. (2009) and Lu et al. (2015), propose a solution to this problem of convergence and divergence in general. In Section 3.2.2, we will introduce their artifact-centric solutions. However, specifically in the domain of financial audit, Gehrke and Müller-Wickop (2010) and Werner (2017) propose a solution that is specific for that field. We cover their solution in Section 3.2.3.

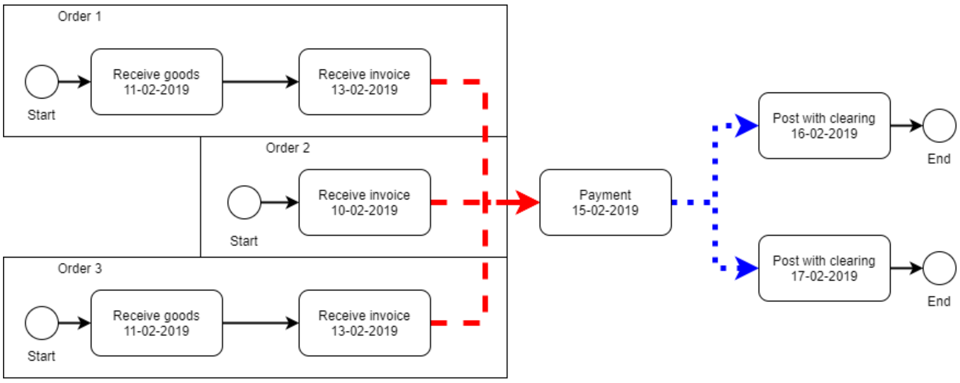


Figure 3.2: Convergence (red - dashed) and Divergence (blue - dotted) in a Typical Accounting Process

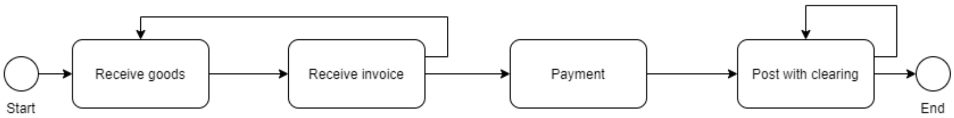


Figure 3.3: Model as Mined by a Naive Temporal Process Mining Technique (Directly-Follows-Graph(DFG))

3.2.2 Artifact-Centric Process Mining

Several attempts have been made to tackle the problem of dimensionality in interpreting process mining results. Nooijen et al. (2013) introduced a first attempt to discover a process model automatically from a document-based; they refer to it as a data-centric process. To do this Nooijen et al. (2013) introduced the concept of artifacts, which allow for the technique to accurately mine the multidimensionality within processes coming from data-centric sources. Lu et al. (2015) then added semi-automatic discovery of interactions and causal dependencies between these artifacts to this technique. This resulted in the ability to mine an end-to-end process model built up of interacting artifacts with causal dependencies.

Li et al. (2018) propose to solve this problem by introducing a new type of event log, an object-centric event log. Aalst (2019) stresses the shortcomings of the single-dimensional event logs once again and supports the case of an object-centric log. However, as these developments are still very recent, not much work was done on implementing it in an audit context yet.

3.2.3 Process Mining in Audit, Accounting-Logic based Approach

Gehrke and Müller-Wickop (2010) propose a different way to deal with the challenge of the multidimensionality of event traces in an audit setting. They propose to exploit the accounting logic explained in Section 2.4 to preprocess event logs in a different way. Instead of looking at timestamps of events to form dependencies in a temporal way, they build these dependencies following the logic of clearing open journal items in accounting. Werner (2013) uses this preprocessing to construct colored Petri nets that model both the control flow and data perspective of the process, be it only in the form of separate instance graphs per separate process. Müller-Wickop and Nüttgens (2014) extend the Event-Driven Process Chain (EPC) notation (Keller et al., 1992) to be used in describing accounting processes, including the data perspective. Werner and Gehrke (2015) introduce an algorithm that can be used to go from process instances to process models, by transforming the number of occurrences of each unique instance graph to weights in a combined process model. Werner (2017) combines previous research in an attempt to transform the colored Petri nets to process models that enable a more straightforward interpretation by their end-users.

To achieve this interpretability, they remove deadlocks and focus on just the control-flow perspective. In recent work, Werner (2019) suggests the use of heat maps to indicate which accounts and entries are material for the auditor.

3.2.4 Continuous Monitoring

Another branch of literature that aims to aid analytical procedures in auditing is the branch of continuous monitoring (Brennan and Teeter, 2010; Singh and Best, 2015; Jans and Hosseinpour, 2019). Singh and Best (2015) propose introducing a continuous monitoring guard that enforces internal controls while the process is being executed. These controls are based on results from earlier process mining research and auditing expertise. The aim of this approach is to prevent as many mistakes as possible by triggering errors while suspicious activities are executed. This way, many anomalies can be corrected before they actually take effect. This continuous monitoring is, therefore, a way process mining could be permanently included in the accounting and audit process in the industry.

3.3 Summary & Conclusion

In this chapter, we introduced work related to the topic of this study. We have shown that there is a large body of literature attempting to improve the analytical procedure in auditing by applying both process mining as well as other data mining techniques to it. Other studies have already proved the capabilities of these techniques in assisting an auditor to execute his work. Furthermore, we have shown the difficulties that arise when a multidimensional problem is attempted to be solved by a single-dimensional solution and attempts that have been made at overcoming these problems. Concluding from our study of this related work, we decide to take the accounting-logic based approach of Section 3.2.3. Meaning that in the rest of this study we will focus on mining process graphs from the clearing structure in journal entries. We made this decision because this approach is closest to the current processes at the Auditing Company and therefore is most likely to be applicable in an APRA context on the short-term.

Chapter 4

Data Description

In this chapter, we will describe the data that is required for the graph mining algorithm of this study in more detail. First, we will give a general description of the data that is required. Next, We will go in-depth into what data we used for our research and briefly mention which fields are required from the source systems. In Section 4.1 we will describe the required data in general. In Section 4.2, we describe the data we studied for this study. Next, in Section 4.3, we provide some descriptive data set statistics. Finally, in Section 4.4, we provide a summary of the chapter.

4.1 Data Format

The graph mining algorithm used in this study works with a specific data input. This input is specific to the domain of financial accounting. Generally, the data that is required for this is the following:

- Journal Entries (JE), preferably all, but minimum all vendor and general ledger related journal entries. The customer related journal entry documents are optional.
- Journal Entry Items (JEI) for all journal entries
- Information related to the journal entry (items), e.g. vendor master data and tables that explain document types or other contextual information.
- Event logs, if available in a source system.

The algorithm attempts to infer dependency between journal entries from the recursive structure this data follows, to mine process graphs. This recursive clearing structure, as explained in Section 2.4, is visualized in Figure 4.1. To transform this data into data that the model can handle, we have to apply data cleaning and preparation steps. The resulting data sources from these processing steps are shown in Figure 4.2. The inner (green) box of this figure shows the data which is required for the generation of the graphs, while the outer box also contains other relevant data for the risk flagging of graphs.

The data in the inner container consists of a clearing documents adjacency list and a starting documents list. The clearing documents adjacency list is a list of sets consisting of each journal entry and all other journal entries one of its items has been cleared by. In the example, $Doc1$ has at least one line that was cleared by $Doc5$, as well as at least one that was cleared by $Doc6$. The starting document list is a list of all journal entries that have at least one item with a clearing document, i.e. all documents that are cleared by at least one other document but do not clear any other items.

The outer box consists of the data of the inner box but with additional metadata, which is linked to the items in the inner box by their numbers. It contains data about the journal entry (e.g. username, posting data, transaction code) and about the journal entry items (e.g. account number, debit/credit indicator, amount), but also all other relevant data for analysis. Other relevant data could, for instance, be data from the changelogs, as explained in Section 2.3.1, or any other data related to any journal entry (item).

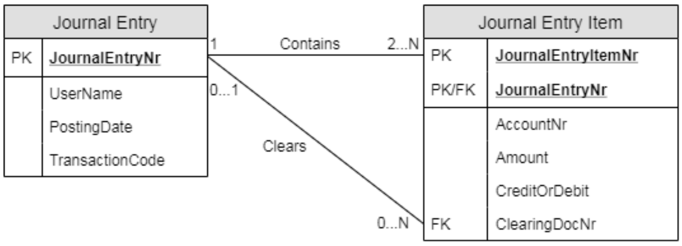


Figure 4.1: ERD Model of Journal Entries and Journal Entry Items (Werner, 2017)

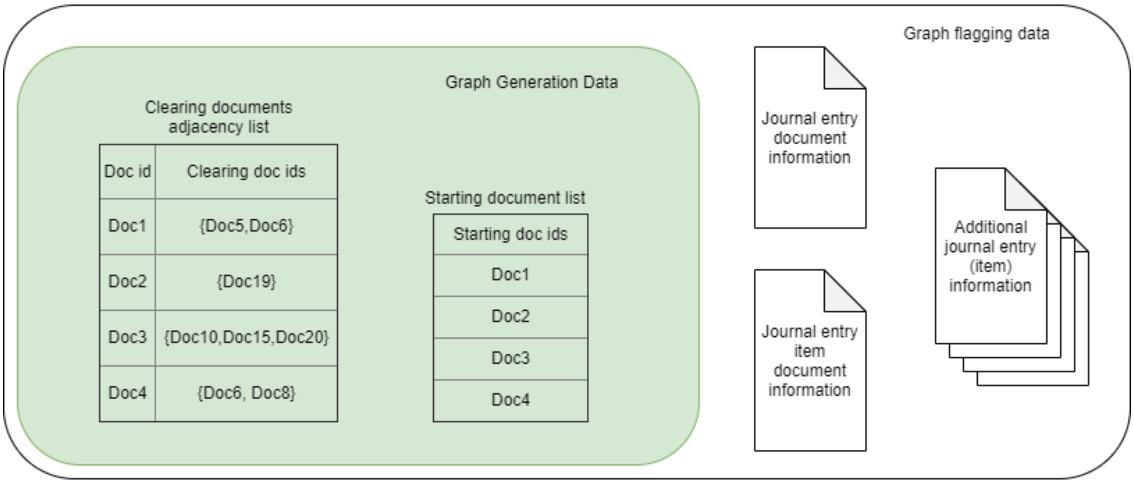


Figure 4.2: Data Sources for the Process Graph Generation Algorithm in Chapter 5

4.2 Case Study Data

The data that was used for this study comes from an auditee's ERP system in one of the the running projects at the Auditing Company. We scoped the data collection to data of just 2018, and 14 business units. The source system is an SAP system, specifically SAP ECC 6.0. The tables and views from which data were extracted are shown in Table 4.1. We get the journal entries that are required from the BKPF table and the journal entry items from combining the FBL1n and FBL3n tables. In this combined table of FBL1n and FBL3n the clearing document is in the clearing doc. field (for BSEG the AUGBL field). Data was extracted through the front end of the client's SAP system and exported as text files. A full overview of all downloaded tables and why they were excluded is included in Appendix A. Note that in this system, the journal entry items are divided into three categories: vendor items, general ledger (G/L) items and customer items. We were unable to extract data through from the BSEG table, which contains all journal entry items. Therefore we extracted the journal entry items using the FBL1n and FBL3n transaction codes. This yields that we do not have customer journal entry items. The studied company only makes use of the SAP Finance (FI), but not of the SAP Material Management (MM) module. This yields that data about purchase orders were not available in the system, but would have had to come from another system.

Table/View	Description	Remarks
BKPF	Accounting Document Header (Journal Entries)	
FBL1N	Vendor Line Items (Vendor part of Journal Entry Items)	FBL1N is a view on the underlying data tables (LFA1, LFB1, BSIK, BSID, KNA1, T001, T005, SKAT, SKB1)
FBL3N	G/L Line Items (G/L part of Journal Entry Items)	FBL3N is a view on the underlying data tables (BSIS, BSID, KNA1, T001, T005, SKAT, SKB1)
SKAT	G/L Account Master Record (Chart of Accounts: Description)	
LFA1	Vendor Master Data	
CDHDR	Change document header Table and data	
CDPOS	Change document items Cluster table and data	
T001	Company Codes Table	
T003	Table fields within data Dictionary	
TSTC	Transaction codes table and data	

Table 4.1: Tables for Source Data Extraction

Other systems

We mentioned before where the clearing relation of Figure 4.1 can be found in a SAP system. To implement the proposed algorithm in this study in a broader context, we also examined data from other systems.Because the general ledger and vendor line data is very similar across all these systems, we focus on where the clearing relations can be found in these systems. An overview of the fields which contain this clearing relation is included in Table 4.2, for completeness of this overview we also included SAP again. It should, however, be noted that many AIS have less elaborate clearing sequences than SAP; they might only have an invoice booking and a payment. For these cases, the added value of the graph generation is questionable as the graphs would be of depth 1.

System	Clearing table	Clearing field
Microsoft Dynamics AX	LedgerTrans	SettledBy
Microsoft Dynamics NAV	G/L Entry	ClosedBy
SAP	BSEG	AUGBL
Exact Globe		

Table 4.2: Other Accounting Information System (AIS) with the clearing fields and tables

4.3 Data Set Statistics

To give a general overview of the data we are working with, we provide some key statistics in this chapter. As mentioned before, the data originates from an auditee served by the AC, over the fiscal year of 2018. We have a total of 14 business units, with 291777 journal entries and 414317 journal entry items[1]. Business units that have a lot of customer transactions, therefore, have a relatively low amount of journal entry items. Furthermore, we also provide the average invoice value for each of the business units. This gives an indication of the type of invoices these business units handle and the impact they have on the overall finances of the company. Figure 4.3 shows the characteristics of the data, divided over the business units. Table 4.3 shows the exact numbers for these charts. If we mention big business units throughout this study, we mean big in terms of journal entries and journal entry items. Thus, we see business unit 1 as the biggest and 4 is the smallest.

[1]Note that these numbers do not follow the 2..N relationship from Figure 2.2. This is because, as mentioned in Section 4.2, the customer journal entry items are not included.

Business unit	# Journal entries	# Journal entry items	Average invoice value
1	44522	83114	€ 13.853,37
2	14879	19404	€ 14.164,90
3	13047	26361	€ 17.649,61
4	407	672	€ 2.831,32
5	50750	59407	€ 7.844,55
6	21019	50051	€ 13.613,46
7	19067	39428	€ 37.659,69
8	43314	59489	€ 4.566,23
9	4883	9892	€ 6.480,92
10	7101	5434	€ 9.603,10
11	44052	19723	€ 14.561,69
12	5348	8082	€ 21.021,26
13	14430	19172	€ 4.092,05
14	8958	14088	€ 14.121,20
Total	291777	414317	€ 13.004,52

Table 4.3: Number of Journal Entries, Journal Entry Items and Average Invoice Amount per Business Unit (BU)

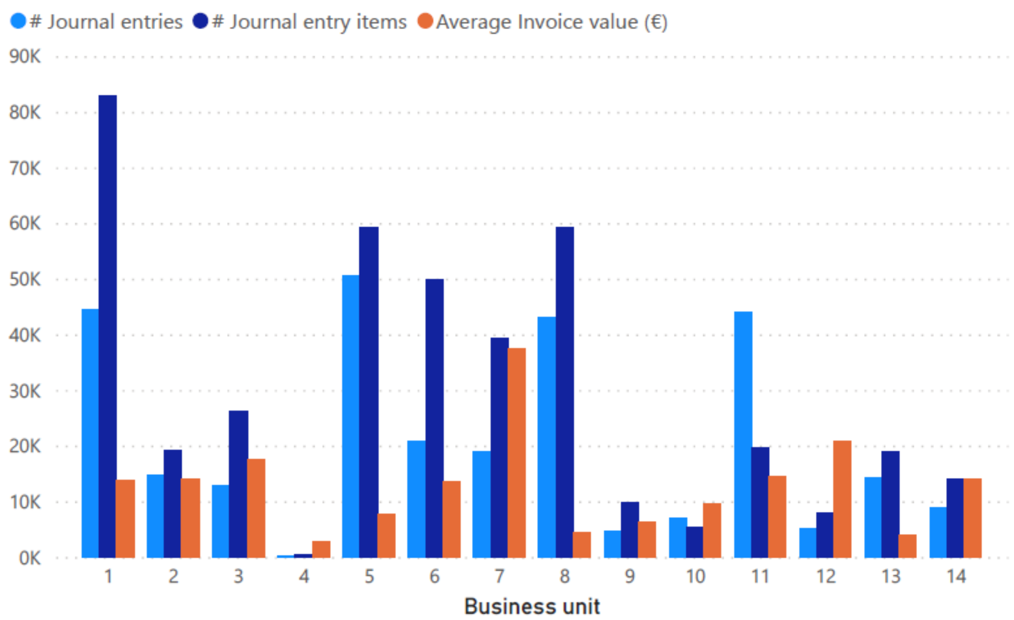


Figure 4.3: Number of Journal Entries, Journal Entry Items and Average Invoice Amount per Business Unit (BU)

4.4 Summary

In this chapter, provided an introduction of the data that we will refer to throughout the rest of this study. First, we introduced the general data which is required for the graph mining algorithm that will be introduced in the next chapter. Next, we described the specific data set that was studied and some of its statistics. The rest of this study will focus solely on data from the SAP system described in this chapter.

Chapter 5

Model Description

In this chapter, we will introduce a model to improve the quality and quantity of risks identified by the Accounts Payable Recovery Audit (APRA). Following the conclusion of Chapter 3, we implement the accounting-logic based process mining approach. In this chapter, we will go through this implementation step-by-step. First, in Section 5.1, we will go into the generation of the process graphs. In Section 5.2 we will show a running example of this generation algorithm. In Section 5.3 we present a way of improving the readability of the graphs. After this, in Section 5.4, we will introduce a way of flagging graphs for risks. And finally, in Section 5.5 we will provide a summary of the chapter.

5.1 Process Graph Generation

The process graph generation model proposed in this study is based on work by Werner and Gehrke (2015). However, we only focus on generating the control flow graph and have a different way of handling exclusive clearing entries[1]. Figure 5.1 shows a high-level overview of this process. First, we collect and pre-process data from a source system. Next, we use our Instance Graph Generation Algorithm (Algorithm 1) to mine instance graphs. Finally, we merge these instance graphs into Clearing Process Graphs, which provide us with insight into which documents clear each other. For this, we use Algorithm 2. Our implementation of the graph generation algorithm is based on the concept of weakly connected components in a directed graph (Cormen et al., 2009). In this section, we will walk through the algorithm step by step and explain why choices were made.

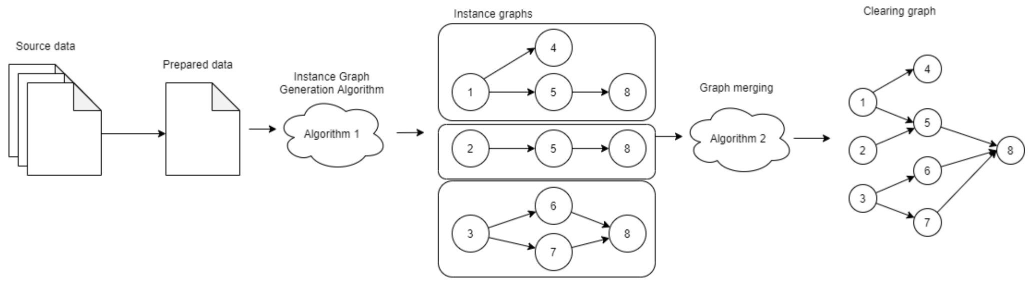


Figure 5.1: High-Level Graph Generation Procedure

The Instance Graph Generation Algorithm (Algorithm 1) takes two inputs, an adjacency list *graph* and a list of start documents *start_nodes* as described in Figure 4.2. It consists of two

[1]An exclusive clearing is a journal entry that is referred to as a clearing document, but has no journal entry items. These types of clearings occur, for instance, when journal entry items have to be cleared by items from a different system.

Algorithm 1 Instance Graph Generation Algorithm

```
Input: graph as adjacency list of all document IDs and clearing IDs
Input: start_nodes as a set of invoice documents to use as start nodes

function GetConnectedSubgraphs(graph, start_nodes)
    count = 0
    visited = ∅
    component_subgraphs = {}
    for node in start_nodes do
        if node not in visited and node in graph then
            count+ = 1
            components_subgraphs[count], graph = DepthFirstSearch(graph, node, set())
        end if
    end for
    Return components_subgraphs
end function

function DepthFirstSearch(graph, start_node, visited)
    visited.add(start_node)
    for next in graph[start] − visited do
        if next not in graph then
            Continue
        end if
        DepthFirstSearch(graph, next, visited)
    end for
    Return visited, graph
end function

components_subgraphs =GetConnectedSubgraphs(graph, start_nodes)
```

Algorithm 2 Graph Merging

```
for subgraph_1 in components_subgraphs do
    for subgraph_2 in components_subgraphs do
        if subgraph_1 ∩ subgraph_2 ≠ ∅ then
            subgraph_1 = subgraph_1 ∪ subgraph_2
            delete subgraph_2 from components_subgraphs
        end if
    end for
end for
```

functions, a GetConnectedSubgraphs and a DepthFirstSearch function. The GetConnectedSubgraphs is a function that generates a dictionary of the results of the DepthFirstSearch. The DepthFirstSearch is a recursive function that, for each start_node, traverses through the graph to find all nodes connected to it (Cormen et al., 2009). One exception from a regular implementation of a depth-first search algorithm can be seen at Algorithm 1 Line 20 of Algorithm 1. To deal with the exclusive clearing entries mentioned before, we decided not to include them. The audit professionals that were consulted as part of this study indicated that these events are almost never relevant for analysis. These exclusive clearings do, however, lead to problems in Algorithm 2 as they lead to undesired graph merges.

Algorithm 2 is a way of merging all graphs resulting from the GetConnectedSubgraphs function into graphs. The reason we do this is that we do not just want to see all documents related to the start document but all related documents to any other document in the graph. Basically, it is a way of overcoming the characteristic of a depth-first search that can only handle one start node. This is achieved by a double loop over all the instance graphs in the dictionary from Algorithm 1, where we check if the intersection of two subgraphs does not return an empty set. If this is true, we take the union of the sets and put that back into the dictionary. The original sets are removed from the dictionary. This way, we obtain a dictionary of subgraphs, where each non-exclusive clearing, node is only in one subgraph.

5.2 Running example

In this section, we will provide a simple running example to show how the graphs are generated. As input for this graph, we use a very basic transaction, with three invoices (19##), two payment runs (20##), one post with clearing activity (12##), and one exclusive clearing (10##) [2] [3]. The input files for the graph are shown in Table 5.1[4]. The step-by-step process is shown in Figure 5.2. The steps are as follows:

- **Algorithm 1 - Iteration 1 to 3:** This shows the three iterations of Algorithm 1. In these iterations, subgraphs are generated for each start node in Table 5.1b. Entries from the adjacency list relevant to each of the edges in these instance graphs are indicated by the boxes with the dotted lines[5].
- **Algorithm 2** The right bottom figure in Figure 5.2 shows how Algorithm 2, the merging algorithm, works. Nodes 2000000799, 2000000800, and 1200000176 are shared by multiple instance graphs. Therefore, the union of these graphs is taken.

[2]Note that for different data sets and different graphs these number ranges can differ.

[3]In this data set a regular process execution is: book invoice - initiate payment run - post with clearing (clear open payment runs).

[4]Self loops occur when a document posts a journal entry item which does not require clearing from another journal entry, thus it is not posted as an open item.

[5]Note that in this example we do not need to use start nodes 1200000176, 2000000799 and 2000000800 as their instance graphs are completely part of another instance graph.

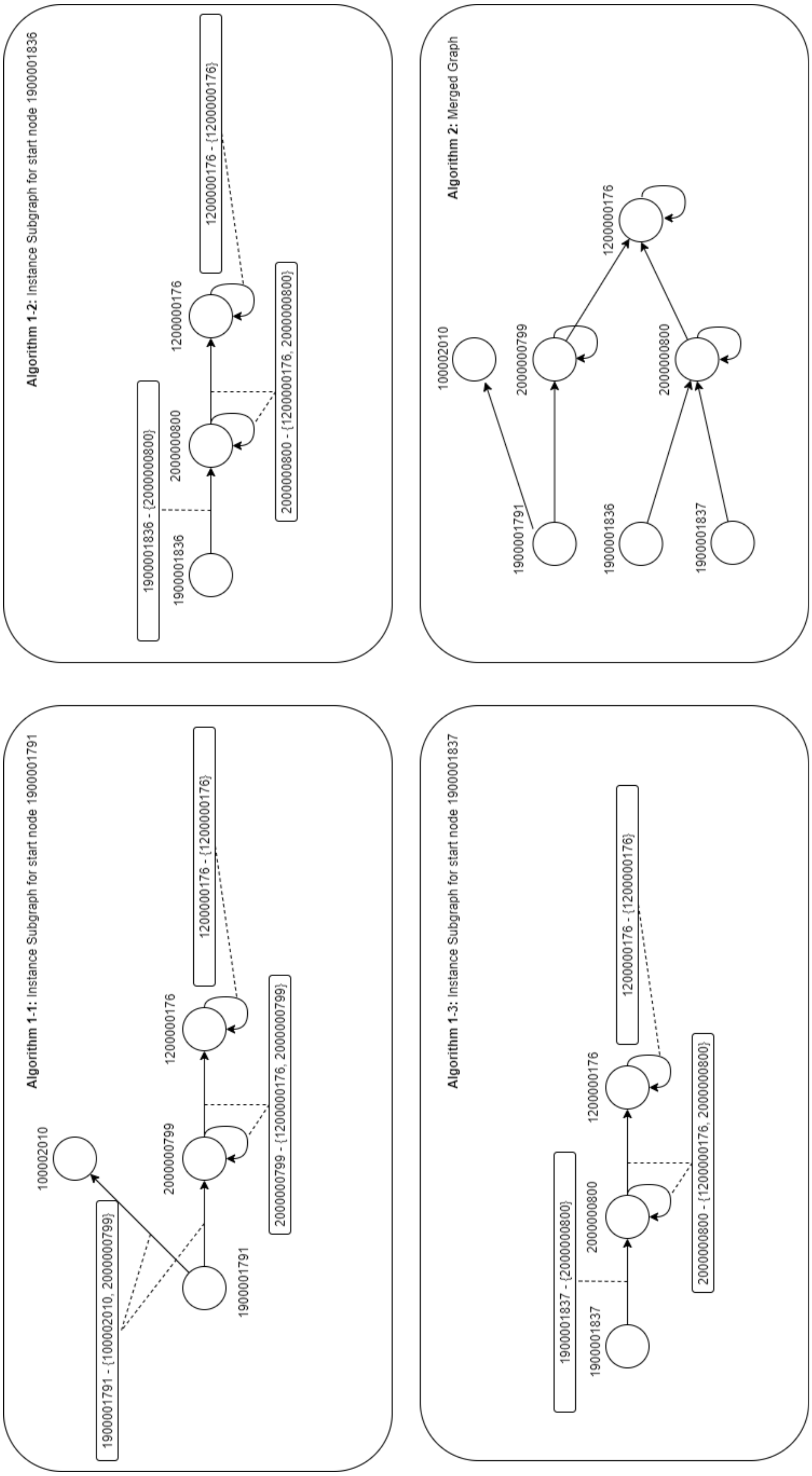


Figure 5.2: Step-by-Step Running Example of Graph Generation

Node	Adjacent nodes
1200000176	{1200000176}
1900001791	{100002010, 2000000799}
1900001836	{2000000800}
1900001837	{2000000800}
2000000799	{1200000176, 2000000799}
2000000800	{1200000176, 2000000800}

(a) Input Graph as Adjacency List

Start nodes
1200000176
1900001791
1900001836
1900001837
2000000799
2000000800

(b) Input Start Documents

Table 5.1: Input Data for Running Example

5.3 Improving Graph Readability

A drawback to the graph mining algorithm introduced in this chapter, is that graphs resulting from the algorithm can become enormous. Especially if the auditee uses certain accounting systems, like SAP, or -principles, like the batching of bank runs. An example of how big these graphs can become is shown in Figure 5.3 (labels in this figure have been omitted). To provide the auditor with an easy overview of patterns in a graph, we implement a way of simplifying the graphs. This process is shown in Figure 5.5, and the result of it on the graph of Figure 5.3 is shown in Figure 5.4.

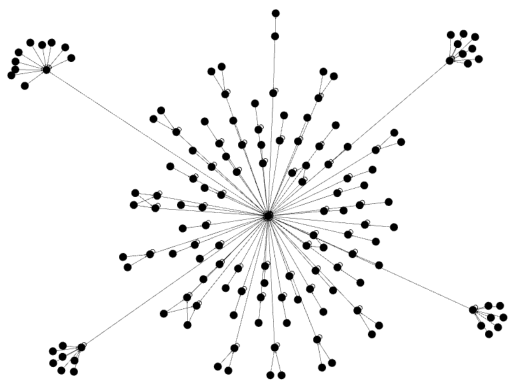


Figure 5.3: Big Generated Graph (147 nodes, 208 edges)

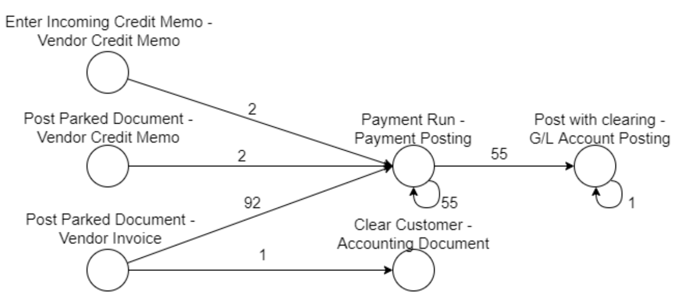


Figure 5.4: Real Simplified Graph with Weight on Edges (6 nodes, 7 edges, 208 edge weight)

The process works with the following steps:

1. **Step 1:** We take the graph mined by the Graph Mining Algorithm and map each type of node to a NodeType[6].

2. **Step 2:** We label each edge in the graph by concatenating the labels of its source and target nodes.

3. **Step 3:** We group by all nodes and edges and count the occurrences of each type of node and edge. Next, we plot the graph again and classify the occurrences of the nodes and edges as their weights. In the example edge, AE has as a label $AE-2$, meaning that in the original graph, there were two edges going from a node of type A to a node of type E. The graph, therefore, is now a grouped graph, meaning that we only have one visual element of each node and edge.

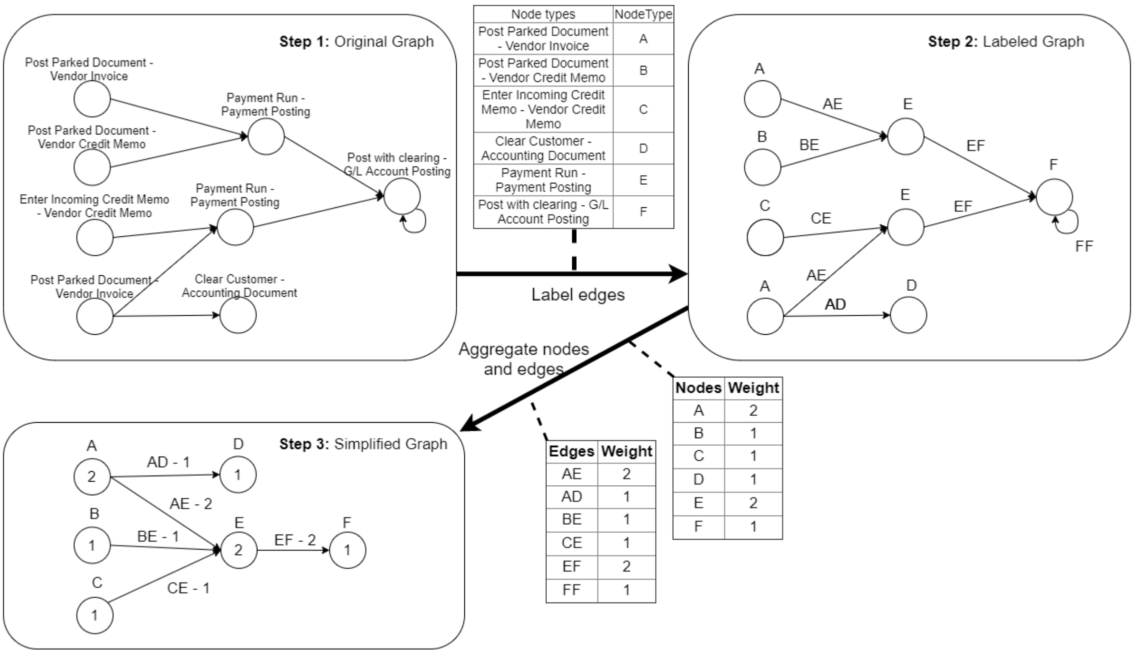


Figure 5.5: Graph Simplifying Process

5.4 Risk Flagging

Next to using the graphs we mined for supporting the auditor in their work, we can also make use of them by introducing a risk flagging process. This process serves as a systematic way of identifying graphs, or sets of documents, that violate risk rules defined by a user. These graphs can then be handed over to the auditors who conduct the APRA to be investigated further. By doing this, we provide these auditors with a more accurate starting point for their research as well as a suggestion of what to look for.

We formulated checks that will trigger for these risks and we divided them into five different categories. These checks and categories are defined in accordance with auditing experts. In Chapter 6, we show the results of the analysis for the implemented checks. In this section, we will discuss these categories and checks and explain why the flags are relevant and how they were implemented. A full overview of all identified checks in these categories is included in Appendix B.

[6] For the nodetypes in this case study we combined the transaction code with which an entry was posted with the document type it has in SAP. We relabel these node names to letters for clarity in explanation.

5.4.1 Segregation of Duties

The first category of checks we identified is on the segregation of duties between different tasks. In accounting, there should be a strong segregation in duties, to fix mistakes, and to prevent fraud (PCAOB, 2017). Even though in most ERP systems, this is enforced with internal controls and procedures, it is still worthwhile checking for it. If an auditee has a process in which this is not enforced or not maintained, it certainly poses a risk to the integrity of their books. Basically, hereby it is checked whether the auditee sticks to the four-eyes principle (PCAOB, 2017).

Check 1: Invoice Booking to Payment Run

For this study, we made the decision to focus on the segregation of duties between the booking of an invoice and the initiating of a payment run. In these payment runs multiple invoices are batched and approved for payment at the same time. If the same employee enters the invoice, as well as initiates the payment run there is a high risk that possible (deliberate) mistakes by this employee go undetected.

This check was implemented using the instance graphs from Algorithm 1, i.e. the graphs before merging. For each instance graph, we isolate the invoice and payment run documents. Because of the way these instance graphs were mined, there can only be one invoice document in each instance graph. To execute this check, we search the rest of the instance graph for a payment run done by the same employee.

5.4.2 Changes to Related Documents During Process

The second category of checks that we identified is related to system recorded changes from the changelog (Section 2.3.1). This category is relevant because changes recorded in this changelog are not visible anymore in the final invoice data. If anyone tried to cover up anything, the changes in these logs can show exactly what has happened, when and by whom. We attempt to extract the changes from these logs, which are possible indicators for mistakes.

Check 2: Changes on Vendor Master Data

One of the checks regarding the changelogs is the check on changes in vendor master data[7]. For this check, all graphs for which changes in the vendor master data have been made between the first and the last document of the graph have been flagged. If changes are made to important vendor information just before or after a document has been booked, this could indicate previous mistakes. An example of a change that could be made to vendor master data that could be relevant for an auditor is the removal of a payment block just before the invoice is booked, or a complete change of the name of a vendor or its IBAN number.

This check was implemented by taking the minimum and maximum date of any document in a graph (active time of the graph), as well as all vendors related to any document in the graph. It is then checked for all of these vendors if changes have been made to their master data in the time window in between the minimum and maximum date of the graph.

Check 3: Changes on External Invoice ID

Another check regarding the changes in the changelogs concerns changes on the external invoice ID of an invoice. The external invoice ID is the identifier that the vendor originally put on the invoice document, which differs from the invoice ID used in the system of the auditee. Changes to the external invoice ID could indicate either a mistake in entering it in the first place, or linking it to a whole different vendor invoice document. Both of these cases increase the possibility that something about a particular invoice is off.

[7] The vendor master data consists of general information for each vendor, e.g. name, country, industry category, and bank account.

The implementation of this check is fairly straightforward. For every document in the graph, it is checked in the changelogs if there are any changes to its external invoice id. If this is the case, a flag is raised for the graph and the documents, as well as the changes to the external invoice ID, are returned.

5.4.3 Non Invoice-Referenced Payments

The third category of checks that we identified is related to payments that are not related to an invoice document. This category is relevant because these payments circumvent control measures put in place in the auditee's ERP system. Also, currently, these documents are not considered in the APRA process, as this focuses solely on invoice documents. If this type of payment occurs frequently, it possibly introduces errors into the process as well, as these payments do not undergo the regular control structure.

Check 4: Payments Directly to General Ledger, Without Invoice

One check in this category is to see if we can find payments directly to a general ledger cost account, without a linked invoice. If this occurs, the costs are not linked to an invoice and thus not linked to a vendor account. This circumvents any regular controls on the vendor level. Because these controls are not enforced, if this type of transaction occurs, it always has an increased risk.

This check was implemented by first identifying all cost accounts in the general ledger of the auditee. Next, all general ledger bookings were retrieved. If these general ledger bookings were not cleared by a payment run or bank run, and if they have a debit entry to one of the identified cost accounts, they are flagged.

5.4.4 Document Amount Additions

The next category concerns the rule that the amounts in all documents that are cleared by the same document should add up to the same amount as the clearing document has. If this is not the case, it could be that either documents are missing or that some other irregularity has occurred.

Check 5: Payment Runs Sum up to Bank Run

For this category, the decision was made to focus on whether the payment runs sum op to the bank runs, i.e. do the proposal payment run and the final payment run match up. There can be different reasons why this is not the case, but most of these warrant further research into the underlying documents. All flags raised by this check should be checked to see whether the differences can be explained.

Implementing this check was done by summing all debit amounts for all payment run documents in a graph and comparing them with the sum of all bank run debit amounts. If these amounts do not match up a flag is raised on the graph and the difference between the payment runs and the bank runs is returned. This can then be used as a starting point for further research.

5.4.5 Other

Some checks were identified that did not go in one of the other categories. These checks can still be relevant for the analysis as they could indicate either parts missing, outliers, or unusual graph structures. Especially if any flags are raised for one of these checks, while concurrently other flags are raised, it increases the risk for the documents in a certain graph.

Check 6: Outliers for Average Invoice Amount

The check that was implemented for this category is one on finding the graphs that have relatively high average invoice amounts. From a control perspective, it is interesting to see which processes concerned high monetary amounts. This can either serve as an extra indication of urgency for any

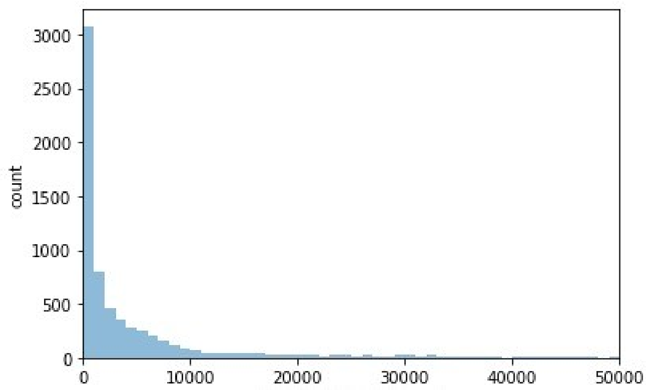


Figure 5.6: Distribution of Average Invoice Amounts in € (Cut Off at €50.000 for Readability)

of the other flags raised, but could also be used to check for instance if the same employees are always involved in these processes.

First, we select only the debit lines of each invoice document in each graph. Next, we group these by document ID and take the average for this group. This way we obtain the average amount that was on the invoices as they were originally posted. Finally, we selected the 95th percentile of this selection. One could also argue to take a different percentile. We decided to take the 95th percentile of the data instead of selecting the outliers by selecting everything that is over the mean + x*standard deviation. We did this as from a closer investigation of all the average invoice amounts in the studied dataset the distribution we could not conclude that it followed a Gaussian distribution (Figure 5.6 clearly does not follow a Gaussian distribution).

5.5 Summary

In this chapter, we have introduced the model that we propose to mine the graph from the recursive clearing data structure in financial accounting. First, we introduced the algorithm itself. Next a way of improving the interpretability of the results to make them easier to use as a tool to support the auditors at the AC. Then, we have shown a running example of how the algorithm would perform on data from a real-life process. Finally, we introduced the manner in which we propose to raise flags on suspicious graphs.

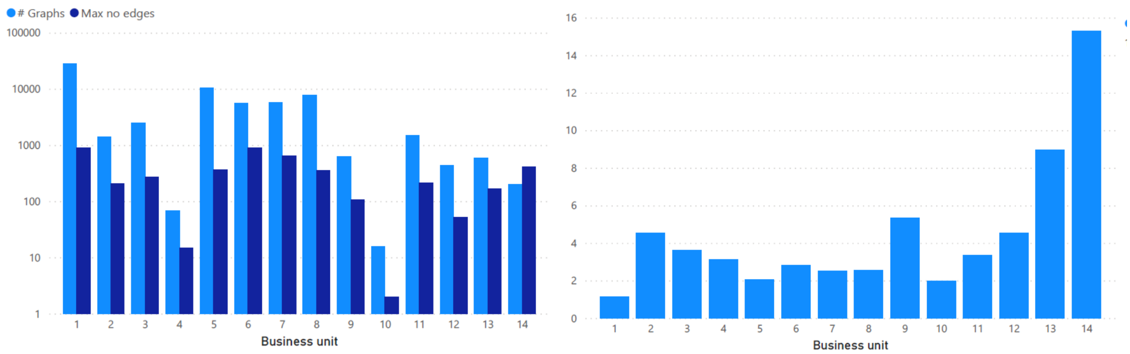
Chapter 6

Results

In this chapter, we will show the results we got by mining the graphs, as well as by flagging these graphs for risks. As a proof of concept, we implemented the six flags, as discussed in Section 5.4. We will discuss the results for each flag and give some examples of them.

6.1 Discovered Graphs

We will begin by describing the results of applying the graph generation algorithm on the data set. The graph generation algorithm introduced in Section 5.1 returns for each business unit a set of graphs. To get a general understanding of these graphs, Figure 6.1 shows some general statistics for these discovered graphs. The number of graphs (Figure 6.1a) and maximum graph size [1], combined with the average graph size (Figure 6.1b), provide a general insight into how bookkeeping is done at the different business units. A high average graph size indicates that most payments are batched in some way, while a low average graph size indicates a lack of batching. If a business unit has a relatively high maximum number of edges with a low average graph size, this likely indicates some invoices are handled in (large) batches while the majority are processed separately. Business unit 1 provides a good example of this. Table 6.1 provides an overview of the same data as Figure 6.1



(a) Number of Graphs and Maximum Graph Size per Business Unit (Logarithmic Y-Scale)

(b) Average Graph Size per Business Unit

Figure 6.1: Statistics for Discovered Graphs In Data Set

[1] We define graph size as the number of edges

Business unit	# Graphs	Max no edges	Average # Edges
1	28235	910	1,1538
2	1447	210	4,5743
3	2534	273	3,6563
4	70	15	3,1429
5	10702	369	2,0640
6	5624	917	2,8512
7	5796	665	2,5330
8	7878	359	2,5590
9	633	108	5,3491
10	16	2	2,0000
11	1502	218	3,3682
12	441	53	4,5669
13	592	171	8,9848
14	203	419	15,3103
Total	65673	917	4,4367

Table 6.1: Number of Discovered Graphs, Maximum Graph Size and Average Number of Edges per Business Unit

6.2 Results on Checks

In this section, we will present the results found by the checks that were identified to be most relevant in Section 5.4. Four out of six of these checks yielded results, while two did not. Because these checks aim to find anomalies in the bookkeeping process it is no surprise not all of them yielded any results. On the contrary, it is a good sign when these checks do not raise flags. It indicates that the company in audit is not exposed to this kind of risk. For each check, we will show how often it has occurred in which business units and show an example of a hit, Finally, we will discuss the checks that did not raise any flags. Every flag that was raised indicates some sort of risk and is therefore potentially useful to investigate in an APRA. In Section 7.1, we will evaluate the factual correctness of these checks and in Section 7.2.1 we will evaluate the perceived readiness for implementation at the AC. Please note that in the coming sections the values on the y-axis differ for each figure. Check 1 and 5 did not return any flags for the given case study and are therefore combined in one section at the end.

Check 2: Changes to Vendor Master While Processing

Check 1 uses the changelogs (Section 2.3.1) of an ERP system to discover whether changes have been made to vendor master data while the invoice is being processed. Figure 6.2 shows how many flags were raised by this check on each of the business units. Changes to the vendor master data could be innocent, like changing the address, but also more suspective like a payment block being removed and later put back on. From this graph, it stands out that Business Units 6 and 7 have a high number of flags for this check, while they are medium-sized business units.

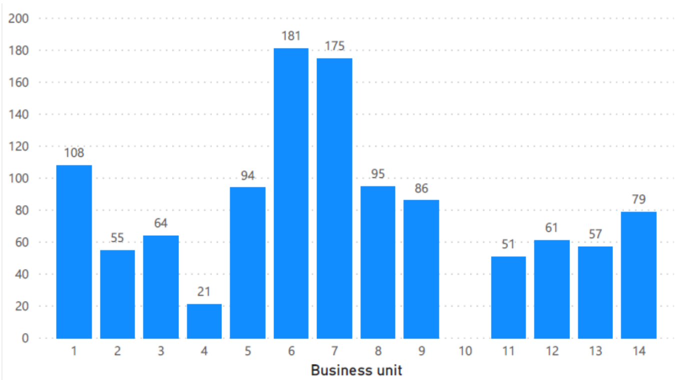


Figure 6.2: Flags for the Check on Change of Vendor Master Data During Booking Process

Check 3: Changes to External Invoice ID

Another check that was implemented using the changelogs is whether changes have been made to the external invoice number of an invoice. Figure 6.3 shows how many flags were raised by this check on either of the business units. Changes to the external invoice number are often suspicious because they could bypass the automated checks set by the ERP system. Changing the external invoice ID after creating the record could lead to the ERP system not triggering in case of an identical external invoice ID, possibly introducing duplicates. From this figure, especially business unit 2 stands out, as it is a small business unit (Figure 4.3) with large graphs on average but still has a lot of flagged graphs.

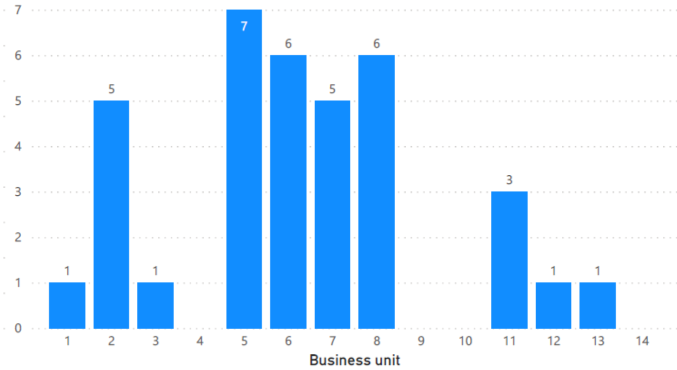


Figure 6.3: Flags for the Check on Change of External Invoice ID

Check 4: Payments Directly to General Ledger, Without Invoice

Check 4 looks for cost bookings that were made directly to a general ledger cost account, without an invoice. Figure 6.4 shows how many flags were raised by this check on either of the business units. Payments directly to a general ledger cost account are a risk as booking payments this way circumvents most control measures put in place to control the invoice handling process. From Figure 6.4, business units 5 and 8 stand out, as many flags were raised for this check in those business units.

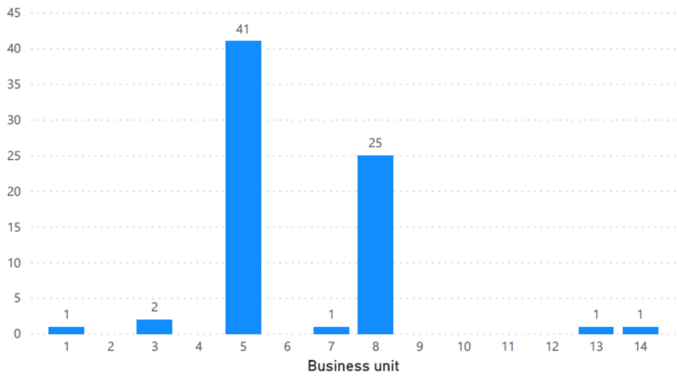


Figure 6.4: Flags for Check on Payments Directly to a General Ledger Account, Without Invoice

Check 6: Outliers for Average Invoice Amount

Check 6 aims to find outliers in the average invoice amounts in graphs, these outliers were determined against the average over averages over all business units. Figure 6.5 shows how many flags were raised by this check on either of the business units. This flag is especially useful in combination with other flags because if a flag is raised for this check in combination with a flag for another check, it indicates a graph with high possible risk and high possible reward for the recovery audit. From this Figure 6.5, business unit 6 stands out as it is a relatively small business unit, but a lot of flags were raised for high average invoice amounts.

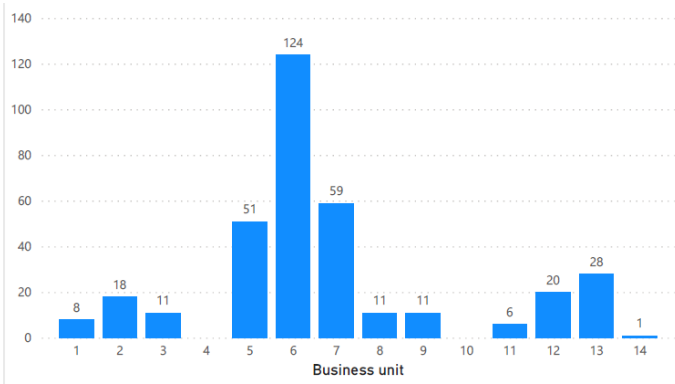


Figure 6.5: Flags for Check on Graphs With Average Invoice Amounts in 95th Percentile (outliers)

Check 1 and Check 5: Checks Without Flags

Two of the checks that were implemented, check 1 and 5, did not return any flags for the data set. Check 1 is the check if all payment runs included in a bank run added up to this bank run. Check 5 is whether there is a clear segregation of duties between the person who books the invoices and the person who initiates the payment runs. It is a good sign for the auditee that these checks did not return any flags, as it indicates that for these controls, their process is well set up. In Section 7.1, we show evidence that these checks did in fact work.

6.3 Summary & Conclusion

In this chapter, we have shown the results of both the graph mining algorithm and the flagging of checks on these graphs. First, we have shown general statistics about the graphs that were mined and what can be learned from them about the process in different business units. Next, we have shown the results of the six checks that were implemented. Four out of six of these checks returned flags, while two did not. From this we can conclude that the process at the auditee for this data set was fairly well regulated with respect to the checks we defined.

Chapter 7

Evaluation

In this chapter, we will evaluate the results we obtained in the previous chapter. This evaluation is split up into two parts. First, in Section 7.1, we will evaluate the factual correctness of the graphs mined and the checks executed. Next, we will evaluate the results with several experts using interviews. This user evaluation in Section 7.2 is split up into an evaluation of the usefulness and applicability of the checks in and an overall survey following the Technology Readiness Level (TRL) and Technology Acceptance Model (TAM).

7.1 Model Accurateness

First, we will evaluate the accurateness of the graphs that we mined and the checks that were implemented. Because we do not have a ground truth for the the data of this case study, we made a clear test set which we were able to check manually for this evaluation. This test set consists of a subset of the data set that was used for this study. We checked this data manually to ensure it did not raise any flags before we introduced them. First, in Section 7.1.1, we manually compared five of the identified graphs against the source data to verify their accurateness. In Section 7.1.2 we deliberately introduced five errors to this sample set for each of the checks we implemented. In this section, we will also evaluate whether these errors are flagged by the algorithm, thereby also evaluating the accurateness of these checks. All of the introduced errors were verified in accordance with one of the audit experts at Auditing Company (AC).

7.1.1 Graphs

To evaluate the creation of graphs, we randomly sampled five graphs out of the evaluation data set. We will check the generation of these graphs by verifying them against the original input data set. These graphs are shown in Figure 7.1. For each of these graphs, we manually checked all nodes to see if the graphs were complete. In checking this, the only clearing edges that were not included in these graphs were the edges that we identified as exclusive clearing edges in Section 5.1. Therefore, it appears that the graph mining algorithm provides accurate results. In these graphs, it is worthwhile noting that the graph in 7.1e is not a complete booking, as it is only of a graph with a diameter of 1[1]. Verifying this in the original data shows that this is a partial booking from the end of the year in scope. This is a result of one of the limitations of the studied data set.

[1] The diameter of a graph is the greatest distance, in number of edges, between any pair of nodes (documents) (Cormen et al., 2009)

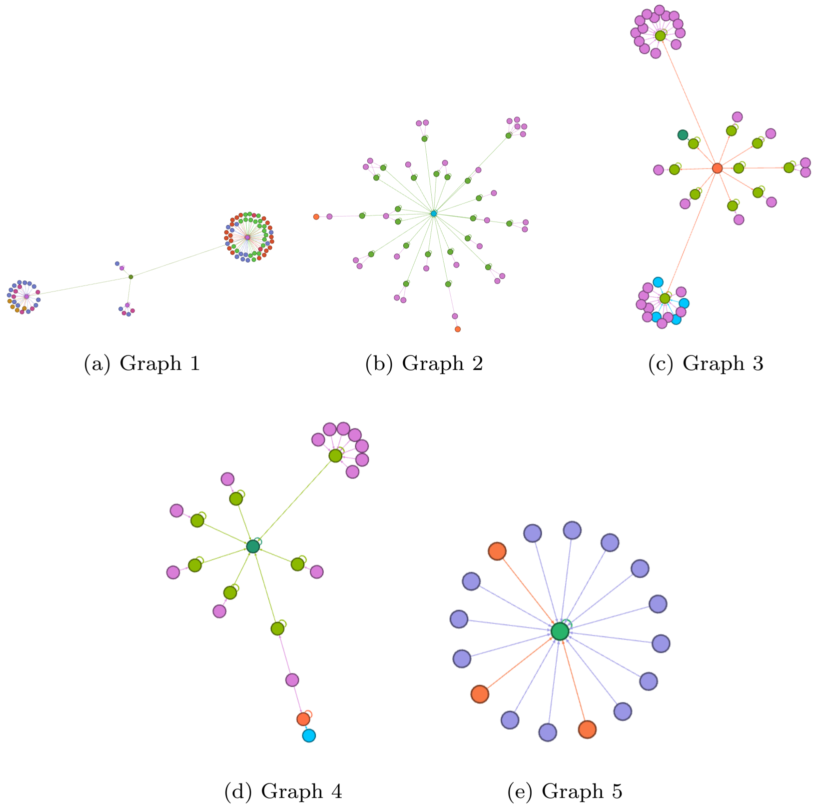


Figure 7.1: Five Discovered Graphs for Evaluation

7.1.2 Checks

To evaluate the precision and recall of the checks that were implemented, we manually added five anomalies to the evaluation data set. We then ran the algorithm and all checks to verify whether the entries that were supposed to be flagged, were indeed flagged. The final check for the outliers in invoice amounts was verified manually as we are not able to . The anomalies we introduced for each check were the following:

- **Check 1: Segregation of Duties: Invoice Booking to Payment Run**

  We randomly took out three invoice bookings of which we changed the entry user to the same user as the payment run. Next, we took two payment runs and changed the entry user of one of their respective invoices to the same user that entered the payment run. We expect to find get flags for all five of the sets of invoice and payment documents.

- **Check 2: Changes to Related Documents During Process: Changes on Vendor Master Data**

  We randomly sampled five vendors out of all entries in the evaluation data set. For three out of these five vendors we added changes that occurred during the time window of the process. For two vendors we added change that were outside of this time window. We expect to find flags for the three vendors with changes within the time window, but no flags for the changes to vendors outside the time window.

- **Check 3: Changes to Related Documents During Process: Changes on External Invoice ID**

  We randomly add changes to the external invoice ID for five of the invoices that were in the evaluation data set. We expect to find all these five changes, as for this check the moment of the change is not a condition.

- **Check 4: Non Invoice-Referenced Payments: Payments Directly to General Ledger, Without Invoice**

  We added five payment bookings to the evaluation data set which are done directly to the general ledger and tagged these with different IDs to make them easy to recognize. We expect to find all five of these extra bookings.

- **Check 5: Document Amount Additions: Payment Runs Sum up to Bank Run**

  We randomly changed the amounts of three payment runs and two bank runs from different sets in the evaluation data set. We expect to find flags for five graphs which each contain one of these changed amounts.

When running our algorithm and the checks we found all flags we expected to. Except for the two changes we made to the amount in bank runs for check 5. However, after manual verification we found that this was due to these bank runs not having any payment runs, but rather being intercompany postings which follow a different process. After taking these graphs out and re-running the checks we found all flags we expected to find. For check 6, the outliers, we manually calculated all averages and verified that the graphs flagged by the check were indeed the outliers.

7.2 User evaluation

For the user evaluation part of this chapter, we conducted six semi-structured interviews with different stakeholders in the APRA process at the Auditing Company (AC). We tried to get as representative as a sample as possible with these stakeholders. In these interviews, we presented the graphs mined from the model as well as the results from the checks we implemented. For the checks that did not return any results for the given data set, an explanation was given, and a sample with manipulated data of what the results would look like if there were any. In these interviews, we used a combination of questions coming from the Technology Readiness Level (TRL) model for the readiness of the implemented checks and the Technology Acceptance Model (TAM) for the user acceptance of the mined graphs and checks.

7.2.1 Readiness of Implemented Checks

The readiness of the implemented checks was evaluated using a method based on the Technology Readiness Level (TRL) model (Héder, 2017). In this method, we first explained the six implemented checks to our interviewees and showed them the results of these checks on the given data set. Subsequently, we asked them to rank each of the implemented checks in to one of five categories:

1. Directly useful (TRL 7-9); this check could immediately be implemented and used in a business context
2. Useful in the future (TRL 4-6); the usefulness of this check is clear. However, it is not clear whether it can be implemented in business straight away.
3. Needs further research (TRL 1-3); this check shows potential, but the usefulness and ability to implement the check are unclear.
4. Not useful; the potential use for the results of this check are unclear.
5. Unknown; The interviewee is not sure about the potential usefulness of this check or the ability to implement it.

The results of this part of evaluation are shown in Figure 7.2. It appears that the users we interviewed have a high degree of confidence in the usefulness of the checks that were implemented. Especially the checks of the booking to a general ledger cost account without invoice reference and the segregation of duties. The explanation for the high trust in the usability for the booking to a general ledger cost account was explained by some of the interviewees because it is something that is not included in the current process of the AC. Currently, only bookings related to an invoice are considered in the analysis, so anything that could include other kinds of payments into their process is welcomed. For the check on the segregation of duties, there is also a consensus amongst the interviewees, as they agree that if this check is violated, it always provides a risk to the correctness of the books and it always warrants for further research.

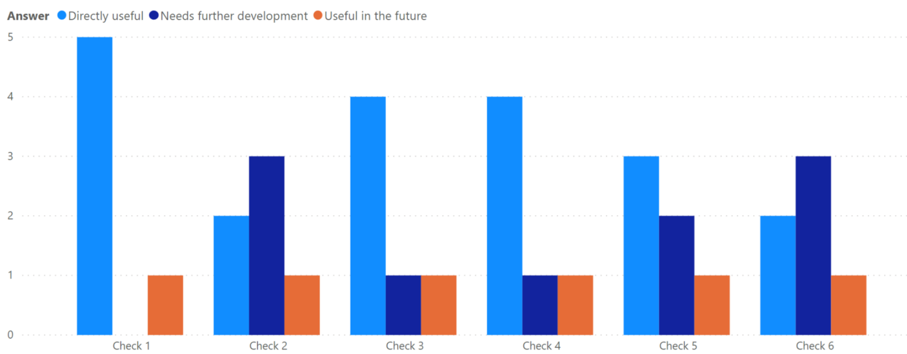


Figure 7.2: Perceived Usefulness/Potential for Implemented Checks

For the check on the changes on vendor master data, some of the interviewees indicated that they saw the potential but were not sure in what way. The most heard comment was that the results from this specific check were too coarse and more development is needed either on the business-end or the development end to figure out which attributes are important. For the check on additions and outliers for graph amounts, it was not clear for the users that were less closely involved in the auditing process how this would help an auditor perform their task. Therefore, they indicate that these needed further development.

7.2.2 User Acceptance

The user acceptance was evaluated using a method based on the Technology Acceptance Model (TAM). This approach was first introduced by Davis (1989). Our survey was, however, based on the work by Lai (2017). In their work Lai (2017) introduce four categories for the questions in the questionnaire for the TAM: perceived usefulness, ease of use, satisfaction and intention to use. For our questionnaire we tried to adapt these questions as good as possible, resulting in the questions in Appendix C. Because we were only able to completely explain and show the results of this study to six stakeholders, we chose to go for a semi-structured approach for gathering results in this model and not for a remote questionnaire. In these interviews, statements were discussed and the interviewees were asked to rate the degree to which they agreed with the statement and provide some explanation for their answers, if they had any. Because the interviews were semi-structured, if interesting explanations were given, we were able to elaborate further on them by asking follow-up questions.

The results of this part of the evaluation are shown in Figure 7.3 to Figure 7.6. Overall it appears that the interviewees mostly agree with the statements as they were posed. For the perceived usefulness scores in Figure 7.3, what stands out most is the difference in perceived improvement in efficiency and effectivity. The main reason that were given for lower scores on these questions were that some of the interviewees were not able to immediately envision how this model would work in a practical setting. Therefore there was also some discussion on how much the quality and quantity of the risks would differ.

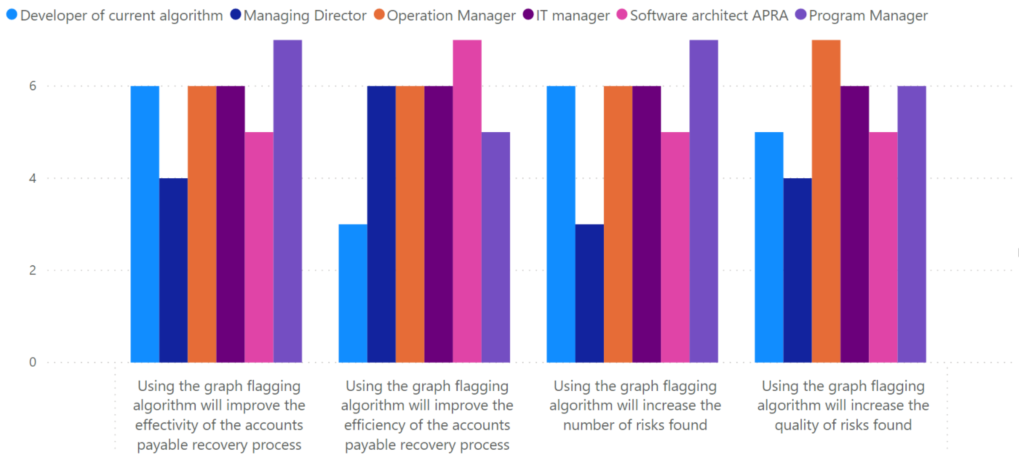


Figure 7.3: Results from Perceived Usefulness Category of Technology Acceptance Model (TAM) Questionnaire (1: Disagree - 7: Agree)

For the ease of use category in Figure 7.4 the question whether the algorithm provides the right information scores lowest. The reason that the interviewees gave for this answer was that the output was still a bit messy and would need some fine-tuning and polishing in order to be used in a business setting.

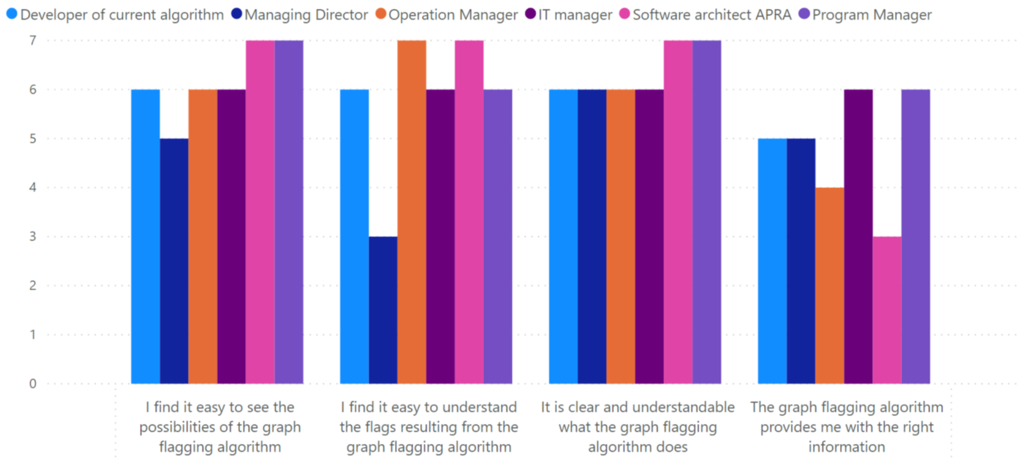


Figure 7.4: Results from Ease of Use Category of Technology Acceptance Model (TAM) Questionnaire (1: Disagree - 7: Agree)

In the satisfaction category in Figure 7.5 we see that scores are overall high (5+), therefore we can conclude that we delivered a satisfying result. Some of the interviewees did indicate that they saw room for improvement in how the results were presented as well as in the terms of the results.

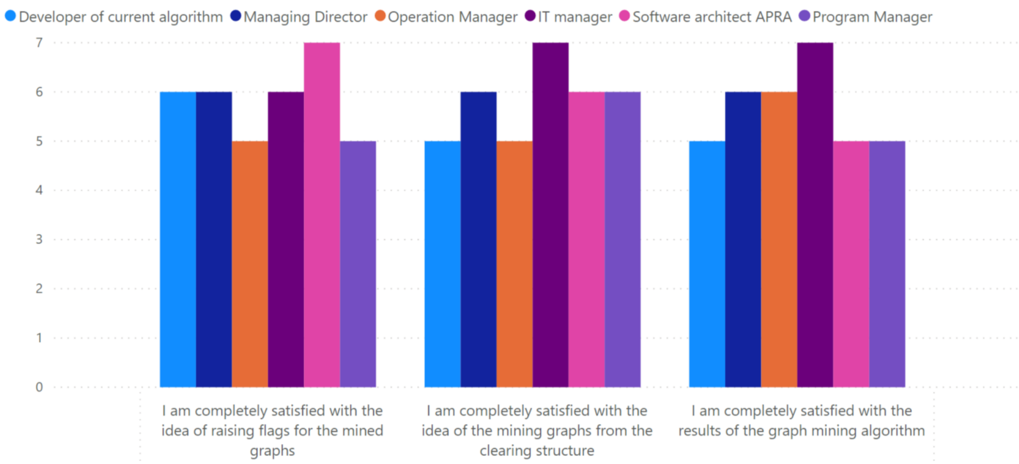


Figure 7.5: Results from Satisfaction Category of Technology Acceptance Model (TAM) Questionnaire (1: Disagree - 7: Agree)

Finally, in the intention to use category in Figure 7.6, we see that one of the interviewees did not expect the algorithm will be used frequently. The reason that was given for this, was that there are some doubts about the willingness to accept it in the business setting because of reluctance to use new techniques auditors might have. However, in general the intention to use, too, gets high scores (all 5+).

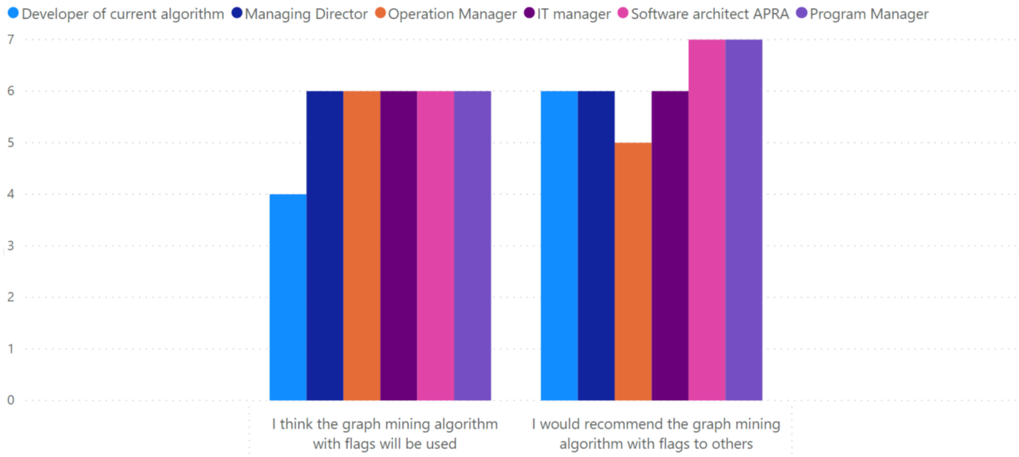


Figure 7.6: Results from Intention to Use Category of Technology Acceptance Model (TAM) Questionnaire (1: Disagree - 7: Agree)

7.2.3 Suggestions from Interviews

A suggestion that came up in every interview was to put the algorithm to the test for many more data sources and data sets. It is necessary to be absolutely certain about the workings of it before it is implemented fully for the business. Another suggestion in this is to add more specific information on which element of a graph exactly triggered a flag to be raised for this graph. Also, the suggestion was given to color code the nodes in the graph with the type of journal entry they represent to provide a faster insight. The remark was also made that the checks that did not return any results were no surprise. The measures that these checks enforce are almost always set as standard controls in the ERP system of an auditee.

Furthermore, some of the interviewees saw the greatest potential in the short term in using the graphs to provide insight to support the current Accounts Payable Recovery Audit (APRA) process. Providing insight into how invoices were handled at the auditee can provide an auditor with very valuable contextual information. An additional remark that was made as well, was that the usage of the information from the vendor master change log could also be valuable for another service that is provided by the AC, the service of reporting vendor master data.

Chapter 8

Summary & Conclusion

8.1 Summarizing Conclusion

In this study, we set out to find how process mining techniques can be used to enhance the quality and quantity of the risks found in an Accounts Payable Recovery Audit (APRA). From a review of related literature in Chapter 3 we concluded that an accounting logic-based process mining technique was most suitable to this end. First, in Chapter 4, we described the data set that was used in this study. Next, in Chapter 5, we introduced an algorithm to mine the process graphs and a practical application of those graphs with the risk checks. Finally, in Chapter 6 we have shown the results of this algorithm and its checks and evaluated these in Chapter 7. In this chapter, we will explain how we came to these results and what are their implications and limitations. Finally, we will conclude with recommendations to the company this study was conducted at and list possible future work.

The first sub research question we formulated was *how we can mine process graphs from the accounting structure to infer dependency, instead of using the temporal logic that is used in many other studies?* In Chapter 5 we have introduced a model that makes use of the recursive clearing logic that is present in many present-day ERP systems. Following this logic, the accounting part of an ERP system has two main tables. A table with the journal entries and one with the journal entry items belonging to these items (Figure 2.2). A journal entry consists of a set of two or more journal entry items that can each be cleared by one other journal entry, hence the recursive structure. This clearing relation provides an indication of the sequence in which these documents were created, thus the sequence in which the processes were executed. Therefore, the sequence obtained from recursively mining these clearings provides us with the order in which all documents regarding the accounts payable process of a payment. This approach differs from the temporal approach, as described in Section 3.2.1, in that it does not need one entity to be followed through the system. Therefore, a more accurate and truthful depiction of a process can be created, especially in those processes that have a high degree of convergence and divergence (Section 3.2.1).

The second sub research question of this study concerns *the data that is required for this accounting logic based mining process, and where to find it in different ERP systems.* In Chapter 4, we described the minimum amount of data that is required for this. As described in the previous paragraph we need at least a table containing the journal entries and one containing the journal entry items. There are many different ways in which this information is stored, as well as many different names it goes by. This study was executed on a data set coming from a company working with SAP. Therefore, the most detailed explanation of how and where to gather this data has been provided for SAP. However, the recursive structure of journal entries and journal entry items is one that many document-based ERP systems share. The overview in Section 4.2 confirms this by providing the tables and fields in which this clearing relation can be found in some other widely used systems, even though in those systems, the relation is not referred to as clearing but for instance as closing.

The third and final sub research question of this study concerns *the possibility of developing checks to improve the quantity and quality of the risks found in an Accounts Payable Recovery Audit (APRA) based on the mined graphs.* In Section 5.4, we have shown the possibility of developing these checks by describing several of those checks. In cooperation with auditors at the company this thesis was written in, we have developed five categories for those checks. For each of those categories we came up with several checks that could provide an auditor with possible risks worth evaluating in an APRA. In Chapter 6, we have shown that running these checks did indeed return results on the studied data set, and in Chapter 7, we have evaluated the accurateness of these checks, their implementation readiness, and the acceptance of the technique by some of the stakeholders. Even though relatively few flags were raised by the formulated checks on the studied data set, we were able to show that there are opportunities in the methods suggested in this study. The responses from the stakeholders were mainly positive in that they have confidence the presented methods will be of added value to their current way of working. The main point of improvement they indicated comes from the presentation of the results of the checks to the end-users.

Concluding, we can state that using the model described in Chapter 5 provides an interesting and valuable addition to the practices of the Auditing Company (AC). First, we have shown that it is possible to infer structure and mine graphs from the way journal entries are stored in an Accounting Information System (AIS). Second, we have shown that it is possible to run checks on the graphs mined with this model and flag entries that violate defined rules. Finally, we have verified that the results from these checks are accurate from a technical viewpoint. But, equally importantly, the perceived usefulness and potential of the technique presented, from the business perspective, is promising.

8.2 Limitations

One of the limitations of this project was the lack of experience in the complex area of audit of the writer before writing this thesis. The focus on research has, from the beginning, been on the more technical, process mining, research. Therefore, probably, research from the audit research area has been missed in this research. Also, because many concepts only became clear after writing a larger part of the thesis, in retrospect some decisions and assumptions could have been made differently. An example of this is the way we sampled data. We sampled data from one year, while looking back it would have been better for this study to sample data for bookings that started in one year. That way we would not have had to deal with the interpretation issues of booking that stretch over the years. Another example is that, in retrospect, we would have opted to download the spreadsheets out of SAP or use SAP's built-in query function to retrieve the data as we cleaning and preparing the data from the extracted text files was a long and tedious process. This process could have been more efficient if we extracted the data in a different way.

Another limitation to the executed research, from the perspective of the company this study was executed in, is that not all of the data required for this type of research is always available. For the studied data set, for example, we were not able to retrieve data from the BSEG table of SAP. We had to overcome this by collecting data from FBL1n and FBL3n instead, leading to the absence of customer journal entries. Another data-related limitation is that we are not certain that every AIS that is encountered by the AC (over 200 different systems) contains the data needed for mining the graphs, even though we based our algorithm on a very basic accounting structure.

8.3 Recommendations

Following the conclusions of this study stated in the beginning of this chapter, some recommendations can be made to the company this study was executed in. The first recommendation is straightforward: try incorporating the process perspective of the financial documents into the APRA process. In this study, we have shown that valuable insights arise when the process structure the accounts payable process of these documents is disclosed. These are valuable insights when trying to understand why mistakes were made. It provides a whole different perspective from which to view at a process. As a legendary Dutch football player (Cruyff) would put it: "You only see it when you realize it." [1]

A recommendation that goes hand in hand with the previous recommendation is to further investigate implementing the checks proposed in this paper and to look for more checks that the process perspective could be used for. With the limited knowledge and expertise the writer of this study had in the way financial audits are conducted, it is likely that the list of checks introduced is not complete. Implementing these checks improves the likelihood that more risks will be found, which in turn potentially increases the recovered funds and therefore increases revenue for the company.

A third recommendation is to do more research into the data that is available in the source systems of the auditees, which might be useful for the APRA. In this study we have shown several of these other data sources; the changelogs and all bookings to the general ledger. But there might be more data that is useful for this kind of audit. Other scholars have, for instance, added data about purchase orders and purchase requisitions, and about the receipt of goods. Data from for instance the parts of ERP systems that are used by the financial controllers also potentially contains valuable information to support the analysis.

A final recommendation to this company is to keep up with recent developments surrounding finding anomalies in an APRA context. During this study, for instance, we stumbled upon some very interesting research on the use of auto-encoder neural networks in detecting unusual journal entries (Schultz and Tropmann-Frick, 2020). This is just one example of a whole range of research that is being conducted in the professional area of AC.

8.4 Future Work

Following the conclusions and recommendations of this study also some directions for future work were identified. Most future work we identified is in identifying extra possibilities the graph structure brings to the types of risk flagging that can be done. One check, for example, that we came up at the very end of this study is to verify whether the temporal order of events aligns with the order mined from the clearing structure. It is likely that there are many more checks that can be formulated. Furthermore, it might be worthwhile to take a step back from the goal of Accounts Payable Recovery Audit (APRA) and put the focus on identifying fraud. When deliberately trying to identify fraud, one might want to look for different patterns than when looking for overall anomalies.

Besides extending the amount or purpose of the checks introduced in this paper, another possibility for future work lies in extending to including more documents of a document based ERP system. For instance, in SAP documents relating to purchase orders are also structured in a similar way as the journal entries and journal entry items analyzed in this study. Including these will give a more complete picture of the accounts payable process that was executed at the auditee.

Next, during the final stages of working on this thesis we stumbled upon a work by Werner (2019). This work takes an approach that is very similar to the approach that was taken in this study. There are a couple of differences between the results of this study and the work by Werner (2019). The first difference is in the focus. This study focuses on the use of the graph flagging algorithm specifically in a accounts payable recovery audit setting, while the study by Werner

[1] In Dutch: "Je gaat het pas zien als je het door hebt"

(2019) focuses on identifying missing internal controls. The difference between these focuses, is that the focus of the APRA and thus the AC is on finding money and of the work by Werner (2019) on missing controls, thus compliance. Furthermore, we have a broader approach to categories of risks, while their focus is mainly on risks in the segregation of duties category (17 out of 19 checks). A suggestion for future work is to combine the efforts made in this study with the efforts made by Werner (2019) and see where they complement each other. Especially to see which internal control checks can also be deployed to recover unjustified payments.

A final direction for future work we identified lies in combining the fields we identified in Section 3.2.2 and Section 3.2.3. If one sees the nodes in the graphs mined by the algorithm in this study (and in the work by Werner (2017)) as artifacts in a artifact-centric process mining approach, the mined clearing structure between those becomes the artifact interactions. Therefore, using the clearing structure in the accounting documents in an ERP could be an additional way to determine the interaction between specifically the accounting artifacts of this ERP system. Combining this technique with the existing artifact interaction discovery techniques is a possibility for future work.

Bibliography

Aalst, W. M. P. v. d. (2011). *Process Mining.* Springer Berlin Heidelberg.

Aalst, W. M. v. d. (2019). Object-Centric Process Mining: Dealing with Divergence and Convergence in Event Data. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11724 LNCS, pages 3–25. Springer Verlag.

Aalst, W. V. D., Adriansyah, A., De Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., Bose, J. C., Van Den Brand, P., Brandtjen, R., Buijs, J., Burattin, A., Carmona, J., Castellanos, M., Claes, J., Cook, J., Costantini, N., Curbera, F., Damiani, E., De Leoni, M., Delias, P., Van Dongen, B. F., Dumas, M., Dustdar, S., Fahland, D., Ferreira, D. R., Gaaloul, W., Van Geffen, F., Goel, S., Günther, C., Guzzo, A., Harmon, P., Ter Hofstede, A., Hoogland, J., Ingvaldsen, J. E., Kato, K., Kuhn, R., Kumar, A., La Rosa, M., Maggi, F., Malerba, D., Mans, R. S., Manuel, A., McCreesh, M., Mello, P., Mendling, J., Montali, M., Motahari-Nezhad, H. R., Zur Muehlen, M., Munoz-Gama, J., Pontieri, L., Ribeiro, J., Rozinat, A., Seguel Pérez, H., Seguel Pérez, R., Sepúlveda, M., Sinur, J., Soffer, P., Song, M., Sperduti, A., Stilo, G., Stoel, C., Swenson, K., Talamo, M., Tan, W., Turner, C., Vanthienen, J., Varvaressos, G., Verbeek, E., Verdonk, M., Vigo, R., Wang, J., Weber, B., Weidlich, M., Weijters, T., Wen, L., Westergaard, M., and Wynn, M. (2012). Process mining manifesto. *Lecture Notes in Business Information Processing*, 99 LNBIP(PART 1):169–194.

Aalst, W. v. d., Hee, K. M. v., Werf, J. M. v. d., and Verdonk, M. (2010). Process Mining to Support Tomorrow s Auditor. 43(3):90–93.

Aalst, W. V. D., Hee, K. V., Martijn, J., Werf, V. D., Kumar, A., and Verdonk, M. (2011). Conceptual model for online auditing. *Decision Support Systems*, 50(3):636–647.

Aalst, W. v. d., Mans, R., and Russel, N. (2009). Workflow Support Using Proclets: Divide, Interact and Conquer. 32(3):111–115.

Amani, F. A. and Fadlalla, A. M. (2017). Data mining applications in accounting: A review of the literature and organizing framework. *International Journal of Accounting Information Systems*, 24:32–58.

Arens, A. A., Elder, R. J., and Beasley, M. (2012). *Auditing and Assurance Services: An Integrated Approach.* Pearson Prentice Hall, 14th edition.

Azevedo, A. and Santos, M. F. (2008). KDD , SEMMA AND CRISP-DM : A PARALLEL OVERVIEW Ana Azevedo and M . F . Santos. *IADIS European Conference Data Mining*, page 182185.

Brennan, G. and Teeter, R. (2010). Aiding the Audit: Using the IT Audit as a Springboard for Continuous Controls Monitoring. *Ssrn*, pages 1–16.

Brewster, B. E. (2011). How a systems perspective improves knowledge acquisition and performance in analytical procedures. *Accounting Review*, 86(3):915–943.

Brinkkemper, S. (1996). Method engineering : engineering of information methods and tools. *Information and Software Technology*, 38:275–280.

Bryer, R. A. (1993). Double-Entry Bookkeeping and the Birth of Capitalism: Accounting for the Commercial Revolution in Medieval Northern Italy.

Carruthers, B. G. and Espeland, W. N. (1991). Accounting for Rationality: Double-Entry Bookkeeping and the Rhetoric of Economic. Technical Report 1.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. MIT press, 3 edition.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly: Management Information Systems*, 13(3):319–339.

Esteves, J. and Pastor, J. (2001). Enterprise Resource Planning Systems Research: An Annotated Bibliography. *Communications of the Association for Information Systems*, 7.

Gehrke, N. and Müller-Wickop, N. (2010). Basic principles of financial process mining: A journey through financial data in accounting information systems. *16th Americas Conference on Information Systems 2010, AMCIS 2010*, 3:1590–1600.

GNU (2019). Change Logs.

Héder, M. (2017). From NASA to EU: The evolution of the TRL scale in Public Sector Innovation. *Innovation Journal*, 22(2):1–23.

Hosseinpour, M. and Jans, M. J. (2016). Categorizing Identified Deviations for financial Statements Auditing. pages 125–129.

Jans, M., Alles, M., and Vasarhelyi, M. (2013). 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., and Vasarhelyi, M. (2014). A field study on the use of process mining of event logs as an analytical procedure in auditing. *Accounting Review*, 89(5):1751–1773.

Jans, M. and Hosseinpour, M. (2019). How active learning and process mining can act as Continuous Auditing catalyst. *International Journal of Accounting Information Systems*, 32(January 2017):44–58.

Jans, M. and Soffer, P. (2018). From Relational Database to Event Log: Decisions with Quality Impact. *LNBIP*, 132:588–599.

Jans, M., Soffer, P., and Jouck, T. (2019). Building a valuable event log for process mining: an experimental exploration of a guided process. *Enterprise Information Systems*, 13(5):601–630.

Jans, M., Van Der Werf, J. M., Lybaert, N., and Vanhoof, K. (2011). A business process mining application for internal transaction fraud mitigation. *Expert Systems with Applications*, 38(10):13351–13359.

Keller, G., Scheer, A.-W., and Nüttgens, M. (1992). *Semantische Prozessmodellierung auf der Grundlage" Ereignisgesteuerter Prozessketten (EPK)"*. Inst. für Wirtschaftsinformatik.

Lai, P. (2017). the Literature Review of Technology Adoption Models and Theories for the Novelty Technology. *Journal of Information Systems and Technology Management*, 14(1):21–38.

Li, F. (2010). Textual Analysis of Corporate Disclosures. *Journal of Accounting Literature*, 29:143–165.

Li, G., de Murillas, E. G. L., de Carvalho, R. M., and van der Aalst, W. M. (2018). Extracting object-centric event logs to support process mining on databases. In *Lecture Notes in Business Information Processing*, volume 317, pages 182–199. Springer Verlag.

Lu, X., Nagelkerke, M., Van De Wiel, D., and Fahland, D. (2015). Discovering Interacting Artifacts from ERP Systems. *IEEE Transactions on Services Computing,*, 8(6):861–873.

Merriam Webster (2019). Audit.

Müller-Wickop, N. and Nüttgens, M. (2014). Conceptual Model of Accounts-Closing the Gap between Financial Statements and Business Process Modeling. *Modellierung*, pages 209–223.

Ngai, E. W., Hu, Y., Wong, Y. H., Chen, Y., and Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*.

Nooijen, E. H. J., Van Dongen, B. F., and Fahland, D. (2013). Automatic Discovery of Data-Centric and Artifact-Centric Processes. In *Bsuiness Process Management Workshops*, pages 316–327. Springer.

PCAOB (2017). AS 2305: Substantive Analytical Procedures.

Schultz, M. and Tropmann-Frick, M. (2020). Autoencoder Neural Networks versus External Auditors: Detecting Unusual Journal Entries in Financial Statement Audits. *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 3:5421–5430.

Segers, I. (2007). Deloitte enterprise risk services.

Singh, K. and Best, P. J. (2015). Design and Implementation of Continuous Monitoring and Auditing in SAP Enterprise Resource Planning. *International Journal of Auditing*, 19(3):307–317.

Werner, M. (2013). Colored Petri nets for integrating the data perspective in process audits. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8217 LNCS(01):387–394.

Werner, M. (2017). Financial process mining - Accounting data structure dependent control flow inference. *International Journal of Accounting Information Systems*, 25(March):57–80.

Werner, M. (2019). Materiality Maps Process Mining Data Visualization for Financial Audits. *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 6:1045–1054.

Werner, M. and Gehrke, N. (2015). Multilevel Process Mining for Financial Audits. *IEEE Transactions on Services Computing*, 8(6):820–832.

Wirth, R. (1995). CRISP-DM : Towards a Standard Process Model for Data Mining. *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, (24959):29–39.

Glossary and Acronyms

AC Auditing Company.

AIS Accounting Information System.

analytical procedures Procedures that help an auditor to understand the business processes of the auditee and to identify possible risks that should be taken into account in further auditing procedures.

APRA Accounts Payable Recovery Audit.

audit The auditing company that this study was conducted in. This company specializes in Accounts Payable Recovery Audit..

audit A methodical examination and review. In a financial context: the formale examination of an organization's or individual's accounts or financial situation.

auditee The person or company that is being audited.

auditor The person or company that conducts the audit.

CPA Certified Public Accounting.

CRISP-DM Cross Industry Standard Process for Data Mining.

EPC Event-Driven Process Chain.

ERD Entity Relationship Diagram.

ERP Enterprise Resource Planning.

ID Identifier.

internal controls A means by which an organization can be directed, monitored and measured. Basically a set of checks that are in place to prevent things from going wrong.

materiality An auditing term to indicate the significance of something for audit purposes. Often related to regulations, amounts or previously known risks.

PO Purchase Order.

SOD Seggregation of Duties.

TAM Technology Acceptance Model.

TRL Technology Readiness Level.

WYSIWYG What-You-See-Is-What-You-Get.

separate acronyms from glossary

Appendix A

Overview of Identified and Collected Data

Accounting data	Description	Collected
BKPF	Accounting Document Header	1
BSEG	Accounting Document Segment	1
BSEC	One-Time Account Data Document Segment	0
RBKP	Document Header: Invoice Receipt	0
RSEG	Document Item: Incoming Invoice	0
Documents related to PO & PR (only if auditee uses SAP MM)		
EBAN	Purchase Requisition	0
EKBE	History per Purchasing Document	0
EKKO	Purchasing Document Header	0
EKPO	Purchasing Document Item	0
EKET	Purchase Order Delivery Schedule	0
Documents related to Goods delivery (only if auditee uses SAP MM)		
MSEG	Purchasing Document Item	0
MKPF	Purchasing Document Header	0
Workflow logs, only if auditee uses SAP workflow		
SWWWIHEAD	SAP Workflow Runtime: Header Table for All Work Item Types Table and data	1
SWW_WI2OBJ	SAP Workflow Runtime: Relation of Work Item to Object Table and data	1
Change logs (changes in values of documents)		
CDHDR	Change document header	1
CDPOS	Change document items	1

Table A.1: Identified Tables SAP (Collected at auditee = 1/ Not Collected/Present = 0)

Appendix B

Overview of identified flags

- **Do the total amounts add up at different steps in the process?**
  - Invoices add up to payment run
  - Payment runs count up to clearing
  - Invoices count up to clearing
- **Are there documents that are not booked as invoices, yet resulted in payments in the graphs?**
  - Only G/L booking without Vendor booking
  - Same amount for vendor booking and G/L booking
  - G/L bookings with manual payments while payment runs are standard
  - Bookings in different graphs that have same amount but no invoice, as an invoice in a graph with a payment
- **Are there changes in the change log on relevant information in the documents of the graph?**
  - Changes on invoice description
  - Changes on amount
  - Changes on external invoice id
  - Changes on vendor master data while invoice handling
- **Are there possible violations of the segregation of duties?**
  - Same user for creating vendor and invoice booking
  - Same user for invoice booking and payment run
  - Same user for invoice booking and post with clearing
  - Same user for payment run and post with clearing
- **Are there exceptional, outlier, graphs?**
  - Average amount for bookings in graph more than average+2x std dev.
  - No VAT lines, while third party booking

Appendix C

Survey questions

The questionnaire for the survey of the TAM in Chapter 7 are divided in four categories. These questions are:

- Perceived usefulness
  - Using the graph flagging algorithm will increase the number of risks found
  - Using the graph flagging algorithm will increase the quality of risks found
  - Using the graph flagging algorithm will improve the efficiency of the accounts payable recovery process
  - Using the graph flagging algorithm will improve the effectivity of the accounts payable recovery process
- Ease of use
  - It is clear what the graph flagging algorithm does
  - I find it easy to understand the flags resulting from the graph flagging algorithm
  - I find it easy to see the possibilities of the graph flagging algorithm
  - The graph flagging algorithm provides me with the right information
- Satisfaction
  - I am completely satisfied with the idea of mining graphs from the clearing structure
  - I am completely satisfied with the idea of raising flags for the mined graphs
  - I am completely satisfied with the results of the graph mining algorithm
- Intention to use
  - I think the graph mining algorithm with flags will be used
  - I would recommend the graph mining algorithm with flags to others