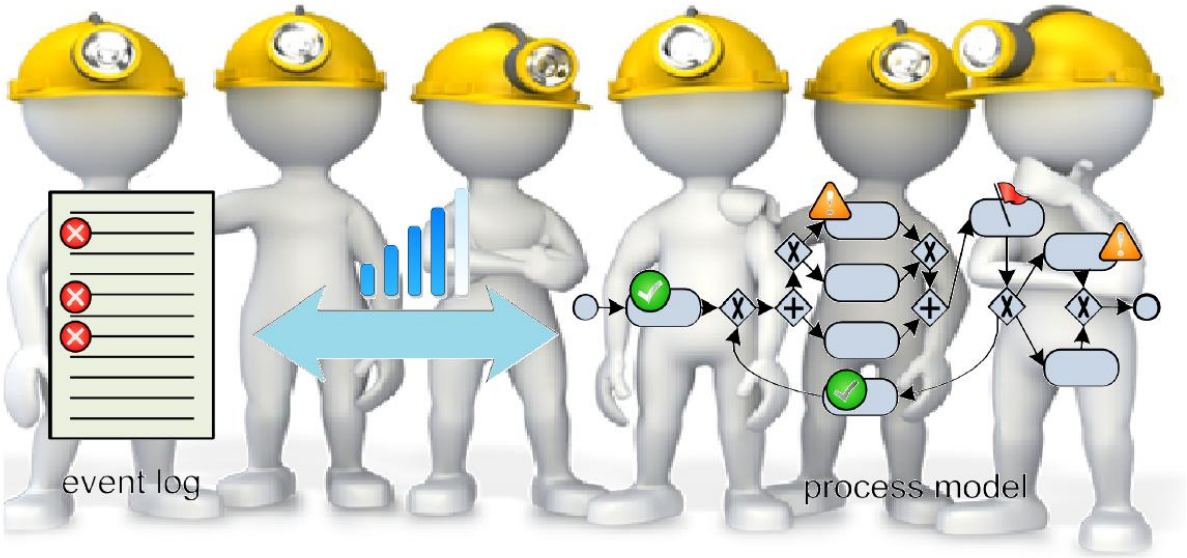
Process Mining and Fraud Detection

A case study on the theoretical and practical value of using process mining for the detection of fraudulent behavior in the procurement process



Masters of Science Thesis
J.J. Stoop
December 2012

Committee:
M. van Keulen – Twente University
C. Amrit – Twente University
R. van Hooff
P. Özer

UNIVERSITEIT TWENTE.

Abstract

This thesis presents the results of a six month research period on process mining and fraud detection. This thesis aimed to answer the research question as to how process mining can be utilized in fraud detection and what the benefits of using process mining for fraud detection are. Based on a literature study it provides a discussion of the theory and application of process mining and its various aspects and techniques. Using both a literature study and an interview with a domain expert, the concepts of fraud and fraud detection are discussed. These results are combined with an analysis of existing case studies on the application of process mining and fraud detection to construct an initial setup of two case studies, in which process mining is applied to detect possible fraudulent behavior in the procurement process. Based on the experiences and results of these case studies, the 1+5+1 methodology is presented as a first step towards operationalizing principles with advice on how process mining techniques can be used in practice when trying to detect fraud. This thesis presents three conclusions: (1) process mining is a valuable addition to fraud detection, (2) using the 1+5+1 concept it was possible to detect indicators of possibly fraudulent behavior (3) the practical use of process mining for fraud detection is diminished by the poor performance of the current tools. The techniques and tools that do not suffer from performance issues are an addition, rather than a replacement, to regular data analysis techniques by providing either new, quicker, or more easily obtainable insights into the process and possible fraudulent behavior.

Occam's Razor: *"One should not increase, beyond what is necessary, the number of entities required to explain anything"*

Contents

1. Introduction

This chapter aims to provide the motivation for this research, the concerns leading to the problem statement, and the research questions that are examined throughout this thesis. Furthermore it provides insight into how the research was conducted, by describing the approach and structure used in this thesis.

1.1 Motivation

In today's business world organizations rely heavily on digital information systems to provide them insight into the way the business is running. The emergence of Workflow Management (WFM) systems, aiming to automate business processes, and Business Process Management (BPM), combining IT knowledge and management science, has put tremendous emphasis on how activities and processes should be performed optimally, how they are modeled, and how analysis of these systems can be used to improve performance. Systems such as Enterprise Resource Planning (ERP) systems or Customer Relationship Management (CRM) produce large amounts of data, which can be analyzed using various techniques and tools such as Business Intelligence (BI), Online Analytical Processing (OLAP) and Data Mining. This whole process, known as the BPM lifecycle, is depicted in Figure 1. The data collected throughout the BPM lifecycle can be used for performance analysis and redesign, but also for detecting (intentionally) deviating behavior.

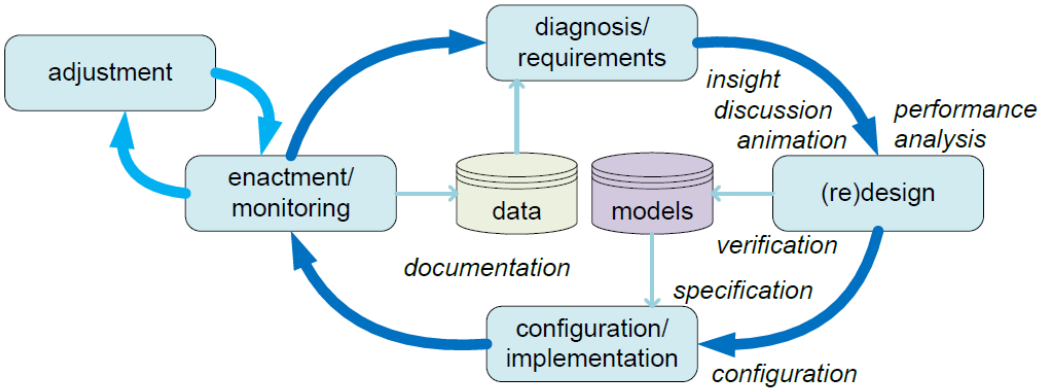


Figure 1: The BPM lifecycle. Taken from (van der Aalst, 2011, p.8).

<<REMOVED DUE TO CONFIDENTIALITY>>

On the cutting edge of process modeling and data mining lays the concept of Process Mining. In short, process mining aims to discover, monitor and improve real, actual processes and their models from event logs generated by various corporate systems, rather than using predefined, manually designed process models (van der Aalst, 2011, p.8). As shown in Figure 2 process mining establishes the link between the recorded result of events during the execution of business processes and how the execution was supposed to happen (i.e. was modeled). Process mining uses data, extracts the information and creates new knowledge. As such, process mining completes the BPM lifecycle (van der Aalst, 2011, p.8).

Figure 2 also shows the three types of process mining: discovery, conformance and enhancement. They are described briefly as follows: discovery is concerned with process elicitation, i.e. it takes some event log and some process discovery algorithm and constructs a process model. Conformance checking is used to check whether or not the events in the event log match some previously determined process model. This model can be created using a process mining discovery algorithm as well as being manually designed. Conformance checking can be used e.g. to see if protocols are followed or which percentages of process executions follow a certain 'path' through the model. Enhancement can be used to improve or repair existing processes, by using both the event log and the (discovered) model to find 'desire lines' in the process model. Enhancement can also be used to extend the model, by adding different properties and adding new perspectives to the process model.

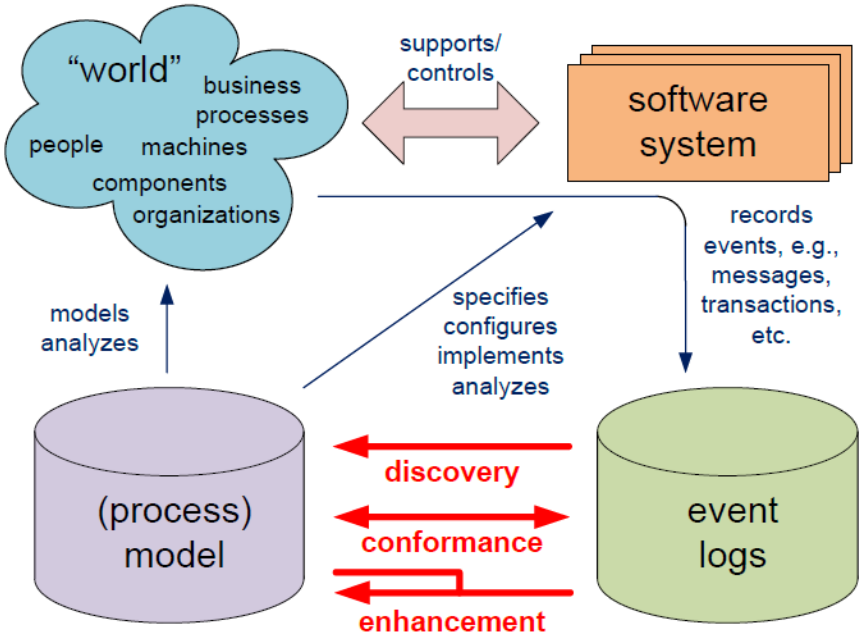


Figure 2: Process Mining overview. Taken from (van der Aalst, 2011, p.9).

There is an obvious link between conformance checking and fraud detection. When fraud is regarded as a deviation from normal procedures and processes, one can easily see how this is similar to conformance checking. With the recent emergence of process mining various authors (Bezerra & Wainer, 2008b; Alles et al., 2011; Jans et al., 2011; van der Aalst et al., 2010) have published research on how process mining may be able to aid both auditing and fraud detection and mitigation. A preliminary analysis of this literature indicates promising results. The remaining question however is how organizations involved in fraud detection can operationalize process mining to incorporate it into their practices.

In this thesis, the possible benefits of using process mining in the field of fraud detection will be examined. Using a literature study and expert interviews into process mining and fraud practices these benefits will be examined. Resulting suggested benefits will be tested by way of practical case studies, to discover which specific aspects and applications of process mining can be utilized and what these

benefits are. These benefits will be synthesized into preliminary operating principles for using process mining for fraud detection in practice.

1.2 Problem Statement

From the introduction in the previous section the following problems can be extracted:

- <<REMOVED DUE TO CONFIDENTIALITY>>
- Therefore, there is no knowledge on how process mining can be utilized for fraud detection and what the specific benefits are of operationalizing process mining for fraud detection.
- As a result, principles on the operationalization of process mining in fraud investigation is lacking.

1.3 Research Questions

Following from the problems stated above, the following research question needs to be answered:

How can process mining be utilized in fraud detection and what are the benefits of using process mining for fraud detection?

In order to answer this question, it can be split up in several smaller questions:

1) *What is process mining and which functional techniques does it encompass?*
2) *What does the process of fraud detection look like and which steps are taken in this process?*
3) *Which functional techniques of process mining can be used in which aspects of the fraud investigation process and what are the benefits?*
4) *Which aspects of process mining can be incorporated into an initial attempt to operationalize process mining in fraud detection based on the case study results?*

1.4 Approach

First, a literature study is conducted to get insights into process mining and its concepts, which aspects of process mining can be used from a fraud detection perspective, and what the possible benefits can be when doing so.

Second, the fraud investigation approach currently used must be examined to get insights into this process. This is done by interviews with employees working in fraud detection as well as other audit-related units. While the main focus in this thesis lies on fraud detection, due to the assumed similarities between fraud detection and auditing it seems plausible that auditing can also benefit from process mining. Also, case studies on the application of process mining to fraud detection are explored to see how other authors have judged the utility of process mining.

Third, this thesis presents the results of a practical case study. In this case study a real-life dataset will be analyzed using various process mining techniques. Two procurement data sets will be analyzed from two different companies. The analysis consists of different tools and techniques that are used and suggested in literature and other case studies. This is done to validate the results of both the literature study and the interviews. The approach is depicted in Figure 3 shown below.

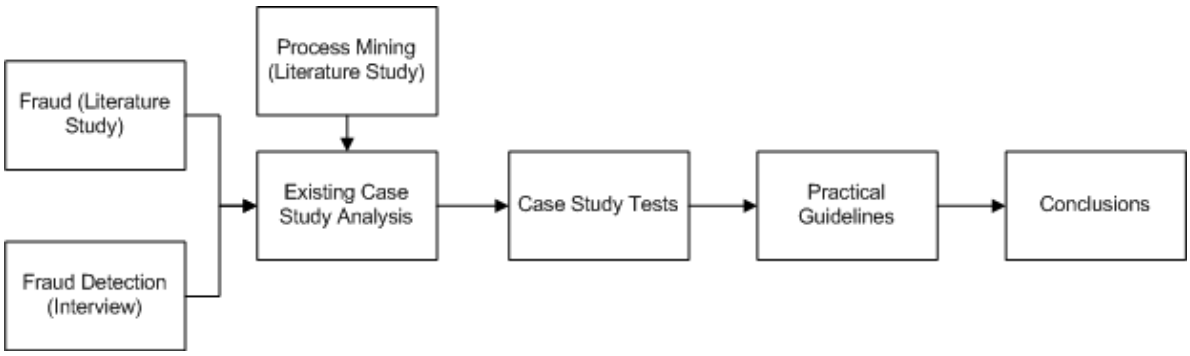


Figure 3: Thesis approach diagram.

1.5 Structure

Following the approach presented in the previous section, the structure of this thesis will be as follows:

- Chapter 2 presents the result of the literature study on process mining and fraud detection to provide the scientific background on the topics and concepts mentioned throughout this thesis.
- Chapter 3 examines the relationship between the theories and concepts presented in Chapter 2. This is extended by an assessment of current available literature on the topic of combining process mining and fraud detection.
- Chapter 4 describes the setup of the case study. The choices made concerning the example data set and the tools used will be elaborated as well as the specific parameter values used while running the analysis.
- Chapter 5 presents the results of the analysis described in Chapter 4. Subsequently it will explain how these results relate to fraud detection indicators and practices.
- Chapter 6 summarizes the findings by presenting a first step towards operationalizing guidelines, with aspects of process mining useful for fraud detection, for employees to utilize in practice.
- Chapter 7 concludes this thesis, by providing the answers to the research questions and providing recommendations for further research.

2. Background

This chapter provides more insight into the concepts mentioned in the introduction, process mining and fraud detection.

The process mining part relies mainly on the concept of process mining as developed by Van der Aalst (2011). The work by Van der Aalst provides a broad variety of articles on different aspects of process mining published, by him and others, in previous years and serves as a guide on the topic.

<<REMOVED DUE TO CONFIDENTIALITY>>

2.1 Process Mining

This section aims to provide an understanding of the concept of process mining; it will briefly discuss the related background topics mentioned in the introduction, as well as a more in-depth discussion of the underlying concepts of the three aspects of process mining: process discovery, conformance checking and process enhancement.

2.1.1 Related Concepts

Process Modeling

As mentioned before, process mining lies on the cutting edge between process modeling and data mining. The BPM lifecycle from Figure 1 usually starts with the design of the model of a process. With a process model, one can reason about models to analyze control flow problems such as deadlocks, run simulations or to optimize and redesign processes. Green and Rosemann (2000, p.78) describe a business process as: *"the sequence of functions that are necessary to transform a business-relevant object (e.g. purchase order, invoice). From an Information Systems perspective, a model of a process is a description of the control flow"*. Process models can further be defined as: *" … images of the logical and temporal order of functions performed on a process object. They are the foundation for the operationalization of process-oriented approaches."* (Becker et al., 1997, p.821). A process model can be *descriptive* or *prescriptive*. Descriptive models try to capture existing processes without being normative, while prescriptive models describe the way that processes should be executed.

Modeling these business processes is usually done by way of workflow models; workflow systems assume that processes consist of the execution of unitary actions, called activities, each with their own inter-activity dependencies (Agrawal et al., 1998, p.469). Greco et al. (2005, p.2) define workflows as: "*A workflow is a partial or total automation of a business process, in which a collection of activities must be executed by humans or machines, according to certain procedural rules".* Throughout this thesis, the term workflow and process will be used synonymously.

The definitions by Agrawal et al., Greco et al. and Blecker et al. are combined by Van der Aalst's description of the relation between processes and process models: *" … processes are described in terms of activities (and possibly subprocesses). The ordering of these activities is modeled by describing casual dependencies. Moreover, the process model may also describe temporal properties, specify the creation*

and use of data, e.g., to model decisions, and stipulate the way that resources interact with the process (e.g., roles allocation rules, and priorities)" (van der Aalst, 2011, p.4).

Despite the development of process modeling, there are some problems with using these models. They are inherent to the concept of modeling and are hence hard to avoid. Consider the definition of 'model' by the Oxford Dictionaries Online: (Oxford Dictionaries, 2010b) *"a simplified description, especially a mathematical one, of a system or process, to assist calculations and predictions"*. This definition illustrates two possible problems; models describe an abstracted, and subjective, view on reality. The designer can omit or include aspects into the model that are considered (un)important; these aspects may only be valid for a certain part of reality. This can further be aggravated by the level of abstraction chosen by the designer. Another important problem is the fact that human emotion and decision-making is hard to incorporate into models (van der Aalst, 2011, p.30).

Event Logs

The information produced by the various processes is saved in event logs. In order to use this data for process mining, it needs to be molded into a usable format, known as Extract, Transform, Load (ETL). The aspect that is most important in this thesis is Transformation: current ERP/CRM/etc. systems use big relational databases, linking different tables by using keys, for reasons such as performance and maintainability. For process mining however, and especially aspects beyond process discovery, it is important to have a complete view on the dataset. Therefore it is important to make sure that all required information concerning the process is combined into the event log; this is called 'flattening' of the data.

An example event log is shown in Figure 4; the various entries are listed in the rows, while the different properties of the process are shown in the columns. It shows the process' cases, events (grouped in traces) and attributes. Figure 5 shows how these notions relate to each other: a process can be run in specific ways; each run is a case. This case has an id, and a specific set of events that were executed, called the trace. Each individual event can have multiple attributes; shown here are the names of the activity, the completion (or start) time, the resource used to execute the event (the actor, or originator, the person who performed it) and the cost.

case id	event id	properties				
		timestamp	activity	resource	cost	...
1	35654423	30-12-2010:11.02	register request	Pete	50	...
	35654424	31-12-2010:10.06	examine thoroughly	Sue	400	...
	35654425	05-01-2011:15.12	check ticket	Mike	100	...
	35654426	06-01-2011:11.18	decide	Sara	200	...
	35654427	07-01-2011:14.24	reject request	Pete	200	...
2	35654483	30-12-2010:11.32	register request	Mike	50	...
	35654485	30-12-2010:12.12	check ticket	Mike	100	...
	35654487	30-12-2010:14.16	examine casually	Pete	400	...
	35654488	05-01-2011:11.22	decide	Sara	200	...
	35654489	08-01-2011:12.05	pay compensation	Ellen	200	...
3	35654521	30-12-2010:14.32	register request	Pete	50	...
	35654522	30-12-2010:15.06	examine casually	Mike	400	...
	35654524	30-12-2010:16.34	check ticket	Ellen	100	...
	35654525	06-01-2011:09.18	decide	Sara	200	...
	35654526	06-01-2011:12.18	reinitiate request	Sara	200	...
	35654527	06-01-2011:13.06	examine thoroughly	Sean	400	...
	35654530	08-01-2011:11.43	check ticket	Pete	100	...
	35654531	09-01-2011:09.55	decide	Sara	200	...
	35654533	15-01-2011:10.45	pay compensation	Ellen	200	...
4	35654641	06-01-2011:15.02	register request	Pete	50	...
	35654643	07-01-2011:12.06	check ticket	Mike	100	...
	35654644	08-01-2011:14.43	examine thoroughly	Sean	400	...
	35654645	09-01-2011:12.02	decide	Sara	200	...
	35654647	12-01-2011:15.44	reject request	Ellen	200	...
...	...	...	...	...	...	...

Figure 4: An example event log. Taken from (van der Aalst, 2011, p.99).

Besides the issue with flattening, Van der Aalst (2011, p.113) mentions five other (sometimes related) concerns regarding the extraction and/or construction of event logs: correlation (assigning events to the right case), timestamp alignment, snapshot problems (incorrectly started or finished traces due to the time of capture), scoping, and granularity.

For a more in-depth and conceptual discussion of the processes and event logs, the reader is referred to (van Dongen & van der Aalst, 2005). For a formal notation of both concepts, the reader is referred to Appendix A.

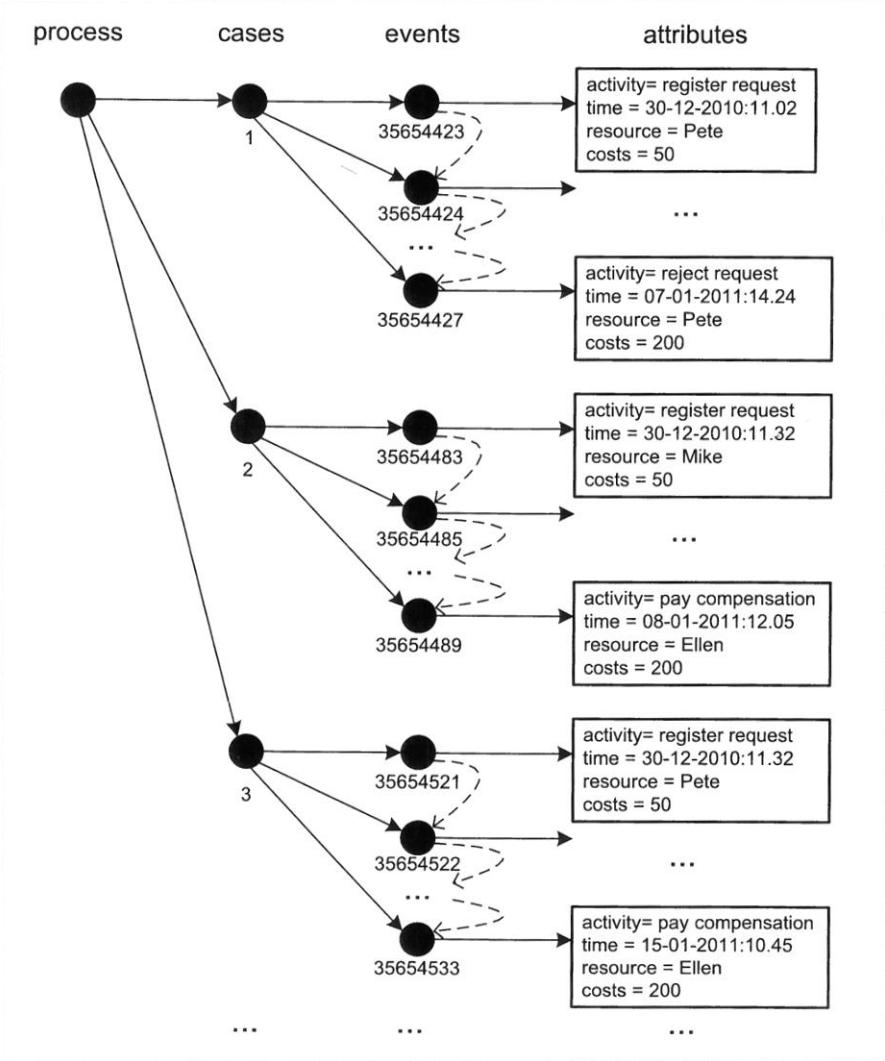


Figure 5: Example event log structure. Taken from (van der Aalst, 2011, p.100).

2.1.2 Process Mining Overview

The three general applications of process mining are shown in Figure 2 indicated by the red arrows: discovery, conformance and enhancement. These three applications each use the event log in a different way. The traditional way of using process models and event logs is Play-out. In Play-out, the process model is used to e.g. run simulations for performance analysis, or verify the model with model checking.

In Play-in, the model and event log are used in an opposite way. Play-in takes the event log and uses it to create a process model, i.e. process discovery. Play-in can also be used in other fields such as data mining, to e.g. develop a decision tree based on available examples.

Replay, shown in Figure 6, takes both the event log and a corresponding process model to perform a variety of analyses. The most interesting from a fraud detection perspective is conformance checking, i.e. detecting deviating traces, and is discussed in Section 2.1.4. Other applications of replay are shown in

Figure 6; finding frequent paths and/or activities, diagnosing bottlenecks, enabling duration predictions, and giving predictions and recommendations on running cases for its attributes.

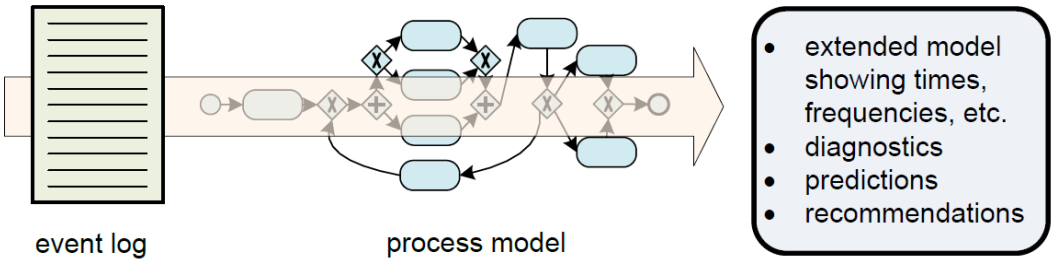


Figure 6: Replay. Taken from (van der Aalst, 2011, p.19).

The developments in the field of process mining have increased its applications over the last years. The last aspects of replay have suggested the use of online, i.e. real-time, data in process mining. There is a number of applications that are aimed towards online, operational support. For a more in-depth discussion of the benefits of process mining on operational support, the reader is referred to Van der Aalst (2010; 2010)

Process mining can be done from three different perspectives: the process, organizational, and case perspective (van der Aalst & Weijters, 2005, p.240). The process perspective focuses on the control-flow of the process and its activities. The organizational perspective focuses on who performed which activity, in order to e.g. provide an insight into the organizational structure or handover-of-work. The case perspective focuses on the properties of cases, e.g. the values of the different attributes shown in Figure 5.

2.1.3 Process Discovery

Although process discovery is a relatively new concept, the idea was considered as early as mid-90s. In Cook & Wolf (1995, p.73) the authors recognized the possibility to *"automate the derivation* [of] *a formal model of a process from basic data collected on the process"*, and called this 'process discovery'. As BPM was quickly gaining popularity, the need emerged to create process models of existing business processes quicker, cheaper, and more accurately. The authors already recognized that process models are dynamic and evolve over time, and hence should be adapted.

In an effort to formalize their previous work, the authors presented a framework that was now event-based, and furthermore went beyond the scope of just software processes. In their conclusions the authors also put emphasis on visualization and the possibility to model using other techniques than just Finite State Machines (Cook & Wolf, 1998, p.246). Meanwhile Agrawal et al. (1998) attempted to further formalize the concept and presented one of the first algorithms to create a Directed Acyclic Graph out of event logs. Similarly, but unrelated, Datta (1998) proposed a probabilistic method to discover Process Activity Diagrams based on the Biermann-Feldman FSM computation algorithm. In Weijters & van der Aalst (2001a; 2001b) the scope of the research was focused towards concurrency and workflow patterns, i.e. AND/OR splits and joins. The authors continued this research towards the discovery and construction of Workflow Nets out of event logs (van der Aalst et al., 2002) and presented the first

process discovery algorithm, the α-algorithm. An extension of the α-algorithm followed shortly, which was able to incorporate timing information, based on timestamps in the event log (van der Aalst & van Dongen, 2002).

The α-Algorithm

The α-algorithm (van der Aalst & Weijters, 2005; van der Aalst, 2011; 2004) is regarded as the first algorithm that was capable of process mining. For a more formal and in-depth description the reader is referred to Medeiros et al. (2007), Wen et al. (2007) and Appendix A. The α-algorithm has various limitations (van der Aalst et al., 2003; de Medeiros et al., 2003). Besides the general issue with log completeness, the α-algorithm is not always able to create a correct model. It can produce overly complex models (resulting in implicit places), it is not able to detect loops of two or less, nor can it discover non-local dependencies resulting from non-free choice process constructs (i.e. some places and transitions are not discovered while they should be possible). Furthermore, frequencies are not taken into account in the α-algorithm; therefore it is very sensitive to noise and can easily misclassify a relation (a log with 100.000 times a→b and one time b→a will result in 'a' parallel to 'b', which is statistically unlikely). Regardless of the issues mentioned, the α-algorithm a relatively straightforward algorithm that provides a good starting point for understanding subsequent algorithms.

Process Discovery Quality

To determine the quality of mined process models, Van der Aalst (2011) describes four metrics, or quality criteria: fitness, simplicity, precision, and generalization. The level of fitness is determined by how big of a fraction of an event log can be replayed on the model. Fitness can be defined at different levels, e.g. case level or event level. Simplicity refers to Occam's Razor: *"One should not increase, beyond what is necessary, the number of entities required to explain anything"*. This indicates that the simplest model being able to explain behavior is the best model. Simplicity could for instance be defined by the number of arcs and nodes in the process model. Precision refers to underfitting, i.e. when the model is over-generalized and allows for different behavior than seen in the event log. Generalization refers to overfitting, the opposite of precision. Models that overfit only allow for the specific behavior seen in the event log, but not any other behavior, however likely it may seem. An example on how these four quality criteria affect models and each other is shown in Figure 7.

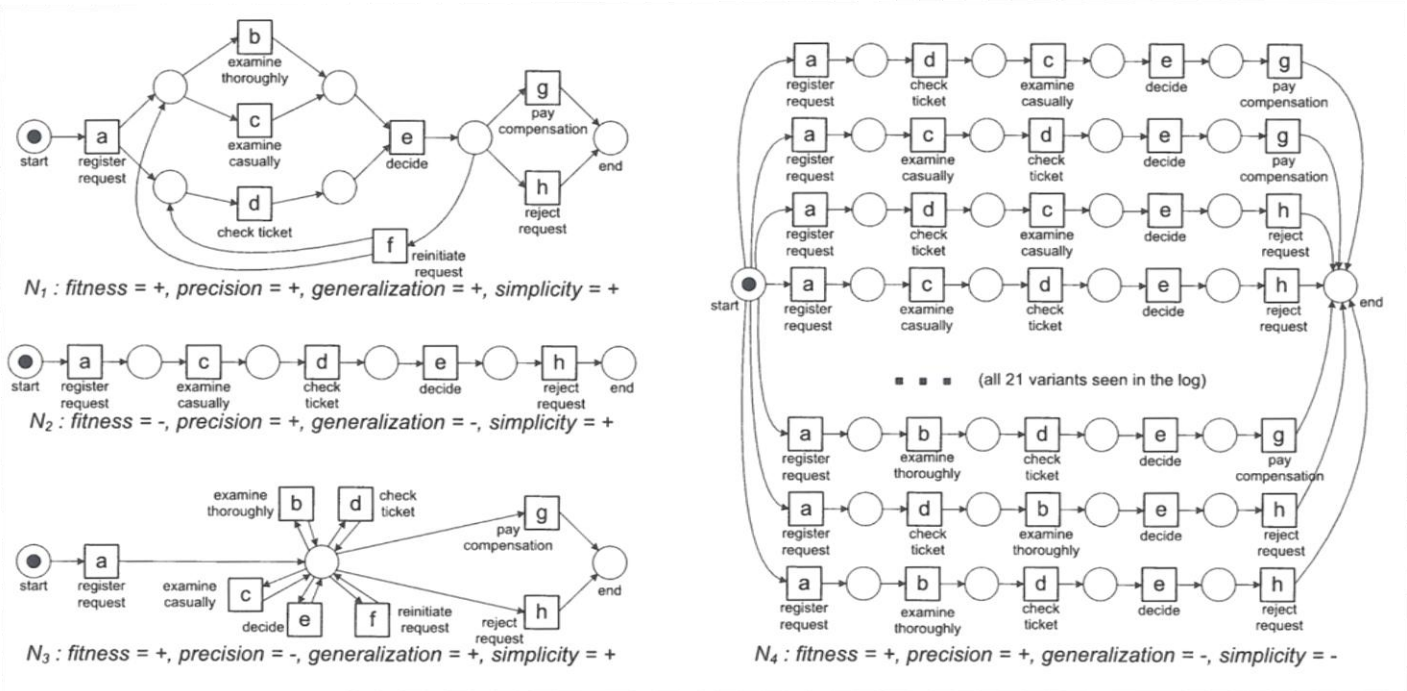


Figure 7: Quality criteria example. Taken from (van der Aalst, 2011, p.154)

Process Discovery Challenges

Process discovery in general has several challenges. The first problem is independent of the approach used: the representational bias, i.e. *"process discovery is, by definition, restricted by the expressive power of the target language"* (van der Aalst, 2011, p.146). Consider e.g. Figure 8, which shows three different representations for the event log {(a,b,c), (a,c)}. When comparing the different models to model Figure 8(a), Figure 8(b) appears to have two activities labeled 'a'. This can lead to both ambiguous behavior (during replay e.g.) as well ambiguous classification of traces (during conformance checking e.g.) Figure 8(c) has different outcomes for activity a; this can lead to similar ambiguity issues. For an overview of representational limitations the reader is referred to Van der Aalst (2011, pp.159-60).

The second problem in process discovery is noise (noise in this sense is regarded as outliers, not incorrectly recorded log entries). As described earlier, infrequent behavior can alter the relations between activities even if they are statistically irrelevant. Solutions to the noise problem are support and confidence metrics know from data mining. Often the 80/20 rule is applicable, in which 80% of the variability in a process model is caused by only 20% of the traces from the event log (van der Aalst, 2011, p.148). Heuristic mining, discussed later, can be used to deal with noise. Note however that, for the purpose of fraud detection, noise (i.e. the deviation from the norm) is what investigators are looking for! There is however an important distinction between the problem of noise during process discovery and noise during conformance checking. Models that contain noise during discovery become complex and unreadable, but will therefore most likely be also able to replay most of the traces. In conformance checks, this can lead to false negatives. Thus, in the context of fraud detection, it is important to keep

all[1] traces when using replay, but for play-in (i.e. process discovery) it can be useful to temporarily remove infrequent ones.

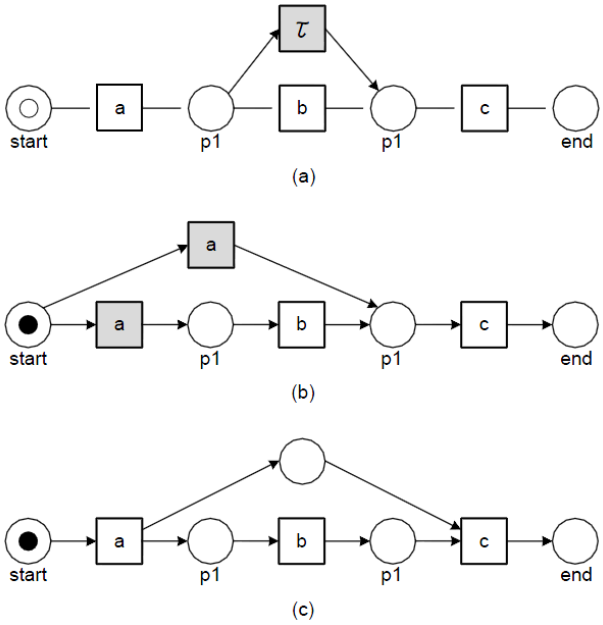


Figure 8: Representational bias example. Taken from (van der Aalst, 2011, p.146)

Completeness can be seen as the opposite of noise; where noise has too much irrelevant data, completeness deals with a lack of relevant data (i.e. possible traces). Consider the situation in which a group of 365 people, the probability of everyone having a different birthday is $365! / 365^{365} \approx 1.455 * 10^{-157}$. Similarly, the chance that an event log contains all possible individual behavior is extremely small. In the context of fraud detection, this leads to the notion that frequency alone might not be a suitable base on which to label a trace as a deviation; the occurring event or trace might have just been improbable.

Other concerns with process mining are related to the field of data mining, such as the lack of negative examples and the complexity and size of the search/state space. In the context of fraud detection, similarly to the noise problem and regardless of frequency, this can again lead to false negatives; the fact a specific trace has not occurred does not always mean it should not be a compliant possibility. Another concern follows from the flattening mentioned earlier: a process model shows its process from a particular angle (e.g. customer, order) and is bounded by its frame (i.e. the information and attributes used), with a particular resolution (i.e. granularity). Therefore, the same process can be depicted by a number of models. Thus, a trace that is labeled as deviant from a particular angle can be compliant from a different angle. This implies that when analyzing data for fraud detection, often different angles should be taken to analyze the data from.

Other discovery techniques

There are various other techniques that can be used to discover process models from event logs. These algorithms can be categorized in various ways and have different underlying characteristics (van der

[1] Obvious erroneously recorded traces (e.g. incomplete) traces exempt.

Aalst, 2011; van Dongen et al., 2009). They are only mentioned briefly in this section; for a more in-depth comparison the reader is referred to Van Dongen et al. (2009). The algorithms that are used in the practical part of this thesis will be further discussed in later sections.

The group of techniques that can be considered algorithmic (α-miner (and several variations), finite state machine miner, heuristic miner) extract the footprint[2] from the event log and create the model. Heuristic techniques (Weijters & Ribeiro, 2011) also take frequencies into account, and are therefore more resistant to noise. Due to the additional use of Causal Nets (a different representation technique) the heuristic approach is more robust than most other approaches (van der Aalst, 2011, p.163). A noteworthy related approach is Fuzzy Mining (Günther & van der Aalst, 2007), which is able to create hierarchical (i.e. aggregatable) models.

Genetic mining is an evolutionary approach from the field of computational intelligence which mimics the process of natural evolution. These approaches use randomization and best model fit to find new alternatives for discovered process models. Characteristics of genetic mining are that it requires a lot of computing power, but can easily be distributed. It is however capable of dealing with noise, infrequent behavior, and duplicate and invisible tasks. Also, it can be combined with other approaches for better results.

2.1.4 Conformance Checking

Conformance checking is the second aspect of process mining. It uses both an event log and a process model (constructed either manually, or using process discovery) and relates the traces and the model by replaying. Through conformance checking deviations between modeled and observed behavior can be detected. This information can then be used for e.g. business alignment (process performance analysis and improvement), auditing (e.g. detecting fraud or non-compliance) or analyzing the results of process discovery algorithms. There are various ways to test conformance (e.g. token replay) and different metrics to measure conformance (e.g. fitness, appropriateness). Furthermore, conformance can be measured on different levels; possibilities are case level, event level, footprint level and constraint level (e.g. using Linear Temporal Logic). Finally conformance can be checked online (during process execution) and offline (after process completion) (van der Aalst, 2011, pp.191-94).

Initially conformance checking was done by two methods, Delta Analysis and Conformance Testing. Delta analysis focuses on model-to-model comparison, but conformance testing directly compares an event log with a model. Using this method it is possible to test the fitness criteria mentioned earlier. It works by replaying the traces from an event log on a Petri Net, and counting the number of times an action was not performed while it was expected to plus the number of times an action was performed while it should not have been possible. Figure 9 shows two examples of the token game being replayed on a process model. Example Figure 9(a) replays the trace (a,c,d,e,h) and fits, example Figure 9(b) replays trace (a,d,c,e,h) and has one missing token and one remaining token.

[2] For more information on the specifics of footprints, the interested reader is referred to Appendix A.

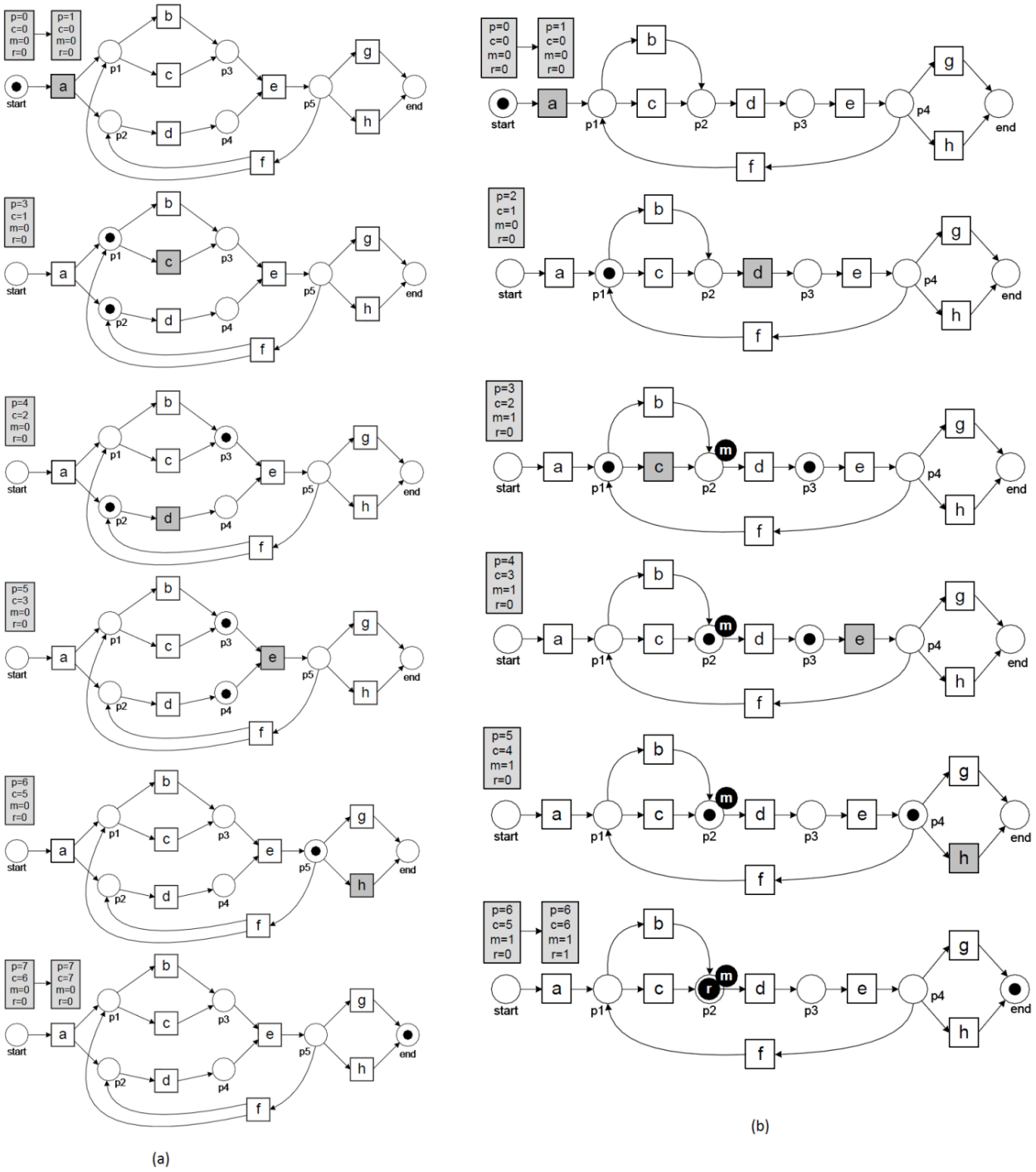


Figure 9: Token Game example. Taken from (van der Aalst, 2011)

Besides fitness, the other metrics to determine the quality of process discovery mentioned earlier can also be used for conformance testing. The fitness metric was improved to incorporate the missing, remaining, produced, consumed token concept, and the appropriateness metrics were introduced (Rozinat & van der Aalst, 2005; 2006a). Structural appropriateness is comparable to the simplicity

criteria mentioned earlier, behavioral appropriateness deals with underfitting and overfitting. For an in-depth analysis of conformance checking and these metrics the reader is referred to (Rozinat & van der Aalst (2008).

The concept of conformance checking can be applied to real-time checks as well. Whereas process mining itself was positioned as part of the BPM concept, the evolution of conformance checking supports BPM significantly. In their conclusion, El Kharbili et al. (2008) present the outlook that *"four main factors that need to be incorporated by current compliance checking techniques: (i) an integrated approach able to cover the full BPM life-cycle, (ii) the support for compliance checks beyond control-flow-related aspects, (iii) intuitive graphical notations for business analysts, and (iv) embedding of semantic technologies during the definition, deployment and executions of compliance checks".*

Conformance testing is one of the most interesting aspects of process mining for fraud detection. Especially token replay can be of high value: discovering certain traces that skip actions, or execute actions that should not have been possible to be executed, can provide solid indicators of fraudulent behavior, without having to analyze each possible path between two activities. Furthermore, conformance testing can potentially be applied to different fields that are in some way involved with human performance. However, non-conformance of traces does not necessarily indicate fraudulent behavior; there may be various acceptable exceptions depending on other case attribute (values).

2.1.5 Other Process Mining Aspects

The organizational, case, and time perspectives, are more concerned with the conformance and enhancement aspects of process mining. Mining and analysis on these perspectives use the attributes from the cases. Figure 4 and Figure 5 show some example attributes: activity, resource, cost. This section discusses the organizational mining and operational support aspects of process mining.

Organizational Mining

The organizational perspective is the subject of organizational mining. It focuses on the resource or originator attribute of an activity to discover e.g. who does which activity most often (focusing on the relation between resource and process) or to discover the Social Network or Handover-of-Work (focusing on the relation between resources themselves). For more details on sociometry, or sociography (referring to methods that present data on interpersonal relationships in graph or matrix from), the reader is referred to Wasserman & Faust (1994). Figure 10 shows an example of a resource-activity matrix, i.e. the mean number of times a resource performs an activity per case. E.g. activity *a* is performed 0.3 times per case by Pete. Based on the numbers, the conclusion could be drawn that e.g. Pete, Mike, and Ellen might have the same role, i.e. tasks and responsibilities.

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
Pete	0.3	0	0.345	0.69	0	0	0.135	0.165
Mike	0.5	0	0.575	1.15	0	0	0.225	0.275
Ellen	0.2	0	0.23	0.46	0	0	0.09	0.11
Sue	0	0.46	0	0	0	0	0	0
Sean	0	0.69	0	0	0	0	0	0
Sara	0	0	0	0	2.3	1.3	0	0

Figure 10: Resource-Activity Matrix Example. Taken from (van der Aalst, 2011, p.222)

In Figure 11 a social network is explained, and in Figure 12 an example is shown. Note that a threshold of 0.1 was used, e.g. work from Pete to Sue or Sean is not shown. A model shown like the one in Figure 12 can be used in a lot of (context specific!) ways. In a bottleneck analysis, one could conclude that Sara should hand over more work to Pete and Ellen to alleviate Mike. On the other hand, the specific cases that were handed over to Ellen could be examined (i.e. combining and checking different case attributes) to see whether there is something special, e.g. if these require specific expertise that only Ellen can provide. For an in-depth discussion of organizational mining and the developed metrics, the reader is referred to Van der Aalst et al.(2005) and Song & van der Aalst (2008).

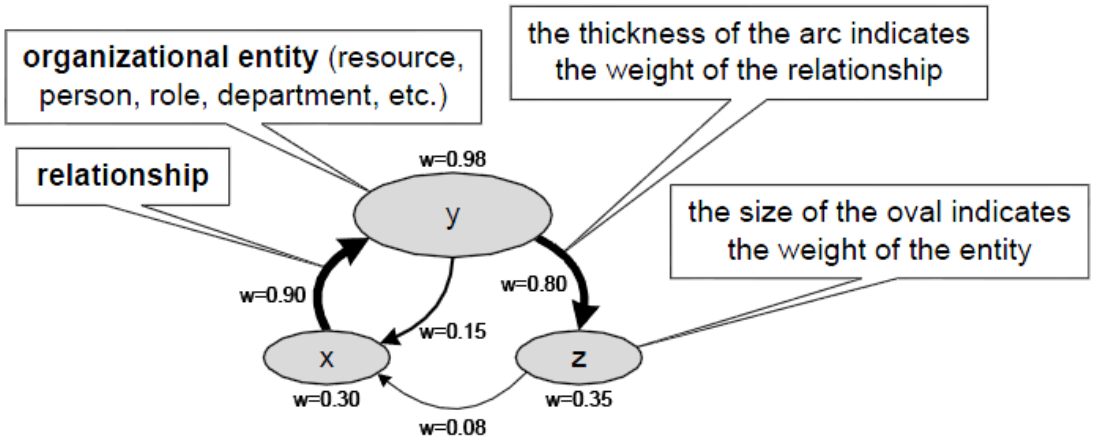


Figure 11: A Social Network. Taken from (van der Aalst, 2011, p.223)

Operational support

The time perspective is concerned with the timing and frequency of events. If activities are not just recorded as atomic event, but have separate timestamps in the log for the different events such as start and complete e.g., it is possible to derive a lot of interesting information from the event log. When the event log is replayed on the model, one could for instance calculate that a certain activity takes X minutes on average to complete with a Y% confidence interval. Other examples of performance related information are (van der Aalst, 2011, pp.232-33): visualization of waiting and service time, bottleneck detection and analysis, flow time and SLA analysis, frequency and utilization analysis.

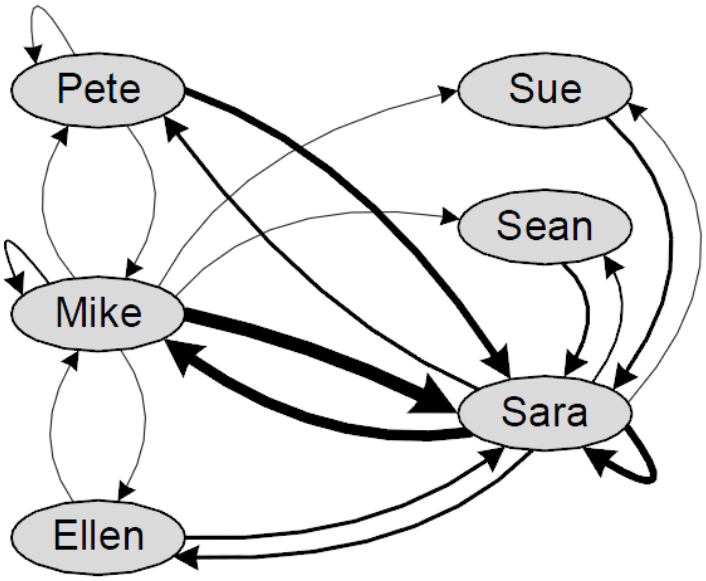


Figure 12: Handover-of-Work Example. Taken from (van der Aalst, 2011, p.224)

The case perspective focuses on properties of the case and how the value of an attribute may affect the routing of a case (Rozinat & van der Aalst, 2006b). After mining the event log, specific rules could be found that e.g. an insurance company always double checks claims of over 100.000 euro. This can then be compared to existing business rules to check conformance, or for audit purposes. Decision mining is not limited to attribute values. Also behavioral information such as the number iterations over a specific activity can be used, timing information can be used (e.g. "cases taking over X minutes are usually rejected") and even non-process-related (i.e. contextual) information (e.g. the weather or stock market information) can be used.

True operational support is the next phase in the development of the application of process mining. With the discussion of the three main types of process mining and the different perspectives, there has been no emphasis on the distinction between types of data and models. Although operational support is out of scope in this thesis, there is some overlap between fraud detection and some aspects of operational support. Compared to regular process mining aspects, operational support is more concerned with online aspects. The concept of *" [] … Business Process Provenance aims to systematically collect the information needed to reconstruct what has actually happened in a process or organization […and …] refers to the set of activities needed to ensure that history, as captured in event logs, cannot be rewritten or obscured such that it can serve as a reliable basis for process improvement and auditing"* (van der Aalst, 2011, p.242). In Figure 13 the concept of business process provenance is shown. The difference between pre mortem and post mortem is concerned with the difference between running and finished cases respectively. The difference between de jure and de facto models is concerned with the difference between normative and descriptive models respectively. The ten activities, grouped by navigation, auditing, and cartography, are concerned with the following:

- Navigation
  - *Explore* running cases at run-time
  - *Predict* outcomes of running cases based on statistical analysis of historical data
  - *Recommend* changes at run-time (like a TomTom car navigation system)
- Auditing
  - *Detect* deviations at run-time
  - *Check* conformance and compliance of completed cases
  - *Compare* in-depth metrics (inter-model checking, no event log is used)
  - *Promote* 'desire lines' (= best practices) to improve processes
- Cartography
  - *Discover* actual models
  - *Enhance* current models with different perspectives (time, resources)
  - *Diagnose* control flow (e.g. process deadlocks, intra-model checking)

For the purpose of fraud detection navigation and especially auditing are of interest. The navigation activities can possibly be used to detect deviations in an earlier stage; this can lower losses incurred due to fraud, or even prevent some fraudulent behavior. Auditing activities are evident; most importantly the extended form of conformance checking, where traces are not checked from control-flow perspective but also from case perspective, can provide very valuable insights.

Consider the following example, in which orders have to be authorized before being sent, depending on their value: if orders that are over amount X have to get past a manager, their trace will show an extra activity. Simple conformance checking will only determine if the activities, including a possible authorization step, were taken in the right order. The case perspective is explicitly required to be able to use the attribute 'order value' and to analyze if the activity was indeed performed for all orders with a value over amount X.

In their current state however, the available tools are not suited to accomplish operational support, and business provenance should be seen as a next step in the development of process mining.

Visualization

Visualization of the processes is an important aspect in process modeling. Regardless of the modeling language, there are some aspects that must be mentioned. First, there is a distinction between so-called spaghetti and lasagna processes. While there is no clear definition and distinction, the two terms indicate the difference between unstructured versus structured processes. A process can be considered a lasagna process if *"within limited efforts it is possible to create an agreed-upon process model that has a fitness of at least 0.8"* (van der Aalst, 2011, p.277). The level of structure greatly influences the readability and analysis possibilities.

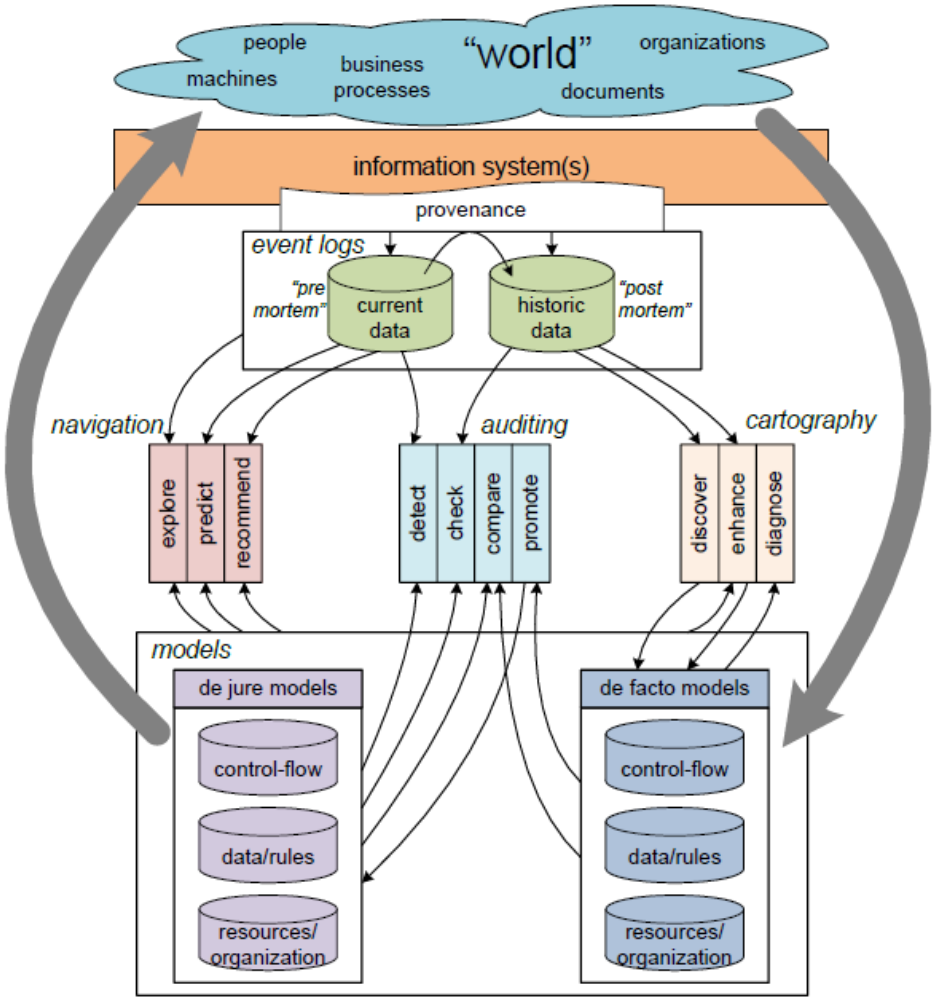


Figure 13: Business Process Provenance. Taken from (van der Aalst, 2011, p.242)

In order to improve model quality in general, some concepts from cartography can be applied: aggregation, abstraction, and seamless zoom. Aggregation incorporates hierarchies into process models. By aggregating low-level events into more meaningful compounded events, process models can be made a lot simpler. Abstraction ignores very infrequent activities and/or traces. This can severely decrease the number of nodes and edges in models, greatly increasing readability. Both approaches can change a spaghetti process into a lasagna process. The most widely used way to accomplish aggregation and abstraction is done by clustering at event log level. This is similar to e.g. the roll-up or drill-down techniques known from Business Intelligence. For more information on trace segmentation, clustering, and abstraction, readers are referred to other references (Bose & van der Aalst, 2009a; 2009b; 2011; La Rosa et al., 2011; Günther et al., 2009).

An alternative way to look at processes is by using dotted charts, shown in Figure 14. A dotted chart depicts events in a two-dimensional plane, where the x-axis represents the time of an event, and the y-axis represents the class. The class can be the activity, but also e.g. the resource. The time dimension can be absolute or relative, and either real or logical. As shown in Figure 14, each case lies on a horizontal

line, where each dot represents an event; the later an event occurs, the more to the right it is displayed. For more information on dotted charts the reader is referred to Song & Van der Aalst (2007)

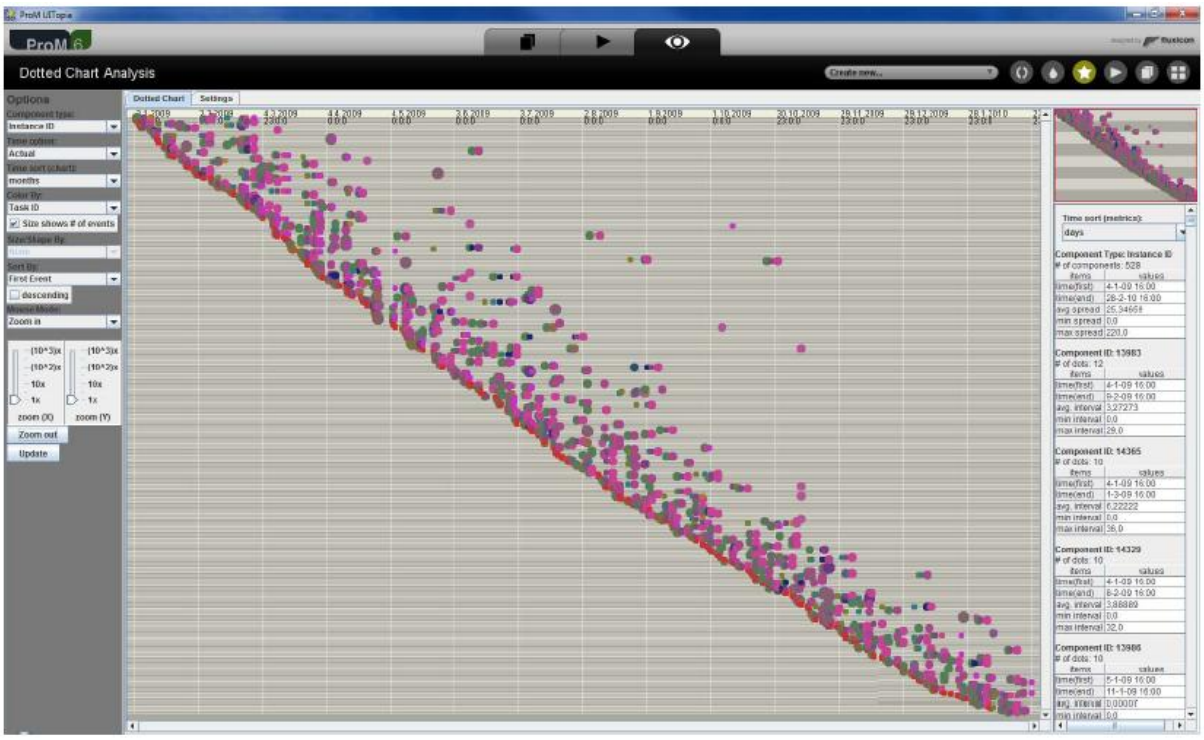


Figure 14: Dotted Chart example.

2.2 Fraud Detection

2.2.1 Fraud Defined

The Oxford Dictionaries Online (Oxford Dictionaries, 2010a) defines fraud as: *"wrongful or criminal deception intended to result in financial or personal gain"*. A distinction can be made between external fraud, i.e. by someone outside the organization, and internal fraud, i.e. by someone from the organization. Internal fraud is similar to occupational fraud; the Association of Certified Fraud Examiners (ACFE) defines occupational fraud as: *"The use of one's occupation for personal enrichment through the deliberate misuse or misapplication of the employing organization's resources or assets* " (ACFE, 2012, p.6). This notion of fraud comprises various different forms, with three primary categories: asset misappropriation, corruption, and financial statement fraud. These categories have several sub-categories, as shown in Figure 15.

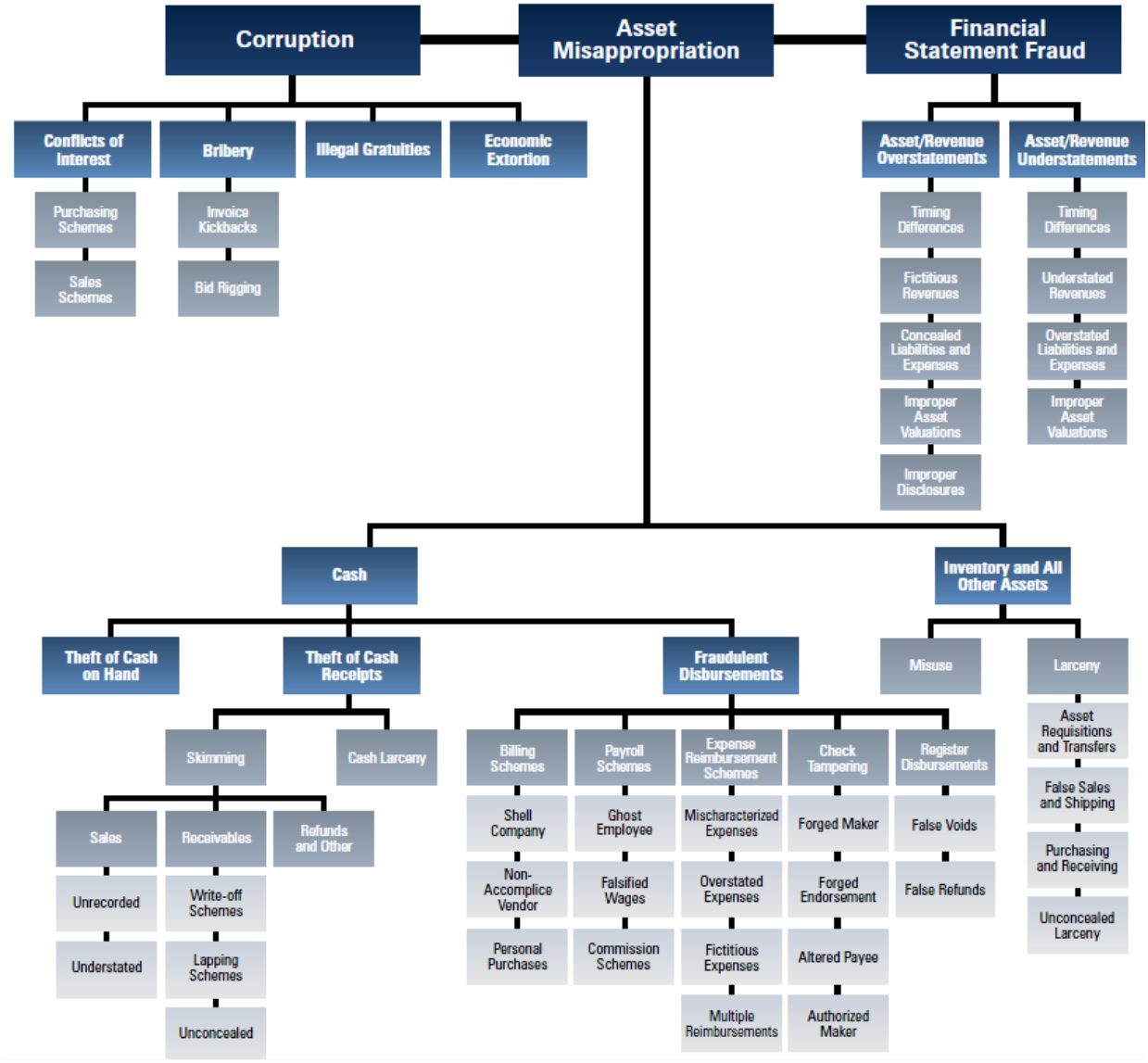


Figure 15: Occupational Fraud. Taken from (ACFE, 2012, p.6)

The costs of fraud are estimated to be a median 5% of an organization's revenues each year (ACFE, 2012, p.4); considering that fraud inherently involves efforts of concealment, the total number cannot be determined. Especially smaller organizations (< 100 employees) are victims of fraud. While the median loss to fraud is comparable to that of bigger sized companies, the impact is more serious due to their (more) limited resources. Combined with the fact that the frequency of anti-fraud controls is significantly lower in organizations with less than 100 employees versus organizations with more than 100 employees (ACFE, 2012, p.34), these smaller organizations are severely more susceptible and vulnerable to fraud.

According to Albrecht et al. (2008a), the so-called fraud triangle, shown in Figure 16, has three elements that are always present in any form of fraud. Perceived pressure is concerned with the motivation for committing the fraud, such as financial need or pressure to perform. The perceived opportunity is

determined by the (perceived) risk of committing the fraud. The bigger the impression of the fraud going undetected and unpunished, the bigger the perceived opportunity. There also needs to be a way to rationalize the fraudulent behavior, comparing the act against internal ("I didn't get a bonus, but I deserve something extra anyhow") or external ("our competitors use the same tricks") moral standards.

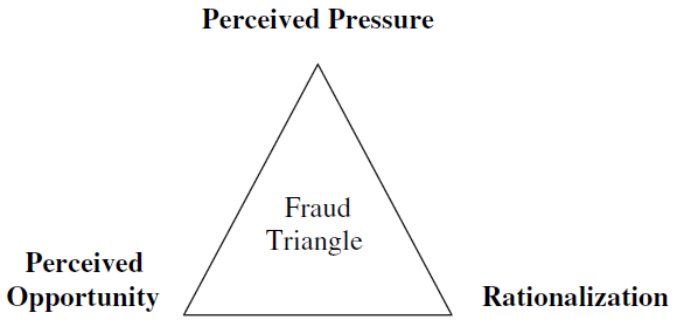


Figure 16: The Fraud Triangle. Taken from (Albrecht et al., 2008a, p.3)

2.2.2 Fraud Detection

Because of the enormous costs associated with fraud, it is evident that prevention and detection are crucial. Forty-nine percent of victim organizations do not recover any losses that they suffer due to fraud. However, the ACFE found that victim organizations that had implemented any of the 16 anti-fraud controls the ACFE defined, experienced considerably lower losses and time-to-detection than organizations lacking these controls (ACFE, 2012, p.8). This is a shared responsibility of both management and audit; whereas management has the best overview of the current state of the organization, auditors are working with the design, implementation and evaluation of (internal) controls on a daily basis (Coderre, 2009, p.7). However, only 40% of occupational frauds is detected by actual detection mechanisms; over 50% is detected by tips or by accident (ACFE, 2012, p.14). Internal audits do not specifically look for fraud, and only analyze a sample due to time constraints. Therefore they can only provide reasonable assurance, which creates a risk for a lot of illegitimate activities to be missed. Albrecht et al. (2008b) suggest the use of fraud audits on order to change the way fraud can be detected. The authors suggest that the major difference with regular audits should be in the purpose, scope and extent, both in method as well as size.

Over the last decades various models have been developed to aid accountants and auditors in the detection of fraud. One of the first people to publish a study that used a statistical model was Altman (Lenard & Alam, 2009, p.4). While this model was developed for the detection of bankruptcy, bankruptcy is closely related to fraud detection because analysis of the financial statements to detect potential bankruptcy can also detect fraud. Altman used financial ratios as variables in his discriminant model to analyze liquidity, profitability, leverage, solvency, and activity. In 1980 Ohlson published a study that used a logistic regression decision model rather than a discriminant model to detect bankruptcy (Lenard & Alam, 2009, p.5). Instead of a score, as in Altman's model, Ohlson promoted his model as one which developed a probabilistic estimate of failure. Besides using various financial ratios similar to Altman, Ohlson had several qualitative variables. Later studies focused specifically on fraud rather than bankruptcy. In 1995 Person used a logistic regression model to successfully identify factors associated

with fraudulent financial reporting and in 1996 Beasley completed an empirical analysis of the relationship between the board of directors' composition and financial statement fraud (Lenard & Alam, 2009).

In order to cope with the increase in effort various authors have proposed increased use of IT in the audit process. In the book 'Computer-Aided Fraud Prevention & Detection' (Coderre, 2009), the author describes a variety of techniques that can aid auditors and investigators in their work. Because of the increased usage of IT, IT will also be a bigger part of both fraudulent behavior and its detection (Coderre, 2009, p.41). By using computer based tools, auditors can conduct analyses on entire datasets, or subsets thereof, rather than selecting a part of the dataset for inspection. The author suggests a variety of techniques that can be applied for fraud detection:

- Filtering can be used to select only a specific part of the data set based on some criteria, that contains records which show indicators of being more suspicious of fraudulent behavior.
- Equations can be used to recalculate e.g. inventory levels to see if all goods are accounted for.
- Gaps can be found in check or purchase order numbers, indicating possible fraudulent behavior.
- Statistical analysis can be used to analyze a number of numerical aspects such as sums, averages, deviations, min-max values, etc. The resulting outliers can then be used to take a better sample for further analysis.
- Duplicates can be a good indicator of fraud, e.g. duplicate vendors, contracts or invoices.
- Sorting can be used to identify records with values outside of normal range, which can be interesting candidates for further analysis.
- Summarization can be used to divide the dataset into specific subsets, which can then be further analyzed using any of the other techniques mentioned.
- Stratification is used to group data based on the numerical values rather than other attributes as done with summarization.
- Pivot tables can be used to analyze data from different angles, and to assess multiple attributes / values of the data in one overview.
- Aging is concerned with the difference in timestamps / dates of the respective data entries. Verifying dates can be a significant part of controls. Furthermore aging can be combined with summarization or stratification.
- Joins can be used to combine different data sources. Data that shows no exceptional behavior might show indicators of fraudulent behavior if combined with other data sources.
- Trend analysis can be a good tool to find fraudulent behavior. Even when someone tried to obfuscate the fraud, a trend analysis can still indicate unusual behavior.
- Regression analysis is used to see if data values are in accordance with expected values. Relations between variables (i.e. data values) are used to determine the expected values.
- Parallel simulation re-enacts the business processes and compares the simulated outcomes with the actual outcome. When there are significant differences, this could indicate fraud.

The author presents a variety of known indicators for fraud; part of these is specifically aimed at purchasing, as this area is particularly vulnerable to fraud (Coderre, 2009, p.185). Examples are

concerned with for instance fixed bidding, wrong quantities of goods received and duplicate invoices. Not all of these indicators are specifically suited for detection by process mining; some indicators are most likely comparable in the effort required to find them and some indicators are easier to find using process mining compared to 'regular' data analysis.

Besides the techniques mentioned by Coderre (2009), other more advanced tools and techniques are also finding their way in fraud detection. Yue et al. (2007) provide a review of 26 articles from the late 1990's till early 2000 researching the application of various data mining algorithms in the detection of financial fraud. In their findings they conclude that most researchers were reasonably successful using either a regression or neural network approach, and that all authors used a supervised/classification approach, where possible fraudulent cases were known beforehand.

Other research continues along the same line: Hoogs et al. (2007) successfully use a genetic algorithm to mine financial ratios in order to detect indicators of financial fraud. However, fraudulent behavior was again known beforehand when training and testing the models. Kirkos et al. (2007) compare decision trees, neural networks and Bayesian belief networks, and conclude that there are indeed indicators of possible frauds in financial ratios. Jans et al. (2007; 2010) use an unsupervised (possible fraud was not known beforehand) clustering technique to find deviations in procurement data, and conclude that the results show a very well usable application into fraud detection and prevention. In later work (Jans et al., 2009), the authors present a framework for using data mining for fraud detection. The authors reused and adapted this framework in subsequent work to incorporate process mining, as discussed in the next chapter.

In addition to developments in the accounting and auditing field, governments and regulators have increased their efforts to prevent and detect fraud. A number of large scale frauds have been uncovered over the last decades, such as Enron, Parmalat, or Ahold. Because of their tremendous impacts on society, politics, and stock markets, there have been a lot of initiatives to counter fraud and improve regulations. The most well-known are probably the Sarbanes-Oxley Act and the establishment of the Public Company Accounting Oversight Board in the United States in 2002, and the SAS 82, updated by the SAS 99, by the American Institute of Certified Public Accountants. In the United Kingdom, the National Fraud Authority was established in 2008, and in The Netherlands the code-Tabaksblat was introduced in 2004. Also, organizations like the Information Systems Audit and Control Association continue to maintain the COBIT (Control Objectives for Information and Related Technologies) framework for IT management and IT Governance.

For a more extensive discussion on (types of) fraud, fraud detection, and auditing, the reader is referred to (Bologna & Lindquist, 1995; Wells, 2005; Davia et al., 2000; Podgor, 1999; Coderre, 2009)

2.2.3 <<REMOVED DUE TO CONFIDENTIALITY>>

<<REMOVED DUE TO CONFIDENTIALITY>>

2.3 Summary

This chapter presented the findings of literature studies and expert interviews on process mining and fraud detection. The three process mining aspects (discovery, conformance and enhancement) and

some of the functional techniques were discussed. Furthermore, the notion of what fraud is as well as practical aspects of its detection were discussed. The existence of certain indicators and techniques such as summarization, stratification or trend analysis to discover these indicators were discussed. In the next chapter the combination of the two topics is examined by looking at case studies performed by other researchers. The tools and techniques mentioned in these case studies will be synthesized to create the setup of the case studies in this thesis.

3. Fraud Detection and Process Mining

This chapter presents an overview of the historical developments of fraud detection using process mining. After a brief overview of related work, earlier case studies and practical approaches are given. These approaches and techniques are then synthesized into initial guidelines for using process mining for fraud detection. The techniques and tools mentioned in this chapter will be explained when used in the practical part of this thesis.

3.1 Developments in Process Mining Supported Fraud Detection

While various authors have researched the use of data mining techniques for fraud detection, Van der Aalst & de Medeiros (2005) were one of the first to combine process mining with anomaly detection. They used token replay (described in Section 2.1.4) to detect process deviations to support security efforts at various levels such as intrusion detection and fraud prevention. Yang & Hwang (2006) claim to use process mining to detect healthcare fraud. However, their approach significantly deviates from the concept of process mining used in this thesis. Based on the steps in ´clinical pathways´ in healthcare, they mine for structural patterns in a way that is comparable to the A-Priori algorithm known from association in data mining. They then used an inference-based approach to predict fraud[3].

In Rozinat et al. (2007) the authors apply the concept of process mining to conduct an audit on a process, focusing beyond just fraud detection. The authors show the use of process mining for various aspects of the audit process. Bezerra & Wainer (2007; 2008a) focus in their work on the detection of fraud using conformance checking of traces. After comparing three different metrics (fitness, behavioral and structural appropriateness) to see which is most useful for fraud detection, they note that the accuracy of the conformance checking is related to the process mining algorithm, the metric used to evaluate the "noise" of a trace in the log, and the threshold value used to evaluate the deviation magnitude (Bezerra & Wainer, 2007). In subsequent work (Bezerra & Wainer, 2008b) the authors take a different view on conformance and anomaly detection. They reason that: *"because some paths in the process model can be enacted more frequently than others, it is probable that some 'normal' traces be infrequent. For that reason, we do not believe in an anomaly detection method based only on the frequency of traces. […] This SIZE metric was defined and used in this study because we believe that a log with anomalous traces induces a process model that is more complex than a model induced by the same log without anomalous traces. That is, we believe that a model mined with normal and anomalous traces will have more paths than a model mined without anomalous traces."* (Bezerra & Wainer, 2008b, p.4)

A first attempt to structure the use of process mining was described by Bozkaya et al. (2009), who proposed a methodology to perform process diagnostics based on process mining. Prior and domain specific knowledge was absent; the only information available was the event log. The methodology consists of five phases: log preparation, log inspection, control flow analysis, performance analysis and role analysis. The authors conclude that, based on a case study, the approach is useful to get a quick first glance of the larger parts of the process, but results have to be handled with care to prevent misinterpretations. The proposed methodology of Bozkaya et al. was further assessed by Jans et al.

[3] For more information on the A-Priori algorithm and inference, the reader is referred to (Tan et al., 2006)

(2008). Initially their focus was on fraud detection and risk mitigation, by adding process mining to their previously developed (data mining) framework (Jans et al., 2009) for fraud detection. The authors describe the various steps they take, and conclude that their approach can be a valuable addition to (continuous) auditing as well as fraud detection. In subsequent work (Jans et al., 2010; 2011; 2011; Alles et al., 2011) the authors reevaluate and refine their approach. They once more conclude that process mining can provide a contribution to business practice, as well as auditing, and suggest that process mining could even fundamentally alter these practices. This is supported by the work of van der Aalst et al. (2010, p.5), who claim that *"Auditing 2.0 - a more rigorous form of auditing based on detailed event logs while using process mining techniques - will change the job description of tomorrow's auditor dramatically. Auditors will be required to have better analytical and IT skills and their role will shift as auditing is done on-the-fly".* An interesting effort into formalizing this idea is presented by Van der Aalst et al (2011). In their work, the authors present a formalized framework for online auditing, consisting of various conceptual tools which use e.g. predicate logic and Linear Temporal Logic (LTL) (van der Aalst et al., 2005a) to check conformance to various (business) rules and compliance aspects.

3.2 Related Case Studies Evaluation

There are two concerns while analyzing the mentioned approaches: the structure of the executed procedures (i.e. what is done, cf. Bozkaya et al.'s (2009) methodology), and the actual tasks and procedures (i.e. how it is done, e.g. process discovery using a Fuzzy miner, conformance checking using token replay).

The work of Bozkaya et al. and Jans et al. is taken as a starting point for determining the structure and procedures for a good process mining methodology. Bozkaya et al. aimed to *"propose a methodology to perform process diagnostics based on process mining …* [that covers] *… the control flow perspective, the performance perspective and the organizational perspective* […] *designed to deliver in a short period of time* […] *a broad overview of the process(es) within the information system"* (Bozkaya et al., 2009, p.1). The authors propose a methodology that is only based on the event log and requires no prior and domain specific knowledge, and therefore presents results that are objective facts. Throughout their work and in their conclusion, the authors put a lot of emphasis on communicating the findings of the analysis to all involved parties, in order to avoid misinterpretation. Note that objective fact finding is quite similar to the tasks of auditors in the fraud detection process: only indicators of fraud are provided, determining and judging actual fraud is done by others. As mentioned before, the methodology consists of five phases: log preparation, log inspection, control flow analysis, performance analysis and role analysis. What these phases consist of and how they were performed in the authors' case study is described as follows:

- Log preparation is concerned with the transformation of the data in the information system into a process mining format. This includes selection of sources, determining the cases, selection of attributes, selection of the time period, etc., and the conversion into a minable format such as XES or MXML.
- Log inspection is used to gain insight into the size of the process and the event log and to filter incomplete cases, which helps the evaluation in later phases. Steps include determining the number

of cases, roles, events, distinct event, events per case etc. In their case study, the authors used the Fuzzy Miner plugin in ProM[4] for process discovery to determine which activities were used as start and end activity, given some threshold. Cases which had other start and end activities were filtered from the log.

- Control flow analysis is used to discover what the actual process in the event log looks like. The authors suggest that this can be done by either checking conformance of a predefined model to the log, discovering the actual model using some process discovery technique, or both. With respect to the specific discovery algorithm, the authors warn for resulting spaghetti models and therefore suggest using the 80/20 rule by cleaning the event log of infrequent traces. This is analogous to the problem mentioned with noise in Section 2.1.3. The authors used the Performance Sequence Analysis plugin in ProM to discover the top 15 patterns that made up around 80% of total observed patterns, and how much of the observed patterns from the filtered log were in those top 15. The model discovered from these patterns (using an undisclosed discovery algorithm) was then checked for conformance.
- Performance analysis is concerned with determining bottlenecks in the process. Cases in the event log and their respective throughput time are analyzed using dotted chart and token replay analysis. Cases that show unusual behavior or performance can subsequently be analyzed further.
- Role analysis is used to determine relations between actors and events, and between actors. The authors suggest using a role-activity matrix (cf. the resource-activity matrix in Figure 10) to discover role profiles and role groups. This can be used to analyze the different work relationships between departments. Furthermore roles can be divided into generalists and specialists, or be used to create hierarchies. Another important part of the role analysis is the social network analysis, to analyze handover of work and subcontracting. In the case study the authors used the Organizational Miner plugin in ProM for the role analysis and social network analysis.

Jans et al. (2008; 2011) used the same approach as presented by Bozkaya et al. during a case study. Their focus was however specifically on internal fraud risk reduction in the procurement process, and was therefore somewhat different from Bozkaya et al. Again the five phases were carried out:

- During the log preparation all relevant activities including start and end activities were determined. Also a random sample was selected to improve computability and performance.
- The log inspection filtered out cases with incorrect start or end activities. The authors do note however that cases with an incorrect end activity are trimmed rather than removed to avoid bias. Again the Fuzzy Miner in ProM was used to get an initial idea of the process model.
- During the control flow analysis the Performance Sequence Analysis plugin in ProM was used to discover all observed patterns. In this case study, the top five and seven patterns made up for 82% and 90% respectively of all behavior. The events in the log forming the top five patterns were then used to create a process model using a Finite State Machine Minder in ProM, which was subsequently used to check conformance. Additionally to Bozkaya et al., the authors used the Fuzzy

[4] ProM is an open-source tool that supports a wide variety of process mining techniques in the form of plug-ins. More information on ProM and other tools used throughout the practical part of this thesis will be provided in the next chapter.

Miner with a lower threshold value to discover additional, less frequent patterns. The extra patterns showed behavior that was possibly deviating from accepted behavior. These deviations were then analyzed using the LTL Checker plugin in ProM to check whether the depicted behavior was actually seen in the event log, or just derived by the algorithm.

- Performance analysis is not included by the authors, as they claim that *"[performance analysis] can be very interesting when diagnosing a process, certainly in terms of (continuous) auditing, it is of less value in terms of internal fraud risk reduction."* This feels somewhat contradictory; it can be considered plausible that fraudulent process deviations differ in performance. Imagine fraudulent cases being stalled, or pushed through the system to divert human attention. Furthermore the authors appear to differentiate between fraud reduction and (continuous) auditing.
- The role analysis was performed in two steps. In the first step the authors created a role-activity matrix. In the second step, the LTL Checker plugin was used to check segregation of duty. The authors did not use the Organizational Miner plugin for either handover of work or subcontracting. It is unclear why these analyses were not performed, as the organizational mining can be considered a valuable tool according as described in Section 2.1.5.
- The authors did perform some other tests not mentioned by Bozkaya et al. Using the LTL Checker plugin other case properties were checked, e.g. 'order value per order type' and 'payment only if signed'

The approach was modified in subsequent work (Jans et al., 2010; 2011; Alles et al., 2011). While the steps in the mining process were altered, they consist for a major part of the same techniques used in the previous methodologies.

- The first step, process discovery, contains the techniques used in log inspection and control flow analysis. While the specific tools are not mentioned, the results appear to be made using the Fuzzy Miner and Performance Sequence Analysis plugins in ProM. Infrequent traces and sequences are identified and considered for later analysis.
- The second step is conformance checking, analyzing the fitness and appropriateness measures mentioned in Section 2.1.4. The authors suggest using the metrics described by Rozinat & van der Aalst (2008), but do not apply these to their case.
- The third step is performance analysis, contrary to Jans et al. (2008). As mentioned in the discussion of the previous paper, this is evident; rather than considering traces as deviations by just process flow, deviations can occur also with respect to time, or the actors involved.
- Social network analysis is the fourth step. Jans et al. (2010) do not present any practical results of applying social network analysis to their case, but refer to the techniques presented in van der Aalst et al. (2005), as described in Section 2.1.5. In Jans et al. (2011) this analysis was added, providing an overall social network, and a social network of cases not conforming to internal controls.
- The fifth step is decision mining and verification, similar to the last two steps of Jans et al. (2008). In these steps assertions regarding trace attributes and flows are verified. While the authors do not explicitly state which tool was used, they refer to van der Aalst et al. (2005a), the basis for ProM's LTL Checker.

The results of each of these case studies were translated into positive conclusions on the application of process mining for the case studies' respective aims. However, the data sets used were sometimes significantly modified before running the respective analysis. In the case studies by Jans et al. (2008, 2011b) the size of the dataset is significantly decreased, and furthermore a big part of the traces were trimmed to contain less activities. In the case studies by Jans et al. (2010a, 2011a) and Alles et al. (2011) the size of the event log was again decreased significantly for computability reasons. From a fraud perspective this is not an acceptable approach: completeness of the analysis is an important requisite in order to obtain usable results. Furthermore, in all case studies certain process mining aspects were only discussed rather than applied, providing no proof of actual applicability and / or utility.

3.3 Methods Synthesis

A summary of the approaches by the three (groups) of authors is provided in Table 1 below to compare approaches and procedures, and to distill a suitable structural approach. Between brackets, the practical techniques (i.e. programs/tools) applied in each step are listed.

Bozkaya et al. (2009)	**Jans et al. (2008, 2011b)**	**Jans et al. (2010a, 2011a) Alles et al. (2011)**
Log preparation	Log preparation	-
Log inspection (ProM Fuzzy Miner)	Log Inspection (ProM Fuzzy Miner)	Process discovery (ProM Fuzzy Miner, ProM Performance Sequence Analysis)
Control Flow analysis (ProM Performance Sequence Analysis)	Control flow analysis (ProM Performance Sequence Analysis, ProM Final State Machine Miner, Petrify, ProM Fuzzy Miner)	
-	-	Conformance check
Performance analysis (ProM Dotted Chart, Token Replay)	Performance analysis, but considered out of scope	Performance analysis (ProM Performance Sequence Analysis)
Role analysis (ProM Organizational Miner, ProM Role Activity Matrix)	Role analysis (ProM Role Activity Matrix)	Social network analysis (ProM Role Activity Matrix, ProM LTL Checker, ProM Social Network Miner)
-	Verifying properties (ProM LTL Checker: segregation of duties, (business) rules, process/control flow)	Decision mining and verification (ProM LTL Checker)

Table 1: Process Mining approaches summarized

From Table 1 it appears that there is a considerable consistency between the steps. Assuming that an event log is correctly created, the following steps with their respective actions should be present when using process mining for fraud detection. Also, the relation to the data analysis as presented by Coderre (2009) is given.

Analogous to Table 1, the initial step is the creation of the event log. Next, there are various analyses which can be grouped into five sets: log analysis, process analysis, conformance analysis, performance

analysis and social analysis. While not necessarily a distinct step, subsequent analysis is often needed to obtain conclusive results after some subsets of cases are identified. These five aspects will serve as the initial set of steps for applying process mining for fraud detection. The 'Iterate and Refocus' part is not so much a step as well as an encompassing notion; it must be acknowledged throughout the entire analysis process. After describing the setup of the practical case study and the details of the tools used, the tools and techniques will be performed to analyze their results and practical use. The results of this case study will determine whether the examined tools and techniques are included in the guidelines presented in Chapter 6.

1. Log Analysis
   - Getting an overall feel for the examined data is important; determining which activities are performed, which activities are start- and stop-activities, which actors are involved, which timeframe is examined, etc. This can lead to an initial filtering of the log, which leads to a more structured process model in later analyses. In order to be able to perform the log analysis, there should be a good understanding of the case itself. So the first part of the log analysis is to actually understand the environment of the case study; the company, its culture, its products, etcetera. While the focus is on the procurement process, other processes such as sales and production can severely impact procurement. Extra emphasis must be put on possible cutoff issues, depending on how the data was extracted.
   - Data analysis techniques involved are filtering, gap analysis, duplicate analysis, summarization and aging.
2. Process Analysis
   - Process Analysis can be considered a combination of 'Control Flow Analysis' and 'Process Discovery'. The goal of this analysis is to gain insight in the process, by way of visualizing the data. This is done e.g. by mining the process model, or using the ProM Performance Sequence Analysis plugin. Note that when mining the process model, the algorithm must be capable of handling noise; the research above suggests using the ProM Fuzzy Miner. The algorithm very much depends on the nature of the process; while the Fuzzy Miner is suitable for the initial (unstructured) process model discovery, the Heuristic Miner is suggested when the process (i.e. event log) is filtered (Rozinat, 2010).
   - Data analysis techniques involved are filtering, gap analysis, duplicate analysis, summarization, statistical analysis, joins, trend analysis and aging.
3. Conformance Analysis
   - From a fraud detection point of view, conformance checking is equivalent to analyzing whether the process (and its actors, data values, activities and/or steps) complies with various rules. Therefore, this step should be an encompassing step, similar to verification steps found in literature. Besides the metrics for fitness and appropriateness, provided by Rozinat & van der Aalst (2008) or Munoz-Gama & Carmona (2010), other conformance techniques are required that are more suited for compliance related conformance. Examples of such techniques are LTL (van der Aalst et al., 2005a), declarative concepts (Montali et al., 2010; 2011; Maggi et al., 2011), Petri net based (Ramezani et al., 2012) and runtime compliance checking (Maggi et al., 2012).
   - Data analysis techniques involved are filtering, gap analysis, duplicate analysis, summarization and parallel simulation.

4. Performance Analysis
   - Performance analysis can provide an insight into deviations that occur on other levels than control flow. As mentioned before, it seems plausible there can be cases that, even though they conform to the correct process flow, are pushed through or delayed in such a way that they escape normal compliance checks. ProM Dotted Chart analysis, ProM Performance Sequence Analysis or Disco's[5] performance view e.g. can provide valuable insights into deviating behavior.
   - Data analysis techniques involved are filtering, summarization, stratification, sorting, statistical analysis, trend analysis and aging.
5. Social Analysis
   - Social analysis can be used for various analyses. Analyses such as segregation of duty testing (which is actually a conformance check on originator level), handover of work, and other previously mentioned checks, can provide key insights into regular process executions and deviations. ProM provides excellent tools for social analysis, but also Disco has some capabilities. Whereas regular data analysis techniques can already provide insights into part of these tests, process mining can possibly provide more specific findings more easily.
   - Data analysis techniques involved are filtering, summarization, sorting, joins and aging.

Iterate and refocus

- When mining for fraud, just as with regular auditing for compliance, traces can be discovered that deviate over any aspect of the process. As mentioned earlier, process mining provides a view from a particular angle. Changing the view (i.e. the specific analysis) or the 'slice' of reality (i.e. the specifically filtered part of the log), can change how relatively fraudulent a specific trace appears. This is similar to the techniques described by Coderre (2009). Whereas he acknowledges that most of the tools and techniques are aimed towards filtering and creating subsets of potentially (more) interesting subsets of the data, the same is true for process mining. Hence, an interesting aspect of process mining for fraud detection is the ability to refocus, on a different slice of the event log, and redo the previous steps over this new slice. Traces that appeared to be deviant from a certain point of view can now be examined more carefully to get a stronger indication of (non-)deviation. Deeper analysis can then provide a better indication whether the discovered deviations are in fact abnormal or just low frequent. One should always 'branch out' and subsequently filter (sub)sets of the event log until definitive indicators are found (or found absent) regarding the respective subset. Sometimes this can be done using process mining techniques, sometimes this can be done with other techniques, and sometimes manual inspection by an auditor is required.

3.4 Summary

This chapter presented the results of a study on the current state of using process mining for fraud detection. Various case studies were analyzed to discover the aspects and techniques of process mining that were used and considered useful by other authors for fraud detection. These applied techniques, grouped into the five analyses in Section 3.3, and their resulting findings were synthesized to create an initial plan of action for the case studies in this thesis. Some of the tools mentioned are described in the

[5] Disco is a commercial tool for process mining. More information on Disco will be provided in the next chapter.

next chapter. In Chapter 5, the tools are subsequently applied in the case studies to evaluate the use of the tools and techniques process mining currently offers.

4. Case Study Introduction

This chapter describes the setup of the case study that is performed as practical part of this thesis. It elaborates on the dataset used, how it was gathered, how it was processed for process mining use, and which tools and techniques were used. Furthermore, the tools and techniques themselves are described to provide a better insight into what their purpose is and how different parameters affect their results.

4.1 Case Study Setup

During the practical part of this thesis, two datasets are used to examine the application benefits of using process mining for fraud detection. For anonymity reasons the specifics of the companies are omitted; it suffices to note that the companies are internationally active and have turnovers of several hundreds of millions of Euros.

The datasets were taken from the procurement process. There are several reasons for choosing this particular area: the procurement process is a reasonably structured process, which makes more suited to use for process mining (van der Aalst, 2011). Due to the many rules and regulations on different levels and from different perspectives, there are a lot of different aspects process mining can potentially check. Also, procurement data sets were the topic of most of the previously mentioned studies in Chapter 3, so it becomes more straightforward to compare results and methods. <<REMOVED DUE TO CONFIDENTIALITY>>

The procurement process entails the purchase of goods by the company from a supplier. As shown in Figure 17, the process starts with a requisition, the request or need for a certain product. Depending on internal policies this requisition may or may not have to be approved in order to be converted into a purchase order (PO), the actual ordering of goods at a certain supplier. When the PO is created, it creates a purchase order header (POH) and for each specific article a purchase order line (POL) indicating the quantity, price, etc. Again depending on internal policies, the PO is approved and then confirmed. Eventually the goods are delivered by the supplier; all goods of a PO can be delivered at once, or this can be done per (batch of) POLs. Goods are usually checked and then put into storage. At some point an invoice is also received; the amounts and prices are checked against the original PO(L) (the 3-way match) and is eventually paid.

A lot of variations to the procurement process exist. Depending on internal policies a lot of checks and controls can be in- or excluded. The above paragraph only provides a general overview of the procurement process.

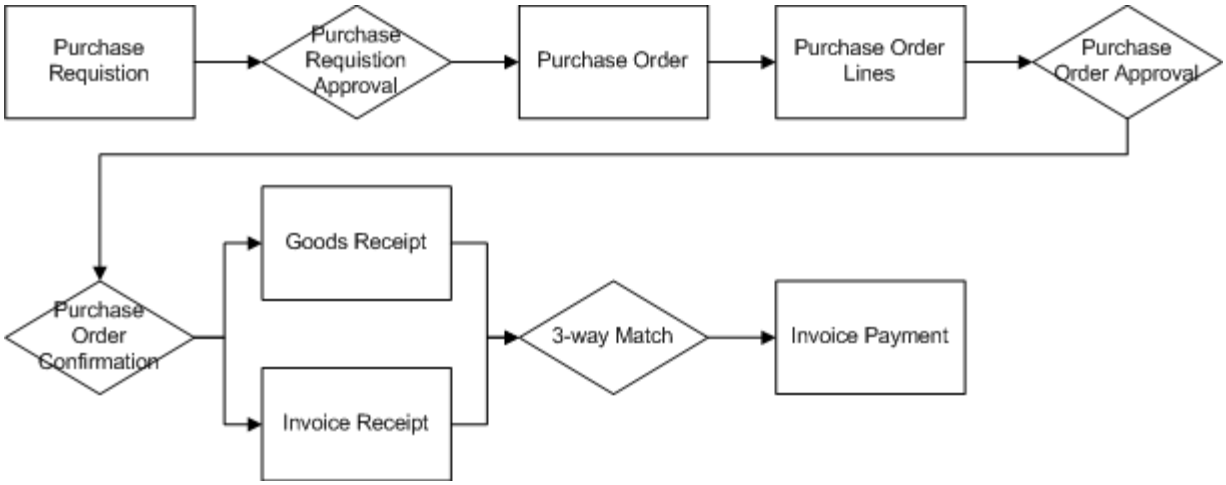


Figure 17: Overview of the general procurement process.

4.2 Event Log Creation

In general, an important aspect of process mining is the creation of the event log. There are many practical aspects regarding how to do this, as mentioned earlier in Section 2.1.1, but for fraud detection the cutoff is important. Corresponding to Van der Aalst's (2011, p.113) snapshot challenge, the cutoff determines the period over which the data is extracted; both the range of the timestamps as well as the moment at which the data is extracted can affect the resulting event log and thus the outcomes of the process mining.

Consider for instance an event log that ranges from January 1st to the 31st of December: if a PO get created on the 29th of December, it is very unlikely that the entire process will be finished before January 1st of the following year. Most likely the receipt of the goods and /or the invoice will not be recorded in the data and will thus be missing in the event log. Similarly, POs that were created at the end of the previous year will be delivered at the start of the current year. Whereas e.g. the receipt of the goods will be recorded, the creation or any other events regarding the PO will not be included in the event log.

The moment at which the data is extracted also affects the eventual process mining results. Because of the relational structure of the data in the source system, it can be possible that some data attributes are changed during the period between the extraction and the end-time of the extraction period. Consider the situation in which a PO is created on December 28th, delivered on January 5th and invoiced on January 28th (and no other event regarding the PO occurs). Depending on the date of the data extraction, the following situations can arise:

- If the data is extracted before January 5th the PO will not be changed. Any data regarding the PO will be correct.
- If the data is extracted between January 5th and January 28th, the delivery event will not be recorded. However, there might be some field in the PO table recording the delivery status of the PO. This field will indicate that the PO is delivered, but the actual event will be missing from the log. The same can happen with the invoicing event and status if the data is extracted after January 28th.

- Similarly, it can be possible for an invoice to arrive during the extraction period, for a PO that was created and delivered before the extraction period. This may result in the impression that the invoice is the only known event regarding the PO and that the invoice is fraudulently fabricated.

From both a process mining and a fraud perspective these issues can cause severe misinterpretations of the data if not accounted for. On the other hand, it can provide grounds for filtering out part of the data (i.e. cases) that is concentrated around the cutoff edges. Unfinished and/or incorrectly started cases can severely distort the process mining results and should therefore be removed from the log. If the filtered cases were not already analyzed during an earlier process mining analysis they should be noted to make sure that these cases are indeed incorrectly recorded and not actually missing events.

Problems with the cutoff can partly be solved by using a soft rather than a hard cutoff. In a hard cutoff only the timestamp is used to determine what is extracted, any event outside of the date range are omitted, resulting in missing events and thus incomplete traces. In a soft cutoff the process is taken into account as well; various choices can be made to include events inside or outside the date range. In Figure 18 the different possibilities are shown. These are actually taken from the Disco Timeframe filter, but the concept is identical. Cutoff types (a) to (d) are soft cutoffs. However, because data source systems are very unlikely to be process-aware up to the point that they know when a process is finished, cutoff type (a), and depending on the time of extraction type (d) as well, will not be possible during data extraction. Types (b) and (c) are an option, depending on how the system and the extraction is set up. Type (e) is the hard cutoff. It immediately shows how the cutoff affects the problems described above.

4.3 Applied Tools

During the case studies, various practical tools will be used. The tools that are described in literature, mentioned in Chapter 3, are likely to be of use for process mining for fraud detection. Therefore they are briefly described in this section.

ProM

The tool most widely recognized in the world of process mining is ProM[6]. ProM was initially developed by the Process Mining Group at Eindhoven Technical University. It is a Java based framework, that can be extended by a variety of plugins to perform different process mining techniques. At the time of writing there are two versions of ProM available, 5.2 and 6.2. The release of version 6.0 denoted a complete overhaul of the program, which effectively means that the 230+ available plugins for ProM 5.2 have to be converted to be usable in ProM 6.2. While this is a still ongoing process, a significant part of the ProM 5.2 plugins are not (yet) available for ProM 6.2. This in combination with the fact that some newer plugins are developed only for the 6.2 version leads to a considerable discrepancy in available plugins and functionality between the two versions. During the execution of the analyses version 5.2 is used if the required plugin functionality is not yet available for version 6.2. For fraud detection, ProM can be used for all of the five aspects mentioned in Section 3.3. Log analysis can be done using the basic log inspection plugins, to detect frequency of activities e.g. Process analysis can be done by using a variety

[6]More information and technical specifications can be found at http://www.processmining.org and http://www.promtools.org/

of different mining plugins, to detect the process model. The Performance Sequence Analyzer plugin can be used to find the most (or least) common patterns. Conformance analysis is done only using ProM, as it is the only tool supporting conformance checking. Various performance analysis plugins such as the Dotted Chart can be used to examine timing / performance based aspects of the process. Social analysis is done using ProM plugins such as the Social Network Miner or the Role-Activity-Matrix.

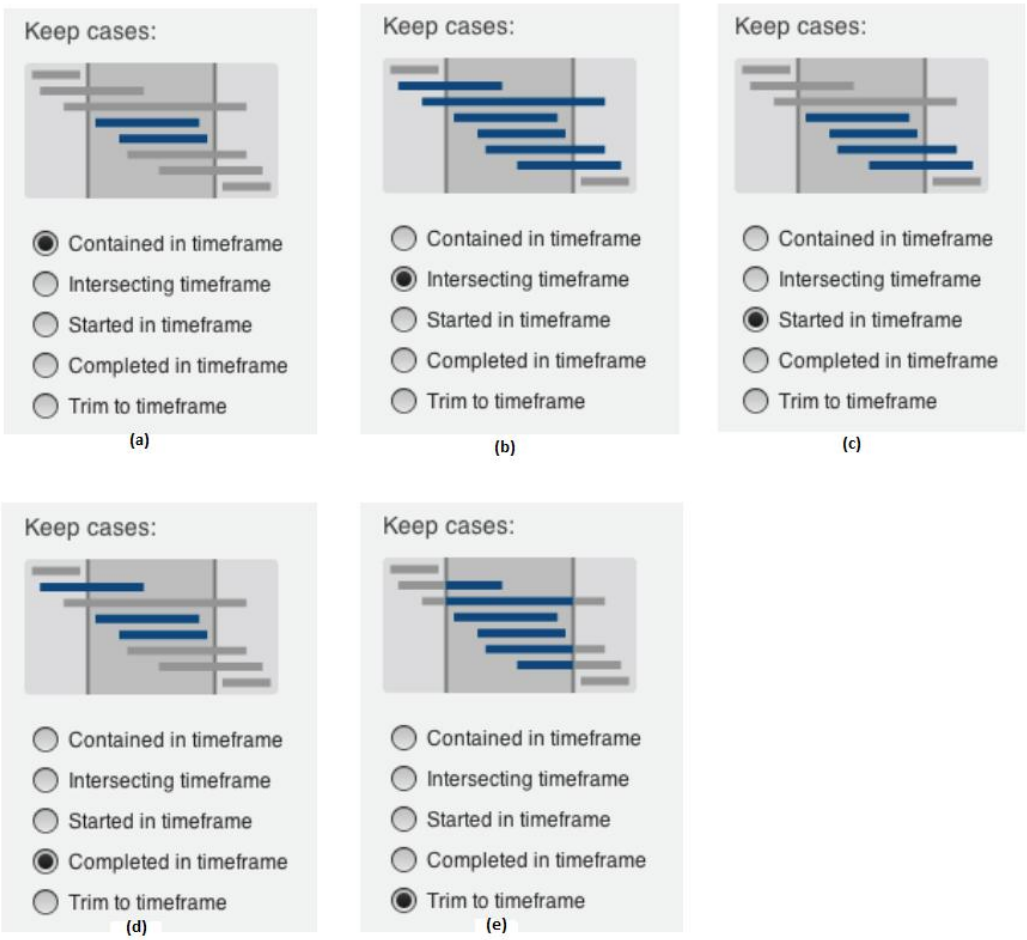


Figure 18: Cutoff types

Disco

Another tool used for process mining is Disco[7]. Disco can be seen as a commercial form of ProM; it is developed by the same company that is also highly involved in the development of ProM. Disco is currently aimed more towards the process discovery and process enhancement aspects of process mining. Even though conformance testing and other more advanced aspects are not yet available, Disco's current capabilities are very user friendly compared to ProM. For process discovery and log inspection, Disco can be preferred over ProM. As it uses the same formats as ProM, it is relatively easy to switch between Disco and the different versions of ProM, or to use both tools simultaneously. Disco can

[7] More information and technical specifications can be found at http://www.fluxicon.com/disco

be mainly used for the fraud detection aspects log analysis and process analysis. Conformance analysis is not really supported by Disco. Performance analysis can be done using filters on case duration or number of events. Filters can also be used to check SoD aspects of social analysis; with a workaround also a handover-of-work network can be created. Disco's biggest advantage is the usability of its filters, which can be based on a variety of case attributes. They can be combined very easy and quickly to filter out special subsets of cases that have indicators of fraudulent behavior. The filters in Disco are comparable to concepts known from BI; drill-down, slice and dice, etc. can be used to split or recombine different subsets of the data.

Heuristic Miner

The ProM Heuristic Miner plugin, and the enhanced ProM (6.2 only) Flexible Heuristic Miner plugin (Weijter & Ribeiro, 2010), is a process discovery plugin that tries to deal with low structured processes, non-trivial constructs, and / or a lot of noise. Like the α-algorithm, the FHM algorithm uses log based ordering relations to determine model semantics. The big difference with the α-algorithm is that frequencies are taken into account when determining the relations. By setting three initial thresholds for dependency, length-one loops and length-two loops, different noise types can be included to or excluded from the model. The FHM algorithm has a few other noteworthy parameters: the 'Positive Observations' parameter is the absolute minimum number of log observations that is required before a relation between two activities is added to the model. The 'Long Distance Threshold' is used to add relations that are not directly evident, but are possibly still present. A very important threshold is the 'Relative-to-best' threshold. When an activity has multiple possible outgoing edges, this parameter determines how much lower the frequency of the activities other than the best one found may still be to be added to the model. The other parameters unfortunately lack documentation. For fraud detection, the Heuristic Miner is only used for the process analysis aspect. By discovering the process model, loops or skipping of activities can e.g. be discovered, which might indicate fraudulent behavior.

Fuzzy Miner

The Fuzzy Miner (Günther & van der Aalst, 2007) plugin is a process discovery plugin in ProM. It was developed to address the problems of large numbers of activities and highly unstructured behavior. Like the Heuristic Miner it uses significance (i.e. 'X follows Y' relation observation frequency), but it combines significance with correlation (i.e. 'X follows Y'-naming or data element correlation). Based on these metrics and their respective sub-metrics, the algorithm is able to aggregate and abstract events into clusters. Furthermore it uses graph simplification techniques to create a more structured process model. While the resulting model cannot be converted into other representation languages, Fuzzy Models can be used to animate the event log on the model to get a better visual interpretation of the process flows. The Fuzzy Miner plugin has too much parameters to describe. If during the practical part of this thesis parameters are changed, they will be discussed when and where needed. For more in-depth information on the different parameters, the reader is referred to Günther & van der Aalst. An important note is that Disco also uses some undisclosed variant of the Fuzzy Miner. While it lacks the user-definable parameters, it is often able to provide a reasonably structured process model very quickly. Just as with the Heuristic Miner, the Fuzzy Miner is only used in fraud detection for discovery of the process model.

Performance Sequence Analysis

The Performance Sequence Analysis plugin from ProM (5.2 and 6.2) can be used to get a better insight into the performance of the different variants of the executed process. It can be used to group similar traces into patterns, or variants, and calculates throughput time metrics. Besides control flow, other data attributes of a trace can be used to base the grouping into patterns on. Furthermore this plugin has a filtering option which lets the user select different (groups) of patterns based both on control flow and / or throughput time. Note that while this plugin is available in ProM, part of its functionality (including advanced filtering) can be mimicked by the 'Cases' view in Disco. More information on the Performance Sequence Analysis plugin can be found at the ProM Online Help webpage[8]. For fraud detection, the use of the Performance Sequence Analysis comes from the grouping of cases into pattern variants. The most common variants can be used to create the general model, used in conformance checking, while the least common variants can be used as an initial filtering for possible subsequent analysis. Also, the plugin's capabilities of showing minimum, maximum, and average throughput time can be used for the performance analysis aspect.

Organizational Miner

The Organizational Miner is only one of five social mining related plugins in ProM. As described in Section 2.1.5, it is concerned with the organizational perspective. There are four other plugins in ProM which focus on this perspective: the Social Network Miner, the Role Hierarchy Miner, the Semantic Organizational Miner and the Staff Assignment Miner. Since it is inconvenient to describe all of them in detail, the plugins, the parameters and results will be discussed when and where needed. Disco has no real organizational mining capabilities. It is however possible to create a model that shows a process model based on resources rather than activities, i.e. a model that shows the control flow of resources involved in a process trace, in other words the handover-of-work network.

Role Activity Matrix

The Role Activity Matrix is equivalent to the Resource Activity Matrix described in Section 2.1.5. While it is a fairly simple analysis tool, it can be used to quickly rule out some compliance / fraud issues regarding segregation of duty. While it does not show the activities per role on a per-trace basis, yet if no single actor executes multiple actions that are possibly a violation, one can ensure that the actions are also never executed by the same actor in a certain trace. Furthermore, it can provide insight into the frequency with which originators perform some activities. If e.g. a person performs activity A 10.000 times and activity B only 10 times, this may be reason for suspicion; it is questionable if this person should be involved in activity B at all.

LTL Checker

The Linear Temporal Logic (LTL) Checker plugin is used to specify and check whether a variety of logic statements hold within a log. It can be seen as an extension of propositional logic, taking order and temporal aspects into account. A complete description of LTL is out of the scope of this thesis, but a small overview is provided. Using LTL, temporal constraints and checks can be specified for next-time (after activity X, activity Y must follow directly) , eventually (after activity X, activity Y must eventually happen), always (specifying invariants), and until (until activity X, activity Y must not be executed). More

[8] http://www.processmining.org/online/performancesequencediagram

importantly for fraud detection constraints can also be specified on other attributes rather than just on activities. For example , this way the 'four eyes' principle constraint can be specified, by checking whether or not, in pseudo-code,

ALWAYS (NOT (Activity_By(Check_1, Person_1) AND (Activity_By(Check_2, Person_1)))).

Note that while Disco does not have an explicit LTL checker, it can use its filters to filter out traces based on LTL-like constraints, e.g. by selecting all traces that contain activity A that are eventually or directly followed by activity X. For more information on LTL and the LTL Checker plugin the reader is referred to van der Aalst et al. (2005b).

4.4 Summary

In this chapter procurement is discussed as the domain of the case studies. Furthermore the concept of cutoff (or snapshot as called by Van der Aalst(2011)) during data extraction and event log creation was discussed. Finally, part of the tools used in the examined case studies from Chapter 3 were discussed. In the next chapter the case studies are performed analogue to the 1+5+1 steps presented in Section 3.3, using the tools mentioned in Section 4.3. This way, the tools and the techniques can be tested in order to assess their usefulness, and can be assessed whether or not the current state of the tools is sufficient for the purposes of fraud detection. This assessment can also present recommendations for further tool improvement and development, but this is beyond the scope of this thesis.

5. Practical Results

<<REMOVED DUE TO CONFIDENTIALITY>>

5.1 Case Study 1

<<REMOVED DUE TO CONFIDENTIALITY>>

5.1.7 Case Study 1 Synopsis

In the previous sections we have identified various subsets of the event log with properties that are potentially more interesting from a fraud detection point of view. Given below are the most noteworthy findings of the analyses and subset identifications.

During the log analysis a lot of cases with only invoicing events were identified. Because of the lack of data most of these were removed from the event log, however 136 and 123 cases were identified that did have entries in the POL and POL plus the PO table respectively. These cases are missing a lot of information and should be investigated to find out what happened. We identified 26 cases that were lacking a 'Closed' event but were set to 'Closed' in the POL table regardless. After investigating these cases there are six cases left that show unexplainable behavior. Using regular data analysis 53 cases were identified that concerned a POL whose PO was cancelled but still invoiced. This would however not have been detectable by process mining, as the cancellation was on PO level and not POL level which was used in our case study. A different design of the event log on PO rather than POL level could possibly have detected this using process mining.

During the process analysis the most common variants were identified and grouped into the general model. The 'DescriptionChanged', 'QuantityChange', 'CopiedFromOrder', 'CustomerOrder' and 'Stopped' activities were not present in this general model. The 'DescriptionChanged' activity does not provide an apparent reason to assume fraud is more or less likely in these cases. Cases with a 'QuantityChange' activity can be more vulnerable to fraud however, since the change in quantity also changes the value of the PO(L). There is no information on the specific meanings of the other three activities, so these will not be considered unusual by this line of reasoning. Looking at **Fout! Verwijzingsbron niet gevonden.** however, the very low event frequency of these activities does however warrant further analysis.

The conformance analysis did not provide results due to performance problems. Consequently no special subsets of cases were identified. If however certain process flows are determined beforehand, Disco's filter capabilities do provides means to check if certain events follow other certain events. With conformance analysis, we would have been able to create a collection of all cases that did not conform to the model, not just cases that deviate from a certain process flow.

During the performance analysis subsets of originators and suppliers were identified that were over-involved in rushed and long-duration cases. While rushed and long-duration cases are not necessarily fraudulent, it can provide additional indicators and / or reason for further analysis. With the identification of subsets of special interest, other tools can be used in order to analyze statistical measures.

As with conformance checking, the theoretical value of using process mining social analysis techniques are evident, but the practical use is questionable due to the very poor performance of the available tools. Of all the possible social analysis techniques discussed earlier, only the Role-Activity-Matrix provides results that are useful for fraud detection. These results serve as an initial step, and must be supplemented by Disco's Follower-filter or ProM's LTL-Checker plugin.

During creation of the event log a number of cases was removed. The cases that were not 'Closed' according to the POL table were removed, as discussed earlier. Also, some events were aggregated and became redundant because of that, and were subsequently removed; if a specific analysis would require these events not to be aggregated the event log has to be rebuilt. Regardless, it is only the events that are removed because of aggregation and grouping and not the cases. Because of both the aggregation and grouping of events and the way the system records its events, a lot of timing issues were resolved to ensure determinism. Furthermore, identical invoicing events were removed as we argued that they are most likely recording errors by the system; regular data analysis can be used to identify these cases if required.

Also related to the event log creation was the fact that the results presented here are in fact a second attempt. During the first execution of the case study, a different choice was made as to how the data from the POLH table should be used to create the event log. Because of this difference, similar issues arose as described earlier with events having identical timings. During the performance analysis it became clear that these issues were essential to the design choice made and they could not be solved. With insights gathered during the previous analyses a new event log was made that proved to solve these issues. Even though previous results were rendered useless, the insights and skills acquired enabled us to redo the same analyses in about a quarter of the time. Even though it is not a direct result of the analyses, this does acknowledge the notion that different views can provide different results, as well as the requirement for a sound understanding of the available data.

The results of the process mining analyses from this case study cannot be compared to results of a regular data analysis. Apart from being beyond the scope of this thesis, not all data required for a regular data analysis was available. Using our process mining analysis techniques, we were able to identify a number of subsets of cases that showed indicators of unusual and possibly even fraudulent behavior. The value of the indicators does however vary; some indicators (such as e.g. the two cases provided in the social analysis) are more meaningful for concluding that a case is likely to be fraudulent than others (such as e.g. the occurrence or absence of certain activities). The results of the case study were discussed with the data owner, who based on the results would investigate some of the subsets further. While any actual fraud is unfortunately not disclosed, the data owner acknowledged that some interesting new insights were provided that were previously not known.

Table 2 summarized the numbers of cases found during the analyses in this case study; when the 'No. of cases' column has a question mark, the indicator was noticed but not explicitly followed up on.

Process mining step	No. of cases	Indicator

Log analysis	123	Missing activities
Log analysis	26 (6 not explained)	Missing 'Closed' event
Log analysis	?	Case continues after 'Closed' event
Performance analysis	2.769	Rushed 'Released'→'Arrived'
Performance analysis	370	Rushed 'Released'→'Arrived' by over-involved originator
Performance analysis	2.072	Delayed 'Arrived→'Received'
Performance analysis	19 + 19 +24 + 16 + 15 + 12 + 79 + 34	Delayed 'Arrived→'Received' by over-involved originator
Performance analysis	95 + 93 + 82 + 81 + 69	Delayed 'Arrived→'Received' by over-involved supplier
Social analysis	172	Possible SoD violation; involvement in 'Released' and ('Arrived' or 'Received')
Social analysis	?	Possible SoD violation; performing low-frequent activities out of regular tasks
Regular data analysis	1.471	Cancelled POL with related invoice

Table 2: Case study 1 results

5.2 Case study 2

<<REMOVED DUE TO CONFIDENTIALITY>>

5.2.7 Case Study 2 Synopsis

During the log analysis several subsets were created and filtered out of the eventual event log. Because of the low frequency and unclear meaning of the activities, the '0-' and '21-' events were removed. Start stop analysis of the events led to the overview in **Fout! Verwijzingsbron niet gevonden.** and the removal of these events from the log. While most of these subsets appeared to be caused by cutoff issues, endpoint analysis provided a number of subsets with unusual behavior. Furthermore, there also appeared to be very long periods between some activities, and long case durations in general, which in suggests problems due to cutoff. There are however some subsets of cases that cannot be reasonably be explained by cutoff issues, so these should be analyzed in detail by an auditor.

The process analysis showed that the process was very structured in general, since 70% of the cases were contained in a single variant, and almost 99% of the cases fitted in the top five variants. Also, the general process model made from these top five variants, shown in **Fout! Verwijzingsbron niet gevonden.**, appears to be quite straightforward. Therefore, cases that are non-conformant to these five patterns are very unlikely and possibly even more likely to contain fraudulent behavior.

Despite the problems with conformance analysis in the first case study, an attempt was made to run the conformance analysis. While it was not possible to analyze the full log, the analysis was ran on the cases started in 2009. This resulted in 313 out of 37.663 cases being non-conformant; the process flow of this subset of cases is shown in **Fout! Verwijzingsbron niet gevonden.**. Due to the relative ease with which the conformance analysis could be performed, compared to the first case study, the utility of conformance analysis for fraud detection has increased. However, it still requires more time and effort than analyzing each potentially fraudulent activity-to-activity relation individually.

During the performance analysis, the times between ’20-naar 1e controle’ and ’40-naar administratie’ were analyzed to see if any cases were rushed during their approval. Combined with a SoD check, we found out that while overall this specific SoD constraint is violated in around 10% of cases, when the time between these respective events is less than five minutes, the share of cases that have a SoD violation increases to 88%; when the time between these respective events is less than one minute, the share of cases that have a SoD violation increases to 97%. While theoretically one can argue that the SoD violation will be noted anyway because of the logging of the originators, this is actually an important finding, as there are a number of cases where the originators of these events are unknown. Even more so, since the likelihood of SoD violations with one or more missing originators will only increase due to obfuscation attempts.

The social analysis consisted of multiple SoD checks, of which the results are shown in **Fout! Verwijzingsbron niet gevonden.**. There are numerous SoD violations that can be considered fraud indicators, and provide grounds for further manual analysis. A similar analysis was previously performed using regular data analysis techniques. The results of this analysis were comparable to a Role-Activity-Matrix, but instead of the frequency of activities, the total invoice value was summed. Unfortunately it is not possible to compare the results of this data analysis to the results presented in the social analysis section. This has various reasons: first of all process mining tools are currently not able to have the Role-Activity-Matrix process the attributes of cases, such as value, but only list the frequency of the respective activities per originator. Furthermore if this would have been possible, since an originator can perform the same activity multiple times in a single case, this would distort the results as the value is counted multiple times. The results are on a per-activity basis rather than a per-case basis. Also the timeframes of both analyses were different. The results of the data analysis were on a much smaller (sub-)set of the data used in our analysis.

Table 3 summarized the numbers of cases found during the analyses in this case study.

Process mining step	No. of cases	Indicator
Log analysis	6	Occurrence of ‘0-’ event
Log analysis	1.082	Case ends with ’20-naar 1e controle’ event
Log analysis	444	Case ends with ’20-naar 1e controle’ event, over-involvement of originator
Log analysis	1.519 + 7	Occurrence of ’21-’ event
Log analysis	580 + 248	Case starts or ends with ’25-naar 2e controle’ event
Log analysis	53	Case starts with ’30-afgekeurd’ event
Log analysis	2	Case has ’30-afgekeurd’ event after ’40-naar administratie’ event
Log analysis	36.245	Case starts with ’40-naar administratie’; possibly normal process flow
Log analysis	251	Case ends with ’40-naar administratie’
Log analysis	160	Case starts with ’50-geprint’
Log analysis	86.173	Case ends with ’50-geprint’; possibly

		normal process flow
Log analysis	401	Case starts with ’60-nieuwe Kbon’ event
Log analysis	466	Case ends with ’60-nieuwe Kbon’ event
Log analysis	6.991	Case starts with ’80-Kbon PL’ event
Log analysis	8	Case has ’80-Kbon PL’ as only event
Log analysis	25	Case starts with ‘100-Kbon admin.’ event
Log analysis	2	Case ends with ‘100-Kbon admin.’ event
Log analysis	4	Case starts with ‘110-Kbon afgekeurd’ event
Log analysis	9	Case ends with ‘100-Kbon afgekeurd’ event
Log analysis	4.691	Case starts with ‘120-definitief’ event
Process analysis	1	Case is send back for 2nd check after having ended
Performance analysis	7.847	Rushed ’20-naar 1e controle’→’40-naar administratie’
Social analysis	388	Missing originator
Social analysis	7.727 + 14.508 + 381 + 10 + 267 + 2.402 + 124 + 21 + 4 + 422 + 5 + 315	SoD violations

Table 3: Case study 2 results

5.3 Summary

In this chapter we presented the results of the case studies. As discussed in Sections 5.1.7 and 5.2.7 there were a number of subsets identified that showed indicators of unusual and possibly even fraudulent behavior. The majority of these indicators are not enough to prove or disprove actual fraud; they are however reason enough for an auditor to perform further analysis as to what occurred. Similar to indicators of fraud found during regular data analysis, some indicators only make the existence of actual fraud more likely, while others can be considered strong indicators of fraud on their own.

6. A First Step Towards Operational Principles

From the results of the different analyses from the case studies, activities and analyses can be synthesized that provided valuable insights when detection unusual or fraudulent behavior. This chapter provides the most important and notable aspects that were encountered during the case studies as a first attempt to provide guidelines for operationalizing process mining for fraud detection in practice. Furthermore, when applicable, these respective insights can be compared to findings of regular data analysis techniques. Similar to Section 3.3, there are again 1+5+1 aspects that are applicable.

6.1 Log creation

During the creation of the event log the following aspects have to be taken into account:

1. Understanding the process
    - Understanding the activities
    - Activity aggregation and grouping
2. Understanding the data
    - Data storage (data formats)
    - Data extraction (cutoff)
    - Event timings
3. Log perspective

1: First and foremost there must be a clear understanding of the analyzed process, of which activities it consists, and especially how the system records the process and its activities. While this might seem obvious, knowing which activities are of interest and which are not can ease the analyses significantly. Some activities e.g. can consist of multiple events; this leads to complex process models, because it increases the number of activities, and furthermore non-atomic events can intertwine and increase the number of paths even though this does not add significant information. The SAP ERP-system for example sometimes (depending on implementation) records a price change simultaneously in multiple tables, which each table-change resulting in an event. Similarly, some activities can be aggregated into groups, if it can be argued that they do not add information. Consider for example the first case study, where multiple activities were grouped. From the perspective of fraud detection it is important to acknowledge this as well. Some activities can be indicators of fraud by themselves (e.g. price changes), while others can sometimes be considered not important (e.g. description changes).

2: There must be a clear understanding on the data; how it is stored, its relational model and how it was extracted. This is important in order to prevent issues with the cutoff, as described in Section 4.2. Considering fraud detection, absence of certain activities can sometimes be explained by the cutoff, but if not it may be considered indicators of fraudulent behavior.

The timings of the events are an important aspect of the log. As seen in the first case study, when events happen simultaneously, this can lead to problems in the resulting event log. This can lead to changes in the order of activities, which is also important considering the fraud perspective.

3: The level on which the cases are constructed in the event log is the last important note on log creation. In the first case study we choose to create the log on Purchase Order Line (POL) level rather than Purchase Order (PO) level. While the choice itself depends on the specific goal of the analyses and the preferences of the analysts, the consequences must be well understood. Results can possible change significantly when taking a different perspective.

6.2 Five Analysis Aspects

Log analysis

During log analysis the following aspects have to be acknowledged:

1. Activity analysis
    - Activity frequency
2. Endpoint analysis
    - Cutoff issues

1: During log analysis the frequencies of the different activities can provide insights into which activities are more common than others. Cases that contain activities that are unusual or suspect in itself can provide insights into cases that show interesting behavior. This can also be done using regular data analysis. Note that, as mentioned before, low frequency in itself does not necessarily indicate fraudulent behavior, as mentioned in Sections 2.1.3 and 3.1.

2: The endpoint analysis obtained valuable insights, by providing subsets and removing incomplete cases. This can as well be done using regular data analysis, but is much easier with process mining (assuming the event log is readily available). Endpoint analysis shows which cases are likely to be incorrectly recorded either due to cutoff issues or due to actual incorrect behavior. Essentially, log analysis is used to clean the log of faulty cases that otherwise would possibly distort the results of other analyses, and it provides further grounds for the creation of special subsets (e.g. based on a period of time or the occurrence of a certain activity) to be analyzed individually.

Process analysis

During process analysis the following aspects have to be taken into account:

1. Process flow analysis
    - Pattern variant analysis
    - General model

Process analysis provides similar insights as log analysis, but rather on trace level rather than event level. The analysis shows which sequences of activities and patterns are more common than others and like this it can provide reasons to analyze a specific subset of the event log, as shown in the case studies in Chapter 5. By taking the most common patterns, a general model of the process can be created, however the non-compulsory nature of certain activities in the first case study caused issues in creating a general model. The most important aspect of process analysis is to analyze whether or not, and with which frequency, some activities precede or follow other respective activities. From a fraud perspective the occurrence in itself, but especially the order of occurring activities is important.

Conformance analysis

Conformance analysis should in theory provide straightforward means to filter out which cases are not compliant with a certain general process model. The benefit is that it can provide these insights for the entire log, rather than having to check each possible activity-to-activity relation and see if this happens. Especially with a lot of different activities, manually analyzing each of these relations would be infeasible. However, it has become clear that the performance with respect to conformance checking of the current tools is poor and we must therefore question its value. Furthermore, from a fraud perspective we can question the value of knowing all process deviations; it is likely that only violations of certain activity-to-activity relations provide valuable insights for fraud detection. These reservations combined with the fact that currently additional attributes cannot be incorporated into the analysis (e.g. PO(L) value) and the fact that the construction of the general model used for the analysis is very time-consuming, leads us to argue that at the moment, there is too little added value of applying conformance when process mining for fraud detection.

Performance analysis

During performance analysis the following aspects have to be taken into account:

1. Case throughput times analysis
    - Rushed cases subset identification
    - Delayed cases subset identification
2. Attribute over-involvement subset analysis

1: Even though performance analysis is in theory more aimed towards performance management and improvement in general than fraud detection, the results of the analyses in the case studies showed that it can aid fraud detection. By taking out specific combinations of activities and analyzing the durations, we identified subsets of cases that were unexpectedly quick or slow. This can indicate cases that are intentionally rushed or lagged through the system to divert human attention.

2: Within these subsets we were able to identify originators and suppliers that were over-involved in certain subsets. These provide grounds for further analysis to determine whether these subsets are susceptible for fraudulent behavior. In the second case study we showed this way that rushed cases were violating SoD constraints in 97% of cases instead of the 'normal' 10% violations.

Social analysis

During social analysis the following aspects have to be taken into account:

1. Tool performance
2. SoD violations
    - Role-Activity-Matrix analysis
    - LTL-Checker or Disco-filter analysis

1: According to both literature and case studies examined in Section 3.2 there are interesting applications of social analysis such as social networks, handover-of-work networks and Dotted Charts. Our case studies the practical performance issues limit its use to testing SoD constraints. The case

studies found in literature had only tens rather than hundreds of actors, which might explain the difference in performance / computability.

2: By way of the Role-Activity-Matrix in ProM this matrix can easily show if a person violates SoD constraints by performing restricted activities. As mentioned before however, when a person is involved in conflicting activities we still have to check if the activities were performed in the same case. This is easily done using Disco, but is also partially possible using regular data analysis. The benefit of using data analysis is that it is easier to use attribute values, whereas process mining can currently only use these attributes to filter on; e.g. the total value of POs created by an originator can be easily computed with data analysis techniques, but this is a cumbersome manual task using process mining. The Role-Activity-Matrix can also provide insights into whether or not people are sticking to their roles by looking at the activity frequency, i.e. we identified numerous examples where some originators performed certain activities hundreds or even thousands of times, while performing some activities only a very few times.

6.3 General remarks

As mentioned in Section 3.3, iterating over the subsets identified during any step of the five analyses must be always acknowledged. On the one hand, combining different analysis techniques over the same subset can increase the value of an indicator when required, as some indicators are more conclusive than others. On the other hand, changing to a different view can also provide conclusive insights into fraudulent or legitimate behavior of cases.

This is related to communicating the results between analyst and business user, as mentioned by Bozkaya et al. (2009) mentioned in Section 3.2. Data analysis, and its results, can be considered more conclusive in some regards. If an analysis is done to e.g. find out which PO values occur most often, this is a clear fact. Process mining analyses (and their results) require an understanding of the process, and its concerned aspects. Because of the flattening of the data, some aspects and relations can be lost and it is therefore even more important for findings to be discussed with the process or data owners.

While cutoff has been treated in detail already, cutoff and completeness deserve more emphasis. The results of the process mining analyses can be significantly affected by incorporation of unfinished and / or incorrectly recorded cases. Any analysis based on averages e.g. provides distorted results when cutoff issues are not solved. On the other hand completeness is one of the most important issues for fraud detection; an investigator has to be sure that no potential fraudulent cases were missed. Removing a number of cases from the event log because they had e.g. incorrect start or end activities, as was done in this case study as well as the first one, thus requires solid substantiation. As argued this is correct from a process mining point of view. We explicitly mentioned these cases for further analysis, however we currently do not know if these filtered out cases contain any other unusual behavior, other than the reasons for which they were removed from the log. If we would have been able to acquire additional data to fix what appear to be cutoff issues, we can obtain a lot more certainty about these cases and greatly increase the completeness of our results.

7. Conclusions

This chapter presents a summary of the topics presented in this thesis, as well as a discussion of the most important aspects presented. This thesis concludes with recommendations for further research.

7.1 Summary

This thesis aimed to provide insights into how the concept of process mining could be operationalized for fraud detection. In the first part of Chapter 2 a literature study is presented on process mining and its techniques. The three aspects of process mining (process discovery, conformance and enhancement) are discussed as well as the currently available tools and techniques used in these respective aspects. In the second part of Chapter 2 a literature study and expert interviews are used to gain insight into the concept of fraud, and how fraud detection is conducted in practice and which techniques it encompasses.

In Chapter 3 six case studies were analyzed to discover which aspects and techniques of process mining are previously used for fraud detection. Based on the results of the analysis of these case studies, an initial structure for the case studies presented in Chapter 5 was created. Besides creation of the event log, this initial structure consists of five types of process mining analysis. When these analyses identify subsets of the event log with cases that contain indicators of unusual behavior, the notion of iterating over these subsets using different analysis techniques is acknowledged. This is done in order to obtain more conclusive insights into the data and the indicators found. These 1+5+1 steps were subsequently applied using the tools discussed in Chapter 4.

In Chapter 5 the results of two case studies are presented. The 1+5+1 methodology was able to identify several subsets of cases in the event log that revealed unusual behavior. While actual fraudulent behavior could not be proven, the analyses provided insights that were not previously known or detected by other analysis methods. Not all analyses could be applied; the available process mining tools are in their current state not able to provide results. Performance becomes very poor as the size of the event log and its attributes increases.

The application of the analyses was synthesized in Chapter 6, to present the first step towards operationalizing principles for using process mining for fraud detection. These principles, the 1+5+1 concept, are discussed to indicate how they can be applied and which aspects can be currently used in practice. Log creation affects the eventual analysis results significantly, as the design choices and focus points chosen during creation determine the view on the process. Log analysis and process analysis can identify cases that are unusual with respect to activities and process flow. Performance analysis can be used to initially identify rushed or delayed cases that can be subsequently analyzed. Social analysis uses the Role-Activity-Matrix to analyzed how often originators are involved in a respective activity. Other applications of social analysis tools currently suffer too much from performance issues. The tools used for conformance analysis are also not able to provide results.

7.2 Discussion

There are four aspects of this thesis that require acknowledgement. First, the operationalization aspects can be divided into four conditions for application. The first condition is the availability of a well-defined

and reliable event log. The data quality as well as the design is a major influence on process mining, as both seen in the case studies. The second condition is tool-support; current tools can perform parts of the analyses presented. No single tool is currently able to perform all analyses, at the moment multiple tools must be combined. The third condition is the availability of operationalization principles as to how to apply the different techniques to obtain indicators of fraudulent behavior. The 1+5+1 concept presented in this thesis is a first step towards these principles. The fourth condition is that the scope of the domain under investigation is fit for process mining. Even though this is closely related to the design of the event log, process miners must consider whether or not the domain is as suitable for process mining as the procurement process in our case studies.

The second aspect is the presented 1+5+1 concept. This concept was developed in this thesis from the examination of previously conducted case studies. The techniques mentioned in these case studies were subsequently evaluated in our own case studies, with varying results. Considering the five different analysis types, some can obtain valuable results with for fraud detection. The current state of the tools however prevents presenting decisive conclusions, but this does not imply that the 1+5+1 concept itself is faulty. In Chapter 2 a sound theoretical basis was presented as to how the different analyses of the 1+5+1 concept can aid fraud detection. Although other authors obtained similar or better results by altering their event log, we predict that improvement of the tools results in an eventual validation of the concept.

The third aspect is related to the last +1 step of the 1+5+1 concept, iterate and refocus. This step is necessary to get a better insight into the value of specific indicators, from both a process mining as well as fraud detection point of view. As process mining takes a certain perspective on the data, a different perspective can provide different insights, with possibly different results. Combining different analyses and perspectives provides more conclusive indicators. With respect to fraud detection, iterating and refocusing is sometimes required because of the inherent value of certain indicators. Some indicators are close to conclusive evidence, or certain violations of business rules or SoD constraints. Other indicators are mere suggestions that unusual behavior is occurring, requiring additional insights by changing either the perspective, the analysis type, or both. Regardless of the strength of the indicator, process miners and / or auditors may indicate the strength and proof of fraud, but have to leave jurisdictional ruling to legal authorities.

The fourth aspect is the issue of completeness. As just mentioned, authors in related case studies sometimes used random samples for computability reasons. From a fraud perspective, this is however not an acceptable possibility. When examining for indicators of fraudulent behavior, 100% completeness must be a guarantee so that no fraudulent behavior slips through. It might appear that in the case studies presented in this thesis this was sometimes lacking; in a lot of situations the choice was made to remove certain subsets from the event log to improve subsequent analyses. When it is stated that these subsets should be examined further it must be emphasized that this is a necessity for completeness. The number of subsets as well as the number of cases in these subsets can however be reduced when data extraction takes cutoff issues into account.

As an overall conclusion, the research in this thesis results in the following answer to the main research question:

How can process mining be utilized in fraud detection and what are the benefits of using process mining for fraud detection?

The 1+5+1 concept presented in this thesis has shown that process mining can be used for fraud detection. By focusing on the process rather than the data, and the different aspects of that process, new insights are obtained that were previously unknown. Additional benefits arise with respect to ease of use, but the most important condition is the availability or creation of a suitable event log, which increases the required effort to apply process mining. The five types of analysis demonstrated in the case studies each result in indicators for non-compliant and possibly fraudulent behavior. Compared to regular data analysis techniques, process mining techniques are sometimes able to identify cases with different indicators of fraudulent behavior. Just as with current fraud detection techniques this sometimes results in subsets of cases that should be subsequently analyzed, rather than conclusive signs of fraudulent behavior. By changing the view on the data and performing subsequent analyses on these subsets, more clear indicators can be found. This can either be done by process mining or data analysis techniques, or by manual inspection by an auditor. In its current state, process mining is limited by the performance of the available tools. If however new tools can be developed capable of handling real-life event logs, process mining can be used as an addition to current data analysis techniques for fraud detection.

7.3 Recommendations

The extraction of the data with respect to cutoff issues and the creation of the event log has been discussed plenty in the previous sections. The extraction itself however deserves some attention as well. In both case studies data was provided by a party other than the process miner. In the first case study this led to errors in the extracted data. Even though most of these errors were minor and could be repaired, the POL table was extracted again. Due to the differences in dates between the two extractions, the second type of cutoff error was introduced in the event log. In order to prevent this, it is recommended to renew all data rather than just a single table, but this was unfortunately not possible due to time constraints.

The problems with tool performance are mentioned multiple times throughout this thesis. Not being able to obtain results with a number of tools undermines the utility of process mining in general, despite the theoretical possibilities of its application as described in Chapter 2. With respect to the tools used, it would be very beneficial if improved tools could be developed. On the one hand it might be possible to solve problems with performance (especially in ProM), while on the other hand the feature set can be greatly improved and tailored more towards fraud detection purposes.

Whereas process mining evolved from process discovery to conformance analysis to operational support, we feel this should be applied to process mining for fraud detection as well. The operational support aspect of process mining was considered out of scope in this thesis. However, given the increased interest in continuous monitoring in both BPM and fraud detection, online operational

support can possibly provide a lot of added value. While there are currently some attempts made in this area, this subject is still a long way off from being used in practice. Furthermore, the interest in artificial intelligence techniques for application in the fraud detection domain mentioned in Chapter 3 are worth mentioning. Whereas regular data analysis techniques are being complemented by data mining techniques, process mining can possibly be also combined with artificial intelligence, to provide insights that were previously hard to detect.

Bibliography

ACFE, 2012. *ACFE's 2012 Report to the Nations on Occupational Fraud and Abuse*. Association of Certified Fraud Examiners.

Agrawal, R., Gunopulos, D. & Leymann, F., 1998. Mining Process Models from Workflow Logs. In *6th International Conference on Extending Database Technology: Advances in Database Technology*. London, 1998. Springer-Verlag.

Albrecht, W.S., Albrecht, C. & Albrecht, C.C., 2008a. Current Trends in Fraud and its Detection. *Information Security Journal: A Global Perspective*, 17(1), pp.2-12.

Albrecht, W., Albrecht, C. & Albrecht, C., 2008b. Current Trends in Fraud and its Detection. *Information Security Journal: A Global Perspective*, 17(1), pp.2-12.

Alles, M., Jans, M. & Vasarhelyi, M., 2011. Process Mining: a New Research Methodology for AIS. In *CAAA Annual Conference*., 2011.

Becker, J., Rosemann, M. & Schütte, R., 1997. Business-to-business process integration: functions and methods. In Galliers, R., ed. *Proceedings of the 5th European Conference on Information Systems (ECIS '97)*., 1997.

Bezerra, F. & Wainer, J., 2007. Towards, Detecing Fraudulent Executions in Business Process Aware Systems. In *Workshop on Workflows and Process Management*. Timisoara, 2007.

Bezerra, F. & Wainer, J., 2008a. Anomaly Detection Algorithms in Business Process Logs. In *International Conference on Enterprise Information Systems - ICEIS*. Barcelona.

Bezerra, F. & Wainer, J., 2008b. Fraud Detection in Process Aware Systems. In *Companion Proceedings of the XIV Brazilian Symposium on Multimedia and the Web*.

Bologna, G. & Lindquist, R., 1995. *Fraud Auditing and Forensic Accounting*. John Wiley and Sons.

Bose, J.C. & van der Aalst, W.M.P., 2009a. Abstractions in Process Mining: A Taxonomy of Patterns. In *Proceedings of the Seventh International Conference on Business Process Management*. Berlin Springer-Verlag.

Bose, J.C. & van der Aalst, W.M.P., 2009b. Context Aware Trace Clustering: Towards Improving Process Mining Results. In Liu, H. & Obradovic, Z., eds. *Proceedings of the SIAM International Conference on Data Mining*.

Bose, J.C. & van der Aalst, W.M.P., 2011. Discovering Hierarchical Process Models using ProM. In Nurcan, S., ed. *Proceedings of the CAiSE Forum 2011*., 2011.

Bozkaya, M., Gabriels, J.M.A.M. & van der Werf, J.M.E.M., 2009. Process Diagnostics: A Method Based on Process Mining. In *Proceedings of International Conference on Information, Process, and Knowledge Management*., 2009.

Coderre, D., 2009. *Computer Aided Fraud Prevention & Detection*. John Wiley & Sons.

Cook, J. & Wolf, A., 1995. Automating Process Discovery through Event-Data Analysis. In *ICSE '95: Proceedings of the 17th international conference on Software engineering*. New York, 1995. ACM.

Cook, J. & Wolf, A., 1998. Discovering Modles of Software Processes from Event-Based Data. In *ACM Transactions on Software Engineering and Methodology*. New York, 1998. ACM.

Datta, A., 1998. Automating the Discovery of As-Is Business Process Models: Probalistic and Algorithmic Approaches. *Information Systems Research*, 9(3), pp.275-301.

Davia, H.R., Coggins, P., Wideman, J. & Kastantin, J., 2000. *Accountant's Guide to Fraud Detection and Control*. Chichester, UK: John Wiley and Sons.

de Medeiros, A.K.A., van der Aalst, W.M.P. & Weijters, A.J.M.M., 2003. Workflow Mining: Current Status and Future Directions. *Lecture Notes in Computer Science*, 2888, pp.389-406.

El Kharbili, M., de Medeiros, A.K.A., Stein, S. & van der Aalst, W.M.P., 2008. Business Process Compliance Checking: Current State and Future Challenges. *Lecture Notes in Informatics*, 114, pp.107-13.

Greco, G., Guzzo, A., Manco, G. & Saccà, D., 2005. Mining and Reasoning on Workflows. *IEEE Transactions on Knowledge and Data Engineering*, 17(4), pp.519-34.

Green, P. & Rosemann, M., 2000. Integrated process modeling: an ontological evaluation. In *Information Systems - The 11th international conference on advanced information systems engineering*., 2000.

Günther, C.W., Rozinat, A. & van der Aalst, W.M.P., 2009. Activity Mining by Global Trace Segmentation. In Rinderle-Ma, S., Sadiq, S. & Leymann, F., eds. *Proceedings of the Fifth Workshop on Business Process Intelligence*. Berlin, 2009. Springer-Verlag.

Günther, C.W. & van der Aalst, W.M.P., 2007. Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics. In Alonso, G., Dadam, P. & Rosemann, M., eds. *International Conference on Business Process Management*. Berlin, 2007. Springer-Verlag.

Hoogs, B., Kiehl, T., Lacomb, C. & Senturk, D., 2007. A genetic algorithm approach to detecting temporal patterns indicative of financial statement fraud. *Intelligen Systems in Accounting, Finance and Management*, 15, pp.41-56.

Jans, M., Alles, M. & Vasarhelyi, M., 2010. Process mining of event logs in auditing: Opportunities and challenges. In *International Symposium on Accounting Information*. Orlando, 2010.

Jans, M., Depaire, B. & Vanhoof, K., 2011. Does Process Mining Add to Internal Auditing? An Experience Report. *Lecture Notes in Business Information Processing*, 81(1), pp.31-45.

Jans, M., Lybaert, N. & Vanhoof, K., 2007. Data Mining for Fraud Detection: Toward an Improvement on Internal Control Systems? In *European Accounting Association - Annual Congress, 30*. Lisbon, 2007.

Jans, M., Lybaert, N. & Vanhoof, K., 2009. A Framework for Internal Fraud Risk Reduction at IT Integrating Business Processes: The IFR² Framework. *The International Journal of Digital Accounting Research*, 9, pp.1-29.

Jans, M., Lybaert, N. & Vanhoof, K., 2010. Internal Fraud Risk Reduction - Results of a Data Mining Case Study. In *ICEIS*., 2010.

Jans, M., Lybaert, N., Vanhoof, K. & van der Werf, J.M.E.M., 2008. Business Process Mining for Internal Fraud Risk Reduction: Results of a Case Study. In *International Research Symposium on Accounting Information Systems*., 2008.

Jans, M., van der Werf, J.M., Lybaert, N. & Vanhoof, K., 2011. A business process mining application for internal transaction fraud mitigation. *Expert Systems with Applications: An International Journal* , 38(10), pp.13351-59.

Kirkos, E., Spathis, C. & Manolopolous, Y., 2007. Data Mining Techniques for the Detection of Fraudulent Financial Statements. *Expert Systems with Applications*, 32, pp.995-1003.

La Rosa, M. et al., 2011. Managing Process Model Complexity via Abstract Syntax Modifications. *IEEE Transactions on Industrial Informatics*, 7(4), pp.614-29.

Lenard, M.J. & Alam, P., 2009. An Historical Perspective on Fraud Detection: From Bankruptcy Models to Most Effective Indicators of Fraud in Recent Incidents. *Journal of Forensic & Investigative Accounting*, 1(1), pp.1-27.

Maggi, F.M., Montali, M. & van der Aalst, W.M.P., 2012. An operational decision support framework for monitoring business constraints. In *Proceedings of the 15th international conference on Fundamental Approaches to Software Engineering*. Berlin, 2012. Springer-Verlag.

Maggi, F.M., Montali, M., Westergaard, M. & van der Aalst, W.M.P., 2011. Monitoring business constraints with linear temporal logic: an approach based on colored automata. In *Proceedings of the 9th international conference on Business process management*. Berlin, 2011. Springer-Verlag.

Medeiros, D., A.K.A., Weijters, A.J.M.M. & van der Aalst, W.M.P., 2007. Genetic Process Mining: An Experimental Evaluation. *Data Mining and Knowledge Discovery*, 14(2), pp.245-304.

Montali, M. et al., 2011. *Monitoring Business Constraints with the Event Calculus*. Technical Report. Bologna: University of Bologna.

Montali, M. et al., 2010. Declarative specification and verification of service choreographiess. *ACM Transactions on the Web*, 4(1).

Munoz-Gama, J. & Carmona, J., 2010. A Fresh Look at Precision in Process Conformance. In *BPM 2010.*, 2010. Springer.

Nederlands Beroepsorganisatie van Accountants, 2012. *Handleiding Regelgeving Accountancy*. [Online] Available at: http://www.nba.nl/hraweb/hra1a/201201/html/45763.htm [Accessed 29 June 2012].

Oxford Dictionaries, 2010a. *Definition for Fraud*. [Online] Available at: http://oxforddictionaries.com/definition/fraud [Accessed 25 June 2012].

Oxford Dictionaries, 2010b. *Definition for model*. [Online] Available at: http://oxforddictionaries.com/definition/model?q=model [Accessed 12 June 2012].

Podgor, E.S., 1999. Criminal Fraud. *American University Law Review*, 48(4), pp.729-69.

Ramezani, E., Fahland, D. & van der Aalst, W.M.P., 2012. Where did i misbehave? diagnostic information in compliance checking. *Lecture Notes in Computer Science*.

Rozinat, A., 2010. *ProM Tips — Which Mining Algorithm Should You Use? — Flux Capacitor*. [Online] Available at: http://fluxicon.com/blog/2010/10/prom-tips-mining-algorithm/ [Accessed 26 July 2012].

Rozinat, A., de Jong, I.S.M., Günther, C.W. & van der Aalst, W.M.P., 2007. Process Mining of Test Processes: A Case Study. *BETA Working Paper*, 220.

Rozinat, A. & van der Aalst, W.M.P., 2005. Conformance Testing: Measuring the Alignment Between Event Logs and Process Models. *BETA Working Paper Series*, 144.

Rozinat, A. & van der Aalst, W.M.P., 2006a. Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models. *Lecture Notes in Computer*, 3812, pp.163-76.

Rozinat, A. & van der Aalst, W.M.P., 2006b. Decision Mining in Business Processes. *BETA Working Paper Series*, 164.

Rozinat, A. & van der Aalst, W.M.P., 2008. Conformance Checking of Processes Based on Monitoring Real Behavior. *Information Systems*, 33(1), pp.64-95.

Song, M. & van der Aalst, W.M.P., 2007. Supporting Proces Mining by Showing Events at a Glance. In Chari, K. & Kumar, A., eds. *Proceedings of the 17th Annual Workshop on Information Technologies and Systems*. Montreal, 2007.

Song, M. & van der Aalst, W.M.P., 2008. Towards Comprehensive Support for Organizational Mining. *Decision Support Systems*, 46(1), pp.300-17.

Tan, P.N., Steinbach, M. & Kumar, V., 2006. *Intorduction to Data Mining*. Addison-Wesley.

van der Aalst, W.M.P., 2005. Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing. *Requirements Engineering Journal*, 10(3), pp.198-211.

van der Aalst, W.M.P., 2009. Using Process Mining to Generate Acuurate and Interactive Business Process Maps. *Business Information Systems Workshops*, 37, pp.1-14.

van der Aalst, W.M.P., 2010. Business Process Simulation Revisited. *Enterprise and Organizational Modeling and Simulation*, 63, pp.1-14.

van der Aalst, W.M.P., 2011. *Process Mining*. Berlin Heidelberg: Springer-Verlag.

van der Aalst, W.M.P., de Beer, H.T. & van Dongen, B.F., 2005a. Process mining and verification of properties: an approach based on temporal logic. In *Proceedings of the 2005 Confederated international conference on On the Move to Meaningful Internet Systems*. Berlin Springer-Verlag.

van der Aalst, W.M.P., de Beer, H.T. & van Dongen, B.F., 2005b. Process Mining and Verification of Properties: An Approach based on Temporal Logic. *Lecture Notes in Computer Science*, 3760, pp.130-47.

van der Aalst, W.M.P. & de Medeiros, A.K.A., 2005. Process mining and security: Detecting anomalous process executions and checking process conformance. *Electronic Notes in Theoretical Computer Science*, 121, pp.3-21.

van der Aalst, W.M.P., Pesic, M. & Song, M., 2010. Beyond Process Mining: From the Past to Present and Future. In Pernici, B., ed. *Advanced Information Systems Engineering, Proceedings of the 22nd International Conference on Advanced Information Systems Engineering*. Berlin, 2010. Springer-Verlag.

van der Aalst, W.M.P., Reijers, H.A. & Song, M., 2005. Discovering Social Networks from Event Logs. *Computer Supported Cooperative work*, 14(6), pp.549-93.

van der Aalst, W.M.P. & van Dongen, B.F., 2002. Discovering Workflow Performance Models from Timed Logs. In Han, Y., Tai, S. & Wikar, D., eds. *International Conference on Engineering and Deployment of Cooperative Information Systems*. Berlin, 2002. Springer-Verlag.

van der Aalst, W.M.P. et al., 2003. Workflow Mining: A Survey of Issues and Approaches.. *Data and Knowledge Engineering*, 47(2), pp.237-67.

van der Aalst, W.M.P. et al., 2011. Conceptual Model for Online Auditing. *Decision Support Systems*, 50(3), pp.636-47.

van der Aalst, W.M.P., van Hee, K.M., van der Werf, J.M. & Verdonk, M., 2010. Auditing 2.0: Using Process Mining to Support Tomorrow's Auditor. *IEEE Computer*, 43(3), pp.90-93.

van der Aalst, W.M.P. & Weijters, A.J.M.M., 2005. Process Mining. In M. Dumas, W.M.P. van der Aalst & A.H.M. ter Hofstede, eds. *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley & Sons. Ch. 12. pp.235-55.

van der Aalst, W.M.P., Weijters, A.J.M.M. & Maruster, L., 2002. Workflow Mining: Which Processes can be Rediscovered? *BETA Working Paper Series*, 74.

van der Aalst, W.M.P., Weijters, A.J.M.M. & Maruster, L., 2004. Workflow Mining: Discovering Process Models from Event Logs. *Transactions on Knowledge and Data Engineering*, 16(9), pp.1128-42.

van der Spoel, S., 2012. *Outcome and Variable Prediction for Discrete Processes*. Enschede: Universiteit Twente.

van Dongen, B.F., De Medeiros, A.K. & Wen, L., 2009. Process mining: overview and outlook of Petri net discovery algorithms. In K. Jensen & W.M.P. van der Aalst, eds. *Transactions on Petri Nets and Other Models of Concurrency II*. Berlin: Springer. pp.225-42.

van Dongen, B.F. & van der Aalst, W.M.P., 2005. A Meta Model for Process Mining Data. In Casto, J. & Teniente, E., eds. *Proceedings of the CAiSE'05 Workshops (EMOI-INTEROP Workshop)*. Porto, 2005. FEUP.

Vonk, H., 2012. *Personal correspondence*.

Wasserman, S. & Faust, K., 1994. *Social Network Analysis: Methods and Applications*. Cambridge: Cambridge University Press.

Weijter, A.J.M.M. & Ribeiro, J.T.S., 2010. Flexible Heuristic Miner (FHM). *Beta Working Paper*, 334.

Weijters, A.J.M.M.l. & Ribeiro, J.T.S., 2011. Flexible heuristics miner (FHM). *BETA working paper*, 334.

Weijters, T. & van der Aalst, W.M.P., 2001a. Process Mining: Discovering Workflow Models from Event-Based Data. In *Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence*.

Weijters, T. & van der Aalst, T.M.P., 2001b. Rediscovering Workflow Models from Event-Based Data. In *Proceedings of the Eleventh Belgian-Dutch Conference on Machine Learning*. Antwerpen.

Wells, J., 2005. *Principles of Fraud Examination*. Chichester, UK: John Wiley and Sons.

Wen, L., van der Aalst, W.M.P., Wang, J. & Sun, J., 2007. Mining process models with non-free-choice constructs. *Data Mining and Knowledge Discovery*, 15(2), pp.145-80.

Yang, W. & Hwang, S., 2006. A process-mining framework for the detection of healthcare fraud and abuse. *Expert Systems with Applications*, 2006, pp.56-68.

Yue, D., Wu, X., Wang, Y. & Chu, C., 2007. A Review of Data Mining-Based Financial Fraud Detection Research. In *International Conference on Wireless Communications, Networking and Mobile Computing*. Shanghai, 2007.

Appendix A Formal Notations

A.1 Process Models

While there are lot of different notations for process modeling (e.g. Petri nets, BPEL, YAWL, etc.) the main approach is to model the process as a directed graph (Agrawal et al., 1998, p.469). Note that in this thesis, there will be no focus on the difference between the respective modeling languages. Expressed formally process modeling can be described as (Agrawal et al., 1998, pp.472-74):

Let $\mathcal{A}$ be the space of all possible activities. A process P consists of a set of activities $_{P,}$ where

$$A_P = \{A_1, \ldots, A_n \, \mathrm{A_i} \in \mathcal{A} \text{ for all i=1, ..., n }\}.$$

The process can be represented as a directed graph (G_P) where the nodes represent the activities (A_P) and the edges (E_P) represent the execution and transition of these activities:

$$G_P = (A_P, E_P).$$

The edges are traversed by execution of the activities, which can be notated by the output function 0_P, with

$$0_\mathrm{P} : A_P \rightarrow \mathcal{N}^k.$$

All activities are considered atomic being either executed fully or not, which can be notated by the Boolean function $f_{(u, v)}$, with

$$f_{(u, v)} : \mathcal{N}^k \rightarrow \{0, 1\} \text{ for all } (u,v) \in E_P.$$

A.2 Event Logs

Van der Aalst (van der Aalst, 2011, pp.100, 104) defines the concepts of case, process, activity, and attribute as following:

Let $\mathcal{E}$ be the space of all possible events and AN the set of attribute names;

$$\forall\ e \in \mathcal{E},\ n \in AN: \#_n(e) \text{ is the value of attribute } n \text{ for event } e,$$
$$\text{or } null \text{ if } e \text{ does not have such attribute.}$$

Typical examples of attributes are the timestamp of the event, the resource (e.g. the person executing the action), or the transaction type (start, complete, wait, abort). As for cases:
Let $\mathcal{C}$ be the space of all possible cases. Analogous to events,

$$\forall\ c \in \mathcal{C},\ n \in AN: \#_n(c) \text{ is the value of attribute } n \text{ for case } c,$$
$$\text{or } null \text{ if } c \text{ does not have such attribute.}$$

Furthermore, each case has the attribute trace

$$\#_{\mathrm{trace}}(c) \in \mathcal{E}^* = \hat{c}.$$

where $\mathcal{E}^*$ is the set of finite sequences over $\mathcal{E}$. Thus, trace $\hat{c}$ is a finite sequence of events σ, in which each event appears only once. Event log $L \subseteq \mathcal{C}$ is a set of traces:

$$L = \{\sigma_1, \ldots, \sigma_n\} \subseteq \mathcal{E}^*.$$

A.3 The α-algorithm

The α-algorithm is a *process discovery* algorithm which that maps an *event log* onto a *process model*. This is done by scanning the event log for particular patterns, i.e. casual dependencies, and for the α-algorithm the result is a Petri Net. These Log-Based Ordering Relations (van der Aalst et al., 2004, p.11; van der Aalst, 2011, p.130) are defined as follows:

Let W be a workflow[9] log over A (a set of activities, as in Section 2.1.1 Related Concepts), i.e., $W \in P(A*)$. Let a, b ∈ A. Now we can define the following relations between a and b

- $a >_W b$ $\Leftrightarrow \forall_\sigma \mid \sigma = t_1t_2t_3 \ldots t_{n-1}$ and $i \in \{1, \ldots, n-1\}$ such that $\sigma \in W$ and $t_i = a$ and $t_{i+1} = b$
  i.e. a is directly followed by b at some point in the event log
- $a \rightarrow_W b$ $\Leftrightarrow a >_W b$ and $b \not> _W a$
  i.e. a is directly followed by b, but b is never directly followed by a
- $a \#_W b$ $\Leftrightarrow a \not> _W b$ and $b \not> _W a$
  i.e. a is never directly followed by b, b is never directly followed by a
- $a \|_W b$ $\Leftrightarrow a >_W b$ and $b >_W a$
  i.e. a is directly followed by b, and b is directly followed by a

Consider as an example event log L_1 consisting of three traces: a→b→c→d, a→c→b→d, a→e→d

$$L_1 = \{(a,b,c,d), (a,c,b,d), (a,e,d)\}$$

Using the relations on the example log yields:

- $>_{L1}$ = { (a,b), (a,c), (a,e), (b,c), (c,b), (b,d), (c,d), (e,d) }
- $\rightarrow_{L1}$ = { (a,b), (a,c), (a,e), (b,d), (c,d), (e,d) }
- $\#_{L1}$ = { (a,a), (a,d), (b,b), (b,e), (c,c), (c,e), (d,a), (d,d), (e,b), (e,c), (e,e) }
- $\|_{L1}$ = { (b,c), (c,b) }

Another way to visualize these relations is by putting them in a table, called a footprint:

	a	b	c	d	e
a	$\#_{L_1}$	$\rightarrow_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$
b	$\leftarrow_{L_1}$	$\#_{L_1}$	$\|_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$
c	$\leftarrow_{L_1}$	$\|_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$
d	$\#_{L_1}$	$\leftarrow_{L_1}$	$\leftarrow_{L_1}$	$\#_{L_1}$	$\leftarrow_{L_1}$
e	$\leftarrow_{L_1}$	$\#_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$

Figure 19: The Footprint of example log L_1. Taken from (van der Aalst, 2011, p.130)

From the footprint, the causal relations can be discovered. E.g. if $a \rightarrow_L b$ (b follows a) and $a \rightarrow_L c$ (c follows a) but $b \#_{Lc}$ (b and c never follow each other) the log contains a XOR-split. If $a \rightarrow_L c$ (c follows a) and $b \rightarrow_L c$

[9] A workflow is a subclass of Petri nets that have dedicated start- and end-nodes and where all nodes are on a path from start to start to end.

(c follows b) but $a\#_{Lb}$ (a and b never follow each other) the log contains a XOR-join. Similarly AND-splits ($a\rightarrow_L b$, $a\rightarrow_L c$, $b \|_L c$) or AND-joins ($a\rightarrow_L c$, $b\rightarrow_L c$, $a \|_L b$) can be derived. Figure 20 shows a graphical representation of the ordering relations and the subsequent logical patterns.

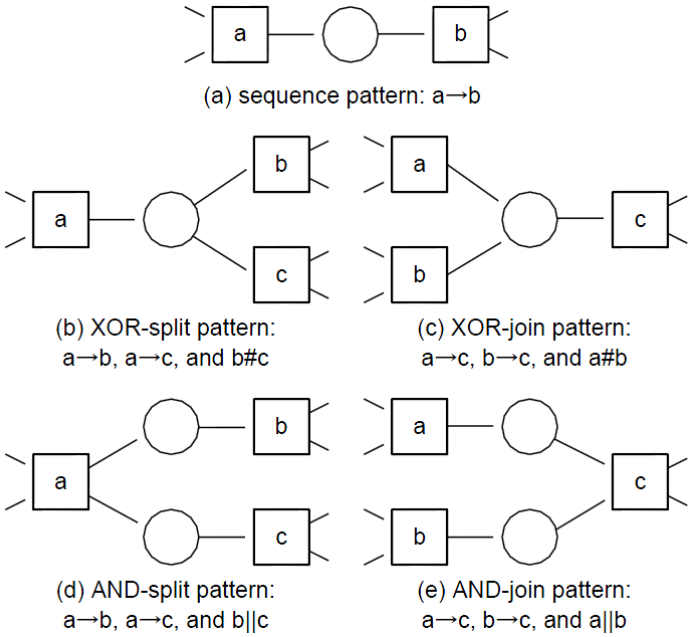


Figure 20: Footprint patterns versus log-based ordering relations. Taken from (van der Aalst, 2011, p.131)

The steps of the α-algorithm are defined as follows: let L be an event log over $T \subseteq \mathcal{A}$; $\alpha(L)$ provides:

1 $T_L = \{ t \in T \mid \exists\, \sigma \in L : t \in \sigma \}$,
i.e. the set of activities appearing in the log. For L_1 this results in $T_L = \{a, b, c, d, e\}$.

2 $T_I = \{ t \in T \mid \exists\, \sigma \in L : t = \mathit{first}(\sigma) \}$,
i.e. the set of activities that start some trace. For L_1 this results in $T_I = \{a\}$.

3 $T_O = \{ t \in T \mid \exists\, \sigma \in L : t = \mathit{last}(\sigma) \}$,
i.e. the set of activities that end some trace. For L_1 this results in $T_O = \{d\}$.

4 $X_L = \{ (A, B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall\, a \in A, \forall\, b \in B : a \rightarrow_L b \wedge \forall\, a_1, a_2 \in A : a_1 \#_L a_2 \wedge \forall\, b_1, b_2 \in B\ b_1 \#_L b_2 \}$,
i.e. the causal relations in the log. For L_1 this results in $X_L = \{ (\{a\},\{b\}), (\{a\},\{c\}), (\{a\},\{e\}), (\{b\},\{d\}), (\{c\},\{d\}), (\{e\},\{d\}), (\{a\},\{b,e\}), (\{a\},\{c,e\}), (\{b,e\},\{d\}), (\{c,e\},\{d\}) \}$.

5 $Y_L = \{ (A,B) \in X_L \mid \forall\, (A',B') \in X_L\ A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
i.e. only the minimal causal relations in the log (which cannot be deduced from other relations), removing nonmaximal pairs. For L_1 this results in $Y_L = \{ (\{a\},\{b,e\}), (\{a\},\{c,e\}), (\{b,e\},\{d\}), (\{c,e\},\{d\}) \}$.

6 $P_L = \{ p_{(A,B)} \mid (A,B) \in Y_L \} \cup \{i_L, o_L\}$
i.e. all places in the log, i.e. all nodes between two actions. For L_1 this results in $P_L = \{ i_L, o_L, p_{(\{a\},\{b,e\})}, p_{(\{a\},\{c,w\})}, p_{(\{b,e\},\{d\})}, p_{(\{c,e\},\{d\})} \}$ (including start and end nodes).

7 $F_L = \{ (a,p_{(A,B)}) \mid (A,B) \in Y_L \wedge a \in A \} \cup \{ (p_{(A,B)},b) \mid (A,B) \in Y_L \wedge b \in B \} \cup \{ (i_L, t) \mid t \in T_I \} \cup \{ (t,o_L) \mid t \in T_O \}$,

i.e. the arcs connecting the places to the actions. For L_1 this results in F_L = { (i_L,a), (a, $p_{(\{a\},\{b,e\})}$), ($p_{(\{a\},\{b,e\})}$, b) . . . , (d, o_L) }.

8 $\alpha(L) = (P_L, T_L, F_L)$,
i.e. the resulting Workflow Net, with places P_L, transitions T_L, and arcs F_L.

The result is shown in Figure 21:

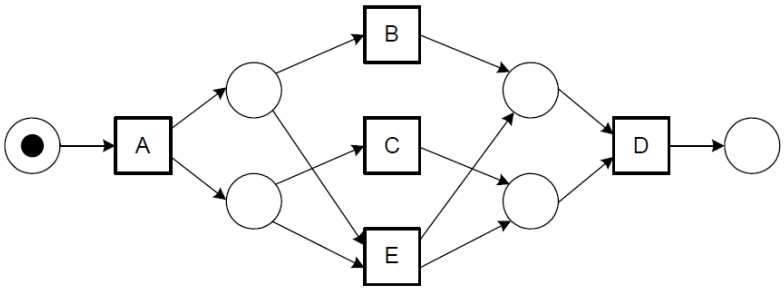


Figure 21: Example Workflow Net. Taken from (van der Aalst et al., 2004, p.22)