Business Process Mining for Internal Fraud Risk Reduction: Results of a Case Study

Mieke Jans, Nadine Lybaert, Koen Vanhoof

Abstract

Corporate fraud these days represents a huge cost to our economy. Academic literature merely concentrates on the fight against external fraud, while internal fraud also represents a major problem. In this paper we discuss the use of process mining to reduce the risk of internal fraud. Process mining diagnoses processes by mining event logs. This way we can expose opportunities to commit fraud in the process design. We present a framework as a complement to the internal control framework of the COSO and apply this framework in a case company.

This is a working paper, please do not quote.

1 Introduction

Everybody can recall some kind of fraud that has been all over the news. If it were Enron, WorldCom, Lernout & Hauspie, Ahold, Société Générale or another case does not matter. Fact is that fraud has become a serious part of our life and hence a serious cost to our economy. Several studies on this phenomenon report shocking numbers: forty-three percent of companies worldwide have fallen victim to economic crime in the years 2006 and 2007 (PwC, 2007). The average financial damage to companies subjected to this survey was US$ 2.42 million per company over two years. Participants of another study (ACFE, 2006)[1] estimate a loss of five percent of a company's annual revenues to fraud. Applied to the 2006 United States Gross Domestic Product of US$ 13,246.6 billion, this would translate to approximately US$ 662 billion in fraud losses for the United States only. These numbers all address corporate fraud.

[1] "The Association of Certified Fraud Examiners (ACFE) is the world's premier provider of anti-fraud training and education. Together with nearly 40,000 members, the ACFE is reducing business fraud worldwide and inspiring public confidence in the integrity and objectivity within the profession." (www.acfe.com)

There are several types of corporate fraud. The most prominent distinction one can make in fraud classification is internal versus external fraud, a classification based on the relationship the perpetrator has to the victim company. Management fraud is an example of internal fraud, where insurance fraud is a classic example of external fraud.

In this paper we present a framework for internal fraud risk reduction. Risk reduction comprehends both fraud detection and prevention and the framework is for both academics to investigate how to reduce internal fraud risk and for organizations. In a previous paper, we already present a framework with data mining being the core of that framework. In this paper we complement that framework with a process mining part. Process mining aims at uncovering a process model based on real transaction logs. This relative new research domain can be applied in several ways for the purpose of internal fraud risk reduction.

We start the paper with an introduction in internal fraud and internal control, since our framework is suggested as a complement to the internal control framework. In the next section we present our framework, followed by an introduction in process mining. Because the concepts of continuous auditing and continuous monitoring have a lot in common with the presented work, these concepts are shortly mentioned in Section 5. In Section 6 we present the application of our framework in a case company. We end with a conclusion.

2 Internal Fraud and Internal Control

In this paper, we consider the threat of internal fraud. For internal corporate fraud we rely on the definition of "occupational fraud and abuse" by the ACFE: "*The use of one's occupation for personal enrichment through the deliberate misuse or misapplication of the employing organization's resources or assets.*" (ACFE, 2006) This definition encompasses a wide variety of conduct by executives, employees, managers, and principals of organizations. Violations can range from asset misappropriation, fraudulent statements and corruption over pilferage and petty theft, false overtime, using company property for personal benefit to payroll and sick time abuses.

Where the academic field does not pay much of attention to internal fraud (merely to external fraud), it has received a great deal of attention from other interested parties. The emergence of fraud into our economic world didn't go unnoticed. In 2002, a US fraud standard (SAS 99) was created and by the end of 2004 also an international counterpart (ISA 240) was effective. Meanwhile, the CEO's of the International Audit Networks released a special report in November 2006: Global Capital Markets and the Global Economy: A Vision From the CEOs of the International Audit Networks.

This report, issued by the six largest global audit networks, is released in the wake of corporate scandals. The authors of this report express their believe in fighting fraud, as they name it "*one of the six vital elements, necessary for capital market stability, efficiency and growth*". The remaining five elements concern investor needs for information, the alignment and support of the roles of various stakeholders, the auditing profession, reporting and information quality.

The threat of internal fraud was first officially recognized in 1985 when the (US) National Commission on Fraudulent Financial Reporting (known as the Treadway Commission) was formed. To study the causes of fraudulent reporting and make recommendations to reduce its incidence, the Treadway Commission issued a final report in 1987 with recommendations for auditors, public companies, regulators, and educators. This report re-emphasized the importance of internal control in reducing the incidence of fraudulent financial reporting and included a recommendation for all public companies to maintain internal controls. The Committee of Sponsoring Organizations of the Treadway Commission (COSO) [2] was formed to commission the Treadway Commission to perform its task. In response to this recommendation, COSO developed an internal control framework, issued in 1992 and entitled *Internal Control - Integrated Framework*. According to the COSO framework, internal control is defined as:

> *A process, effected by the entity's board of directors, management, and other personnel, designed to provide reasonable assurance regarding the achievement of objectives in the following categories:*
>
> - *Effectiveness and efficiency of operations*
> - *Reliability of financial reporting*
> - *Compliance with applicable laws and regulations*

In meanwhile, COSO issued in 2004 a revision of the *Internal Control - Integrated Framework* under the title of *Enterprise Risk Management Framework*, expanding on internal control to the broader subject of enterprise risk management. (Cosserat, 2004; Davia et al., 2000; Whittington and Pany, 1998) Following this broad definition, internal control can both prevent and detect fraud. And although this definition is stemming from the foundation of the National Commission on Fraudulent Financial Reporting, also other classes of fraud than fraudulent financial reporting can be encountered.

[2]The sponsoring accounting organizations include the American Institute of Certified Public Accountants (AICPA), the American Accounting Association (AAA), the Financial Executives Institute (FEI), the Institute of Internal Auditors (IIA), and the Institute of Management Accountants (IMA).

Also the studies of PwC and the ACFE mentioned before, reveal some information concerning the detection of internal fraud. Internal control seems to deliver an effective tool in the fight against internal fraud. So from different angles, internal control is considered to be a means that has the ability to fight internal fraud. Likewise, in a business environment internal fraud is currently dealt with by internal control. As mentioned before, internal control encompasses a wide variety of tasks and settings. Next to a qualitative approach (like for example creating a control environment), quantitative data analyzing is required. It is at this point we believe there lies an opportunity to combine academic research with practical insights. In another paper by Jans et al. (2008) a data mining approach is proposed as a complement to the internal control framework. We hereby focus on fraud risk reduction, which includes both fraud prevention and fraud detection, just like internal control. The suggested framework of that study (and applied in a case study) can be found in Figure 1. We refer to Jans et al. (2008) for a detailed description of this framework.

In this paper, we wish to introduce yet another complement to the internal control framework, a second path. Where the first complementary advise for internal fraud risk reduction is to apply a data mining approach, we now suggest to also apply a process mining approach. Process mining is a relative new research domain and aims to extract an "a posteriori" process model from stored transaction logs. This enables *Delta analysis*, i.e. detecting discrepancies between the process design constructed in the design phase and the actual execution in the enactment phase (van der Aalst et al., 2003). This kind of analysis is important in the light of defining opportunities to commit fraud. We will discuss the framework and the underlying ideas in the following section.

3 Framework for Internal Fraud Risk Reduction

In this section we introduce our framework with its underlying concept. This framework provides both a guidance for the empirical part of our study and a framework for other researchers to help in their approach to reduce internal fraud risk. In Figure 2 one can find an extended version of Figure 1. The left branch of the framework is the part which was introduced in a former paper so we will not go into detail about this. In this paper we wish to present the common part and the right branch of the framework.

Our framework, presented in Figure 2, starts with **selecting a business process with an advanced IT integration**. An organization should select a business process which it thinks is worthwhile investigating. The implementation of advanced IT is, according to Lynch and Gomaa (2003), a breeding ground for employee fraud. So selecting a business process with an

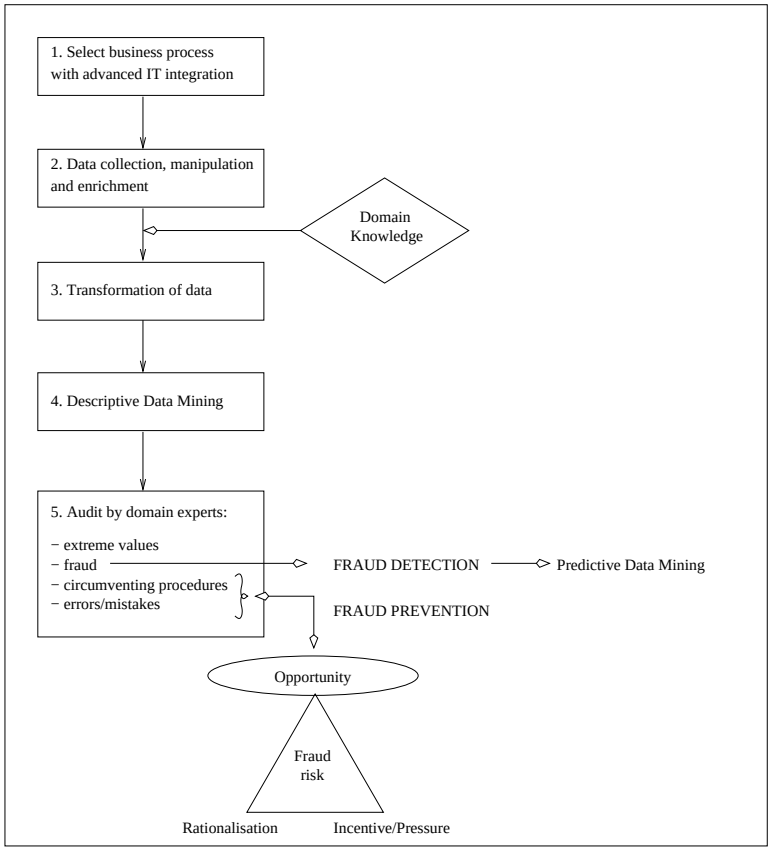

Figure 1: Framework for internal fraud risk reduction, applying a data mining approach

advanced IT integration is a good starting point to encounter this stream of frauds. Of course, for applying a data or process mining technique, we need a process of which data is electronically stored. This practical note can however not be the decisive reason for selecting a specific business process.

After the selection of an appropriate business process, **data has to be collected, manipulated and enriched** for further processing. This is comparable to the step "Data preparation" in Chien and Chen (2008)'s framework for personnel selection. The manipulation of data refers to the cleaning of data, merging connected data, transforming data into interpretable attributes and dealing with missing values. These are mainly technical transactions.

During the third step, a **transformation of the data** occurs. For the data mining branch, the technical data is translated into behavioral data. This

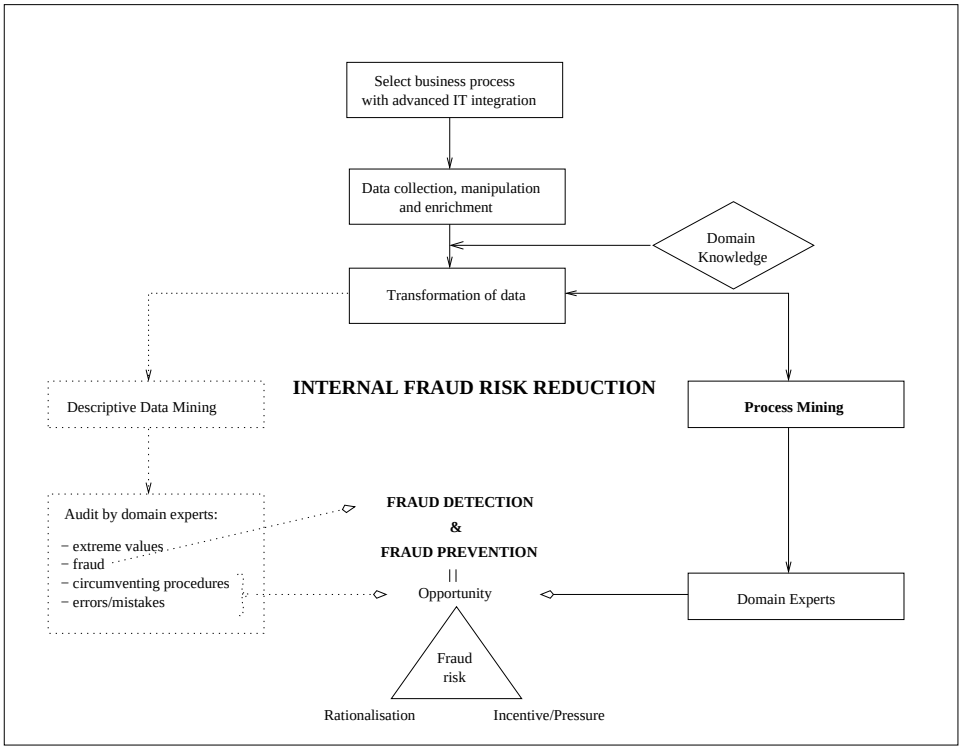

Figure 2: Extended framework for internal fraud risk reduction, integrating process mining

translation builds upon **domain knowledge** and is not just a technical transformation. For the process mining branch, the transformation of data merely refers to the creation of an event log. This event log will be the subject of the process mining step. Although the researcher may dispose of the desired log information, it is seldom available in the required format, hence the transformation of data into the event log.

The fourth step contains the **process mining** itself. As indicated before, this gives the ability to perform a *Delta analysis*. An organization has business processes mapped out in procedures, guidelines, user guides etcetera. In this step, we visualize the *actual* process that occurs in a certain business unit instead of the *designed* process. This way one can detect flows or sub flows that for example were not meant to exist. This can give insights in potential ways of misusing or abusing the system. It is the element 'Opportunity' of Cressey's fraud triangle that makes it interesting to gain these insights. Cressey's hypothesis, better known as the "fraud triangle", sees three elements necessary for someone to commit fraud. There has to be

pressure (or a "perceived non-shareable financial need"), a perceived *opportunity* and the perpetrator must be able to *rationalize* its acts. (Wells, 2005) The fraud triangle is cited many times in fraud literature and has become an important hypothesis. Opportunity is the only fraud triangle element an organization can exert influence on and hence is important in our framework. Also according to Albrecht et al. (1984)'s "fraud scale" opportunity is an element of influence on fraud risk. The results of the process mining step is eventually discussed with the **domain experts** to uncover the most important opportunities.

As can be seen, the process mining part of the framework works primarily on fraud prevention. However, the information gathered from this analysis, can be used as exploratory research and implemented in the data mining part. This way, process mining can indirectly also lead to fraud detection.

Before turning to the case study where our framework is applied, we give a short introduction to process mining and the ProM framework.

4 Process Mining

Nowadays many different information systems, like ERP, WFM, CRM and B2B systems, are characterized by the omnipresence of event logs. These can be referred to as 'audit trails', 'transaction logs', 'history' etcetera. Traditionally, an organization stores a lot of this kind of information as background information, but does not actively use this information to analyze the underlying process. This is where process mining aims to make a difference. "*The basic idea of process mining is to diagnose processes by mining event logs for knowledge*" (van der Aalst and de Medeiros, 2005).

With process mining, several assumptions are made. First of all, one assumes it is possible to record events such that at least four characteristics can be identified. An event 1) refers to an *activity*, 2) refers to a *case*, or process instance, 3) can be appointed to a performer, or *originator*, and 4) a *timestamp* can be identified. For each process under investigation these are the constraining assumptions. If available data fulfills these assumptions, process mining can be applied on that particular process. Table 1 shows a classic example of an event log, used by van der Aalst et al. (2007), van Dongen et al. (2005) and van der Aalst and de Medeiros (2005) amongst others. The event log shows an example with 19 events, allocated to five cases, describing five different activities, performed by six persons.

Event logs are the starting point of process mining. The data of the event log can be mined and different aspects about the underlying process can be analyzed. In general, three different perspectives can be distinguished: the process perspective, the organizational perspective and the case perspective.

Table 1: An example of an event log, used by van der Aalst et al. (2007).

Case id	Activity id	Originator	Timestamp
case 1	activity A	John	9-3-2004:15.01
case 2	activity A	John	9-3-2004:15.12
case 3	activity A	Sue	9-3-2004:16.03
case 3	activity B	Carol	9-3-2004:16.07
case 1	activity B	Mike	9-3-2004:18.25
case 1	activity C	John	10-3-2004:9.23
case 2	activity C	Mike	10-3-2004:10.34
case 4	activity A	Sue	10-3-2004:10.35
case 2	activity B	John	10-3-2004:12.34
case 2	activity D	Pete	10-3-2004:12.50
case 5	activity A	Sue	10-3-2004:13.05
case 4	activity C	Carol	11-3-2004:10.12
case 1	activity D	Pete	11-3-2004:10.14
case 3	activity C	Sue	11-3-2004:10.44
case 3	activity D	Pete	11-3-2004:11.03
case 4	activity B	Sue	14-3-2004:11.18
case 5	activity E	Clare	17-3-2004:12.22
case 5	activity D	Clare	18-3-2004:14.34
case 4	activity D	Pete	19-3-2004:15.56

The *process perspective* or the "How?" question focuses on the ordering of activities. Which paths are followed? This will typically be expressed in terms of Petri Nets or Event-driven Process Chain (EPC). The *organizational perspective* or the "Who?" question uses the input data in the field 'originator'. In this perspective, underlying relations between performers or between performers and tasks can be exposed. The *case perspective* or the "What?" question focuses on a single case. It will be more interesting to analyze if other data elements than in the event log are added in a separate table, for example the size of an order or the related supplier etcetera. (van der Aalst et al., 2007)

In the context of internal fraud risk reduction and the broader framework we place process mining in, the most important perspective is the process perspective. In a later stage, we still can turn to the organizational and the case perspective. So in this study we will start with the process perspective to expose opportunities to commit fraud within a company. Afterwards, we turn to the other perspectives, mostly in the light of monitoring controls (see Section 5)

For this study, the open-source ProM framework is used. This framework is developed by a group of researchers at the *Eindhoven University of Technology*. The ProM framework is a flexible framework that hosts several earlier developed mining tools such as EMiT (van der Aalst and van Dongen, 2002), Little Thumb (Weijters and van der Aalst, 2003), and MiSoN

(van der Aalst and Song, 2004). In this framework different algorithms for each of the three perspectives mentioned above can be plugged in. Also new plug-ins can easily be developed and added into this framework. For more details about this "pluggable" environment we refer to van Dongen et al. (2005) and to `www.processmining.org` where the software, several tutorials and useful publications can be found.

Before we continue with our case study, another contribution to both the academic literature and the accounting community raises to the surface. Although we wish to apply process mining in a broader context, it is also worth mentioning that this idea can or should be used in a context of continuous monitoring too. In the following section the concepts of continuous auditing and continuous monitoring are introduced.

5 Continuous Auditing and Monitoring

A new age of data storage goes along with new demands. Traditionally, internal audits and their related testing of controls are executed on a cyclical basis. However, with the electronic storage of all kinds of data, easily accessible and available in large volumes, new methods of internal auditing are implemented. Advanced technology has been employed to perform continuous auditing. Continuous auditing is defined as "*a framework for issuing audit reports simultaneously with, or a short period of time after, the occurrence of the relevant events*".(CICA/AICPA, 1999) An important subset of continuous auditing is the continuous monitoring of business process controls (Alles et al., 2006). Continuous monitoring of controls is defined by the Institute of Internal Auditors as "*a process that management puts in place to ensure that its policies and procedures are adhered to, and that business processes are operating effectively. Continuous monitoring typically involves automated continuous testing of all transactions within a given business process area against a suite of controls rules.* (IIA, 2005) Notice that continuous monitoring is a responsibility management bears, while continuous auditing is a task of the internal audit department. However, there is an interaction effect between the efforts put into place concerning continuous monitoring and continuous auditing. When management performs continuous monitoring on a comprehensive basis, the internal audit department can partly rely on this and no longer needs to perform the same detailed techniques as it otherwise would have under continuous auditing. (IIA, 2005)

In her framework, COSO also identifies the monitoring of controls as one of the five components of internal control. The remaining four components are the control environment, the entity's risk assessment process, the information system and control activities. Employees need to know that noncompliance with controls is likely to be detected (deterrence effect). Moni-

toring controls also provides feedback concerning these controls. (Cosserat, 2004)

We can conclude that the (continuous) monitoring of controls is certainly an activity that contributes to internal fraud risk reduction. The reason that we introduce the concept of continuous monitoring here, is that process mining provides a way of implementing such a continuous monitoring system. For example, segregation of duties is a common control that in many ERP systems is included. If one takes the procurement business process for instance, one person may have the authority to create a purchasing order and another person to approve the invoice. This is a control on the transactional level. It can however occur that one person has both authorities (both to create a purchasing order and to approve an invoice). It is not interesting for an organization that a person with such authorities can perform both activities on one single case (one purchasing order in this example). Therefor it would be interesting to enforce a segregation of duties on the case level instead of on the transactional level. Process mining has this potential. With process mining, one can control different explicit controls, such as the segregation of duties for example.

6 Case Study at Epsilon

6.1 Creating the Event Log

For the application of our suggested framework, the corporation of a case company was acquired. This company, which chooses to stay anonymous and is called Epsilon in this study, is ranked in the top 20 of European financial institutions. The business process selected for internal fraud risk reduction is procurement, so data from the case company's procurement cycle is the input of our study. More specifically, the creation of purchasing orders (PO's) was adopted as process under investigation. This is inspired by the lack of fraud files (at the compliance department) in this business process within the case company, while one assumes this business process is as vulnerable to fraud as every other business process.

For the process perspective, it is not necessary to have a specific fraud in mind. It is the objectivity with which the process mining techniques work, without making any presuppositions, that gives these techniques surplus value. We see the Delta analysis as a starting point to evaluate with an open mind what opportunities these deviations can mean for a perpetrator. When one has a specific fraud in mind when interpreting the analysis and looking if there are opportunities to commit this specific fraud, one can be blind for other opportunities. On the other hand, when mining the organizational and the case perspective, it can be beneficial to have some specific frauds in

mind. This is certainly the case at the case perspective, as the monitoring of internal controls fits into this perspective. At this stage specific internal controls, motivated by specific frauds in mind, are monitored and checked.

As a start, a txt-dump is made out of their ERP system, SAP. All PO's that in 2007 resulted in an invoice are the subject of our investigation. We restricted the database to invoices of Belgium. This raw data is then reorganized into an event log and a random sample of 10,000 process instances out of 402,108 was taken (for reasons of computability). Before creating the event log, the different activities or events a case passes through, have to be identified, in order to meet the assumptions.

An important assumption at process mining is that it is possible to describe the process under consideration by sequentially recording events. These events are the activities that all together constitute the process. Aside from the possibility to determine such sequential events, it is also assumed that these events are all linked to one particular case, called a *process instance*.

It is beyond the scope of this paper to fully describe the procurement process at Epsilon, supported by SAP. What it boils down to (based on interviewing domain experts) is that a PO is made, signed and released, the goods are received, an invoice is received and it gets paid. During this process all different kind of aspects are logged into the ERP system, from which we now have to create an event log. The first question we must ask ourselves is *'What would be a correct process instance to allocate events to?'*.

A natural choice of process instance would be a PO, since this seems to be the central document where everything relates to. But do we have data available to link all steps to a PO and to construct as such event logs per process instance, being a PO? The answer is short: yes, we have this information. We know exactly who made a PO; who signed and released it and when; we know when the Goods Receipts and Invoice Receipts are obtained and by whom; and we know when these invoices are paid. Still, we cannot use a PO as process instance. This rejection is on grounds of the dynamics of a PO. We know for example which PO is signed or released, we do not know however anything about the content of the PO at that time. This means that we can see for example that a PO has been signed and released for ten times, but we do not know the exact content of what has been approved each time. The same holds for the related Goods Receipts and Invoice Receipts. We know there is a link, but we do not know if the content of the invoice was for example also part of the PO when it was signed and released. These lacunae are created by the specifics and the two dimensionality SAP R3 uses in saving and linking data. An invoice line is for example matched with a line item of a PO. This is also the base of the ERP system to control the approval. So a line item could be a better candidate for process instance.

After examining the feasibility of using a PO item line as process instance, a

Table 2: Model example of event log of the purchasing process

PI-ID	WFMElt	Event Type	Timestamp	Originator
450000000190	Create PO	Complete	02 Feb 2006	John
450000000190	Change Line	Complete	30 Nov 2006	John
450000000190	Sign	Complete	05 Dec 2006	Paul
450000000190	Release	Complete	06 Dec 2006	Anne
450000000190	GR	Complete	05 Jan 2007	John
450000000190	IR	Complete	15 Jan 2007	Matt
450000000190	Pay	Complete	16 Feb 2007	Marianne
450000000210	Create PO	Complete	23 Jan 2007	Doug
...				

line item of a PO was indeed selected as process instance to allocate events to. We established the following events as activities of the process:
- Creation of the PO (parent of item line)
- Last change of the particular item line
- Sign(s) of parent PO after last change of item line
- Release of parent PO after last change of item line
- Goods Receipt on item line (GR)
- Invoice Receipt on item line (IR)
- Payment (or Reversal) of item line

These events are also called Work Flow Model Elements (WFMElt). After reorganizing the raw data (performed in SAS software), the event log contains per *Process Instance* (PI, being a PO line item) different events, being a *WFMElt*, with a particular *Timestamp* and *Originator* for each event. Also the *Event Type* must be stated, but this will be set default to 'Complete', since we do not have information to distinguish further. In Table 2 a model event log is given. Of course, the event log based on real life data will look differently and not as clean as this example.

For modeling the process underlying these activities and expecting flows, we use a Petri Net. A Petri net is a dynamic structure that consists of a set of *transitions*, *places* and *directed arcs* that connect these transitions and places in a bipartite manner. Transitions are indicated by boxes and relate to some task, while places are indicated by circles and represent passive phases. Places may hold one or more *tokens*, indicated by black dots. If all input places of a transition contain a token, this transition is *enabled* and may *fire*. When a transition fires, it consumes a token of each of the input places and produces a token for each of its output places. The Petri Net in Figure 3 represents in this way the procurement process at the case company.

The first activities (Create PO and Change Line) flows are straightforward.

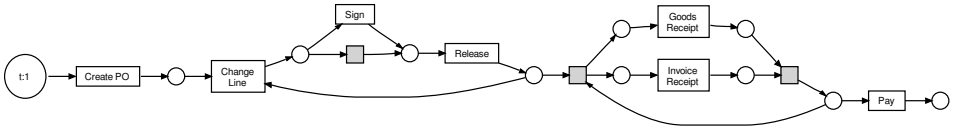

Figure 3: Process model of procurement in Petri Net

After a last change is made to a PO item line (our process instance), the parent PO can be signed and released, or only released. If only one signature is needed, one only has a release, otherwise this release is preceded by a sign. In reality and also depicted in our Petri net, the item line can be changed afterwards and a new sign and release are triggered. This will however not be visible in our event log, since we only have the last changes made to an item line to our possession. Only after the release, the Goods and Invoice Receipts can occur. This is an AND-relation, without a specified order. Afterwards the payment can occur. Normally, both a Goods and Invoice Receipt are prerequisites, so we depicted it this way. However, in some circumstances no goods Receipt is necessary. In these cases the goods receipt indicator must be turned off.

The designed model and its belonging activities formed the starting point of creating the event log. Notice that the decisions made during the composition of the event log are all related to the data structure the company has at its disposal. The decisions we make in this research, cannot be copied to another company, as each company stores its data on a different way. Even when the same ERP system is in use, there can be differences. The aim in this stage must always be the composition of an event log with the mandatory fields of *ProcessInstance, WFMElt*, *Event Type*, *Originator* and *Timestamp*. The combination of these assumptions and the available data (structure) will impose in each research its specific constraints.

In this particular case study we started from the table of PO line items. This table contained only items that were mentioned in invoices that were paid during 2007. A link between the parent PO and the last change on the item line could be made. The timestamp of this last change was used to link this PI to the 'Sign' and 'Release' activity. Apart from that, we started from the information about invoices, which gave us the opportunity to associate the 'Pay' events with unique process instances. Working backwards, it was possible to find the connection between a 'Pay' and an 'IR', followed by a link between an 'IR' and a 'GR'. This is the methodology we followed to select only those invoice and goods receipts that actually triggered the payments in our event log.

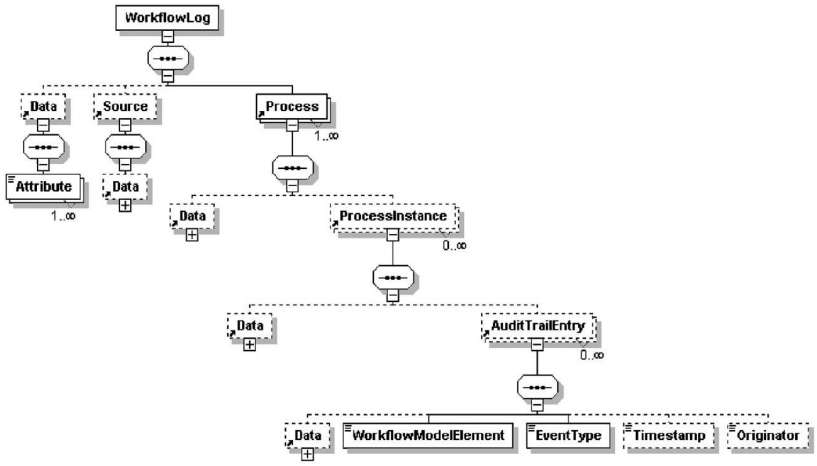

Figure 4: MXML mining format

Together with the ProM framework, a generic XML format to store event logs in is developed, called the MXML format. Figure 4 illustrates this MXML format. One can find the fields mentioned above. Aside from these fields, there is also some additional space for extra data, in the form of attributes. These attributes can be inserted at each level. The attributes created in our event log are listed in Table 3. On the level of a process instance, we added the following information: the document type of the parent PO, the purchasing group that entered this parent PO (on two different levels, hence the 'a' and 'b' version), and the associated supplier. Although these four attributes are actually linked to the parent PO and not to a separate item line, this is useful information. Aside from these first four attributes, we also included the order quantity and unit of the PO item line, the resulting net value and whether or not the goods receipt indicator was turned off. This is important to verify if the ERP system's internal control on this part is working efficiently. (A 'Pay' should not occur without a 'GR', unless the goods receipt indicator is turned off).

On the level of the audit trail entry, a work flow model element also carries unique information. In particular three events are enriched with additional information. When the event concerns a 'Change Line', we store information about this change: what was the (absolute) modification? This field contains a numeric value concerning a change in net value. If not the net value was changed, but another field, for example the delivery address, this field contains a modification of zero. The other attribute gives us, in case of a change in net value, the size of the modification, relative to the net value before the change (hence a percentage).

Table 3: Attributes of event log

Level	Attribute	WFMElt
Process Instance	Document type	
	Purchasing Group a	
	Purchasing Group b	
	Supplier	
	Order Quantity	
	Order Unit	
	Net Value	
	Goods Receipt Indicator	
Audit Trail Entry	Modification	Change Line
	Relative Modification	Change Line
	Reference GR	IR
	Reference Pay	IR
	Quantity IR	IR
	Value IR	IR
	Reference IR	Pay
	Value	Pay
	Reference IR	GR
	Quantity GR	GR
	Value GR	GR

When the event concerns an 'IR', four attributes are stored. We store the references that make the link to the 'GR' and 'Pay' possible, the quantity of the units invoiced, and the credited amount, the value. Notice that this information is not collected from an entire invoice, but only from the specific line that refers to the PO item line of this process instance. Similar to the 'IR', three attributes are stored when the event concerns a 'GR': the reference to link this Goods Receipt to the associated 'IR', the quantity of goods received and the resulting value that is assigned to this Goods Receipt. This value is the result of multiplying the Goods Receipt quantity with the price per unit agreed upon in the PO.

After collecting all the data necessary for the event log, ProM*Import* is used to convert our event log into the desired MXML format.

6.2 Descriptives

As already stated, we start with a random sample event log of 10,000 Belgian process instances. A process instance is a PO item line. The process analyzed in this paper contains seven real activities (see Table 4, original log). Notice that the event 'Reverse' does not occur in this log.[3] The log at hand contains 65.931 events in total and 297 originators participated in the

[3]'Reverse' is apparently not present at all in the log for Belgium (not even before random sampling).

Table 4: Log events

WFMElt	Occurrences (absolute)		Occurrences (relative)	
	original log	cleaned log	original	cleaned log
Pay	11,426	11,426	17.33%	17.558%
IR	11,282	11,172	17.112%	17.167%
Create PO	10,000	10,000	15.167%	15.366%
Change Line	10,000	9,505	15.167%	14.606%
Release	8,641	8,540	13.106%	13.123%
Sign	7,590	7,489	11.512%	11.508%
GR	6,992	6,945	10.605%	10.672%

process execution. All audit trails (the flow one process instance follows) start with the event 'Create PO', but they do not all end with 'Pay'. The ending log events are 'Pay' (95%), 'Change Line' (4.5%), 'Release' and 'GR'. Since not all audit trails end with 'Pay', we could add an artificial 'End' task before we start mining this process. However, we might better clean up the event log further, so we have left only those audit trails that end with 'Pay'. We kept the process instances randomly selected, but left out all the audit trail entries after the last payment since we then have the entire process covered, from creating a PO until the payment of the associated goods. This resulted in an event log with 65.077 audit trail entries and 293 originators. The occurrences of the audit trail entries can be found in the 'cleaned log' part of Table 4. As can be seen are all 'Pay' activities maintained, and there are still 10,000 process instances involved (there every audit trail starts with 'Create PO'). The log summary confirms that all audit trails end with the activity 'Pay'. This cleaned log will be our process mining input.

Analyzing a bit more the event log at hand, yields that 216 different patterns are present. This is a very high number, certainly for such a relatively simple process model design. This gives us already an idea of the complexity of this process and the noise on this event log.

6.3 Mining the Procurement Process

6.3.1 Introduction

After presenting our event log and some descriptive statistics, we start process mining. As already mentioned, we will focus first on the process perspective or the control-flow perspective. This results in a graphical representation of the process underlying the transactions of the event log. To construct such a process model, causal dependencies have to be exposed. The causal dependency between activity A and activity B, denoted by $A \rightarrow_W B$, means that A is directly followed by B, but B is not directly followed by A.

This dependency is deducted by looking at the (timed) order of activities per process instance (or case). Looking at Table 1, we see that $A \rightarrow_W B$, $A \rightarrow_W C$, $A \rightarrow_W E$, $B \rightarrow_W D$, $C \rightarrow_W D$, and $E \rightarrow_W D$. In this model example these dependencies are easy to deduce, but in real life logs this is not only harder, there are also two important complicating factors: noise and completeness.

For a causal dependency between A and B, B must follow A directly, but A may never follow B. So if an event log contains for example 99 out of 100 times the sequence A - B, and one time B - A, it will not give the dependency $A \rightarrow_W B$. So we need a mining algorithm that can handle *noise*.

Further can *completeness* be a problem. If there are n activities that can be executed in parallel, the total number of possible causal dependencies is $n!$, growing faster than an exponential function. Hence it is not realistic that every log contains all paths possible for the underlying process. This leads us to a supplementary condition: the mining algorithm needs to be able to handle low frequent behavior.

To tackle the problems of noise and completeness, the HeuristicsMiner Plug-in of ProM is used in this study. The Heuristics Miner, described in Weijters and van der Aalst (2002), Weijters and van der Aalst (2003), and Weijters et al. (2006), and applied in a similar study in van der Aalst et al. (2007), shows to deal with incompleteness and is robust for noise. For a better understanding of the heuristic approach, we will shortly discuss the underlying ideas in the following subsection.

6.3.2 HeuristicsMiner

The starting point of the HeuristicsMiner is a frequency based metric to indicate the probability of a dependency relation between two activities A and B. If activity A is often directly followed by activity B, and B is never followed by A, there is a high probability that there is a dependency relation between A and B. This probability is expressed through $A \Rightarrow_W B$. These dependency values $\Rightarrow_W$ between the events of an event log become the input of the metric. Afterwards this metric is used in a simple heuristic in the search for reliable dependency relations ($A \rightarrow_W B$ relations). Let us first explain how the dependency values are calculated.

Let W be an event log over period T, and $a, b \in T$. Then $\mid a >_W b \mid$ is the number of times a is followed directly by b (denoted by $a >_W b$) in event log W, and

$$a \Rightarrow_W b = \left(\frac{\mid a >_W b \mid - \mid b >_W a \mid}{\mid a >_W b \mid + \mid b >_W a \mid +1} \right)$$

As can be seen, the value of $a \Rightarrow_W b$ will always lie between -1 and 1. We quote a simple example of van der Aalst et al. (2007) to demonstrate

the rationale behind this definition. If five traces of $A >_W B$ are found, but the other way around never occurs, the dependency value $A \Rightarrow_W B$ equals $5/6 = 0.833$. This value indicates that we are not completely sure of the dependency relation. The five observations could have been caused by noise. If however 50 traces of $A >_W B$ are found, and again the other way around never occurs, the dependency value $A \Rightarrow_W B$ equals $50/51 = 0.980$. This higher value gives us pretty much assurance of the dependency relation $A \rightarrow_W B$. Even if there is one trace, caused by noise, where A follows B, $A \Rightarrow_W B$ equals $49/52 = 0.942$, still a high value. So a high $A \Rightarrow_W B$ value strongly suggests that there is a dependency relation between activity A and B, $A \rightarrow_W B$.

The rationale the authors of the HeuristicsMiner follow is concerning the threshold value of this dependency value. A high value strongly suggests a causal relation between two activities, but what is a high value? The solution the authors present is not to set an absolute threshold at all. The authors rely on the knowledge that each non-initial activity must at least have one other activity that is its cause, and that each non-final activity must have at least one dependent activity. This is the information that is captured in the heuristic to limit the search for reliable dependency relations. The heuristic approach takes *the best* candidate with the highest $A \Rightarrow_W B$ score. Alternatively, the best candidate plus all candidates with a $A \Rightarrow_W B$ score close to the value of the best candidate are selected.

Although the heuristic formulated above is not complete and has to be extended to recognize complicating factors such as recursion, short loops and the type of joins and splits, we can now interpret dependency values and can start with a first application of the HeuristicsMiner.

6.3.3 Results

We start the HeuristicsMiner with high thresholds, revealing the core process. The expected result is a graph that is fully explicable by domain experts, without any flows that raises questions. For this analysis, the default thresholds were maintained with exception of the 'Positive observations', this parameter was set '300' (instead of '10'). Other combinations of high thresholds were used and all yielded the same Heuristic Net. The result is displayed in Figure 5. The number in the activity box indicates the frequency of that activity; the arcs between activities depict dependency relations; with close to the arrow the relevant dependency value and the number of audit trails that followed this path. The model has an 'improved continuous semantics fitness' of 0.6203 and 557 "wrong" observations[4]. We

[4]"wrong" observations are observations that do not correspond to the presented direction of arcs.

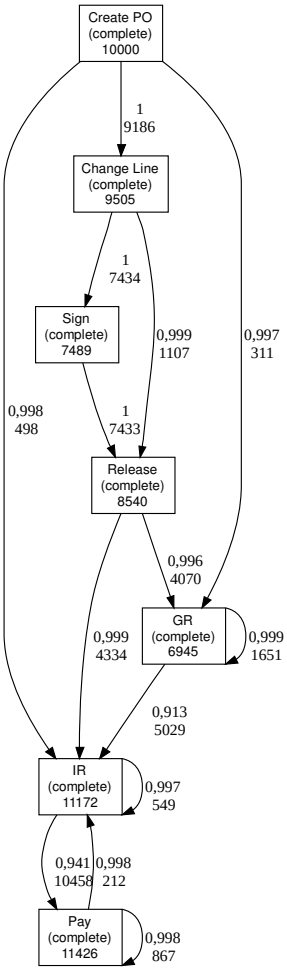

Figure 5: The result of HeuristicsMiner with positive observations = 300

Table 5: Thresholds allowing for more unfrequent flows

Parameter	Threshold
Relative-to-best threshold	0.3
Positive observations	1
Dependency threshold	0.6
Length-one-loops threshold	0.6
Length-two-loops threshold	0.6
Long distance threshold	0.6
Dependency divisor	1
AND threshold	0.1
Extra Info	false
Use all-events-connected-heuristic	true
Use long distance dependency heuristics	false

discuss this model with a domain expert of the case company. The outcome of this discussion is presented in the following paragraphs.

The designed process model is put forward, however under slightly different circumstances. First a PO is created, afterwards the item line is changed. In a next step this PO is signed and released, or released immediately. Whether a PO is released immediately or first signed, depends on several parameters like purchasing group, document type and amount. There is however not one general rule to test whether these conditions are met. After the PO is released, a Goods Receipt on the particular line item is introduced, followed by an Invoice Receipt. We here see a deviation, namely that not always a Goods Receipt is entered. This is not depicted in the designed model, but as already mentioned, this is indeed a possible variant. If the goods receipt indicator is turned off, a Goods Receipt is not conditional to a payment. The last step is the payment itself after the Invoice Receipt. Also here is a slight deviation from the designed model: there is an interaction between Invoice Receipt and payment. This can be the case, when several invoices or invoice lines are linked to one Goods Receipt line. Then these invoices are entered separately and paid accordingly. Finally, the loops that we find on 'GR', 'IR' and on 'Pay' are also not surprising for the domain expert and there is no harm seen in this practice.

Two arcs however are not yet explained. Apparently, some PO's are created, and the next step is a Goods or Invoice Receipt. This is due to the data collection. The step 'Change Line' we have at our disposal, is only the last change that happened on this item line. This change then triggers the activities 'Sign' and 'Release'. Due to this particular structure, for example the following flow order could be represented in our event log: Create PO - GR - IR - Pay - Change Line - Sign - Release. After cleaning the log, these last three activities are deleted. This data structure explains the two arcs from 'Create PO' to 'GR' and to 'IR'.

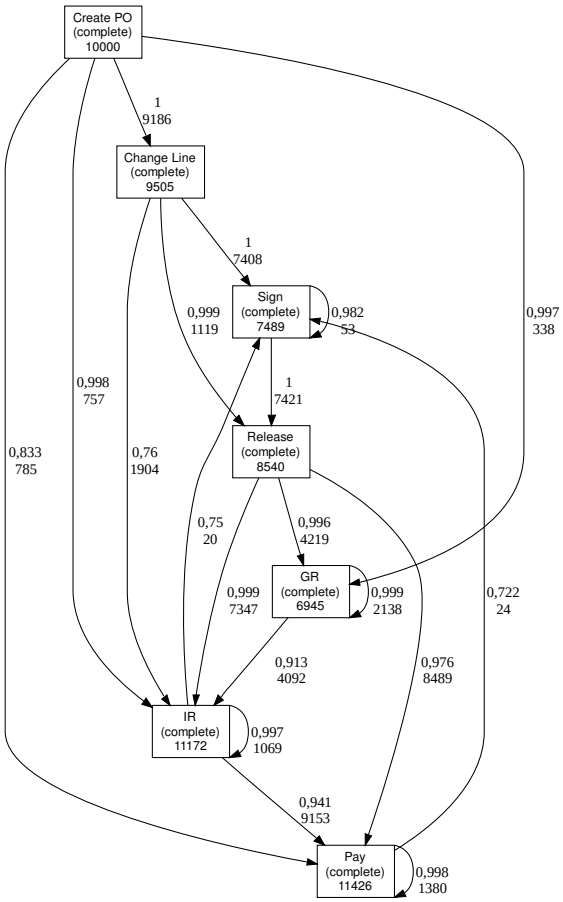

Figure 6: The result of HeuristicsMiner with lower thresholds (see Table 5)

When we loosen our thresholds, we will get a model with far more flows. We set the thresholds summarized in Table 5 with the result depicted in Figure 6. As expected, more flows are present, representing less frequent patterns. This is also indicated by the lower dependency values near the arcs. The extra flows and their dependency values are:

- Create PO → Pay 0.833
- Change Line → IR 0.76
- Release → Pay 0.976
- IR → Sign 0.75
- Pay → Sign 0.722
- loop on Sign 0.982

Except for 'Release → Pay' and the loop on sign, the dependency values are quite low. Probably these two flows will be quite normal. The 'Release → Pay' flow should be interpreted in an AND-relationship with 'Release → IR' and 'Release → GR'. Apparently, the foreseen order of Release - GR - IR is not always respected. This however should still be inspected, since a payment should not be able to occur without an Invoice Receipt.

Invoice Always Precedes Payment?
Like just mentioned: each 'Pay' should be preceded by an 'IR'. The flows 'Create PO → Pay' and 'Release → Pay' should therefor not be present. However, 'Create PO → Pay' could again be explained away by the fact that we only have the last changes of line items. So this could be the reason that we do not have the confirmation that there was a sign and release. This still does not take away the fact that the 'Pay' in this flow was not preceded by an Invoice Receipt. Since we made a selection of document types out of the SAP tables (like invoices and goods receipts), it could be the case that other document types (like *Subsequent Debits*) triggered the payment. We did not take this document type into account for a reason, namely because this is not supposed to be used quite often. So although this could be a possible explanation for some (or several) cases, it is optimistic to assume this explains all these specific audit trails. Further investigation in these process instances is required.

Mining the event log in a *case perspective*, can gives us more insights on this matter. We use the *LTL-Checker* plug-in to check whether each process instance eventually has an Invoice Receipt, and later a payment. The formula tested for this was "eventually_activity_A_then_B" with parameters A and B set to 'IR' and 'Pay' respectively. Out of the 10,000 cases, 9,900 process instances were categorized as 'correct', 100 instances as 'incorrect'. When analyzing further these instances, seven patterns were underneath these 100 cases, of which three pattern jointly represented 92 cases (see Figure 7). Pattern 1 has a frequency of 40, pattern 2 of 33 and pattern 3 occurs 19

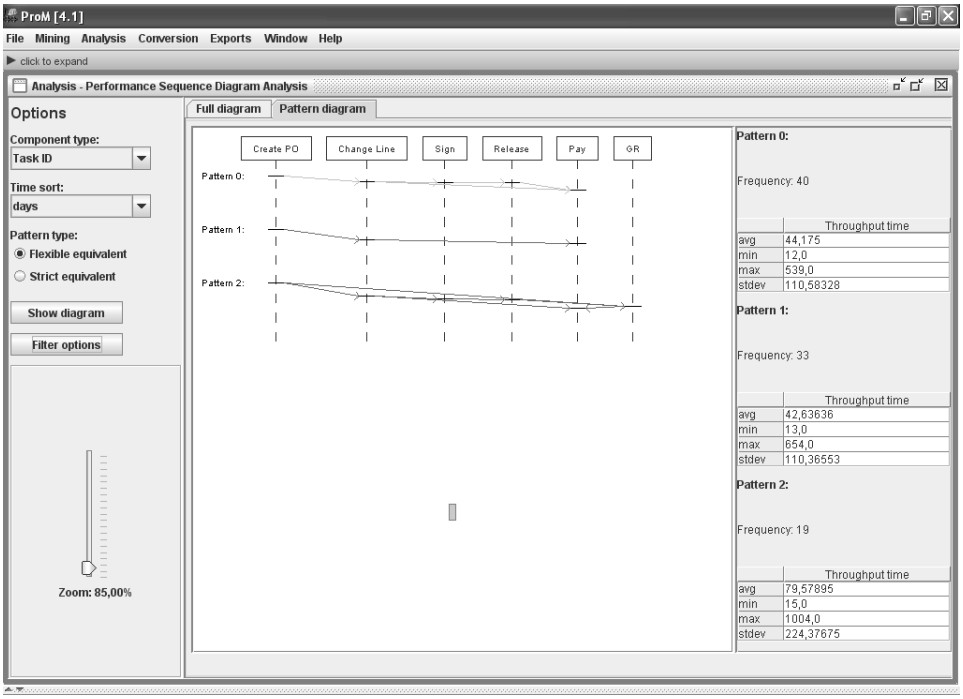

Figure 7: Three patterns of 92 cases without 'IR'

times. These patterns will be discussed with the domain expert. The remaining eight cases were selected to examine individually. For the moment we are still waiting on the domain expert's comment on these cases.

Is Every Change Line Released According Policy?
The flow 'Change Line → IR' is not an order of activities that is foreseen in the designed model. The domain expert sees two explanation for this flow. A first explanation is that the change of the line item was not value related (or within the allowed margins) so that no new sign and/or release was mandatory. Notice that we cannot make any assumptions about the presence of a Goods Receipt in this matter. Perhaps the Goods Receipt was before this final last modification. Maybe there was no Goods Receipt at all, justified or not. The other explanation of the 'Change Line → IR' flow is that this PO item line is changed each time a new order is placed (instead of creating a new line item or PO). If this is the case, we cannot assure -in this specific event log- that the Invoice Receipt is still concerning the latest content of this PO item line. If this is the case, we are dealing with procedures that are not followed.

Also the flow 'IR → Sign' is related to the same structure: a significant

change should first be signed, before an Invoice could be received. Only after the release, a PO can be sent to a supplier, hence the name 'release'. The problem is that, given the limitations of the data structure and as a consequence of the event log, we cannot be sure these two actions ('IR' and 'Sign') concern the same content of the PO item line (the same reasoning as in 'Change Line → IR'). However, the possibility exists. This would mean that only after the invoice is received, the approval of the PO occurs. This would mean further that the order is placed at the supplier, even before the PO is authorized. This is certainly not according the procurement policies.

Further investigation is recommended to check whether a change line of significant value is (signed and/or) released before a Goods or Invoice Receipt is introduced.

Sign Always Precedes Payment?
The flow with the lowest dependency value, 'Pay → Sign', is also worth investigating. Again, we do not have certainty that this payment is related to the same content of the PO that is signed for at that moment. Maybe the item line is changed in mean while to place a new order -again, this is against policy- and the triggered sign takes place after the payment of the previous order. However, this is the best case scenario, with 'only' procedures that are violated. The worst case scenario is that an invoice is paid and the related PO is only signed afterwards. Normally, this should not be possible in SAP, if the internal controls work efficiently.

To check the question if every payment is preceded by a sign, is not as straightforward as the check whether a payment is always preceded by an Invoice Receipt. This is due to the data structure in SAP. There is no direct link between the activities 'Sign' and 'Pay'. This is because a sign concerns a complete PO, while a payment concerns only an item line of the PO. Additionally, SAP keeps no history of the content of the PO. We do not know for instance what the content of the paid item line was. In meanwhile, this content could have been changed. In order to check the flow 'Pay → Sign' we suggest an audit of a random sample of these instances.

7 Discussion

In this work we introduce the new field of process mining into the business environment. For the case of data mining, it took some decades before the application of this research domain was projected from the academic world into the business environment (and more precisely as a fraud detection mean and as a market segmentation aid). As for the case of process mining, we wish to accelerate this step and recognize already in this quite early stage which opportunities process mining offers to business practice. In our extended framework, we point out the usefulness of process mining in

the light of internal fraud risk reduction. Process mining offers the ability to objectively extract a model out of transactional logs, so this model is not biased towards any expectations the researcher may have. In the light of finding flaws in the process under investigation, this open mind setting is a very important characteristic. Also the ability of monitoring internal controls is very promising.

Not only for internal fraud risk reduction, but also for the field of continuous auditing and continuous monitoring, process mining has valuable characteristics. We hope to cause a chain of further research in the usefulness of process mining in the business practice; both in the context of fraud risk reduction, as in the context of continuous auditing and/or monitoring. We also aim to stimulate business practice to recognize the opportunity process mining offers.

The results presented in the last section are only a starting point for further investigation. In this paper mainly the *process perspective* of process mining was addressed (except for the short introduction of the *LTL-Checker*), more precisely the *discovery* of a process (as opposed to the *conformance* of a process). In a further stadium of this paper we wish to fully examine all indicated flaws of the process, using different techniques.

Although room for further investigation is left, there are already some interesting aspects of the procurement process discovered. Another important issue is the data structure of SAP. We are confronted with many limitations in our research, just because of the way the data is stored in the SAP tables. This could be a good lesson to learn from for SAP, if it wants to be a part of the upcoming process mining era.

8 Conclusion

In this paper we present an extended framework, based on a previous work of Jans et al. (2008), to apply process mining in the context of internal fraud risk reduction. Process mining offers a lot of possibilities to examine a business process. Different aspects can be investigated, with all perspectives being interesting in terms of risk reduction. Also the explicit possibility to check internal controls, offers a new way of looking at continuous monitoring, a part of internal fraud risk reduction.

References

ACFE (2006). 2006 ACFE Report to the nation on occupational fraud and abuse. Technical report, Association of Certified Fraud Examiners.

Albrecht, W. S., K. R. Howe, and M. B. Romney (1984). *Deterring Fraud: The Internal Auditor's Perspective*. Institute of Internal Auditors Research Foundation.

Alles, M., G. Brennan, A. Kogan, and M. A. Vasarhelyi (2006). Continuous monitoring of business process controls: A pilot implementation of a continuous auditing system at Siemens. *International Journal of Accounting Information Systems 7*, 137–161.

Chien, C.-F. and L.-F. Chen (2008). Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry. *Expert Systems with Applications 34*(280-290).

CICA/AICPA (1999). Continuous auditing. Technical report, The Canadian Institute of Chartered Accountants.

Cosserat, G. W. (2004). *Modern Auditing* (2 ed.). John Wiley & Sons, Ltd.

Davia, H. R., P. Coggins, J. Wideman, and J. Kastantin (2000). *Accountant's Guide to Fraud Detection and Control* (2 ed.). John Wiley & Sons.

IIA (2005). Continuous auditing: Implications for assurance, monitoring, and risk assessment. *Information Technology Controls - Global Technology Audit Guide (GTAG)*.

Jans, M., N. Lybaert, and K. Vanhoof (2008, June). Internal fraud risk reduction: Results of a data mining case study. In *Forthcoming Proceedings of 10th International Conference on Enterprise Information Systems*, Barcelona, Spain.

Lynch, A. and M. Gomaa (2003). Understanding the potential impact of information technology on the susceptibility of organizations to fraudulent employee behaviour. *International Journal of Accounting Information Systems 4*, 295–308.

PwC (2007). Economic crime: people, culture and controls. the 4th biennial global economic crime survey. Technical report, PriceWaterhouse&Coopers.

van der Aalst, W. and A. de Medeiros (2005). Process mining and security: Detecting anomalous process executions and checking process conformance. *Electronic Notes in Theoretical Computer Science 121*, 3–21.

van der Aalst, W., H. Rijers, A. Weijters, B. van Dongen, A. de Medeiros, M. Song, and H. Verbeek (2007, July). Business process mining: An industrial application. *Information Systems 32*(5), 712–732.

van der Aalst, W. and M. Song (2004). Mining social networks: Uncovering interaction patterns in business processes. In *Lecture Notes in Computer Science*, Volume 3080, pp. 244–260. Springer-Verlag, Berlin.

van der Aalst, W. and B. van Dongen (2002). Discovering workflow performance models from timed logs. In *Lecture Notes in Computer Science*, Volume 2480, pp. 45–63. Springer-Verlag, Berlin.

van der Aalst, W., B. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. Weijters (2003). Workflow mining: A survey of issues and approaches. *Data & Knowledge Engineering 47*, 237–267.

van Dongen, B., A. de Medeiros, H. Verbeek, A. Weijters, and W. van de Aalst (2005). The ProM framework: A new era in process mining tool support. Volume 3536, pp. 444–454. Springer-Verlag, Berlin.

Weijters, A. and W. van der Aalst (2002). Process mining: Discovering workflow models from event-based data. pp. 78–84.

Weijters, A. and W. van der Aalst (2003). Rediscovering workflow models form event-based data using little thumb. *10*, 151–162.

Weijters, A., W. van der Aalst, and A. de Medeiros (2006). Process mining with the HeuristicsMiner algorithm. *Beta Working Paper Series*.

Wells, J. (2005). *Principles of Fraud Examination*. John Wiley & Sons.

Whittington, O. R. and K. Pany (1998). *Principles of Auditing* (12 ed.). Irwin McGraw-Hill.