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

Mieke Jans   Nadine Lybaert   Koen Vanhoof
Jan Martijn van der Werf

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. This suggestion results in an extension of the IFR^2 framework, presented by Jans et al. (2009). Process mining diagnoses processes by mining event logs. This way we can expose opportunities to commit fraud in the process design. We present the extended IFR^2 framework as a complement to the internal control framework of the COSO and apply this framework in a case company.

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 the 'Extended IFR^2 framework', based on Jans et al. (2009)'s IFR^2 framework, aimed at 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, the IFR^2 framework with data mining being the core to reduce internal fraud risk is presented. (Jans et al., 2009) In this paper we complement that framework with a process mining part, resulting in the extended IFR^2 framework. 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 mitigating 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

[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).

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.

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. (2009) a data mining approach is proposed as a complement to the internal control framework, leading to the IFR^2 framework. The focus hereby lies on fraud risk reduction, which includes both fraud prevention and fraud detection, just like internal control. For a detailed description of the IFR^2 framework, we refer to Jans et al. (2009).

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. Our suggestion is poured into an extension of the IFR^2 framework, leading to the extended IFR^2 framework. We will discuss the framework and the underlying ideas in the following section.

3 The Extended IFR^2 Framework

The extended IFR^2 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 1 one can find the extended version of the original IFR^2 framework. The left (shaded) branch of the framework is the part which was introduced in the former paper so we will not go into detail about this. We refer to our previous work for more information on that part. In this paper we wish to present the right branch of the framework.

The core of our extension is to apply process mining. As indicated before,

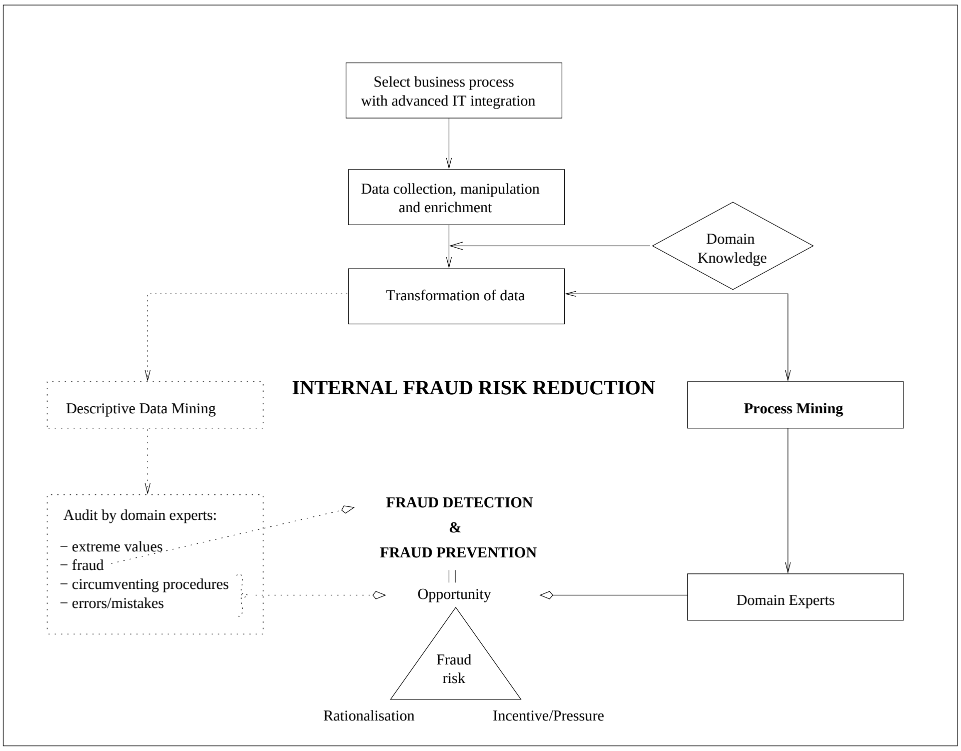


Figure 1: The extended IFR^2 framework, integrating process mining

this gives -for starters- the ability to perform a *Delta analysis*. An organization has business processes mapped out in procedures, guidelines, user guides etcetera. In the process mining 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. Aside from a *Delta analysis*, process mining also provides the possibility to specifically monitor internal controls, like for example the four-eyes principle or the segregation of duty. As opposed to currently wide used internal control tests, the process mining approach for monitoring internal control is data oriented, and not system oriented. In other words: we are able to test whether the true transactional data (the output of the internal control system) are effectively submitted to the presumed internal controls. Instead of testing whether the internal control settings function by means of performing a set of random tests, we mine the actual submitted data and are able to test whether all conditions are met. This provides a whole new view on monitoring internal controls.

Another advantage of applying process mining is that, just like with the data mining application, it is not necessary to have a specific fraud in mind. Further surplus value is delivered by the objectivity with which the process mining techniques work, without making any presuppositions. 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 (see below), it can be beneficial to have some specific fraud(s) in mind. This is certainly the case when monitoring internal controls. At this stage specific internal controls, motivated by specific frauds in mind, are monitored and checked.

After applying process mining, feedback from and to the domain experts is needed to interpret the results. This will eventually lead to new insights whether or not there are opportunities to commit fraud. It is the element 'Opportunity' of Cressey's hypothesis 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 most important in our framework. Also according to Albrecht et al. (1984)'s "fraud scale" opportunity is an element of influence on fraud risk.

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 extended IFR^2 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 logs. Typically, these information systems record information about the usage of the system by its users. These logs contain information about the instances, also called cases, processed in the system, the activities executed for each instance, at what time the activities were executed and by whom. Some systems also

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

contain information about the data users entered for each activity. However, this data is not actively used by the organization to analyze the underlying processes supported by the system.

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). It allows to analyze these event logs, sometimes also referred to as 'audit trail', 'transaction log' or 'history'. Records in these logs are called *events*. In process mining, each event needs to refer to an *activity* for a specific *case*. Preferably, each event also refers to the performer, the *originator* of the event, and a *time stamp*. 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. The *process perspective* or the "How?" question focuses on the ordering of activities, i.e. it tries to answer the question "Which paths are followed?"

This is typically expressed in graphical process models, using a formalism like Petri Nets, Event-driven Process Chain (EPC) or BPMN. The *organizational perspective* or the “Who?” question focuses on the users, the originators, that play a role within the process. 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 case in isolation. Typically, for this analysis, the log needs to be enriched by extra data about the case. This can be data about the complete case, or data for a specific event, like the data submitted at the event (van der Aalst et al., 2007).

In the context of internal fraud risk reduction and the broader framework we place process mining in, an important perspective to start with is the process perspective. In a later stage, we turn to the organizational and the case perspective. Therefore, 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 tool ProM (??, Aal) is used. ProM consists of many different algorithms to cover the analysis of the three perspectives. ProM is designed in such a way that researchers and users can easily develop their own plugins and add them to the framework. For more information about ProM, we refer the reader to van Dongen et al. (2005) and to `www.processmining.org`.

5 Continuous Auditing and Monitoring

Traditionally, internal audits and their related testing of controls are executed on a cyclical basis. Auditors typically check random samples, and use simple checklists to audit an organization. However, with the electronic storage of all kinds of data, easily accessible and available in large volumes, new methods of internal auditing can be developed and implemented. Already, 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 non-compliance with controls is likely to be detected (deterrence effect). Monitoring 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. Process mining can help in different aspects of monitoring and auditing. As all actions of each case are recorded by the system, one can check the complete process, rather than taking random samples. Secondly, process mining can help in the discovery, analysis, implementation and verification of business controls. One example is the segregation of duties. It is a common control included in many ERP systems. If one takes the procurement business process for instance, one person may have the authority to create a purchasing order and another person has the ability to approve the invoice. This is a control on the transactional level. It can however occur that one person has both authorities, e.g. the person is allowed to create a purchasing order and to approve an invoice. A control should prohibit that this person is approving invoices of purchase orders that person created himself. Process mining can verify this property, i.e. it is checked whether there is no purchase order where an invoice is approved by the same person that created the purchase order. This example shows that process mining has the potential to assist the auditor.

6 Case Study at Epsilon

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.

In a first part of the case study, we want to support the ideas of the domain experts about the process. For this purpose, we perform a process diagnostic step. A good methodology for process diagnostics by process mining can be found in Bozkaya et al. (2008), which will be the applied methodology in the next paragraphs. It consists of five phases: log preparation, log inspection, control flow analysis, performance analysis, and role analysis.

In a second part of the case study, we turn to a verification step. During this step, we wish to check whether certain aspects and conditions of the process hold or not. This will be elaborated in section **??**. We now start with the five phases of process diagnostics.

7 Log Preparation

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 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?'*.

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

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
...				

- (Change of the particular item line)
- Sign 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

The Change of an item line is no imperative event and could occur on several different moments in the process. This change can trigger a new 'Sign' and 'Release', but this is not always the case. Also important to note is the double dimensionality of the events. 'Create PO', 'Sign' and 'Release' are activities that occur on the header level of a PO. The remaining events are on the level of a PO line item. This can lead for instance to a 'Sign' and 'Release' in an audit trail of a particular PO line item (the process instance), while these events are not actually related to this line item, but perhaps to another line item of the same parent PO. This aspect is important to be aware of when interpreting the results.

The established events in our event log 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 representation. 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 **??** (*Not presented yet*) represents in this way the procurement process at the case company.

The first activity flows are straightforward. After the parent PO of an item line (our process instance) is created, this 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 between the creation and the Sign - Release activity. It is also possible that the item line is changed afterwards and a new sign and release could be triggered. Only after a release, an order can be sent to the supplier which will eventually result in a Goods Receipt and an Invoice Receipt. 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.

After turning the information from the SAP data base into the suggested events and event log, this event log was converted to the MXML format, a generic XML format to store event logs in. At this format, there is also 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, and the associated supplier. Although these three attributes are actually linked to the parent PO and not to a separate item line, this is useful information. Aside from these first three 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.[3] Next to this PO related information, we also included the total quantity and total value of all Goods Receipts that are linked to this PO item line. We did the same for the related Invoice Receipts and the total value of all Payments that are associated with this process instance.

On the level of the audit trail entry, a work flow model element also carries unique information. In particular four events are enriched with additional information: 'Change Line', 'IR', 'GR', and 'Pay'. When the event concerns a 'Change Line', we store information about this change: If it was a change of the net value, what was the size of this modification? If not the net value was changed, but another field, for example the delivery address, this field

[3] 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).

Table 3: Attributes of event log

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

contains a modification of zero. The other stored 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).

When the event concerns an 'IR', four attributes are stored. We store the references that contain the (possible) link to the 'GR' and 'Pay', the quantity of the units invoiced, and the credited amount, the value. Notice that these quantities and values only concern this specific Invoice Receipt, as opposed to the Invoice Receipt related attributes of the Process Instance. Those attributes provide summarized information of all Invoice Receipts attached to the Process Instance. Also beware 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 possibly link this Goods Receipt to the associated 'IR' (this is not always possible, only in a specific number of cases), 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. The last event that is provided of attributes is 'Pay'. The value of this payment is captured, as well as the key to create a link to an associated

'IR'.

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

8 Log Inspection

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.[4] The log at hand contains 62.531 events in total and 290 originators participated in the 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' (93.85%), 'Change Line' (5.02%), 'Release', 'IR', 'GR' and 'Sign'. 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'. There are two ways we can obtain this. Or we filter out all process instances that do not end with 'Pay', or we keep the randomly selected process instances, but cut off the audit trail after the last 'Pay' activity of that trail. We have chosen the latter option. This choice is inspired by the fact that if we filter out all PO's that do not end with 'Pay', we might filter out a certain group of PO's that behave in a different manner. We think for example of PO's that are being used over and over again. The audit trail of such a PO may look as follows: *Create PO-Sign-Release-GR-IR-Pay-Change Line-Sign-Release-GR-IR-Pay-Change Line-...* By filtering PO's on 'end task equals 'Pay" we could create a bias on the proportion of this kind of PO's in the total data set. By cutting off the audit trail after the last payment, we preserve the original representation of PO behavior. As said before, we kept the process instances randomly selected, but left out all the audit trail entries after the last payment because 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 61.562 audit trail entries and 285 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.

[4] '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	11,157	11,157	17.842%	18.123%
Release	10,651	10,471	17.033%	17.009%
IR	10,648	10,608	17.028%	17.231%
Create PO	10,000	10,000	15.992%	16.244%
Sign	9,794	9,616	15.663%	15.62%
GR	5,235	5,213	8.372%	8.468%
Change Line	5,045	4,497	8.068%	7.305%

For getting a first glance, the *Fuzzy Miner* (a plugin in the tool ProM) is used on the log, shown in Figure 2. The thickness of an arc indicates the frequency, which reveals *Create PO-Sign-Release-IR-Pay* as most frequent path. This is corresponding to the designed process model. Also the side paths are perfectly explicable. The digress onto *Change Line* and the use of a Goods Receipt before the Invoice Receipt are part of the designed model. Also the path of having a Goods Receipt after a payment is easy to understand in the light of a split delivery.

9 Control Flow Analysis

9.1 Uncovering the core process

The third step of the followed process diagnostics methodology is analyzing the control flow. In a first part, we wish to uncover the core process that is embedded in the event log, and find confirmation that the business process functions in a way that corresponds to the designed model. When using the *Performance Sequence Analysis* plugin of ProM, we have a view on the patterns followed in this log. The analysis reveals 161 patterns. 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. However, five or seven patterns suffice for covering respectively 82% and 90% of the entire log (see Table 5). Looking at these patterns with the domain expert tells us already that all these patterns are completely according Epsilon's procedures. However, to discover a process model that covers the run of the mill, it is necessary to filter out the unfrequent patterns. That is why, in this section, we will only use the first five patterns (describing 82% of the log) for further analysis. This way we can extract a process model from the event log that describes the overall process. This model will in turn be compared with the designed process model, in order to assure the process in general is executed as desired.

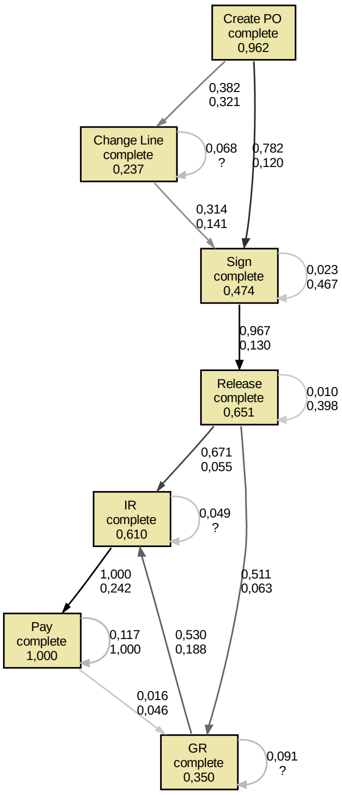


Figure 2: Fuzzy Miner result

Table 5: Top 7 of most occurring sequences

		Occurrences		Total	Throughput Time (days)			
Pattern	Sequence	#	%	%	Average	Min	Max	St.dev.
0	*Create PO - Sign - Release - IR - Pay*	3,066	30.7%	31%	16.75	3	176	9.59
1	*Create PO - Sign - Release - GR - IR - Pay*	2,528	25.3%	56%	34.77	2	327	26.11
2	*Create PO - Change Line - Sign - Release - GR - IR - Pay*	1.393	13.9%	70%	29.74	4	328	36.46
3	*Create PO - Change Line - Sign - Release - IR - Pay*	633	6.3%	76%	25.28	3	241	27.15
4	*Create PO - Release - Change Line - IR - Pay*	599	6.0%	82%	68.88	4	264	39.23
5	*Create PO - Sign - Release - Change Line - IR - Pay*	546	5.5%	88%	21.4	9	299	16.2
6	*Create PO - Release - IR - Pay*	232	2.3%	90%	20.04	2	197	24.16

Table 6: Most unfrequent sequences

Pattern(s)	Occurrences per pattern	Total occurrences	Representation of log
29 - 30	10	20	0.2%
31 - 35	9	45	0.5%
36	8	8	0.1%
37	7	7	0.1%
38	6	6	0.1%
39 - 43	5	25	0.3%
44 - 57	4	56	0.6%
58 - 66	3	21	0.2%
67 - 91	2	50	0.5%
92 - 160	1	69	0.7%
		307	**3.1%**

Taking the selection of the log with only patterns 0 till 4 (8,219 cases) and applying the *Final State Machine* (FSM) miner, results in the process model depicted in Figure 3. Running a conformance check reveals that 8,000 cases, or 80% of the total log, is covered by this process model. This result is used as a feedback to the domain experts. It was concluded that the general outlines of the process are clearly coming forward in the event log. This is seen as a reassuring start.

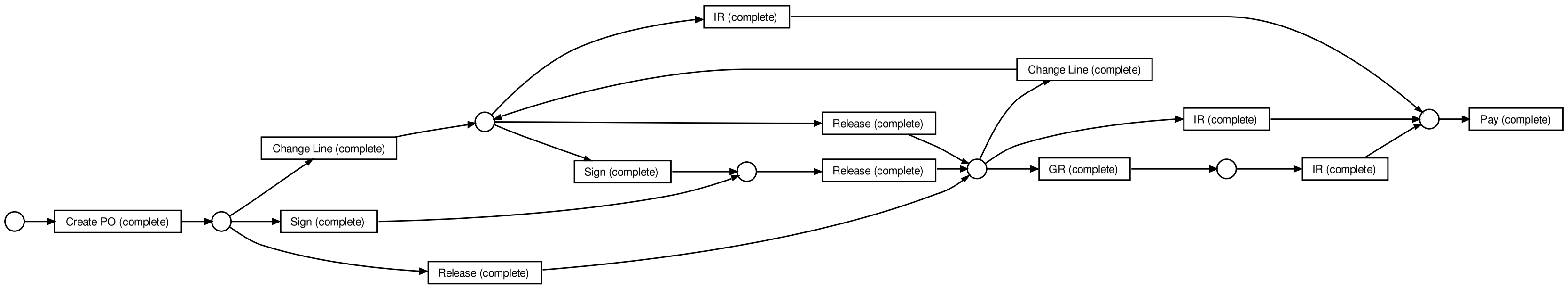


Figure 3: FSM Miner result

9.2 Exposing less frequent flows

Another contribution the control flow analysis can provide, is to use the complete event log (i.e. without selecting only the frequent patterns) and to have a look at the resulting flows when lower thresholds are used. Lowering the threshold settings will result in a graph with more edges, exposing flows that are less frequently followed. This is a nice and convenient way (visual) of looking at the most important unfrequent paths. Turning back to the application of the *Fuzzy Miner*, we change the settings in such a manner more flows become apparent. Concretely, we change the 'Cutoff' edge filter to the values 0.70 and 0.85. These different settings indeed result in models with more edges. Elevating the 'Cutoff' to 0.70 (compared to the default setting of 0.20) revealed two extra flows: 'Create PO → Release' and 'Sign → GR'. Elevating the 'Cutoff' further to 0.85 (depicted in Figure 4) revealed even four more extra flows (on top of the other two):

- Create PO → GR
- Release → Pay
- Sign → IR
- GR → Change Line

Before discussing the extra six flows, visible at the graph in Figure 4, an important aspect of interpreting these results has to be highlighted. The arcs from one event to another in a resulting graph of the Fuzzy Miner, need to be seen in an AND/OR relationship, which is not visible at this output graph. This means that for instance an arc from activity A to activity B does not per definition mean that B directly follows A. Perhaps this arc should be interpreted along with another arc, from activity A to activity C. The two flows 'A → B' and 'A → C' *may* represent an AND (or OR) relationship (after A, B and/or C follow) without having B per definition directly after A, the same for C. So looking at the Fuzzy Miner result gives us ideas of extra flows, but deducing direct flows between one activity and another, needs to be explicitly checked.

In the next paragraphs the six extra flows are discussed with the domain experts and if necessary explicitly checked. Two flows are very normal: 'Create PO → Release' and 'GR → Change Line'. A 'Change Line' can occur at every stage of the process and the fact that the PO is not first signed, before it is released is a realistic possibility. However, there are certain conditions attached to leaving out the 'Sign'. These cab be verified in a later stage.

The flows 'Create PO → GR', 'Sign → GR' and 'Sign → IR' each have the same problem. A release is a prerequisite for ordering goods at a supplier (hence the name). Normally speaking, only after placing an order at

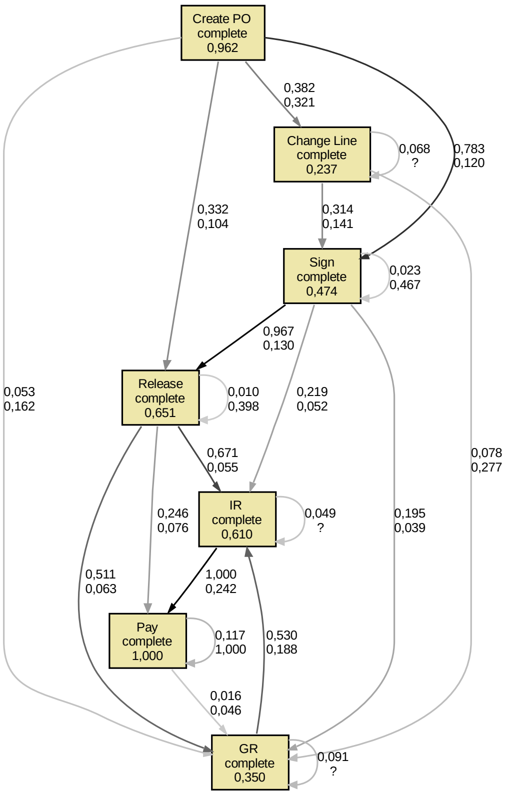


Figure 4: Fuzzy Miner result with 'Cutoff'=0.85

a supplier, a Goods Receipt or an Invoice Receipt could be received at a purchasing department. Following this train of thought, all three flows are contrary to the designed process, and should also not exist if the SAP settings function as they should. So before looking for explanations or going over to investigation, we need to confirm whether these flows really occur in these specific orders, or that they are part of an AND/OR relationship that takes care of the above mentioned restriction. Herefor we use the *LTL-Checker* plugin in ProM and test whether 'Eventually activity A next B' with A and B being the events in question we want to check. The LTL checks reveal that out of the 10.000 process instances, none showed the direct flow of 'Create PO → GR', three instances had a direct flow 'Sign → GR' and again none had a flow 'Sign → IR'.

We take the three process instances with the flow 'Sign → GR' under investigation. The first one shows a pattern of *Create PO* - *Sign* - *Release* - *Sign* - *GR* - *Release* . So a release has been taken place before the Goods Receipt is entered into the system, confirming the SAP control settings. Because the events 'Sign' and 'Release' are both on the header level of a PO and hence not per definition linked to the process instance (only one line item of a PO), it could be that the 'GR' in this case fell in between a *Sign - Release* flow, triggered by another line item. The other two process instances we looked into showed the same situation.

The last flow, 'Release → Pay', raises the question whether for these payments an according invoice is received. Normally, each 'Pay' should be preceded by an 'IR'. Again we start with checking whether there exists a direct flow of 'Release → Pay' for our process instances. We check the same formula 'Eventually activity A next B' with A and B being 'Release' and 'Pay'. There are 55 instances (out of the 10.000) showing this direct flow. There are two possible scenarios for this flow: (1) the 'IR' has taken place before 'Release'. This can again be explained as the 'Sign → GR' flow: a *Sign - Release* flow, triggered by another line item, popped in between an *IR - Pay* flow of this process instance. Or (2), there is no 'IR' related to this 'Pay'. This condition can be tested and looked into later, at the verification step.

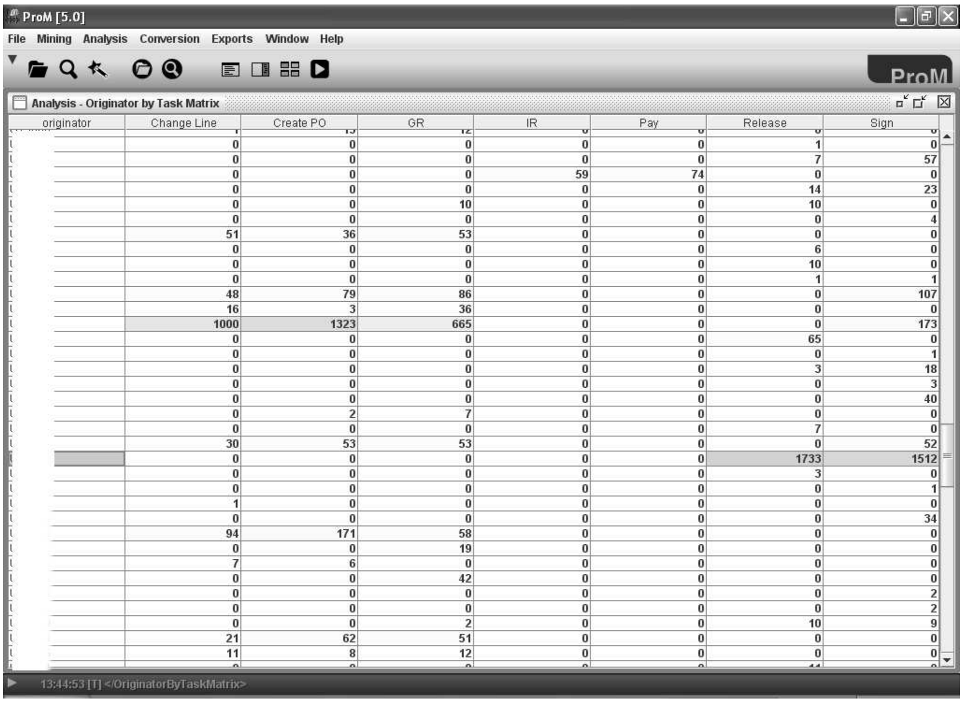
10 Performance Analysis

At the phase of performance analysis, questions like "Are there any bottlenecks in the process?" are answered. (Bozkaya et al., 2008) In this phase the average and maximum throughput times of cases are looked into and analyzed. Although this 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 is why we do not include this fourth phase in this case study.

11 Role Analysis

At the fifth phase of process diagnostics, role analysis, the roles in a process are analyzed. A role should be seen as a person (in this case study)that is involved in the process, by executing activities of that process. Role analysis attempts to answer questions like "Who executes what activities?" and "Who is working with whom?". (Bozkaya et al., 2008) In this phase, it is interesting to check on the efficiency of the segregation of duty.

The segregation of duty is a principle to reduce potential damage from the actions of one employee. (Elsas, 2008) Therefore it is hindered that one single employee has control over a critical combination of business transactions, such as there are for example a 'Sign' and a 'Release' authority in one single purchase. By looking at the role-activity matrix, we can have a first look whether a person executing the activity 'Sign', also executes the activity 'Release'. A print screen of a part of the matrix can be found in Figure 5. At this screen we find for instance one originator that executed 1.733 times a release, and also signed 1.512 PO's. This is the most extreme case of the event log, but other originators also combine these two tasks. This matrix however does not tell us something about whether this should be a problem or not, because if these signs and releases concern different PO's, there is nothing wrong with having both authorities in one person. We find however confirmation for the necessity to investigate this further. These checks require a case perspective of process mining, which brings us to the verification step, the second part of our analysis.



originator	Change Line	Create PO	GR	IR	Pay	Release	Sign
	0	0	0	0	0	1	0
	0	0	0	0	0	7	57
	0	0	0	59	74	0	0
	0	0	0	0	0	14	23
	0	0	10	0	0	10	0
	0	0	0	0	0	0	4
	51	36	53	0	0	0	0
	0	0	0	0	0	6	0
	0	0	0	0	0	10	0
	0	0	0	0	0	1	1
	48	79	86	0	0	0	107
	16	3	36	0	0	0	0
	1000	1323	665	0	0	0	173
	0	0	0	0	0	65	0
	0	0	0	0	0	0	1
	0	0	0	0	0	3	18
	0	0	0	0	0	0	3
	0	0	0	0	0	0	40
	0	2	7	0	0	0	0
	0	0	0	0	0	7	0
	30	53	53	0	0	0	52
	0	0	0	0	0	1733	1512
	0	0	0	0	0	3	0
	0	0	0	0	0	0	1
	1	0	0	0	0	0	0
	0	0	0	0	0	0	34
	94	171	58	0	0	0	0
	0	0	19	0	0	0	0
	7	6	0	0	0	0	0
	0	0	42	0	0	0	0
	0	0	0	0	0	0	2
	0	0	0	0	0	0	2
	0	0	2	0	0	10	9
	21	62	51	0	0	0	0
	11	8	12	0	0	0	0

```
13:44:53 [T] </OriginatorByTaskMatrix>
```

Figure 5: Role-activity matrix

12 Verifying Properties

After mainly looking at the *process perspective*, we turn to the *case perspective* of process mining by the verification of certain properties. In this section we wish to check whether certain conditions hold or if certain internal controls efficiently function. We classify the checks to execute in three categories: checks on the segregation of duty, case specific checks and checks on internal control. For all these checks, we use the *LTL Checker* plugin of ProM. In the following paragraphs the ideas are set forward, but the empirical results still need to be inserted.

12.1 Checks on segregation of duty

As already was confirmed by the role-activity matrix, there is a need to further investigating whether the segregation of duty is respected in this business process. After, together with the domain expert, discussing what controls are interesting for a company to check whether this segregation of duty is efficient, we came to the following three checks:

- Are 'Sign' and 'Release' always executed by two different persons?
- Are 'GR' and 'IR' always entered by two different persons?
- Are 'Release' and 'GR' always executed by two different persons?

When designing the right formula to execute the first check, it is important to take into account that this has to be checked pairwise. If a release takes place, then a 'change line' occurs, and the next sign is performed by the previous releaser, this does not have to pose a problem. As long as the release, following the last sign, is given by another employee, the segregation of duty is intact.

12.2 Case specific checks

Also some very specific checks, related to the company under investigation, can be formulated. For Epsilon for example, there is always a 'Sign' needed before a release can be given, except in two situations:

- The PO document type has a certain 'value A'
and the total PO value is less then 'amount B'.
- The supplier is 'X' and the total PO value is less then 'amount C'.

12.3 Monitoring internal control

Several internal control settings are possible at an ERP environment. Rather than just checking if these settings are in place at a specific moment, we can test the output data whether the internal controls function properly. In this case study, we selected two controls. The first internal control we wish to test is if it is ensured that no payment can occur without having an (approved) invoice entered into the system. The second control checks whether the change of a PO line item appropriately triggers a new sign and/or release.

For the first control, we have to use the attributes 'Reference Pay' and 'Reference IR' of the events 'IR' and 'Pay' respectively and check whether these are tuned to each other. The second control builds upon the attribute 'Net Value' of the process instance and the 'Relative Modification' and 'Modification' attributes of the activity 'Change Line'.

13 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 IFR^2 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.

14 Conclusion

In this paper we present the extended IFR^2 framework, based on a previous work of Jans et al. (2009), 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 monitor 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.

Bozkaya, M., J. Gabriels, and J. M. van der Werf (2008). Process diagnostics: A method based on process mining.

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.

Elsas, P. I. (2008). X-raying segregation of duties: Support to illuminate an enterprises's immunity to solo-fraud. *International Journal of Accounting Information Systems ?*(?), ?

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 (2009). A framework for internal fraud risk reduction at IT integrating business processes: The IFR^2 framework. *International Journal of Digital Accounting Research 8*(14).

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., 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.

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.