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

Mieke Jans, Nadine Lybaert, and Koen Vanhoof

Hasselt University, Agoralaan Building D, 3590 Diepenbeek, Belgium
http://www.uhasselt.be/

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 [1]. The average financial damage to companies subjected to this survey was US$ 2.42 million per company over two years. Participants of a study of the Association of Certified Fraud Examiners (ACFE)[1] estimate a loss of five percent of a company's annual revenues to fraud [2]. 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.

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 and apply a framework for internal fraud risk reduction, where risk reduction stands for both fraud detection and prevention. In a previous paper, we already presented a framework with data mining being the core of that framework. In this paper we complement that framework with a process mining part (see Figure 1). 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.

[1] "The 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)

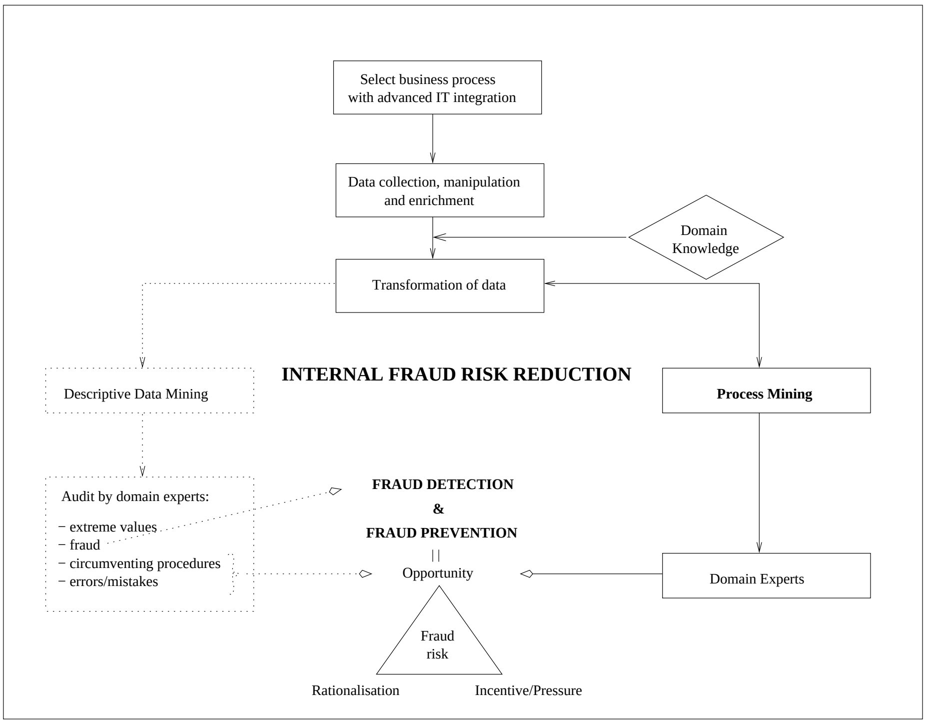

Fig. 1. Extended framework for internal fraud risk reduction, integrating process mining

2 Process Mining as Complement on Internal Control

The emergence of fraud into our economic world did not go unnoticed. In 1985 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)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*.

Where the implementation of internal control up till today is merely system oriented (what are the procedures, what are the associated risks, how are the processes guided etc.), we believe there is an opportunity to complement this framework with a data oriented approach. As complement, we present the use of process mining in the light of internal fraud risk reduction. The two most important contributions to introduce process mining in the fight against internal fraud are: the open mind setting of the investigator and the use of available output data of the organization. The open mind setting refers to the absence of any presuppositions concerning possible frauds before starting process mining. In fact, the objective character is what it makes this unique. The other element, using output data of the organization, implies the highest possible step in internal control, namely checking if the internal control is effectively.

3 Mining the Procurement Process at a Case Company

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 process flow from the creation of purchasing orders (PO's) to the payment of associated invoices was adopted as process under investigation. To this end data was collected and an event log was created. We use *ProM*'s HeuristicsMiner for a first analysis.

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 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 1 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.

Table 1. 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
...				

3.1 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 2, original log). Notice that the event 'Reverse' does not occur in this log.[2] The log at hand contains 65.931 events in total and 297 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' (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 2. 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.

Table 2. 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%

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.

3.2 Mining the Procurement Process with HeuristicsMiner

We start the HeuristicsMiner (a plug-in of *ProM*) with high thresholds, revealing the core process. The expected (and received) result is a graph (not depicted)

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

that is fully explicable by domain experts, without any flows that raise 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.

Table 3. 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

When we loosen our thresholds, we will get a model with far more flows. We set the thresholds summarized in Table 3 with the result depicted in Figure 2. As expected, more flows are present, representing less frequent patterns. This is also indicated by 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.

3.3 Discussion with Domain Experts

The outcomes of the HeuristicsMiner are discussed with the domain experts. Some flows are, as expected, possible to explain away by the data structure or process characteristics. Other flows need to be inspected more closely. This requires another perspective than the process perspective which was used up till

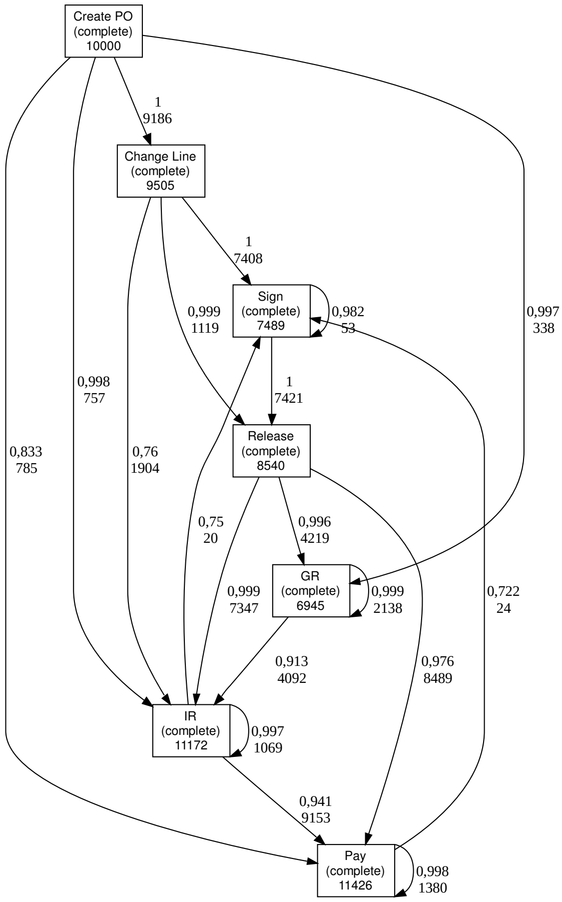

Fig. 2. The result of HeuristicsMiner with lower thresholds (see Table 3)

now, namely a case perspective. For example to check whether each payment is preceded by an invoice receipt is a specific case based check we can run in a next round of investigations. The results of these extra investigations are discussed in a later version of this paper.

4 Conclusion

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.

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.

References

1. PriceWaterhouse&Coopers: Economic crime: People, culture and controls. The 4th biennial Global Economic Crime Surve (2007)
2. Association of Certified Fraud Examiners: 2006 ACFE Report to the nation on occupational fraud and abuse (2006)