PROCESS MINING OF EVENT LOGS IN AUDITING: OPPORTUNITIES AND CHALLENGES

Mieke Jans
Hasselt University
Belgium

Michael Alles
Miklos Vasarhelyi
Rutgers Business School
Newark, NJ, USA

Version: February 15, 2010[1]

[1] Comments are welcome and may be addressed to mieke.jans@uhasselt.be.

PROCESS MINING OF EVENT LOGS IN AUDITING: OPPORTUNITIES AND CHALLENGES

Abstract In this paper we discuss the value that process mining of event logs can provide to internal and external auditors. Process mining aims to extract knowledge from event logs recorded by an information system. What makes an event log such a unique and potentially invaluable resource for auditing is not only that it provides the auditor with more data to analyze, but also because that additional data is recorded automatically and independently of the person whose behavior is the subject of the audit. In other words, an event log helps achieve the classic audit principle of "four eyes", or in modern parlance, act as the equivalent of a surveillance camera, peering over the auditee's shoulder. Until recently, the information contained in event logs was rarely used by auditors. In this paper is considered how process mining can add value to auditing, perhaps even to fundamentally transform it.

Keywords Event logs, process mining, auditing, continuous auditing.

1. Introduction

In this paper we discuss the value that process mining of event logs can provide to internal and external auditors. The Business Process Mining Center describes **Process Mining** in the following terms:[2]

The basic idea of process mining is to extract knowledge from event logs recorded by an information system. Until recently, the information in these event logs was rarely used to analyze the underlying processes. Process mining aims at improving this by providing techniques and tools for discovering process, control, data, organizational, and social structures from event logs. Fuelled by the omnipresence of event logs in transactional information systems… process mining has become a vivid research area.[3]

As the quote indicates, the source of data for process mining is an "event log", also called an "audit trail", which is defined as *"a chronological record of computer system activities which are saved to a file on the system. The file can later be reviewed by the system administrator to identify users' actions on the system or processes which occurred on the system."*[4] Despite the presence of the word "audit" in the term audit trail, it does not refer to auditing in the accounting sense, but to the general potential an event log provides to reconstruct past transactions. In fact, there has been little use made by financial auditors of process mining to examine the data contained in event logs, and the term audit trail is mostly confined to IT and cyber-security circles.[5]

To understand why process mining offers such promise for auditing, we have to recall the way in which accounting used to be done. Until only a few decades ago, accounting meant filling out giant ledger books by hand, a world where accounting could accurately be described as bookkeeping. In that world of manual accounting the data which auditors had to rely on when checking what transactions the client firm had undertaken and how it had accounted for those transactions came entirely from paper based ledgers. These ledgers could further have been

[2] The BPM Center is a collaboration between the Information Systems groups (IS@CS and IS@IEIS) at Eindhoven University of Technology) and the Faculty of Information Technology of Queensland University of Technology. See http://is.tm.tue.nl/staff/wvdaalst/BPMcenter/index.htm.
[3] http://is.tm.tue.nl/staff/wvdaalst/BPMcenter/process%20mining.htm.
[4] http://www.fas.org/irp/congress/1996_hr/s960605a.htm.
[5] Hence, to avoid confusion, we avoid the term "audit trail" in this paper and confine ourselves to "event log". Others terms also used more or less synonymously for event logs are "transaction logs" and "history logs".

supplemented by other pieces of paper, such as written letters of confirmation and invoices hand stamped as "paid".

The problem with such archaic procedures is not just their lack of technology, but also with the limitations the reliance on manual procedures poses for the process of auditing. Hand written ledgers suffer from what we call the "what you see is what you get" or WYSIWYG problem: the only information that the auditor has is what they can literally observe in front of them. Hence, the auditor has no way of verifying who made those ledger entries and when they did so. If those entries have been falsified, erased and overwritten, added to or modified at another date and time by the same or other party, the auditor can detect that only through scrutiny of the physical evidence of the books, in much the same way, and for the same reason, that bankers in that era had to rely on an experienced bank teller's familiarity with the signature of a customer to determine whether a check was genuine or forged.

We describe this constraint facing auditors using the term WYSIWYG—which is more commonly encountered in computer science—deliberately because it captures the fundamental difference that event logs can make to auditing.[6] An event log is far more than a simple "*chronological record of computer system activities"* because a ledger is also a chronological record. Rather, what makes an event log such a unique and potentially invaluable resource for auditing is not only that it provides the auditor with more data to analyze, but also because that additional data is recorded automatically and independently of the person whose behavior is the subject of the audit. In other words, an event log helps achieve the classic audit principle of "four eyes", or in modern parlance, act as the equivalent of a surveillance camera, peering over the auditee's shoulder.[7]

With access to an event log the auditor is no longer restricted to the WYSIWYG ledger of transactions entered by the auditee, but also possesses an independent set of what we describe as "meta-data" about the circumstances under which the auditee made those entries. That meta-data encompasses far more than simple time stamps for transactions, for by taking advantage of that tracking data, the event log enables the auditor to reproduce the history of any given

[6] The origins of the term WYSIWYG actually predates its use in IT. See http://en.wikipedia.org/wiki/WYSIWYG.
[7] We don't consider the possibility where an auditee can access the IT system and alter the event log itself. Any kind of auditing would be vulnerable in such circumstances.

transaction. In other words, the auditor is now able to potentially trace the relationship of that particular entry and its author to all prior recorded transactions by that or related parties.

It is this inherent relationship-creating aspect of event logs that gives process mining its great power and its name: the ability through analysis of event logs to recreate the business processes of the firm. For example, through process mining the auditor has the ability to compare how processes such as purchase to pay were actually conducted as opposed to how they are supposed to be, or to ascertain how the layoff of key workers impacted segregation of duty controls. Such process views of the business are much more difficult to discern from transactional data alone, but feasible when that transactional data is supplemented by the meta-data and history contained in the event lots and made visible by the techniques of process mining.

For the potential of process mining of event logs to be realized, however, the log must first be created, must have integrity, and then the auditor must possess the desire, skills and tools to analyze it for audit-relevant information. It is important to understand that the term event "log" promises more than what is the reality today: the event log is a database of time, process, and originator stamps, not a chronicle that an auditor can simply read as if it were a story. Thus, the creation of an audit usable event log that can be process mined is not a straightforward action, at least not yet. Future information systems, anticipating the value of process mining, may facilitate the extraction of event logs from the firm's ERP databases, but for the moment this step requires considerable manual effort by the auditor—and by researchers examining the value that process mining of event logs can provide to auditing.

It is also worth noting that the definition given above for an audit trail, drawn from 1996 Congressional Hearings on Intelligence and Security, goes on to add the caveat that *"because audit trails take up valuable disk space and can slow the computer system down, many system administrators do not use them or use only limited ones."* While that concern is less pertinent today—with the power of IT systems having increased exponentially thus giving virtually unrestricted ability to store and analyze data—it is not entirely absent.

Process mining cannot be done if the administrators of the client firm's IT systems choose not to create them or to restrict what information is stored in them. Fortunately, many of the Enterprise Resource Planning Systems (ERP), such as SAP®, that forms the technological backbone of most midsize and large businesses today creates event logs automatically and without undue

strain on their computing resources. In such cases, the potential is there for the auditor to follow up and make use of the data that is available in the client firm's event logs, but thus far process mining has not formed part of the procedures of the traditional audit. It is to remedy that deficiency which prompts us to consider how process mining can add value to auditing, perhaps even to fundamentally transform it.

While process mining has not featured prominently in the audit research literature either, using automatically created logs to help auditors determine how accounting and auditing errors arose was proposed by (Alles, et al., 2004). They advocated the creation of: *"a ''black box (BB) log file'' that is a read-only, third-party-controlled record of the actions of auditors, especially in regard to their interactions with management and choice of audit metric and models. Comprehensive and secure in a way that the current system of working papers is not, and accompanied by sophisticated search and analytic algorithms, the log files will serve as an ''audit trail of an audit'', thus enabling an efficient and effective tertiary assurance system."* The inspiration for a BB log came from the black boxes carried on all passenger aircraft and which are intended to assist investigations if the plane crashed, by recording the last 30 minutes of cockpit conversation and instrument settings. In the wake of the failings of Arthur Andersen at Enron and WorldCom, Alles et al (2004) felt that a BB log could perform a similar function for auditing, albeit with records that were longer in duration and far more comprehensive, since the storage of such logs would not be constrained by size and survivability issues as they are on an aircraft.

However, while the BB log was focused on the working of the audit and would require developing a way of automatically recording audit work papers, process mining analyses data created by the auditee firm itself and which is often already being recorded and stored by its ERP systems. Hence, this paper focuses on process mining which provides the "sophisticated search and analytic algorithms" of event logs, rather than the technology needed to create logs in the first place.

Another related paper is (Alles, et al., 2010) that proposes taking advantage of the universality of disaggregate data that modern IT systems now make available to auditors to create sophisticated benchmark models called "continuity equations" of business processes for analytic procedure testing. Despite their emphasis on processes, however, that paper uses only

transactional data to estimate continuity equations and not the broader meta-data contained in event logs.

In order to provide the reader with real life problems and possible solutions, a case study is presented throughout this paper to illustrate both the creation of an event log and its analysis using process mining to provide assurance. This case study includes examples of analyses made possible by process mining. However, because the scope of this paper goes beyond this case study, not all discussed possibilities or opportunities are applied on the case study. The main reason is that not all opportunities are in their current format applicable to real life data. The scope of the paper is to inform the audit profession about current developments that could have a high impact on the profession. Consequently, all relevant developments in process mining at this time are presented, even when they are not yet applicable in their current form. This way, a contribution is also made to the computer science research domain in that an overview of valuable research questions is provided from the auditing point of view.

In the next section of the paper we discuss the emergence of a process view of businesses and process aware information systems and their potential for auditing. Section 3 of the paper then examines how event logs are created and includes a case study showing how an event log was extracted from an ERP database of an actual financial services company, and its content and general structure organized to facilitate process mining. In section 4 we demonstrate with examples and general principles how process mining of event logs can add value to auditing. We do not claim to be exhaustive in covering all possible applications of process mining to auditing, instead aiming to indicate the promise of the methodology. Section 5 adds to that objective by listing the numerous ways in which event log data can be process mined, and which offer different perspectives and opportunities to auditors. Section 6 offers concluding comments.

2. Opportunities for Auditing of Process Aware Information Systems

The emergence of the digital economy has fundamentally altered both the way of running businesses and of performing audits. Most businesses of any significant size today store their data electronically thanks to the maturing of technologies for databases and computer networks. Systems for Enterprise Resource Planning (ERP), Workflow Management (WFM), Customer Relationship Management (CRM), Supply Chain Management (SCM) or Busines-to-Business (B2B) are all consequences of this evolution in technology and business practice. These systems

deal, explicitly (like WFM systems) or implicitly (like ERP or CRM systems) with business processes.

Business process mining or in short **process mining**, is a term subsuming all methods of distilling structured process descriptions from a set of real executions. (van der Aalst, et al., 2004) The idea of mining the process in a context of workflow processes was introduced by Agrawal et al. in 1998 (Agrawal, et al., 1998). Around that time, Datta looked at the discovery of business process models and Cook et al. investigated similar issues in the context of software engineering processes. (Datta, 1998) (Cook, et al., 1998) Herbst was one of the first to tackle the issue of inducing concurrent workflows. (Herbst, 2000) The last decade research in this domain has expanded as several different aspects of business process analyses are investigated by researchers from different disciplines. Process aware information systems (PAIS) induce this domain field even further. Process mining starts from executions logs and typically, PAIS systems register the start and/or completion of activities in an **event log**.

Figure 1 gives a concrete example of the data that can be stored about an invoice in an event log. The left hand side shows the data about the invoice that is entered by the person making the data entry, which is the auditee in the case of auditing. In past times, this is what was hand entered out by the client firm's accountant in its ledger. We will call this 'input data' as it is characterized by the controllable act of a person inputting the data. This input data is the type of data available at the moment this data is stored, such as the invoice number, the posting date, the supplier etc. This is also the type of data that is currently used for auditing and monitoring.[8]

The right hand side of Figure 1 shows the data that is stored in the event log of that same invoice. Formally speaking all input data is also part of the event log, but clearly it is the entries that are unique to the event log that are of particular interest to an auditor because that data is recorded automatically by the system and not inputted by the auditee. It is this meta-data which makes an event log of larger dimension than the set of input data. The instance in the example of the invoice is uniquely identified by the invoice number. Aside from the instance identifier and input data, meta-data on all activities that were conducted on this instance is added. These

[8] This is also referred to as 'account data' (where account-or instance-in this example refers to an invoice). But to avoid confusion we do not use this term since 'account data' is not characterized by the fact that this data is entered manually in the system (or that it is related to financial accounting), but by the fact that it describes the account (or instance), in this case the invoice.

activities, also called *events*, are captured in the log, together with the originator and the timestamp of each activity.

As the visualization of the event log in Figure 1 indicates, the event log's true power is not just as another database of information related to the invoice. That contextual data also enables the auditor to reconstruct the history leading up to that transaction by identifying relationships between this transaction and all other transactions in the database that share parameters with this one. This includes changes to the invoice and the identity of other individuals who "touched" the invoice in any way during its progress through the purchase to pay business process.

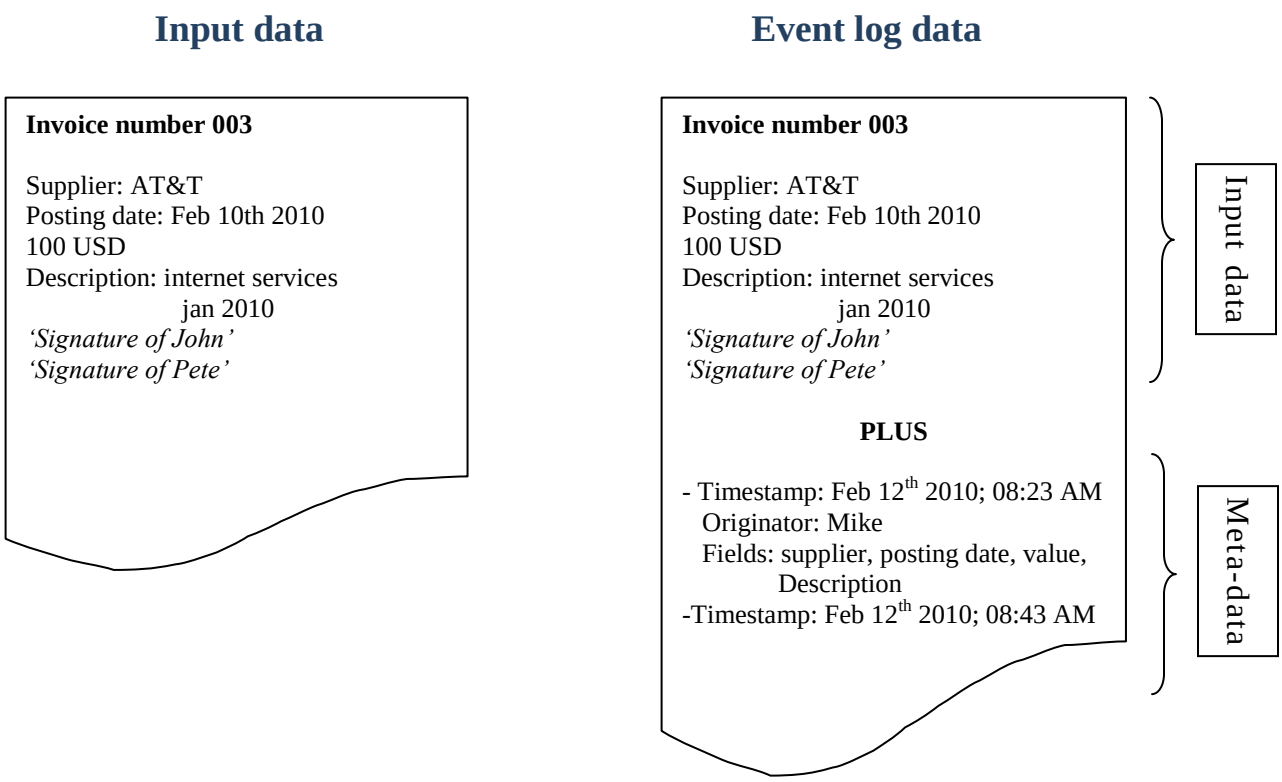


Figure 1: Visualization of Input Data and Event Log Data of an Invoice

For example, where we see a posting date of February 10th in the input data (assuming that this is controllable by the employee), this could differ from the system's timestamp which is recorded in the event log. The meta-data in the event log contains many pieces of information, like at what times which fields are changed by which originators. This meta-data, when combined with the input data, enables the auditor to reconstruct the history of a particular transaction. Thus, the following activities can be reconstructed from the data shown in the event log example of Figure 1:

1. on Feb 12, 8:23 AM: Mike entered invoice No. 3 in system, filling out the supplier (AT&T), posting date (02-10-2010), invoice value (100 USD) and description (internet services Jan 2010)
2. on Feb 12, 8:43 AM: John changed 'Value' from '120USD' to '100USD'
3. on Feb 12, 8:44 AM: John signed invoice No. 3

Figure 1 demonstrates where auditors can expand the scope of its evidence by using process mining of event logs. Where auditors used to have only the paper-written general ledgers with the data on it that the auditee (or anyone else) had written down, there is now event log data available that records everything that happened to that ledger. For example, where the auditor used to rely on the signature at the bottom of a document to verify who was responsible for the document, the event log keeps track of, amongst other things, the person opening/altering/signing the document. This event log is independent of the person entering the input data. Thus the event log can provide meta-data for auditors about the context of the transactional data and enable them to get beyond the WYSIWYG constraint of the input data.

This event log forms the start and also the opportunity to enact process mining. Based on the log of these events, a process model can be distilled to reconstruct the observed behavior. This observation can be used to gain insights about what *actually happens* in a process, in contrast to what people *think* happens in a process. In addition, the actual process can be compared with a predefined process, if one has been specified. Since ERP systems have user configurable settings as is often the case in (when using for instance the so-called reference models in SAP, which are models that describe the preferred way processes should be executed), the deviations from the actual process to the 'preferred' way can be extracted. (Rozinat, et al., 2006) (van der Aalst, et al., 2004) In section 6 of the paper we discuss the different ways in which process mining can analyze the data contained in the event log. But we first turn to a detailed examination of event log creation and structure.

3. Event Log Creation

The previous sections dealt with the opportunities the existence of event log data and their analysis through process mining provides to auditing, but it should be noted that this data is not trivial to extract from the system. The event log data that is captured by the ERP system is potentially vast in magnitude and dispersed over numerous tables (with a certain logic schema depending on the ERP system and company settings). In order to mine the event log and, hence, the process, a rigorous and defensible method of structuring the data needs to be developed.

This configuration should enhance and facilitate the analysis of the event log. In this section the structure of the event log as input for process mining is analyzed and a case study using data extracted from the ERP system of a financial services firm in Europe is used to show how an event log is created.

The aim in creating an event log is to enable process mining of that data. Hence, the scope and power of that process mining is dependent on how comprehensive that event log is in including data on all activities relevant to the process being analyzed. Thus the two critical steps when creating the event log are the identification of activities and the selection of a process instance.

The first preparatory step is for the auditor to develop a holistic understanding of the activities that constitute the process being audited. For example, when the log data is about the process preceding a paid invoice, the underlying activities can be 'create purchase order', 'sign purchase order', 'release purchase order', 'enter Goods Receipt', 'enter Invoice Receipt', and 'pay invoice'. Extra activities could be 'alter purchase order', 'send goods back to supplier', and so forth. Identifying these underlying activities is the first preparation step. Clearly, to some extent activity identification is a judgment call by the auditor, trading off the comprehensiveness of the process understanding versus the desire to reduce the dimensionality of the accompanying event log.

The second step in event log creation is the selection of a *process instance*, or a *case*. A process instance is the subject that undergoes the identified activities. In the case of the paid invoice, the process instance can be the invoice itself, an item line of the invoice, the corresponding purchase order or maybe the item line of the purchase order that triggers the invoice. In order to make the event log, a decision has to be made on the selection of the process instance.

Once the process activities are identified and the process instance is selected, the manipulation of log data into the required structure can be effected.[9] The suggested event log structure is based on four related tables. Figure 2 visualizes the event log structure, which is a relational database with process instances being the cases that undergo activities. Each process instance (PI) has a unique ID, the PI-ID, which is recorded in the PI-ID table. Each activity a process

[9]The term manipulation refers to the restructuring of the data and should not be confused with altering the data itself.

instance is subjected to is called a *unique event entry* (UEE). The UEE table stores all these activities. The activity itself and a unique identifier (UEE-ID) are stored, along with the instance they are belonging to (PI-ID) and two extra fields of meta-data: the timestamp when this activity occurred and by which originator. Meta-data is shown as green in Figure 2.

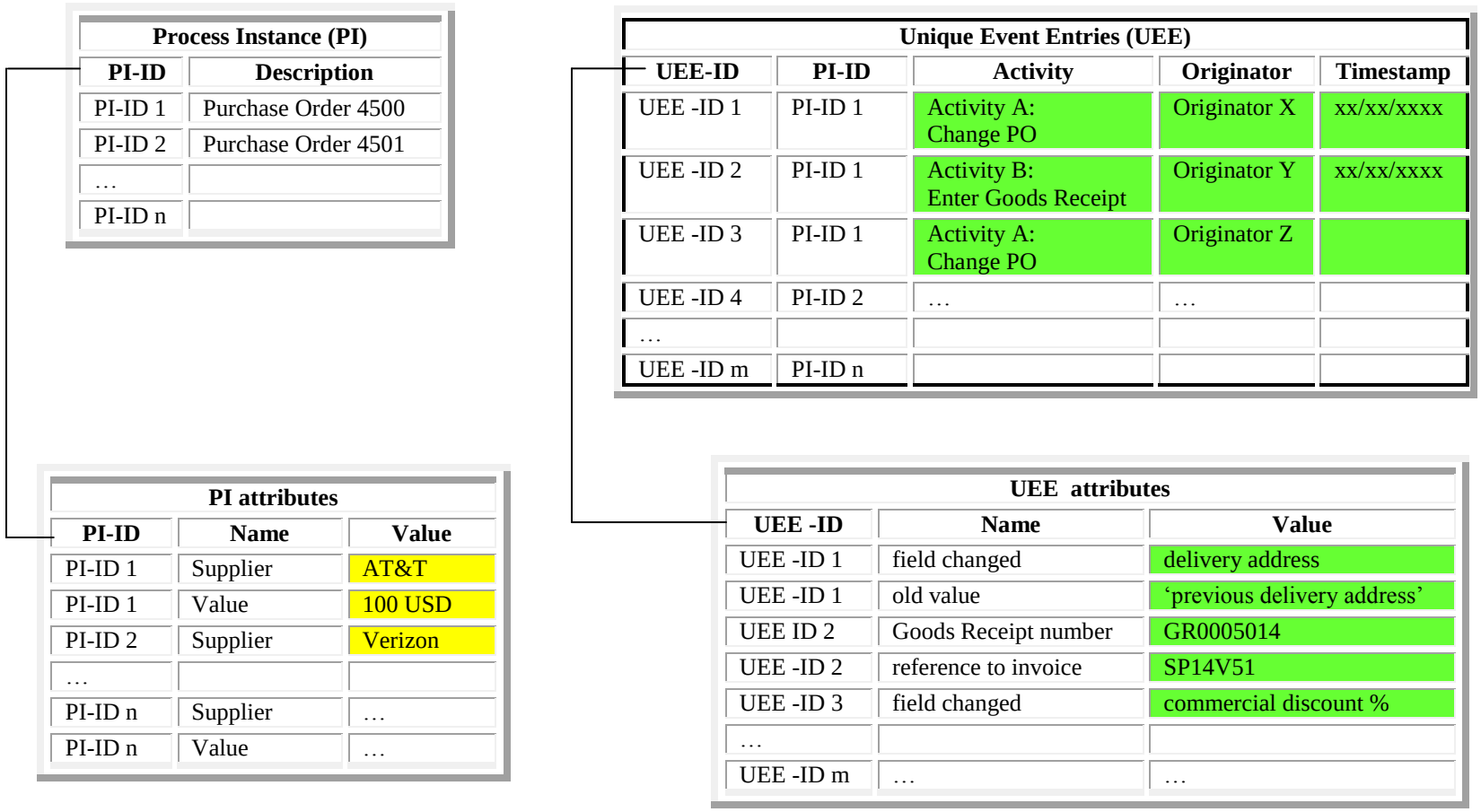
The activity is performed by the auditee and hence intuitively may be seen as input data (for example the activity of changing a PO results indeed in newly entered information), but the storage of the act itself is beyond the control of the auditee and hence should be seen as meta-data. All unique event entries belonging to one process instance constitute the *log trace* of this process instance. In figure 2, PI-ID 1 shows a log trace *Activity A – Activity B – Activity A*, being UEE -ID 1, UEE -ID 2, and UEE -ID 3.

In two separate tables, extra information on both the process instances and the unique event entries is captured. This is done by means of *attributes*, which is a term for the variables that describe the process instances and unique event entries. For each process instance, the same list of attributes is stored. In this example the supplier and the value of each purchase are attributes and examples of input data. Input data is shown as yellow in Figure 2. However, attributes can also have a meta-data nature. With regard to the unique event entries, the attributes that are stored depends on the activity itself. For example: with activity A (Change PO) the attributes 'field changed' and 'old value' (of the field changed) are stored, both meta-data; with activity B (Enter Goods Receipt) attributes 'Goods Receipt number' and 'reference to invoice' are stored. The Goods Receipts number and the reference to invoice are usually meta-data since these linkages are mostly automatically in current ERP systems. Following this assumption, the list of UEE attributes in this example is purely meta-data based. However, a mixture of input data and meta-data could also occur. As is visually clear by looking at the colors in Figure 2, by supplementing the input data (yellow-colored) with meta-data (green-colored), the auditor has far more data to monitor the auditee when using an event log than when analysis is restricted to input data alone.

The decision on which activities and attributes will be captured in the event log is determined by what attributes are available to be logged by the system and the judgment of the auditor as to the scope of the event log. It also has to be kept in mind that while ERP systems can automatically log a very large set of variables, for example, the table settings that control the system,

comprehensive logging is computationally demanding and often the set of variables that are logged is constrained by having some of the logging capability turned off.

To illustrate the processes and judgments required to create an event log, we turn to a case study using data provided by a multinational financial services firm in Europe. The firm agreed to cooperate with this research and provided extracts of their ERP system, which we use to demonstrate how the event log was composed, using its procurement business process as the subject of analysis.



Process Instance (PI)

PI-ID	Description
PI-ID 1	Purchase Order 4500
PI-ID 2	Purchase Order 4501
…	
PI-ID n	

Unique Event Entries (UEE)

UEE-ID	PI-ID	Activity	Originator	Timestamp
UEE -ID 1	PI-ID 1	Activity A: Change PO	Originator X	xx/xx/xxxx
UEE -ID 2	PI-ID 1	Activity B: Enter Goods Receipt	Originator Y	xx/xx/xxxx
UEE -ID 3	PI-ID 1	Activity A: Change PO	Originator Z	
UEE -ID 4	PI-ID 2	…	…	
…				
UEE -ID m	PI-ID n			

PI attributes

PI-ID	Name	Value
PI-ID 1	Supplier	AT&T
PI-ID 1	Value	100 USD
PI-ID 2	Supplier	Verizon
…		
PI-ID n	Supplier	…
PI-ID n	Value	…

UEE attributes

UEE -ID	Name	Value
UEE -ID 1	field changed	delivery address
UEE -ID 1	old value	‘previous delivery address’
UEE ID 2	Goods Receipt number	GR0005014
UEE -ID 2	reference to invoice	SP14V51
UEE -ID 3	field changed	commercial discount %
…		
UEE -ID m	…	…

Figure 2: Clarifying example of event log structure

As discussed above, in order to create an event log two important preparatory steps are necessary: the identification of activities and the selection of a process instance. Based on interviews with the domain experts on the procurement process, the following activities were identified as constituting the procurement process: ‘create a purchase order (PO)’, a possible ‘change of a line item’, ‘sign’, ‘release’, ‘input of the Goods Receipt (GR)’, ‘input of the Invoice Receipt (IR)’, and ‘pay’. The process instance we selected to undergo these activities is a line item of a PO. This choice of process instance is based on the typical structure the data is stored in within SAP, namely: the ‘sign’ and ‘release’ activities are linked to a complete PO instead of to a single line item, but the ultimate payment refers to a PO line item. That is why we chose a line item of a PO as process instance.

Table 1 shows how the table with process instances looks like in this case study. This table contains all the identifiers of the process instances under examination with a brief description.

Process Instance (PI)	
PI-ID	**Description**
450000000110	Purchase Order 4500000001, line 10
450000000120	Purchase Order 4500000001, line 20
…	
450000025110	Purchase Order 4500000251, line 10

Table 1: Exemplary Process Instance table of case company

In Table 2, an example of how the unique event entries look like with the selected activities is shown. As explained before, the data in this table is all meta-data.

Unique Event Entries (UEE)				
UEE -ID	**PI-ID**	**Activity**	**Originator**	**Timestamp**
1	450000000120	Create PO	John	March 21, 2009
2	450000000120	Change Line	John	March 21, 2009
3	450000000120	Sign	Katy	March 30, 2009
4	450000000120	Release	Paul	April 1, 2009
5	450000000120	GR	Sarah	May 4, 2009
6	450000000120	IR	Mike	May 25, 2009
7	450000000120	Pay	Peter	May 30, 2009
8	450000000130	Create PO	John	…
9	…			
…	…			

Table 2: Exemplary unique event entry table of case company

Aside from the information in the table with the unique event entries, an additional table is created with data attributes of each process instance. This table contains information about the parent PO and about the item line itself, due to the double dimensionality of data in SAP. The following information was recorded about the parent PO: the document type, the purchasing group that creates this PO and the supplier involved. The selected information about the process instances concerns the value (in EUR) of the item line (Net value), the unit in which the quantity is expressed (Unit), the amount of ordered units on this line (Quantity PO) and whether or not the GR indicator was flagged. If this indicator is flagged, a choice which in this case is controlled by the employee, the input of a GR is mandatory for the payment of the invoice. If GR is turned off, an invoice can be paid without a receipt. Next to this PO related information, the total quantity and total value of all Goods Receipts that are linked to this PO item line are stored at the attributes table. The same is done for the related Invoice Receipts and the total value of all payments that are associated with this process instance. Notice that in this case study all process instance attributes are input-data. In Table 3 an example of the recorded data attributes of a process instance is shown.

PI attributes		
PI-ID	**Name**	**Value**
450000000120	Document type	DI
450000000120	Purchasing Group	B01
450000000120	Supplier	45781
450000000120	Net value	10,025
450000000120	Unit	EA
450000000120	Quantity PO	1
450000000120	GR indicator	X
450000000120	GR total quantity	1
450000000120	GR total value	10,000
450000000120	IR total quantity	1
450000000120	IR total value	10,000
450000000120	Pay total value	10,000
450000000130	Document type	FO

Table 3: Exemplary PI attributes table of case company

The data attributes in Table 3 concern attributes of the process instances, but also a table with extra attributes of the unique event entries is created. In particular four activities are enriched with additional information: ‘Change Line’, ‘IR’, ‘GR’, and ‘Pay’. If the event concerns a ‘Change Line’, the following information about the change is stored: when it was a change of the net value, what was the absolute size of this modification? If not the net value was changed but another field, for example the delivery address, this attribute stores a modification of zero. The second stored attribute of a ‘Change Line’ gives us, in case of a change in net value, the relative size of the modification (hence a percentage). If the event concerns an ‘IR’, four attributes are stored: the references that contain the (possible) link to the ‘GR’ and ‘Pay’, the quantity of the units invoiced, and the credited amount, called 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 which were overall sums. 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 activity 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 activity that is provided with attributes is 'Pay'. The value of this payment is captured, as well as the key to create a link to an associated 'IR'. Table 4 is an example of how the table with UEE attributes could look like, based on the exemplary unique event entry table in Table 2. In Table 2, only the unique event entries with UEE-ID 2, 5, 6 and 7, representing a 'Change Line', a 'GR', an 'IR', and a 'Pay', would trigger the storage of extra attributes in the UEE attributes table.

UEE attributes		
UEE-ID	**Name**	**Value**
2	Modification	100
2	Relative modification	0.01
5	Reference IR	41358
5	Quantity GR	1
5	Value GR	10,000
6	Reference GR	41358
6	Reference Pay	5100000832
6	Quantity IR	10,000
6	Value IR	10,000
7	Reference IR	5100000832
7	Value Pay	10,000
…		

Table 4: Exemplary UEE attributes table of case company

4. Process Mining as an Audit Tool

We do not intend in this paper to write the last word on the subject of how process mining of event logs can add value to auditing. This is the beginning of the transformation that process mining will bring about in auditing, not the end, and both practitioners and researchers have much to do to explore what process mining can accomplish as a tool in the hands of auditors. Hence, in this section we put forward various examples of what that tool can accomplish and hope that this will inspire and guide future research. Audit researchers should also be aware that leading audit firms are investing in process mining and it is important that the academic community strive to retain the intellectual leadership in the application of process mining to auditing.

In auditing the information that is analyzed to issue an audit opinion is essentially the same as when an audit was performed in a paper-based setting: input data. This data may well now be analyzed in an electronic way by using search queries in contrast to manual examination as before, but that is simply automating an existing manual procedure, not reengineering audit practice to take full advantage of the capabilities of digital businesses.

For example, auditors look at the value of an invoice, whether it is signed or not, etc. Only, instead of looking at source documents this information is captured by searching databases. Aside from the evolvement to a less labor intensive method of working, there are also extra features made possible by this electronic storage which were not within reach before, like calculating maxima, minima, means, comparing information, performing three-way matches and so on. These calculations certainly give extra insights, but again, the type of data which is used as input for these calculations is the same as before: the data the auditee filled out him/herself. However, incorporating a more process based view and embracing the possibilities of process mining the event log would more fully utilize the capability of IT systems to provide continuous assurance.

How can process mining of event logs add value to auditing? Consider the following example: it is a routine assumption in forensic accounting that fraud typically takes place at times where there are fewer other employees around to ask questions, such as at lunchtime. Some forensic accountants thus monitor the firm's ledgers at lunchtime to see who else is on the system, whereas the person committing the fraud might wish to cover their tracks by entering a different time for the fraudulent transaction they are inputting, even assuming that the system requires a time to be entered in the first place. But an event log will store the actual time of this transaction, regardless of what the person fills out, and that information is available to be detected by the auditor without recourse to monitoring at precisely the same time as the data is being entered.

Another example of an anomaly that could be revealed by mining the event log data and which may not be detected so readily through other means is violations of segregation of duty controls. When an invoice has no signature or approval at all, or if the invoice is signed twice by one person instead of by two persons, this would come forward with or

without applying process mining. But employees not following required procedures like first getting approval and only then ordering the goods, would not be apparent by only looking at transactional data alone. When using the timestamps in the event log, however, this circumvention of procedures would become immediately apparent. Of course, such a violation of procedures could be prevented if the firm's ERP system is configured in such a way that this cannot take place, but even if that were the case the auditor would still wish to ensure that in fact such processes in violation of procedure have not taken place.

Indeed, even when the ERP settings are configured in such a way that a release should be given before a goods receipt can be entered into the system, there is no guarantee these settings are always in place, since these are configurable settings (necessary for operational efficiency). Hence, it would be useful for auditors to test whether procedures are followed, even when the ERP settings force employees to follow a particular order of activities. Put differently, reliable tests to check whether the control settings are *always* in place, and not only at the time when tested would be useful from a monitoring point of view.

Another example of how process mining can audit the auditor is by providing a means to check whether the flexibility in the ERP system may be systematically abused for personal gain. Think for instance of the collaboration between a supplier and an employee, systematically changing a purchase order within the sustained margins after a last approval to this purchase. The supplier gets paid more than was agreed upon, and can provide kickbacks to the involved employee. This abuse can be discovered by analyzing event log data since it captures changes to the invoice. Because of the stored originators of activities it is also possible to analyze collaboration between employees and other parties.

An analogy that can be applied to event logs is that of video surveillance used in businesses to safeguard assets and deter crime. While, as with all analogies, the parallels are not perfect, pursuing this line of thought enables us to begin to appreciate the new capabilities that process mining of event logs can provide to auditing—capabilities that are difficult to reproduce through auditing of input data alone.

The major difference between event logs and video surveillance recordings is that storing transactional data is cheap and that it is time and location stamped so that it is feasible for an auditor to search for anomalies and track back history in the event of a detected problem, such as theft or fraud. By contrast, many surveillance cameras in large business facilities, such as museums, are actually non-functioning replicas since there is no possibility of cost effectively monitoring their feeds. Of course, even searching an indexed event log is non-trivial given the magnitude of data they are likely to contain and particularly when the auditor does not have something very specific to search for, and that is why we call for further research into the application of process mining techniques in auditing. But the potential to use event logs forensically is a very real one.

And it is that potential that gives rise to perhaps the most important benefit of creating event logs and process mining them: the resulting deterrence effect. That is the same reason why many businesses purchase replica surveillance cameras and feel that it is worthwhile to install them. Just the chance that someone might be watching a person's behavior can serve to constrain that behavior. How much more effective this deterrence effect would be then if an auditee knew that event logs were indeed being automatically and continuously monitored for anomalies and subject to tests of analytic procedures? The protection surveillance cameras offer, after all, is trivial compared to that promised by process mining of event logs. The latter is potentially the equivalent of a situation when all the installed surveillance cameras are real and their feeds are actually being monitored all the time.

Of course, when considering the value added that process mining of event logs can provide to auditing, an important caveat to be kept in mind is the possibility of types of frauds that stay undetected even when processes are continuously monitored. For example, obviously frauds that leave no electronic trace will not be detectable by process mining, or indeed, with any other IS based detection method. Also, despite the deterrence effect of having and/or mining the event log, the strength of that deterrence effect depends on the personal incentives and opportunities of those parties susceptible to committing fraud.

5. Methods of Process Mining in Auditing

Within process mining there are different approaches to find answers to different questions. These include “how”, “who” and “what” of the business process of relevance to the auditor. These three categories of questions represent three fundamental process mining *perspectives*: the process perspective, the organizational perspective and the case perspective, respectively, representing the subject of the analysis:

The *process perspective* uses the time and location stamps in the event log to help answer the question of “How the process was undertaken?” The process paths revealed by the process analysis can be expressed in visual terms, for example by using Petri Nets or Event-driven Process Chains (EPC). This perspective can be used by auditors to compare the process as it is meant to be performed against how it actually is and thus identify control failures and weaknesses.

The *organizational perspective* uses the data in the 'Originator' field in the event log to help answer the question of “Who was involved in the process?” In this perspective, underlying relations between performers or between performers and tasks can be made visible. The obvious use of this perspective in auditing is in checking segregation of duty controls, either retrospectively, to check existing procedures, or prospectively, to verify integrity of controls when personnel changes are expected (for example, due to layoffs or expansion).

The *case perspective* or the “What happened with this particular transaction?” question focuses on a single case, tracing back its history and relationships of parties that are involved in that history. This will be useful to analyze the separately stored attributes, for example the size of an order or the related supplier. (van der Aalst, et al., 2007)

Another way of classifying process mining is by the approach followed to search for answers to these three questions. Broadly speaking, there are at least five different such *tasks* in process mining: a. process discovery, b. conformance check, c. performance analysis, d. social networks analysis, e. decision mining and verification.

In the remainder of this section each approach is shortly described along with the possible opportunities the task presents for the monitoring task of an auditor. While we have applied some of these tasks to the data in our case study, exploring the potential for

auditing of numerous different techniques of process mining requires a substantial research effort by academics and practitioners. (Jans, 2009; Jans, et al., 2008)

a. Process Discovery

The bulk of business process mining research, and also the first examined aspect of mining workflows, is focused on deducing process models from executed transactions. This is done in terms of relationships of precedence and/or in terms of various routing constructs such as parallelism, synchronization, exclusive choice, and loops. (Folino, et al., 2009) In other words, in process discovery the event log is mined to reveal paths with no a-priori process to guide the discovery process.

A few years after, amongst others, Cook and Wolf's idea to discover models from event-based data, Weijters and van der Aalst (Weijters, et al., 2001) presented a simple heuristic approach to discover workflow models from a transaction log on which they later built further, incorporating issues as noise, time, practical experience and the construction of a tool 'EMiT'. (Cook, et al., 1998; Weijters, et al., 2001; Weijters, et al., 2003; Weijters, et al., 2006; van der Aalst, et al., 2002; van der Aalst, et al., 2004) Other research present additional approaches to the challenge of process discovery, including the use of multi-perspective metrics (Günther, et al.), clustering log traces (Greco, et al., 2006; Alves de Medeiros, et al., 2008; Bose, et al., 2009), and using both structural and non-structural elements (Folino, et al., 2009) amongst others.

The added value of this task in a context of monitoring is that it can both assure that processes take place as is preferred and on the other hand reveal processes which are not supposed to take place. When employees circumvent procedures by not following the preferred following order of activities, this will become visual in the process discovery output, which can be a graph in various business process as shown in Figure 3. An example of a circumvented procedure could be placing an order with a supplier before getting an approval. It is important for a company to get insights in whether or not procedures are followed. When procedures are circumvented, this can be interpreted as a window of opportunity by an employee. This employee might turn this window of opportunity into fraud at a later point when he comes for instance in a situation of

financial distress. For a further discussion on the importance of discovering circumvented procedures in the context of fraud prevention we refer to (Jans, et al., 2009).

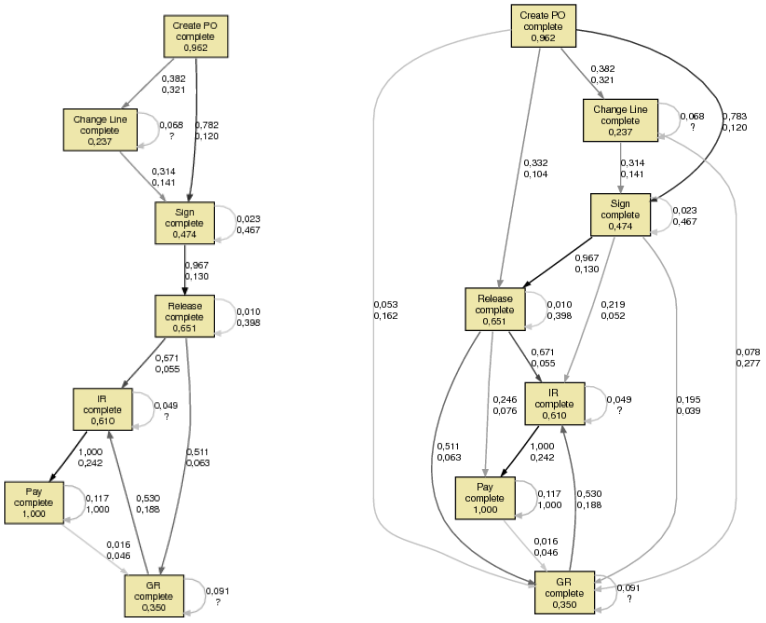


Figure 3: Examples of process discovery output. Left: default settings. Right: less severe settings resulting in more, less frequently followed flows.

Apart from a graphical output, the process discovery can also result in a summary of followed sequences. In Table 4 the most frequent sequences of our case study are presented. A sequence is a log trace a process instance follows, like for example '*Create PO-Sign-Release-IR-Pay'*. This particular instance was followed 3,066 times in the analyzed log, covering almost 31% of the 10,000 mined instances. Apart from a general assurance the event log captures the designed process (8 regular patterns suffice to cover 91% of the log), the frequent sequences can be the input for testing assertions in a later stadium. This is for instance the case when sequences show that shortcuts to the process, allowed when certain conditions are met, are used. For instance, finding that the activity 'Sign' is sometimes passed over (as in pattern 3), is input for testing later on whether the specified conditions for passing this activity over were met in these particular cases.

		Occurrences		***total***
Pattern	**Sequence**	**#**	**%**	**%**

0	Create PO - Sign - Release - IR - Pay	3066	30,7%	31%
1	Create PO - Sign - Release - GR - IR - Pay	2048	20,5%	51%
2	Create PO - Change Line - Sign - Release - GR - IR - Pay	1393	13,9%	65%
3	Create PO - Change Line - Release - IR - Pay	636	6,4%	71%
4	Create PO - Change Line - Sign - Release - IR - Pay	633	6,3%	78%
5	Create PO - Sign - Release - IR - GR - Pay	555	5,6%	83%
6	Create PO - Sign - Release - Change Line - IR - Pay	546	5,5%	89%
7	Create PO - Release - IR - Pay	232	2,3%	91%

Table 4: Most frequent sequences in case study log.

Looking at the *infrequent* sequences on the other hand is also interesting. Log traces that are unique and were not followed by a lot of process instances, can provide the auditor with outliers to subject to manual examination. These process instances form the noise on the process and are for that reason interesting to investigate. This builds on the assumption that cases that behave differently have a greater possibility to represent some fraud than cases that follow the bulk of behavior.

b. Conformance Check

The second process mining task with considerable opportunities for the audit profession is the conformance check. A conformance check tries to answer the following question: "Is there a good match between the recorded events and the model?" The 'model' stands for the process model as is it was made up during the process design phase, hence this technique of process mining the event log assumes an a-priori model guides the mining. This model can be of a *descriptive* nature or of a *prescriptive* nature. (Rozinat, et al., 2006) The descriptive models are the 'preferred' models, affecting the structure of the PAIS, but not imposing it to the system and user without leaving any freedom. Prescriptive models on the other hand describe how the process should be executed (typically used in WFM systems). However, even then the employees sometimes have to deviate from this prescriptive model. Furthermore, in most situations, descriptive models are used by the information system. In that case it is valuable to compare this model to the discovered process model from the event log (by means of the process discovery task). In this context, Rozinat and van der Aalst introduce the suitable term of "business alignment": 'are the real processes and the process model aligned properly?'. (Rozinat, et al., 2006) In their work, they introduce the conformance checker, an algorithm to perform this exercise. The algorithm, fine tuned in (Rozinat, et al., 2008), provides two

metrics in order to check the conformance of a model and an event log: the *fitness* and the *appropriateness* measure, both possible to be calculated by means of three analysis methods (state space analysis, structural analysis, and log replay analysis), each with their own pro's and con's. Future work of these authors will focus on yet other techniques for the metrics, but also on the visualization of non-conformance. For example, other modeling languages than the commonly used Petri Nets will be examined.

From a business perspective point of view, the attributes of unique events could be taken into account in order to confirm or reject conformance. For instance: an invoice of a certain document type only needs to be released, while invoices of other document types need both to be signed and released. It would be interesting if a conformance check could include the value of the attribute 'document type' to check whether the sequences in the event log conforms to the preferred model, depending on their document type.

c. Performance Analysis

Performance analysis techniques such as Process Performance Analysis, Business Performance Analysis and Business Activity Monitoring, focus on the measurement of business process' performances. There are numerous commercial tools available to perform performance analysis of event logs (for example, Aris PPM, Business Objects, HP BPC), along with academic tools like EMiT, developed at Eindhoven University of Technology. Typically, performance analysis creates reports on Key Performance Indicators (KPI) for a business, such as throughput time of a process (providing the minimum, the maximum and average throughput time).

While performance analysis is not a new methodology, extending the techniques to take advantage of the meta-data in event logs is still a work in progress. An example of how to use this type of analysis in the monitoring role of an auditor can be found in identifying cases that go extremely quickly or slowly through the process. Further analyzing these cases with regard to the involved persons may reveal potential malpractices or failures in controls. However, more research is needed on how to exploit the opportunities that performance analysis of event logs provides to auditing.

d. Social Network Analysis

Examining social networks is a process mining approach that is situated in the organizational perspective which uses the event log information that is stored at the originators tab. Apart from drawing various types of reports on who performs how many times which tasks, the organizational perspective provides also the possibility of examining the social networks among employees. In their work, (van der Aalst, et al., 2005) apply methods of sociometry in the context of workflow management. They present metrics and ways to visualize several types of collaboration such as the handover of work between employees and subcontracting. We refer to (van der Aalst, et al., 2005) for further details.

e. Decision Mining and Verification

Decision mining, a next promising process mining task is situated in the third perspective, the case perspective. This perspective mines the event log on the level of process instances. It is suited to test assertions case by case, including the use of the attributes stored in the attributes tables of both process instances and audit trails entries. Decision mining and verification are two tasks that take into account more than only the timestamp and the originator.

Decision mining focuses on decision points in a discovered process model. For instance, after an activity 'Change PO', the process can go two directions: 1) the change triggers a new approval and consequently the next activity is a 'Sign', or 2) the 'Change PO' does not have to do anything with the value of the PO, and the next activity could be, for example, 'Input of the Goods Receipt'. The decision miner task focuses on such decision points in the model and uses machine learning algorithms to uncover the independent attributes at those points. Functioning correctly in the given example, the algorithm could uncover that if the attribute 'relative modification' is beyond 2%, the activity following 'Change PO' is mostly 'Sign'. (Rozinat, et al., 2006) present some work on an algorithm for this task. However, this start needs to be taken to a further level in order to apply this on real life data.

When the output of the previous example is, as stated, 'mostly' a sign after a relative modification beyond 2%, we have a good example of what the verification task serves for. The verification task is the last task to perform in an event log analysis. This task is

also a type of a conformance check, but not on the combination of an event log and a complete model, but on the conformance between an event log and a set of requirements. During the previous steps, several inputs for the verification task may come forward. Consider our example of changes beyond a threshold of 2%. Assume this would normally always be followed by a new sign. In the verification task, the auditor has the opportunity to formulate the assertions he wants to verify, like '*when a 'Change PO' has a relative modification of beyond 2%, is the next activity 'Sign'?*' For each instance, the algorithm checks whether this assertion is correct or not. The output of this task is very straightforward (correct vs. incorrect) and not subjected to interpretable metrics. Consequently, the verification task is the ultimate task for the auditor to check whether all internal control settings functioned correctly during the complete period of time. Running several verification checks on our case company confirmed amongst other things that each payment was preceded by an invoice with the same reference number and that all sing-release duos were performed by distinct persons. However, there were 21 cases where the originator of the 'Release' was the same as the originator of the 'Goods Receipt'. It turned out that this internal control was indeed not configured in an enforcing way. For further information on the used algorithm, the reader is referred to (van der Aalst, et al.).

6. Conclusion

Almost all information systems keep track of some history, stored at the log. A log can be seen on various levels, going from meta-level stored information of the information system itself to a less abstract level of stored information on what occurs on business instances like invoices. Regardless of what level we are looking at, the event log keeps track of everything that takes place in a certain environmental setting and, most importantly, it contains contextual data that is beyond the control of the person entering input data. It is this independent data gathering facility that enables auditors through process mining of event logs to transcend the WYSIWYG problem that is inherent with systems that have input data alone.

Event log data can be compared with a surveillance camera in a store or the black box flight recorder on an aircraft, apart from the fact that the storage capacity of the event log is essentially unconstrained, in contrast to either one of the other two systems. When a company finds an efficient way of using these logs, it can be used as the ultimate monitoring tool, as it tracks every action of an employee without the employee being able to influence the data that the monitoring records. But like video cameras in a store, monitoring comes with a cost. Just as a store may not record every single rack of the store because the store owner is not able to watch all videos, the company needs to take a position on the extent of logging—for instance, while the employee is required to login with a unique identifying password before entering data into the firm's ERP system, typically they would not be required to do so before each and every subsequent keystroke, though in theory their place could have been taken by some other party.

Indeed, a company may turn off the creation of logs on some points, but it could also keep track of everything in especially critical areas. Returning to the surveillance camera analogy, the store owner could hire someone to continuously watch all their tapes in order to deter pilfering, for example, but this would be prohibitively costly. So the store owner has to decide on which products he wants to have a video camera on, and use it only as a means to provide evidence in case theft occurs, rather than as preventive control designed to catch thieves in the act. The fact that employees know the store owner has this evidence afterwards does, however, have a deterrence effect.

A company can achieve an even greater deterrence effect by communicating the existence of the event log to the employees and its ability to comprehensively process mine that data. The available data in the event log is superior to that which can be stored and credibly analyzed in the context of the surveillance camera because of their digital nature and its continuous time, location and originator stamping.

By making the comparison of event log data with surveillance camera tapes, one may think of the considerable storage space needed, and what incentives there are to make all of this worthwhile. The added value of keeping and analyzing an event log lies in the insights into the process that are made possible and in the possibility to detect anomalies otherwise not detectable. Unlike the store owner, by mining the event log one can learn

and monitor the process at hand. New insights about the process preceding the final input (like an invoice being paid) are possible.

Most important of all for auditing, there are anomalies or frauds that cannot be captured by analyzing input data alone. For instance when procedures leave room for flexibility in order to be operationally efficient, this flexibility can be misused, leading to windows of opportunity to commit fraud or leading to additional indirect costs for the company. On the other hand, too narrowly constrained systems are a major drain on corporate client responsiveness. As we have argued in this paper, frauds brought about by such circumvented procedures can be detected by process mining the event log data.

While in this paper we have made the case that process mining of event logs can add value to auditing and provided examples of what that tool can do, we cannot claim to have exhausted all possibilities for how it may do so. The fundamental reason for this is that many of the examples of process mining as applied to auditing that we present in this paper take as their starting point familiar manual audit procedures. In doing so, we are following the standard route in technology adoption, which is to first automate manual processes and only then, once a level of comfort is attained, to reengineer those processes to take full advantage of the capabilities of the technology.

Audit practice as we know it has evolved in a world where the auditor only had access to input data. Those of us who grew up in that world can only imagine how different auditing would have been if the starting point was the meta-data in the event log and auditors were as familiar with the tools of process mining to exploit that data, as they are with such data mining techniques as regression analysis used with input data. Technology moves on, and so, we hope, will auditing.

References

Agrawal R., Gunopulos D. and Leymann F. Mining Process Models from Workflow Logs [Conference] // Sixth International Conference on Extending Deatabas Technology. - 1998. - pp. 469-483.

Alles M. Kogan A., Vasarhelyi M. and Wu, J. The Implications of Unconstrained Data Avaliability and Aggregation Choice on Continuous Auditing Procedures // Unpublished working paper, Rutgers Business School. - 2010.

Alles M., Kogan A. and Vasarhelyi M. Restoring Auditor Credibility: Tertiary Monitoring and Logging of Continuous Assurance Systems [Journal] // International Journal of Accounting Information Systems 5(2). - 2004. - pp. 183-202.
Alves de Medeiros A.K. [et al.] Process Mining based on Clustering: A Quist for Precision [Book Section] // BPM 2007 Workshops, Lecture Notes in Computer Science 4928 / book auth. ter Hofstede A., Benatallah B. and Paik H.-Y.. - Berlin Heidelberg : Springer-Verlag, 2008.
Bose R.P. Jagadeesh Chandra and van der Aalst W. M.P. Context Aware Trace Clustering Towards Improving Process Mining Results [Book Section] // Proceedings of the SIAM International Conference on Data Mining. - 2009.
Cook J.A. and Wolf A.L. Discovering Models of Software Processes from Event-Based Data [Journal] // ACM Transactions on Software Engineering and Methodology 7(3). - 1998. - pp. 215-249.
Datta A. Automating the Discovery of As-Is Business Process Models: Probabilistic and Algorithmic Approaches [Journal] // Information Systems Research 9(3). - 1998. - pp. 275-301.
Folino Francesco [et al.] Discovering Multi-Perspective Process Models [Book Section] // Enterprise Information Systems: 10th International Conference, ICEIS 2008 / book auth. Filipe Joaquim and Cordeiro José. - Berlin Heidelberg : Springer-Verlag, 2009.
Greco Gianluigi [et al.] Discovering Expressive Process Models by Clustering Log Traces [Journal] // IEEE Transactions on Knowledge and Data Engineering 18(8). - 2006. - pp. 1010-1027.
Günther Christian W. and van der Aalst W.M.P. Fuzzy Mining - Adaptive Process Simplification Based on Multi-Perspective Metrics [Journal].
Herbst J. A Machine Learning Approach to Workflow Management [Journal] // LNCS. - Berlin : Springer-Verlag, 2000. - Vol. 1810. - pp. 183-194.
Jans Mieke Internal Fraud Risk Reduction by Data Mining and Process Mining: Framework and Case study [Book]. - Diepenbeek : Hasselt University , 2009. - Vol. PhD Thesis.
Jans Mieke, Lybaert N. and Vanhoof K. Business Process Mining for Internal Fraud Risk Reduction: Results of a Case Study [Conference] // International Research Symposium on Accounting Information Systems. - Paris : [s.n.], 2008.
Jans Mieke, Lybaert Nadine and Vanhoof Koen [Journal] // International Journal of Digital Accounting Research. - 2009. - pp. 1-29.
Rozinat A. and van der Aalst W.M.P. Decision Mining in ProM [Book Section] // Business Process Management, Lecture Notes on Computer Science 4102 / book auth. Dustdar S., Fiadeiro J.L. and Sheth A.. - Berlin Heidelberg : Springer-Verlag, 2006.
Rozinat Anne and van der Aalst W.M.P. Conformance Checking of Processes Based on Monitoring Real Behavior [Journal] // Information Systems 33. - 2008. - pp. 64-95.
Rozinat Anne and van der Aalst W.M.P. Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models [Conference] // BPM 2005 Workshops. - Berlin Heidelberg : Springer-Verlag, 2006. - pp. 163-176.
van der Aalst A.M.P., de Beer H.T. and van Dongen B.F. Process Mining and Verification of Properties: An Approach based on Temporal Logic [Journal].

van der Aalst W.M.P. [et al.] Business Process Mining: An Industrial Application [Journal] // Information Systems 32(5). - 2007. - pp. 713-732.
van der Aalst W.M.P. [et al.] Workflow Mining: A Survey of Issues and Approaches [Journal] // Data and Knowledge Engineering 47(2). - 2003. - pp. 237-267.
van der Aalst W.M.P. and van Dongen B.F. Discovering Workflow Performance Models from Timed Logs [Book Section] // International Conference on Engineering and Deployment of Cooperative Information Systems (EDCIS 2002), Lecture Notes in Computer Science 2480 / book auth. Han Y., Tai S. and Wikarski D.. - Berlin Heidelberg : Springer-Verlag, 2002.
van der Aalst W.M.P., Reijers Hajo A. and Song Minseok Discovering Social Networks from Event Logs [Journal] // Computer Supported Cooperative Work 14. - [s.l.] : Springer, 2005. - pp. 549-593.
van der Aalst W.M.P., Weijters A.J.M.M. and Maruster L. Workflow Mining: Discovering Process Models from Event Logs [Journal] // IEEE Transactions on Knowledge and Data Engineering 16 (9). - 2004. - pp. 1128-1142.
Weijters A.J.M.M. and van der Aalst W.M.P. Process Mining: Discovering Workflow Models from Event-Based Data [Book Section] // Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001) / book auth. Kröse B. [et al.]. - Amsterdam : [s.n.], 2001.
Weijters A.J.M.M. and van der Aalst W.M.P. Rediscovering Workflow Models from Event-Based Data Using Little Thumb [Journal] // Integrated Computer-Aided Engineering 10. - Antwerp : [s.n.], 2003. - pp. 151-162.
Weijters A.J.M.M., van der Aalst W.M.P. and Alves de Medeiros A.K. Process Mining with the HeuristicsMiner Algorithm [Journal] // Beta Working Paper Series. - 2006.