Multilevel Process Mining for Financial Audits

Michael Werner and Nick Gehrke

Abstract—The relevance of business intelligence increases with the growing amount of recorded data. The research on business intelligence has led to a mature set of methods and tools that are used in many application areas, but they are almost absent in the auditing industry. Public accountants face the challenge to audit increasingly complex business processes that process huge amounts of transaction data. Process mining can be used as a business intelligence approach in the context of process audits to exploit this data. We introduce a process mining algorithm to improve such audits. Key requirements for this purpose are the reliability of the mining results, the integration of a data flow perspective and the ability to inspect data from the point of origin to the final output on the financial accounts. The presented algorithm integrates the control flow and data flow perspective. It operates on different abstraction levels to enable the auditor to follow the audit trail. The algorithm creates precise and fitting process models to prevent false negative and false positive audit results, accepts specific unlabeled event logs as input, and considers data relationships for inferring the control flow. It was evaluated by using extensive real world data.

Index Terms—Business Intelligence (BI), financial audits, business process intelligence, process mining, data mining, data analysis, business process modeling, ERP systems, design science research

1 Introduction

Financial audits are important for the smooth functioning of economic markets. They are a control mechanism to prevent the publication of false financial information. The reliability of published financial statements is crucial for stakeholders to direct their decisions. Governments have issued laws and regulations to ensure that companies prepare their financial statements truthfully and fairly. Public accountants act as referees ensuring the adherence to laws, regulations and accounting standards by auditing financial statements. A significant part of a financial audit is the auditing of business processes. The rationale of auditing business processes is the assumption that well-controlled business processes lead to complete and correct postings on the financial accounts. The International Standard on Auditing (ISA) 315 (Revised) mandates the consideration of business processes and related information systems for financial audits: "The auditor shall obtain an understanding of the information system, including the related business processes, relevant to financial reporting (. . .)" [1, p. 7]. It is much more efficient to investigate the structure and embedded controls in business processes than to inspect single transactions. The task of auditing business processes is getting more and more challenging with the increasing integration of information systems for the automation of transaction processing and the growing amount of produced data. Traditional audit procedures like interviews and inspections of selected documents become inefficient or even ineffective in such audit environments [2]. Interview partners may no longer have overall information about a business process if part of it is operated automated and nontransparent in the information system without any human interaction. Furthermore it is questionable if the inspection of relatively few samples is a sufficient audit procedure when millions of transactions are processed.

The current situation in auditing engagements leads to an imbalance. Companies on the one hand use highly integrated information systems to support and automate the operation of business transactions leading to huge amounts of processed data. The auditors on the other hand apply traditional and mainly manual audit procedures that are not appropriate anymore in highly automated environments. The results are inefficient or even ineffective audits.

Problems arising from the increase of processed and available data are not idiosyncratic to the field of financial audits. Data explosion is a general phenomenon that accompanies the expanding availability of digital data storage capacity [3]. The challenges and prospects that arise from the handling and analysis of huge data amounts are currently being discussed and intensively investigated under the umbrella of the term Big Data [4].

Human recipients are only able to handle a certain amount of information before information overload hinders any additional information reception [5, pp. 53–57]. Business intelligence (BI) provides solutions to handle Big Data and to prevent information overload. It is a scientific domain that traditionally researches how digital data can be analyzed and presented to enhance decision making processes [6]. Chen et al. discuss the evolution of business intelligence and analytics (BI&A) and state that it is commonly "referred to as the techniques, technologies, systems, practices, methodologies, and applications that analyze critical business data to help an enterprise better understand its business and market and make timely business decisions." [4, p. 1166].

BI provides mature methods and tools that are used in many application areas. But they are almost absent in the auditing industry. Software tools that assist the auditor are

- *M. Werner is with the Accounting Department, Auckland University of Technology, Auckland 1010, New Zealand. E-mail: michael.werner@aut.ac.nz.*
- *N. Gehrke is with the Department of Computer Science, NORDAKADEMIE, Elmshorn 25337, Germany. E-mail: nick.gehrke@nordakademie.de.*

Manuscript received 1 Aug. 2013; revised 27 Feb. 2015; accepted 1 July 2015. Date of publication 24 Sept. 2015; date of current version 11 Dec. 2015. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TSC.2015.2457907

called computer assisted audit tools (CAAT). Braun and Davis state that CAAT "include any use of technology to assist in the completion of an audit" [7, p. 726]. This is a very broad definition and it would mean that simple project management and documentation tools could also be considered as CAAT. Gehrke [8] describes a more specific approach by illustrating how information on controls that are embedded in information systems can be tested using a prototypical software tool for the purpose of financial audits. But the presented approach only uses a very small part of the data that is stored in information systems and merely focuses on the control perspective. The richest source of information is the recorded transaction data stored in enterprise resource planning (ERP) systems. It exhibit characteristics of Big Data due to its volume and the velocity of data accumulation [4]. This data source remains mostly untouched by existing CAAT.

A promising solution to exploit the available transaction data in financial audits is the use of process mining. Process mining deals with the discovery, monitoring and enhancement of business processes by extracting information from event logs [9]. The usage of process mining would enable an auditor to receive reliable information for the audited business processes very efficiently and effectively. Van der Aalst provided a conceptual model for the integration of process mining into auditing [10] and scholars have applied process mining in the context of internal audits [11]. But process mining tools still have not been widely accepted and applied in the auditing industry.

A main reason is the specific characteristic of the application domain. Process mining algorithms can only be applied usefully if they fit the requirements of the application domain. The research area of process mining has matured during the last decade with the development of powerful general purpose mining algorithms such as heuristic [12], fuzzy [13] or genetic [14] mining algorithms. However, a significant aspect has not been investigated intensively yet. The vast majority of mining algorithms focuses on the control flow perspective. Other perspectives like the data flow perspective are neglected [15]. But the data flow is very important for financial audits. If a process is not compliant the auditor needs to assess the impact on the financial accounts. This requires the integration of financially relevant information. Process mining is commonly used to condense information by abstraction. Process models abstract from the observed behavior of single process executions. It is generally necessary to weigh competing quality criteria against each other when mining process models. Simple process models are normally preferred to complex ones even if this means that these simple models are not as precise and fitting. Auditors have to rely on the correctness of mined models to identify and to assess compliance violations. They therefore require perfectly fitting and highly precise process models. The auditor also has to be able to inspect individual process executions to find out which business transactions caused a violation. Common mining algorithms use labeled event logs. These contain ordered events that are mapped to cases [16]. Financially relevant transaction data stored in ERP systems cannot be used to create such an event log without prior preparation. But specific data relationships in this data can be exploited for process mining purposes [17]. The goal of this paper is the presentation of a process mining algorithm that can be used to improve process audits and that is able to satisfy the specific requirements for the application in financial audits. The presented algorithm combines and visualizes the control flow and the data flow perspective. It accepts unlabeled event log data from ERP systems as input, uses data dependencies to determine the control flow, and produces perfectly fitting and very precise process models. It is especially suitable as a special purpose mining algorithm for financial audits. But the presented approach for combining the control flow and the data flow perspective in a single model might also be valuable for other application domains that exhibit similar requirements.

2 Research Methodology and Structure

The research presented in this paper follows a design science research approach (DSR). This approach was chosen because of the proximity of the investigated research questions to the practical problems and the intention to develop artifacts that have a high contribution and relevance for the application domain. Österle et al. [18] suggest following a research process that consists of the four phases analysis, design, evaluation and diffusion. The structure of this article follows these phases. The analysis phase is represented in the subsequent sections that illustrate the state-of-the-art of related scientific work and discuss the requirements for the development of the presented mining algorithm. The design of the mining algorithm as the core contribution of the paper is presented in section five. Scholars like Hevner et al. [19] stress the importance of research rigor in DSR. We have therefore included an evaluation section that illustrates the results that have been achieved by applying the presented algorithms to extensive real world data. The paper closes with a discussion of the research results, relevant limitations, outlook to future research and a brief summary.

We used different research methods with varying paradigmatic orientation during the research process to reduce the risk of paradigmatic bias and to investigate the research problem from different angles [20]. The results presented in this paper are part of larger research effort that has been carried out by several researchers over the recent years. The development of the presented algorithm bases on the application domain requirements that were identified in empirical studies using expert interviews and surveys [21]. The main research methods that were used to develop the presented algorithm were method engineering [22] and prototyping [23]. Method engineering is an approach for assembling novel methods based on already existing or newly developed method fragments. Research outcomes from prior research have been incorporated as input for the presented research to develop a novel algorithm. Relevant research results that have been considered in the course of method engineering include solutions for the calculation of instance graphs, projections, aggregation graphs [24], and causal matrices [14] as well as previously published research results from the authors concerning the integration of the data perspective into process models [25], case matching [17], complexity reduction of mined models [26], and data dependent control flow inference [27]. The designed algorithm was instantiated in a software prototype to enable

laboratory simulation experiments. Data sets from industry partners were used as input for the prototype to evaluate the proper functioning of the implemented algorithm and to analyze the mining results.

3 Related Work

The research presented in this paper deals with the question of how process mining can be used as a BI approach for financial audits. Process mining is a research domain that has matured over the past decades. It would go beyond the scope of this paper to provide a complete overview of process mining research. Instead we refer to literature that provides a good overview. Tiwari et al. [28] provide a survey of the state-of-the-art and future trends in process mining until the year 2008. Basic and advanced process mining concepts have been comprehensively summarized by van der Aalst [3]. He provides an extensive collection of main research results that have been achieved in recent years and presents an overview of contemporary opportunities and challenges [29].

Compliance checking is a process mining application area that is of particular interest to the research at hand. Compliance can be defined as the adherence to internal or external rules. The main objective of financial audits is to ensure that companies adhere to accounting standards, laws and regulations. Debreceny and Gray [30] suggest the application of data mining methods for analyzing journal entries and provide an extensive case study. Becker et al. [31] have researched the applicability of model-based business process compliance-checking approaches and developed a classification framework. They provide a literature review and distinguish between forward- and backward-compliance checking approaches. Process mining for financial audits is a backward-compliance checking approach because its objective is the disclosure of compliance violations after they have occurred and been recorded in the event log. The application of process mining as a compliance checking approach has already been addressed by different scholars. Alles et al. [32] propose the application of process mining in accounting information systems. Jans et al. highlight opportunities and challenges for using process mining as an audit tool [33] and provide interesting case studies [11], [34]. They focus on the control flow and organizational perspective. An important difference between internal and external audits is the relevance of the data perspective. Müller-Wickop et al. [21] conducted empirical research which shows that internal and external auditors do not necessarily share the same perspective on the importance of different application domain constructs. We will discuss the requirements needed to use process mining in financial audits in greater detail in the subsequent section. But it is important to mention that the inclusion of the data perspective is a key requirement which has not been considered by prior research. The lion's share of process mining research deals with the discovery of the control flow whereas the integration of the data perspective in process mining has generally not been investigated extensively in the academic community yet, apart from very few scientific publications. This observation is supported by Stocker [35] and de Leoni and van der Aalst [15]. De Leoni and van der Aalst use the data flow perspective to discover rules that explain why instances of the same process follow different execution paths. They introduce variables as net components for the extension of Petri Nets (DPN-nets). Trcka et al. [36] follow a different research question to discover data flow errors but apply a similar approach by using extended workflow nets (WFD-nets). Accorsi and Wonnemann [37] choose a different approach to identify information leaks in process models. They include data objects as colored tokens in Colored Petri Nets (CPN).We follow a similar approach and use CPN to include the data perspective by modeling data objects as colored tokens [25]. This allows us to present the control flow and data flow perspective in a single model.

A fundamental challenge for process mining is the balancing between competing quality criteria [9]. Process mining is generally used to reduce complexity by visual representation and abstraction. A process model represents a set of process executions which are called process instances.[1] Multiple executions of a business process commonly do not occur in exactly the same manner. Variety in the execution leads to differing process instances. Every organization needs flexibility to adapt business activities to changing customer demands and market influences. A certain degree of deviation is therefore neither surprising nor damaging. But variance in the course of execution means that it can get impossible to create a model that unambiguously describes the represented process. This leads to the phenomenon of miss-fitting process models. A model is under-fitting if it allows execution paths in the process model that are not represented in the event log and over-fitting if they do not allow for any additional behavior that is not included in the event log.[2] Rozinat et al. provide a framework for the evaluation of process mining algorithms. They identify four quality criteria for the evaluation: fitness, precision, generalization and structure [38]. Fitness indicates if a model is able to represent all cases in the event log. Precision is the complementary criterion. It indicates if a process model does not allow additional behavior that was not observed in the log. Generalization addresses the capability of a model to express more behavior than recorded in the log. It is generally desirable to create a process model that shows an adequate degree of generalization. Structure refers to the graphical representation of a business process and depends on the graphical components of the target language. Other scholars use the closely related criterion of simplicity instead of structure [9]. The following section will show that contrary to many other application domains financial audits require perfectly fitting and as precise process models as possible. De Medeiros et al. suggest the clustering of cases that exhibit similar traces in the event log.[3] Process models are then generated for each cluster [39]. This approach prevents over-generalization and is very useful because it does not require using a specific mining algorithm. A similar trace clustering

1. The term "process instance" and "case" are used ambiguously among scholars. We refer to "process instances" as real world executions of business processes whereas "cases" represent a record of a process instance in an event log. A case is therefore a purposeful abstraction of a process instance represented as a data record.

2. The concepts of under- and over-fitting process models refer to the definition established in [3]. Under- and over-fitting refer to characteristics of mined process models whereas noise and incompleteness refer to characteristics of the event log.

3. A trace is the recorded sequence of executed activities in a process instance. Every case has a specific trace but different cases can exhibit identical traces.

approach is used by Song et al. [40]. Van der Aalst et al. [41] provide a powerful mining algorithm for balancing between over- and under-fitting by using the theory of regions to create Petri Nets from transition systems. The approaches by de Medeiros et al., Song et al. and van der Aalst et al. are very valuable but they do not consider the data perspective and the idiosyncratic structure of journal entries that form the basis for the event log in financial audits.

4 Requirements

This section describes the requirements for using process mining algorithms in financial audits. A business process consists of activities. Information systems that support or automate the execution of activities create journal entries that are posted to the relevant financial accounts. Fig. 1 illustrates this relationship and provides an example of a simple purchase process that consists of four activities.

Three of the four illustrated activities (B, C, D) create journal entries on different accounts. The process starts with the ordering of goods (A). This activity does not create any journal entry on a financial account because the ordering itself has no impact on the balance sheet or income statement. This changes when the ordered goods are received. The corresponding activity (B) creates two journal entries, a debit posting on the *Raw Materials* account and a credit posting on the *Goods Received / Invoices Received (GR/IR)* account. The next activity (C) clears the open entry posted by activity (B) with a debit posting on the *IR/GR* account and a corresponding credit posting on the *Trade Payables* account. The process finally terminates with the payment of the received invoice (D) that creates a debit posting on the *Trade Payables* and a credit posting on the *Bank Account*. All the postings are stored in the information system that is used to operate the business process.

Müller-Wickop et al. [21], [42] conducted empirical investigations using expert interviews and surveys to identify key concepts and information requirements for process audits. They show that the process flow is a central concept and highly relevant from an audit perspective in practice. To audit a process it is necessary to understand how it is structured and which control activities are included that safeguard the correct processing of transactions. This requirement can be satisfied by modeling the control flow perspective. Two further important concepts for external auditors are financial statements and materiality[4]. Knowledge about the interaction among activities alone is not sufficient. For the auditor it is necessary to understand how the activities relate to the financial accounts as illustrated in Fig. 1.

Only those business transactions are inspected in a financial audit that can have a material effect on the financial statements. It is therefore necessary to receive information on the value flow that is created by the audited business process to decide if it needs to be audited from a materiality perspective or if it can be neglected.

Another critical requirement in financial audits is the preservation of the audit trail. The audit trail is a

4. Materiality is defined in ISA 320: "Misstatements, including omissions, are considered to be material if they, individually or in the aggregate, could reasonably be expected to influence the economic decisions of users taken on the basis of the financial statements." [43].

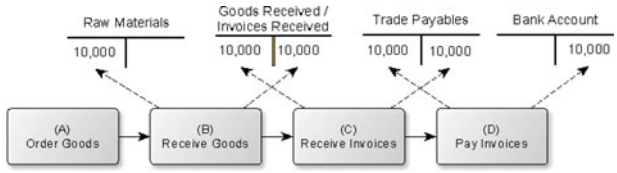

Fig. 1. Simple purchase process.

fundamental concept in financial accounting. It is a path in an information system that allows tracing a transaction from the point of origin to the final output. It is used to verify the accuracy and validity of journal entries [44]. Translating this requirement into the context of process mining implies that a mining algorithm may not alter the original data during the mining process. A suitable mining algorithm must further be able to present the unchanged source data to the auditor for investigation purposes. But on the other hand the mining algorithm should also be able to present information at an adequate abstraction level to provide an overview of the control and data flow as discussed before.

If process mining is used in financial audits the generated models are used to discover incompliant behavior. The provided process models should therefore be as precise and fitting as possible. If the produced process models are overfitting certain behavior recorded in the event log is not represented in the process model. The auditor would therefore assume that no incompliant behavior has occurred. But in reality it is just not represented in the miss-fitting process model which would eventually lead to false positive audit results. If the process model is too general, process behavior is illustrated that actually did not occur. This would lead to false negative audit results and unnecessary investigations by the auditor. The quality criteria identified by Rozinat et al. [38] are useful to express the requirement of accurate process models in terms that are applicable for the process mining research domain. Simplicity of a process model is a preferred characteristic but it is not a key requirement in financial audits. Auditors currently spent weeks trying to understand a business process with the use of traditional audit procedures and by reviewing hundreds of documents. It is therefore acceptable if process models are complex and in extreme cases only comprehensible to experts. Nevertheless a mining algorithm should be able to deliver process models as simple as possible. Generalization should be minimized and precision maximized to prevent false negative compliance testing results. Process models should be perfectly fitting to possibly represent all recorded behavior and to prevent that incompliant behavior that actually occurred remains undetected and therefore reduce the audit effectiveness. Different metrics can be used to measure fitness (*completeness* [45], $PF_{complete}$ [14], *fitness* (*f*) [46], *parsing measure (PM)* and *continuous parsing measure (CPM)* [12]. The metrics *completeness* and *PM* calculate the percentage of traces in the log that can be replayed by the model. The other three metrics consider both traces and tasks in a model. A process model has a perfect fitness if the metrics have a value of one. The precision dimension can also be measured by using different metrics (*soundness* [45], *behavioral appropriateness* [46], and *behavioral precision* [14]). A perfect precision is reached if the relevant metrics also take on the value of one.

5 Multilevel Process Mining

5.1 Mining Algorithm

The requirements discussed in the previous section are partially conflictive. Listing 1 shows an abstract and simplified version of the mining algorithm that was developed based on the identified requirements. A fundamental aspect of the presented solution is to satisfy different requirements at different abstraction levels. The algorithm is called Multi-Level-Process-Mining algorithm (MLPM) due to its main feature of being able to produce process models at different abstraction levels.

The MLPM first matches event data to cases (line 1 to 11), then creates instance graphs (line 12 to 21), calculates process instance models (line 22 to 34) and aggregates instance models to process models (line 35 to 45). Each section is discussed in detail in the following subsections.

5.2 Case Mining and Event Log Structure

The mining algorithm uses recorded transaction data from ERP systems as the data source for the mining. This kind of data only has a medium maturity level from a process mining perspective because ERP systems like SAP or JD Edwards do not use a systematic approach to link and store process relevant data [9]. The event log data in such systems is stored in different database tables and needs to be composed in a meaningful manner before it can be used as input for process mining algorithms. Data related to financially relevant business transactions exhibits specific characteristics that can be used for process mining purposes [25]. The execution of financially relevant transactions in an ERP system creates specific data records. Every execution of an activity creates a posting document. A posting document is a data record that represents a journal entry. Each document

LISTING 1 Multilevel Process Mining Algorithm

1. *Mine Cases*
2. D set of all posting document numbers
3. J set of all journal entry item numbers
4. ID set of all case IDs
5. $D_i = \emptyset$ initially empty set of document numbers belonging to case $i \in ID$
6. $J_i = \emptyset$ initially empty set of journal entry item numbers for case $i \in ID$
7. While $D \neq \emptyset$
8. Remove $d \in D$ from D and insert d into D_i
9. Insert all $j \in J$ posted by d into J_i and remove j from J
10. Insert all $d \in D$ that cleared $j \in J_i$ into D_i and remove d from D
11. Repeat 9. and 10. for all $d \in D_i$ and $j \in J_i$
12. *Reconstruct Instance Graphs*
13. $IG = \emptyset$ initially empty set of instances graphs
14. IG_i instance graph (N_i, E_i, L_i, l_i) for case i with the set of nodes
15. $N_i \neq \emptyset$, $E_i \subseteq N_i \times N_i$ is the set of arcs, L_i the set of task labels
16. and $l_i: N_i \rightarrow L_i$ is a labeling function mapping nodes onto L_i
17. For all $i \in ID$
18. Create $n \in N_i$ for each $d \in D_i$ with $l_i(n) =$ transaction code of d
19. Create $e(n_j, n_k) \in E_i$ for each d_j and $d_k \in D_i$ if d_k cleared an item
20. $j \in J_i$ that was posted by d_j
21. Insert IG_i into IG
22. *Reconstruct Instance Models*
23. $IM = \emptyset$ initially empty set of instances models
24. IM_i instance model $(T_i, P_i, A_i, \sum_i, V_i C_i G_i, E_i, I_i)$ for case $i \in ID$
25. For all $i \in ID$
26. Set $T_i = N_i$
27. For each $e(n_j, n_k) \in E_i$ create $p \in P_i$, $a(t_j, p)$, $a(p, t_k)$
28. For each $d \in D_i$
29. Create $p \in P_i$ for each $j \in J_i$ that was posted by d
30. Create $a(t, p) \in A_i$ for each $j \in J_i$ that was posted by d and
31. $a(t, p)$, $a(p, t) \in A_i$ for each $j \in J_i$ that was cleared by d
32. Aggregate all places p_k and $p_j \in P_i$ if $C_i(p_k) = C_i(p_j)$
33. Aggregate all transitions t_k and $t_j \in T_i$ if $l_i(t_k) = l_i(t_j)$
34. Insert IM_i into IM
35. *Mine Process Models*
36. $PM = \emptyset$ initially empty set of process models
37. Compute causal matrix $CM(IM_i)$ for all $i \in ID$
38. While $IM \neq \emptyset$
39. Remove IM_j from IM and insert IM_j into PM
40. For each $IM_k \in IM$
41. If $CM(IM_j) = CM(IM_k)$
42. Merge IM_j and IM_k with $T_{jk} = T_j \cup T_k$, $P_{jk} = P_j \cup P_k$, $A_{jk} = A_j \cup A_k$, $\Sigma_{jk} = \sum_j \cup \sum_k$
43. Aggregate all places p_l and $p_m \in P_{jk}$ if $C_{jk}(p_l) = C_{jk}(p_m)$
44. Aggregate all t_l and $t_m \in T_{jk}$ if $l_{jk}(t_l) = l_{jk}(t_m)$
45. Remove IM_k from IM

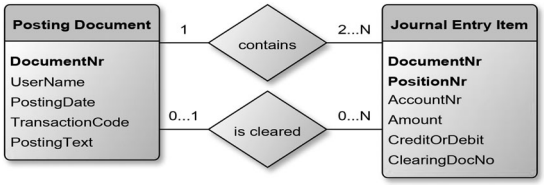

Fig. 2. Entity-relationship-model for accounting data structure.

contains at least two journal entry items. This connection is illustrated by the *'contains'* relationship in the ER model presented in Fig. 2. Journal entries that follow an open-item-accounting principle are linked to each other if they belong to the same process instance. If a process instance is terminated each open journal entry item has been cleared by another posting document.[5] This dependency is modeled in Fig. 2 via the *'is cleared'* relationship.

It is possible to exploit this data chain for each process instance. The procedure is illustrated in line 1 to 11 in Listing 1. The algorithm maps event log entries to cases. It starts with a document number and mines all related journal entry items by using the *'contains'* relationship illustrated in Fig. 2. It then searches for all document numbers that have cleared these items by using the *'is cleared'* relationship. All posted items for every clearing document are then searched in further iterations. The loop terminates when all documents and items that belong to the same process instance have been found. The loop itself is repeated until all case IDs have been mined. The outcome is an event log $\cup_{i=1}^{n}\{D_i, J_i\}$ with $D_i = \{Document\ Nr_{d1}, \ldots, Document\ Nr_{dm}\}\ \forall\ posting\ documents\ d_1 \ldots d_m \overset{match}{\Longrightarrow} i$ and $J_i = \{ItemNr_{j1}, \ldots, \quad ItemNr_{jm}\} \forall items\ j_1 \ldots j_m \overset{match}{\Longrightarrow} i$. The event log also includes the data attributes associated to each posting document and journal entry item that are listed as entity attributes in Fig. 2. In contrast to traditional event logs the events belonging to a single case do not follow a strict linear order. The causal dependencies between the events have to be determined in a separate step.

5.3 Instance Graphs and Abstraction Levels

The application of process mining for financial audits requires algorithms that enable the investigation of a financially relevant transaction from its point of origin to the posting on the financial accounts. On the other hand an auditor should be able to receive an overview to get an all-encompassing understanding of the process structure and its effect on the financial accounts. Both requirements can be met by using different abstraction levels. Scholars commonly distinguish between four levels of horizontal abstraction in business process management [47]. The instance level represents tangible entities that are involved in business processes which include executed activities, resources, concrete data values etc. A set of similar business processes are represented as business process models on the model level. A model is expressed using constructs of a meta-model. These can themselves again be defined at the meta-meta-model level. Process mining algorithms commonly operate on the model level.

5. This principle is also illustrated in Fig. 1. Activity (B) creates an open credit posting on the *IR/GR* account that is cleared by a debit posting on the same account from (C). The open item posted on the *Trade Payables* account from (C) is cleared by (D). The process terminates with a posting on the *Bank Account*. This account is not enabled for open-item-accounting and hence all posted items are cleared after the execution of (D).

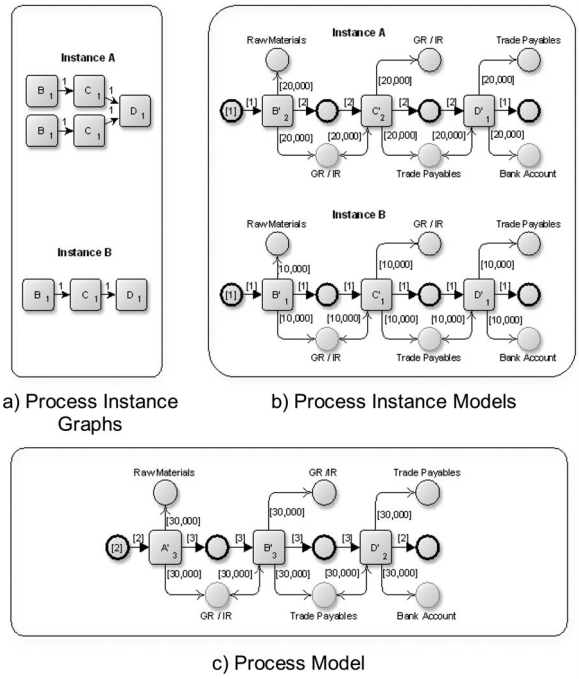

Fig. 3. Abstraction levels in the context of process mining[7].

They generate process models that are abstractions of the individual process instances they represent.[6] Fig. 3 illustrates how the different abstraction levels that were used to develop the MLPM relate to each other and what kind of models the algorithm generates. Process instance graphs reside on the lowest level of abstraction. They are graphical representations of the source data from executed and recorded process instances. Fig. 3a shows two instance graphs. They represent two executions of the purchase process illustrated in Fig. 1. Each rectangle represents the execution of an activity and the arcs between two activities denote the causal relationships between executed activities. The numbers indicate how often an activity was executed and how often the path from one activity to another was chosen. Instance A in Fig. 3a shows the distinctive but typical structure of process instances recorded in ERP systems because it has two branches meaning that two invoices (C) for two different received goods (B) were paid by the same payment (D). The process instance model is an abstraction of the instance graph (Fig. 3b). Executions of identical activities are aggregated into activity models but an instance model still represents just a single execution of a business process. A process model is an abstraction of a set of similar process instances (Fig. 3c). The presented process instance and process models also provide information on the involved financial accounts and hence incorporate the data flow perspective (compare Sections 5.4 and 6).

6. An exception is the multi-phase process mining approach that was developed to generate event-driven-process-chains [24]. It operates on the instance and model level. A similar approach is used for the design of the MLPM.

7. The models in Fig. 3a represent simple directed graphs without logical operators. It is not necessary to model choice at the instance graph level because all decisions have already been made and all relationships in Fig. 3a have the semantic of an *AND* split or join. The models presented in Figs. 3b and 3c represent abstractions of the lower level models. The relationships on these level can generally represent *AND*, *OR* and *XOR* splits and joins. An algorithm to transform instance graphs into EPC or Petri Nets is described in [24].

The process model is important for the auditor to get an overview about the structure of the process, its relationships to the financial accounts and to assess the overall materiality of a business process. Process instance models and process instance graphs are useful for following the audit trail and for inspecting individual process instances with respect to involved activities, users, data values etc.

The second part of the mining algorithm ranging from line 12 to 21 creates instance graphs. The algorithm first creates a node for every document number labeled with the transaction code that was used to create the posting document in the ERP system (line 18). Different nodes in the instance graph can carry the same label (compare Fig. 2a) if they were created by the same transaction of the ERP system. The algorithm then infers the causal dependency between activities (line 19 and 20). Traditional mining algorithms rely on the time stamp of events to infer the control flow. This is not suitable for the event log created by the MLPM algorithm. The same posting document can clear journal entry items belonging to different other posting documents leading to parallelism in the event log. Using the temporal ordering of events can lead to intertwining in parallel branches resulting in models that are of little use to the user.[8] The algorithm uses instead a data dependent approach for determining the control flow. Sun and Zhao [49] introduced an approach to derive the control flow by modeling the data flow. The data dependencies of an activity v can be expressed as $\lambda^d{}_v(I^d_v, O_v)$, where I_v is the input for v, and O_v the output. d denotes the type of data dependency. The algorithm analyses in lines 19 and 20 how activities relate to each other based on their data relationships. It checks if activity A has cleared $Item_B$ posted by activity B. If the condition is true a control arc from B to A is inserted. A can only have occurred if B took place before. Otherwise there would have been no $Item_B$ that could have been cleared by A. This is equivalent to a mandatory dependency between A and B denoted as $B \rightarrow_m A$ because of $O_B \cap I^u_A \neq \emptyset$ with $Item_B \in O_B \wedge Item_B \in I^u_A$. The results of these operations are instance graphs in form of directed graphs equivalent to the graph shown in Fig. 3a.

5.4 Instance Models and Colored Petri Nets

The third section of the mining algorithm reconstructs process instance models as CPN. CPN are suitable for the modeling of business processes and offer a formal as well as graphical notation that can even be understood by non-experts [50]. They provide a sound mathematical foundation for the simulation and verification of Petri Net models [51]. The majority of process mining algorithms rely on low-level Petri Nets. An exception is the approach used by Accorsi and Wonnemann [37]. They use CPN and model data objects as colored tokens. We use a similar approach to generate process models that model the control flow and data flow perspective simultaneously in a single model. A Colored Petri Net is formally expressed by the tuple $\text{CPN} = (T, P, A, \Sigma, V, C, G, E, I)$ [51, p. 87], with:

1) T is a finite set of transitions.
2) P is a finite set of places.
3) $A \in P \times T \cup T \times P$ is a set of directed arcs.
4) Σ is a set of non-empty color sets.
5) V is a finite set of typed variables such that $\text{Type}[v] \in \Sigma$ for all variables $v \in V$.
6) $C : P \rightarrow \Sigma$ is a color set function that assigns a color set to each place.
7) $G : T \rightarrow EXPR_V$ is a guard function that assigns a guard to each transition t such that $Type[G(t)] = boolean$.
8) $E : A \rightarrow EXPR_V$ is an arc expression function that assigns an arc expression to each arc a such that $Type[E(a)] = C(p)_{MS}$, where p is the place connected to the arc a.
9) $I : P \rightarrow EXPR_\emptyset$ is an initialization function that assigns an initialization expression to each place p such that $Type[I(p)] = C(p)_{MS}$.

We integrate the data perspective by modeling colored places in the CPN that represent financial accounts [25]. An instance graph produced by section *Reconstruct Instance Graphs* is first transformed into a CPN. The nodes of the instance graph become the transitions of the CPN (line 26).[9] We distinguish between two types of places. *Control places* model the control flow and *account places* model the data flow. Each arc from the instance graph is transformed into a combination of a *control place* and two connecting *control arcs* in the instance model. A source place p_{source} is inserted with arcs $a_j(p_{source}, t_j) \forall\ t_j \in T \wedge \bullet\ t_j = \emptyset$ and a sink place p_{sink} is with arcs $a_k(t_k, p_{sink}) \forall t_k \in T \wedge t_k \bullet = \emptyset$. The data perspective is integrated and visualized by creating *account places* for each journal entry item in the event log (line 29). The color set function C assigns different color sets to places depending on whether they belong to the group of control or account places.[10] The set of color sets Σ includes the color sets for all possible journal entry values, account numbers, account types, credit or debit indicators and execution numbers.

Account places are connected to related transitions (line 30 and 31). A simple arc $a(t, p)$ is inserted from transition t to the place p if the transition posted a journal entry item on the represented financial account. They are referred to as *posting arcs*. Two arcs $a(t, p) \wedge a(p, t)$ are inserted if the transition has cleared an item on the respective account. These arcs have the semantic of testing arcs because they do not consume the tokens on the connected places. Double-headed arcs are used for visualization as a syntactical abbreviation for two arcs $a(t, p) \wedge a(p, t)$. They are called *clearing arcs*. The arc inscriptions are modeled as constants ($V = \{\}$). The arc expression function E assigns to each *posting* and *clearing arc* a set of constants that denote the posted or cleared value, the account type, account number and an indicator whether it is a credit or debit posting. Each *control arc* is assigned a number that indicates how often this represented control path was chosen in the instance.[10] The source place is initialized by the

8. A discussion of this aspect is beyond the scope of this paper but it is illustrated in detail in [27]. The event logs can be converted into linear event logs [48]. But this transformation is accompanied with undesired side-effects like the duplication of events in the log, and it is not able to deal with the data perspective.

9. Guards are not needed and the guard function G is therefore defined as $G(t) = \text{true for all } t \in T$.

10. A detailed description of the color set function C and arc expression function is provided in [25].

initialization function I. The initialization expression for p_{source} generates n tokens in the initial marking $M_0(p)$, one for each connected start transition.

After having integrated the data perspective by creating a CPN it is necessary to aggregate model components to create instance models. The first step is aggregating places representing financial accounts. Two places p_k and p_j can be aggregated if they represent the same account. This is the case if they carry the same color $C(p_k) = C(p_j)$ (line 32). The relationships between the places and connected transitions need to be maintained in respect to the type and inscription of each arc.[11] The second step aggregates the transitions (line 33). The used method for aggregating transitions is based on the algorithm described by van Dongen and van der Aalst [24]. Transitions are aggregated if they carry the same label $l(t_j) = l(t_k)$. The result of the third section of the mining algorithm is a set of instance process models represented as CPN (compare Fig. 3b).

5.5 Process Models and Clustering

The final section of the mining algorithm produces process models (lines 35 to 45). A key requirement for the application of process mining in the context of financial audits is the creation of perfectly fitting and highly precise process models. Some researchers have developed methods to create precise and fitting process models by clustering traces in the event log [39], [40]. Their approaches are not directly applicable because they do not take into account the data perspective and because they are not suitable for the given event log structure. But clustering is an approach that is also used for the creation of process models in the final section of the presented mining algorithm. But instead of clustering traces we cluster and aggregate process instance models. We use causal matrices to identify isomorph process instances. Causal matrices are key components for heuristic [12] and genetic mining algorithms [52, p. 6]. A causal matrix describes the causal relation between activities in a model. An entry in a causal matrix in column j and row k designates that a causal relation exists between the activities j and k. The causal matrices of two instance models are identical if they show exactly the same behavior. The algorithm computes the causal matrices for all instance models (line 37) that were created by the previous operations. It then searches for identical causal matrices (line 41) and aggregates two models if they have the same causal matrix (lines 42 to 44). The aggregation procedures are identical to the procedures used at the instance model level (line 32 and 33). The aggregation procedure is repeated for all instance models until the set of instance models is empty (line 38).

Aggregating process instance models that exhibit identical control flow patterns does not mean that the resulting process model is identical to the source process instance models. The data values in each process instance model are unique and are preserved during the aggregation process. A process model therefore shows the aggregated data values and data flow of all represented process instances. The final outcome of the mining algorithm is a set of process models where each model represents process instances with the same control flow pattern. Each process model provides the aggregate view of the data flow and relationship of activities with the financial accounts (compare Figs. 3c and 4).

6 Mining Results and Evaluation

The aim of the conducted evaluation is to test whether the designed algorithm is able to produce the estimated results and to get insights into the characteristics of mined process models from real world data. The estimated results are models representing information at different abstraction levels that visualize the control and the data flow, the ability to learn models from unlabeled event logs from ERP systems, perfectly fitting models to prevent false positive audits results, and as precise process models as possible to avoid false negative audit results.

An artificial evaluation is appropriate for the research at hand because it is not necessary to observe the behavior in organizational contexts at this stage. We have chosen an artificial laboratory experiment as one of the evaluation methods suggested for this setting by Venable et al. [53]. Such an evaluation requires the existence of instantiated artifacts that can be used for the experiment. The different sections of the mining algorithm were therefore implemented in a software artifact in iterative cycles. The experiment itself was divided into the phases data extraction, mining and results analysis. We used a separate software module for extracting relevant data from ERP systems[12] that can be adjusted to different source systems. We checked the data for first and second order defects [6, pp. 29-32]. Three data sets were used for the evaluation.

We extracted data from the productive SAP systems of three companies operating in the retail, manufacturing and media industry. The MLPM was used to mine all process instance graphs, process instance models and process models for the three data sets. The data set characteristics and mining results are summarized in Table 1.

The models were inspected on a sample basis by observation and comparison with the original event log data. The software yEd—Graph Editor [54] provides powerful automatic layout functionality and is free for use for noncommercial purposes. It uses the GXML and GraphML formats as input and was used in the experiment to graphically represent the process models. The models were further tested for soundness (proper completion, option to complete, absence of dead transitions, and safeness for all places except account places) by using the CPN simulation tool Renew [55].

Fig. 4 shows an example of a mined process model from data set 1. It is modeled as a CPN as specified in Section 5.4. The transitions illustrated as rectangles represent the activities. The mined model represents a purchasing process that was executed by 8 identical process instances. It contains the activities *Post Goods Receipt*, *Enter Incoming Invoice*, *Clear Account*, *Payment* and *Post with Clearing* that were executed in a sequential order. Control places are rendered with a bold black border. Simple arrows between transitions and control

11. A complete description of this procedure is beyond the scope of this paper. It is described in [26].

12. The introduction of this module is beyond the scope of this paper. The relevant data is stored in the tables BKPF and BSEG in the case of SAP ERP systems.

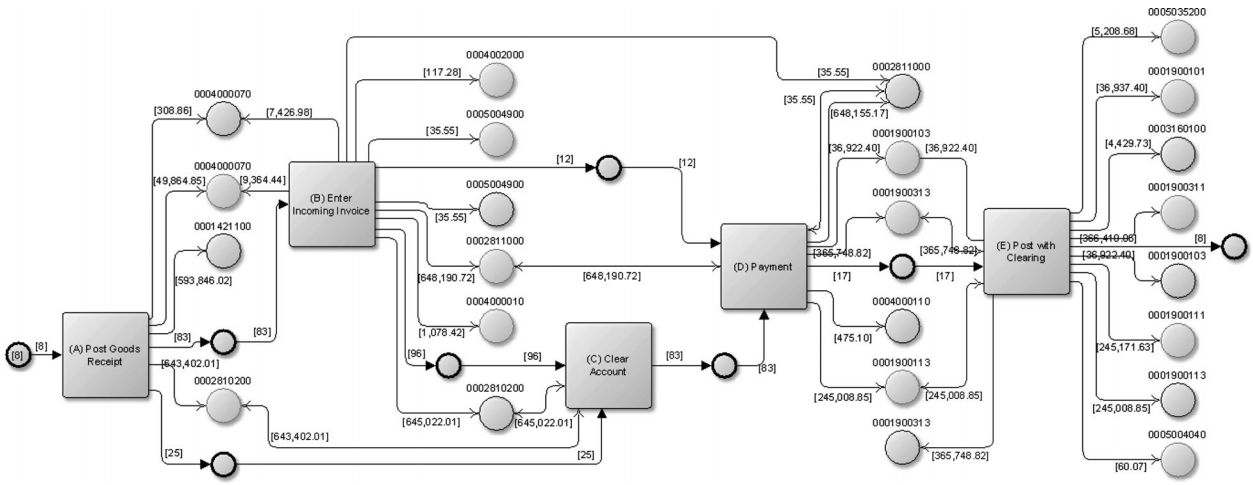

Fig. 4. Example of a mined process model[13].

TABLE 1
Evaluation Data Sets

	Set 1	Set 2	Set 3
Industry	Manufacturing	Media	Retail
Posting Documents	1,764,773	156,604	92,487
Journal Entry Items	7,395,434	559,506	222,901
Process Instances	1,035,805	18,975	40,634
Process Models	841	516	307

places model the control flow and sequence of activities. The inscriptions for these arcs show how often the path was chosen in the represented instances. Account places have a solid or dashed border. The type and color of the border line indicate if the represented account is a balance sheet or profit and loss account and if it represents the debit or credit side.[14] Account places and transitions are connected by dotted arcs. A simple dotted arrow is a posting arc. Double-headed dotted arcs are clearing arcs. Posting and clearing arcs visualize the data flow in the process model. The *Post Goods Receipt* activity for example creates a token with the value of *593,846.02* representing a journal entry item posting on the raw materials account *0001421100*. Another token is created on the account *0002810200*. This is cleared by the subsequent activity of *Clear Account* but without consuming the token on the respective account. The modeled CPN mimics the behavior of the represented process. Transitions create colored tokens on the account places representing the posted journal entries. The control places define the control flow in the model.

The model is perfectly fitting and precise because it can replay all process instances and does not allow any additional behavior. Fitness and precision can be measured using different metrics such as *fitness* (f) and *behavioral appropriateness* (a_B) [46]. These measures can be calculated by replaying cases from the event log and by counting if tokens are missing in the CPN in order to execute the simulation or remain unconsumed in the model after the simulation is terminated. These matrices are not directly applicable to the used CPN. Colored data tokens remain in the model by definition even after all transitions have fired. The used events in the event log do not follow a strict linear order for each case. It is therefore unclear which sequence of events should be used for replaying a case. The traces can be transformed into strict linear sequences [48] but these sequences would not fit to the mined model anymore. The possible transformation into linearized traces would introduce choice at the instance level that did not occur in reality. The model illustrated in Fig. 4 does not show any choice. The paths $p_1 = B \rightarrow D$ and $p_2 = B \rightarrow C \rightarrow D$ are parallel and not optional paths. The trace $A \rightarrow B \rightarrow D \rightarrow E$ for example, which would be an output of the transformation, cannot be replayed by the model presented in Fig. 4.[15]

Calculating token based measures like (f) and (a_B) for the produced process model would require many artificial adjustments which are not reflected by the actual source data.

We alternatively use metrics that are suitable to take into account possible parallelism at the instance level and that directly compare the execution paths in the different models. We use the percentage of the control flow paths from $p_{source} \rightarrow p_{sink}$ that are present in the instance graphs to those in the process model for measuring the fitness denoted as f_{Path}. And we compute the percentage of the control flow paths in the process model that are not present in the instance graphs for measuring the precision denoted as p_{Path} with:

$$f_{Path} = \frac{|\{p|p \in P_{PM} \wedge \in P_{IG}\}|}{|P_{IG}|},$$

$$p_{Path} = \frac{|P_{PM}| - |\{p|p \in P_{PM} \wedge \notin P_{IG}\}|}{|P_{PM}|},$$

$P_{PM} = \{p_1, \ldots, p_n\}$ is the set of distinct control flow paths in the process model PM and $P_{IG} = \{p_1, \ldots, p_m\}$ is the set of all distinct control flow paths from all instance graphs IG that belong to PM.

13. The arc inscriptions for posting and clearing arcs only display the assigned constant for the posted or cleared value. The inscriptions for the account type, account number and credit or debit indicator are omitted to improve readability. The same is the case for the inscriptions of the connected account places that only show the account number.

14. Solid line = balance sheet account, dashed line = profit and loss ac-count, black line = debit side of an account, gray line = credit side of an account.

15. The model itself can be transformed in such a way that it is able to replay all linearized traces. But this would result in a much more complex models and a negative effect on the model precision.

TABLE 2
P_{PATH}[16]

p_{Path}	Set 1	Set 2	Set 3
Mean	0.7925	0.8581	0.8034
Median	1	1	1
Minimum	0.1250	0.1165	0.2222
Maximum	1	1	1
Standard Deviation	0.2404	0.2229	0,2491
Variance	0,0578	0.0497	0.0620
Interquartile Range	0.4286	0.3000	0.4167

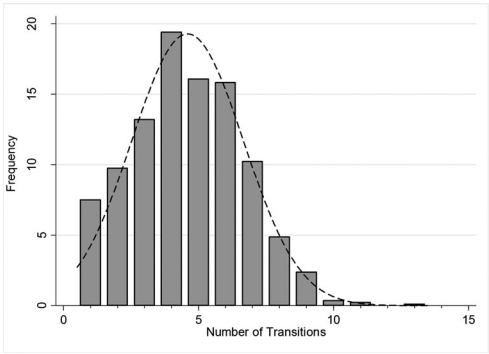

Fig. 5. Frequency distribution for data set 1.

Van Dongen and van der Aalst [24] proof that the used aggregation procedures are path preserving. This means that all control flow paths are also represented in the process model with $f_{Path}= 1$. We calculated p_{Path} for the mined models from data set 1 to 3. The results are listed in Table 2. The average value for this measure over all three data sets is 0.81. It does not equal 1 due to the potential parallelism of branches on the process instance graph level and differences in the execution paths of parallel branches.[17] A value of 0.81 means that 19% of the paths in the process models may actually represent behavior that was not recorded in the event log. It is an acceptable result compared to outcomes of artificial simulations from other scholars that use comparable metrics [38]. But it has to be validated in further research if this precision is high enough for the application in real world scenarios.

The models produced by the MLPM from data set 1 to 3 were also analyzed using descriptive statistics. Figs. 5, 6, 7, and 8 show selected results for data sets 1 and 2. Figs. 5 and 7 show the frequency distributions of the mined process models depending on their model complexity measured as the number of included transitions. They show that the process models are distributed comparably to a normal distribution. Figs. 6 and 8 present scatter diagrams for data set 1 and 2. They illustrate the distribution of the number of represented instance models in a process model depending on the model size. The diagrams show that the vast majority of process instances actually belong to very simple process models that contain only a few transitions. This observation confirms preliminary results from prior research work [56].

16. Process models including loops were excluded from the calculation.

17. This is the case, for example, if too parallel branches like in Fig. 3a show different execution paths with $P_{IG} = \{(B \rightarrow C \rightarrow D), (C \rightarrow E \rightarrow D)\}$ but involving the same activities (in this case C). Then $P_{PM} = \{(B \rightarrow C \rightarrow D), (C \rightarrow E \rightarrow D), (B \rightarrow C \rightarrow E \rightarrow D), (C \rightarrow D)\}$ with$p_{Path} = 0.5$.

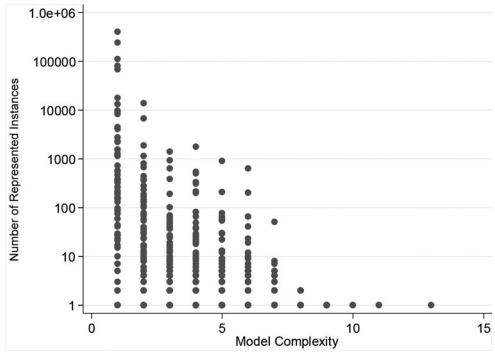

Fig. 6. Scatter diagram for data set 1[18].

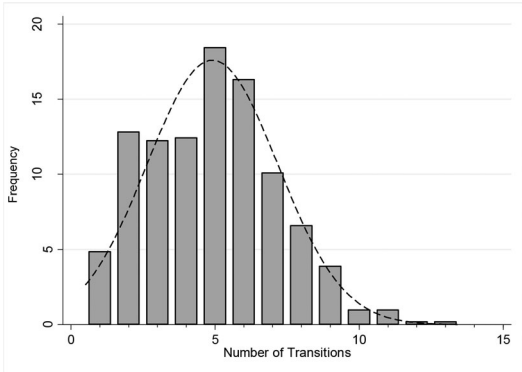

Fig. 7. Frequency distribution for data set 2.

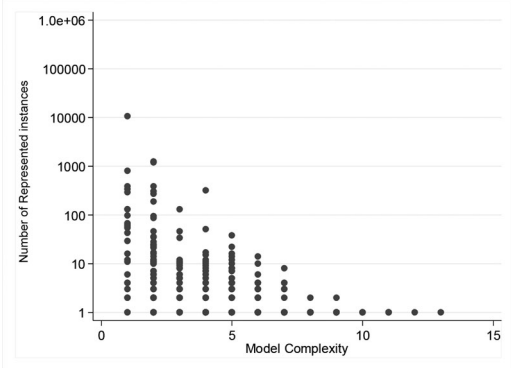

Fig. 8. Scatter diagram for data set 2.

The diagrams for the third data set are not included due to space restrictions. They follow similar patterns. Selected descriptive statistical values are listed in Table 3 for all data sets.

7 Discussion

The overall aim of the presented research is the improvement of financial audits by providing process mining solutions to discover process models from recorded event log data in ERP system. The main contribution of this paper is the introduction of a process mining algorithm that is able to discover the control flow and the data flow perspective in process models by simultaneously providing perfectly

18. The dependent variables in Figs. 6 and 8 use a logarithmic scaling.

TABLE 3
Analyzed Process Models

Transitions per instance	Set 1	Set 2	Set 3
Mean	4.56	4.95	3.98
Median	5	5	4
Standard Deviation	2.07	2.43	2.24
Maximum	15	13	21
Minimum	1	1	1

fitting and highly precise models at different abstraction levels. The innovation of the presented artifact is achieved by combining already existing and newly developed methods that lead to a novel solution for a new application area. The presented research followed a DSR approach. Gregor and Hevner [57] provide a useful framework for the categorization of knowledge generated by DSR and categorize research into four quadrants. The presented research work can be assigned to the exaptation quadrant because the main objective is to provide a solution for a new application area by partly using already existing knowledge. But it also affects the improvement quadrant by introducing a new method to model the control flow and data simultaneously in mined process models. Gregor and Hevner further differentiate between three levels of contribution types that range from abstract, complete and mature knowledge on the highest level to more specific, limited and less mature knowledge on the lowest level. The research results presented in this paper are mainly located on the second level providing constructs and methods for the mining of process models and on the first level presenting an instantiated software artifact. The results that can be achieved by analyzing the mining outcomes can also be input for the third and highest knowledge contribution level. The distributions of the number of instances over the number of transitions in Figs. 6 and 8 for example can lead to the assumption that their distribution curves are very similar. But the descriptive statistics in Table 3 shows that the mean values for the number of transitions in the process model differ quite significantly from each other with 4.56 for data set 1, 4.95 for set 2 and 3.98 for set 3. It could hypothesized that the complexity of the mined process models relates to the maturity of the mined business processes following the assumption that a mature process is more integrated into information systems than a less mature. This research question surely needs further investigation, but it highlights how the presented results can be the starting point for further theoretical research.

The presented mining algorithm is able to discover process models in accordance with the identified requirements to a large extent. The mined models are not absolutely precise. It needs to be validated in further research if the achieved level of precision is sufficient in practice. Several other limitations need to be taken into account. The process models do not represent sound workflow nets according to commonly used definitions [3, p. 39]. This handicap is not too severe because the objective of process mining for financial audits is the adequate modeling of the control and data flow perspective with precise and fitting process models. Formally well-structured process models are of minor interest. But soundness would be achieved if account places are neglected. The remaining models would then represent sound workflow nets but without representing the data flow perspective. The data sets were all extracted from SAP systems. It can therefore not be concluded that the research results also hold true for other data sources. But an important advantage of the used mining algorithm is its independence from the implemented data structures of a particular ERP system because it bases on the general structure of accounting entries. Some process models showed loops that occur when a transaction has cleared a journal item that was posted by the same transaction or by a transaction with the same transaction name located in the subsequent execution path. This constellation leads to a deadlock in the process model which is not critical for the interpretation of the model but generally not desired for the modeling of correct process models. A solution could be the prevention of aggregating transitions carrying the same label if this would result in a loop.

The mining algorithm produces precise and fitting process models at the cost of lacking generalization. It is therefore not applicable for scenarios with highly variable business processes. In the worst case scenario all process instances show a different behavior. The mining algorithm would then not be able to aggregate any instance models and the set of process instances models would be identical to the set of process models resulting in no or little information gain. The data presented in Table 1 shows that this risk is not acute for the given application area. Business processes are usually standardized to a certain degree when they are operated via ERP systems. The data shows that the number of process models ranges from 307 for the smallest data set to 841 models for the largest. This may still seem to be a big number. But 63 process models in data set 1, for example, only consist of one transition. These represent trivial processes. The activities in these processes were mostly carried out by using a single general purpose transaction. They are of little interest from a process perspective but highly important from an audit perspective because this category of process models represent 96% of process instances in data set 1, 70% in set 2 and 51% in set 3. It is clear that further analytical procedures are necessary to address this category. A starting point could be the clustering of process models that use the same accounts. The process models that reflect the major business processes are those that contain many transitions and represent a high number of process instances. Data set 1 contains 135 process models consisting of 5 transitions. But just two of them already represent 62% of the instances of this category. It can therefore be assumed that the majority of instances for more complex process models only represent very infrequent behavior and can be tested traditionally by inspecting individual journal entries. The process models that represent many process instances and create a high value flow are interesting from a materiality perspective and can be audited by including the testing of embedded application controls [58].

8 Conclusion

The amount of available data increases with the integration of information systems for the support and automation of business activities. BI is a research domain that provides mature methods and tools that can be used to exploit and

handle the growing amount of data. While it is commonly used in many application scenarios it is almost absent in the auditing industry. Traditional audit procedures are not efficient and effective in audit environments with highly integrated information systems and an increasing amount of processed data. The audit of business processes is a significant part in the financial audit. Process mining can be applied as a BI approach to support and automate the audit of business processes. The selection of a process mining algorithm should be founded on the analysis of relevant application domain requirements. For the case of financial audits it is crucial that a mining algorithm is able to model both the control flow and data flow. The algorithm should preserve the audit trail and produce perfectly fitting and as precise process models as possible. We have designed and evaluated a multilevel process mining algorithm that meets these requirements to a large extent. It introduces novel constructs and methods for the mining of process models and an instantiated software artifact. The results derived by exposing the designed artifact to extensive real life data can be the starting point for future theory building.

Data sets from the SAP systems of three different companies operating in diverse industries were used for the evaluation of the designed artifact. It cannot be concluded that the results hold true for other ERP systems and industries but the implemented mining algorithm exploits the structure of accounting entries that is system-independent and should therefore be generally applicable. The extension to other ERP systems will be covered in future research.

Public accountants face the challenge to audit increasingly complex and integrated business processes that process huge amount of data. The presented mining algorithm exploits large data sets that are created during the operation of business processes. It provides a suitable solution for analyzing business processes in financial audits but it can also be applied in application contexts that exhibit similar requirements.

A basic limitation of the presented algorithm is the lack of generalization. This is a desired characteristic for financial audits but it is not appropriate in settings with highly variable business processes. Other mining approaches using a two-step approach [41] or fuzzy mining [13] might be more appropriate in these settings. Some restrictions still exist in certain process constellations that create deadlocks in the process models. This phenomenon and adequate solutions still need to be researched in the future.

REFERENCES

[1] *Identifying and Assessing the Risks of Material Misstatement through Understanding the Entity and Its Environment*, IFAC, ISA 315 (Revised), 2012.

[2] M. Werner and N. Gehrke, "Potentiale und grenzen automatisierter prozessprüfungen durch prozessrekonstruktionen [Chances and limits of automated process audits using process reconstruction]," (in German), in *Forschung für die Wirtschaft*, G. Plate, Ed., Aachen, Germany: Shaker-Verlag, 2011.

[3] W. M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, 1st ed., Berlin, Germany: Springer, 2011.

[4] H. Chen, R. H. L. Chiang, and V. C. Storey, "Business intelligence and analytics: From big data to big impact," *MIS Quart.*, vol. 36, no. 4, pp. 1165–1188, Dec. 2012.

[5] H. Krcmar, *Informationsmanagement [Information Management]*. (in German). Berlin, Germany: Springer, 2010.

[6] H.-G. Kemper, W. Mehanna, and H. Baars, *Business Intelligence—Grundlagen und praktische Anwendungen. [Business Intelligence—Foundations and Practical Applications]*. (in German). Wiesbaden, Garmany: Vieweg+Teubner, 2010.

[7] R. L. Braun and H. E. Davis, "Computer-assisted audit tools and techniques: Analysis and perspectives," *Manag. Auditing J.*, vol. 18, no. 9, pp. 725–731, 2003.

[8] N. Gehrke, "The ERP Auditlab—A prototypical framework for evaluating enterprise resource planning system assurance," in *Proc. 43th Hawaii Int. Conf. Syst. Sci.*, Kauai, HI, USA, 2010, pp. 1–9.

[9] W. M. P. van der Aalst, A. Andriansyah, A. K. de Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. van den Brand, R. Brandtjen, and J. Buijs, "Process mining manifesto," in *Proc. BPM Workshops*, 2012, pp. 169–194.

[10] W. M. P. van der Aalst, K. M. van Hee, J. M. van Werf, and M. Verdonk, "Auditing 2.0: Using process mining to support tomorrow's auditor," *Computer*, vol. 43, no. 3, pp. 90–93, Mar. 2010.

[11] M. Jans, M. Alles, and M. Vasarhelyi, "Process mining of event logs in internal auditing: A case study," in *Proc. 2nd Int. Symp. Accounting Inf. Syst.*, 2011, pp. 1–30.

[12] A. Weijters, W. M. P. van der Aalst, and A. K. A. de Medeiros, "Process mining with the heuristics miner-algorithm," Technische Universiteit Eindhoven, Eindhoven, The Netherlands, Tech. Rep. WP, vol. 166, 2006.

[13] C. Günther and W. M. P. van der Aalst, "Fuzzy mining—Adaptive process simplification based on multi-perspective metrics," in *Proc. 5th Int. Conf. Business Process Manage.*, 2007, pp. 328–343.

[14] A. K. A. de Medeiros, "Genetic process mining," Ph.D. dissertation, Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2006.

[15] M. de Leoni and W. M. P. van der Aalst, "Data-aware process mining: Discovering decisions in processes using alignments," in *Proc. 28th Annu. ACM Symp. Appl. Comput.*, Coimbra, Portugal, 2013, pp. 1454–1461.

[16] D. Ferreira and D. Gillblad, "Discovering process models from unlabelled event logs," in *Proc. 7th Int. Conf. Business Process Manag.*, pp. 143–158, 2009.

[17] N. Gehrke and N. Müller-Wickop, "Basic principles of financial process mining a journey through financial data in accounting information systems," presented at the *16th Amer. Conf. Inform. Syst.*, Lima, Peru, 2010.

[18] H. Österle, J. Becker, U. Frank, T. Hess, D. Karagiannis, H. Krcmar, P. Loos, P. Mertens, A. Oberweis, and E. J. Sinz, "Memorandum on design-oriented information systems research," *Eur. J. Inf. Syst.*, vol. 20, no. 1, pp. 7–10, 2010.

[19] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quart.*, vol. 28, no. 1, pp. 75–105, Mar. 2004.

[20] J. Mingers, "Combining IS research methods: Towards a pluralist methodology," *Inf. Syst. Res.*, vol. 12, no. 3, pp. 240–259, 2001.

[21] N. Müller-Wickop, M. Schultz, and M. Peris, "Towards key concepts for process audits—A multi-method research approach," presented at the *10th Int. Conf. Enterprise Syst., Accounting Logistics*, Utrecht, The Netharlands, 2013.

[22] S. Brinkkemper, "Method engineering: Engineering of information systems development methods and tools," *Inf. Softw. Technol.*, vol. 38, no. 4, pp. 275–280, 1996.

[23] T. Wilde and T. Hess, "Forschungsmethoden der wirtschaftsinformatik [Research methods in information systems]," (in German), *Wirtschaftsinformatik*, vol. 49, no. 4, pp. 280–287, 2007.

[24] B. F. van Dongen and W. M. P. van der Aalst, "Multi-phase process mining: Aggregating instance graphs into EPCs and Petri Nets," in *Proc. PNCWB 2005 Workshop*, 2005, pp. 35–58.

[25] M. Werner, "Colored petri nets for integrating the data perspective in process audits," in *Proc. 32nd Int. Conf. Conceptual Model.*, Hong Kong, 2013, pp. 387–394.

[26] M. Werner, M. Schultz, N. Müller-Wickop, N. Gehrke, and M. Nüttgens, "Tackling complexity: Process reconstruction and graph transformation for financial audits (research in progress)," presented at the *33rd Int. Conf. Inf. Syst.*, Orlando, FL, USA, 2012.

[27] M. Werner and M. Nüttgens, "Improving structure—Logical sequencing of process models," in *Proc. 47th Hawaii Int. Conf. Syst. Sci.*, Waikoloa, HI, USA, 2014, pp. 3888–3897.

[28] A. Tiwari, C. J. Turner, and B. Majeed, "A review of business process mining: State-of-the-art and future trends," *Business Process Manage. J.*, vol. 14, no. 1, pp. 5–22, 2008.

[29] W. M. P. van der Aalst, "Process mining: Overview and opportunities," *ACM Trans. Manage. Inf. Syst.*, vol. 99, no. 99, pp. 1–16, Feb. 2012.
[30] R. S. Debreceny and G. L. Gray, "Data mining journal entries for fraud detection: An exploratory study," *Int. J. Accounting Inf. Syst.*, vol. 11, no. 3, pp. 157–181, Sep. 2010.
[31] J. Becker, P. Delfmann, M. Eggert, and S. Schwittay, "Generalizability and applicability of model-based business process compliance-checking approaches—A state-of-the-art analysis and research roadmap," *BuR—Business Res.*, vol. 5, no. 2, pp. 221–247, Nov. 2012.
[32] M. Alles, M. Jans, and M. Vasarhelyi, "Process mining: A new research methodology for AIS," in *Proc. CAAA Annu. Conf.*, 2011, pp. 1–16.
[33] M. Jans, "Process mining in auditing: From current limitations to future challenges," in *Proc. Business Process Manage. Workshops*, 2012, pp. 394–397.
[34] M. Jans, N. Lybaert, K. Vanhoof, and J. M. Van Der Werf, "Business process mining for internal fraud risk reduction: Results of a case study," presented at the *Int. Res. Symp. Accounting Inf. Syst.*, Paris, France, 2008.
[35] T. Stocker, "Data flow-oriented process mining to support security audits," in *Proc. Service-Oriented Comput. Workshops*, 2012, pp. 171–176.
[36] N. Trcka, W. M. P. van der Aalst, and N. Sidorova, "Data-flow anti-patterns: Discovering data-flow errors in workflows," in *Proc. 21st Int. Conf. Adv. Inform. Syst. Eng.*, 2009, pp. 425–439.
[37] R. Accorsi and C. Wonnemann, "InDico: Information flow analysis of business processes for confidentiality requirements," in *Proc. 6th Int. Conf. Security Trust Manage.*, 2011, pp. 194–209.
[38] A. Rozinat, A. K. A. de Medeiros, C. W. Günther, A. Weijters, and W. M. P. van der Aalst, "The need for a process mining evaluation framework in research and practice," in *Proc. Business Process Manage. Workshops*, 2008, pp. 84–89.
[39] A. K. A. de Medeiros, A. Guzzo, G. Greco, W. M. P. van der Aalst, A. Weijters, B. F. Van Dongen, and D. Saccà, "Process mining based on clustering: A quest for precision," in *Proc. Business Process Manage. Workshops*, 2008, pp. 17–29.
[40] M. Song, C. W. Günther, and W. M. P. van der Aalst, "Trace clustering in process mining," in *Proc. Business Process Manage. Workshops*, 2009, pp. 109–120.
[41] W. M. P. van der Aalst, V. Rubin, H. M. W. Verbeek, B. F. Dongen, E. Kindler, and C. W. Günther, "Process mining: A two-step approach to balance between underfitting and overfitting," *Softw. Syst. Model.*, vol. 9, no. 1, pp. 87–111, Nov. 2008.
[42] N. Müller-Wickop, M. Schultz, N. Gehrke, and M. Nüttgens, "Towards automated financial process auditing: Aggregation and visualization of process models," presented at the *Enterprise Model. Inf. Syst. Archit.*, Hamburg, Germany, 2011.
[43] IFAC, *ISA 320 Materiality in Planning and Performing an Audit*, 2009.
[44] M. B. Romney and P. J. Steinbart, *Accounting Information Systems*, 11th ed., Englewood Cliffs, NJ, USA: Prentice-Hall, 2008.
[45] G. Greco, A. Guzzo, L. Pontieri, and D. Sacca, "Mining expressive process models by clustering workflow traces," in *Advances in Knowledge Discovery and Data Mining*. New York, NY, USA: Springer, 2004, pp. 52–62.
[46] A. Rozinat and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Inf. Syst.*, vol. 33, no. 1, pp. 64–95, Mar. 2008.
[47] M. Weske, *Business Process Management Concepts, Languages, Architectures*. Berlin, Germany: Springer, 2012.
[48] N. Müller-Wickop and M. Schultz, "ERP event log preprocessing: Timestamps vs. accounting logic," in *Proc. 8th Int. Conf. Design Sci. Res. Inf. Syst. Technol.*, Berlin, Germany, 2013, vol. 7939, pp. 105–119.
[49] S. X. Sun and J. L. Zhao, "Formal workflow design analytics using data flow modeling," *Decision Support Syst.*, vol. 55, no. 1, pp. 270–283, Apr. 2013.
[50] W. M. P. van der Aalst and C. Stahl, *Modeling Business Processes: A Petri Net-Oriented Approach*. Cambridge, MA, USA: MIT Press, 2011.
[51] K. Jensen and L. M. Kristensen, *Coloured Petri Nets*. New York, NY, USA: Springer, 2009.
[52] A. A. De Medeiros, A. Weijters, and W. M. P. van der Aalst, "Using Genetic algorithms to mine process models: Representation, operators and results," BETA Working Paper Series WP 124, Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2005.
[53] J. Venable, J. Pries-Heje, and R. Baskerville, "A comprehensive framework for evaluation in design science research," in *Proc. 7th Int. Conf. Design Sci. Res. Inf. Syst., Adv. Theory Practice*, 2012, pp. 423–438.
[54] yWorks GmbH. (2014). "*yEd–Graph Editor*. [Online]. Available: http://www.yworks.com/de/products_yed_about.html
[55] University of Hamburg. (2014). *Renew—The Reference Net Workshop*. [Online]. Available: http://www.renew.de/
[56] M. Werner, N. Gehrke, and M. Nüttgens, "Towards automated analysis of business processes for financial audits," presented at the .11th Int. Conf. Wirtschaftsinformatik, Leipzig, Germany, 2013.
[57] S. Gregor and A. R. Hevner, "Positioning and presenting design science research for maximum impact," *MIS Quart.*, vol. 37, no. 2, pp. 337–355, 2013.
[58] M. Werner, N. Gehrke, and M. Nüttgens, "Business process mining and reconstruction for financial audits," in *Proc. 45th Hawaii Int. Conf. Syst. Sci.*, Maui, HI, USA, 2012, pp. 5350–5359.

Michael Werner received the Dipl-WirtInf and MMgmt degrees, and CISA, CISM, and CGEIT certifications. He has been a lecturer for accounting information systems at the Accounting Department, Business School, Auckland University of Technology, since 2015, and is a research fellow at the Chair for Information Systems, Business School, University of Hamburg. He worked at PwC from 2006 to 2011. His current research interests include accounting information systems, process mining, business process management, and compliance. He is a member of the Information Systems Audit and Control Association (ISACA), the ISACA CISM Quality Assurance Team, and the Association for Information Systems.

Nick Gehrke received the CISA certification. He has been a professor for database systems at the Department for Computer Science, NORDAKADEMIE, since 2010. He has been a German tax advisor since 2008 and worked for PwC from 2004 to 2009. His current research interests include business process modeling and compliance. He is a management board member of the Financial Expert Association and a member of the German Informatics Society, the Information Systems Audit and Control Association, and the German Association of University Professors and Lecturers (VHB).