TACKLING COMPLEXITY: PROCESS RECONSTRUCTION AND GRAPH TRANSFORMATION FOR FINANCIAL AUDITS

Research-in-Progress

Michael Werner
University of Hamburg
Chair for Information Systems
Max-Brauer-Allee 60
D-22765 Hamburg
michael.werner@wiso.uni-hamburg.de

Martin Schultz
University of Hamburg
Chair for Information Systems
Max-Brauer-Allee 60
D-22765 Hamburg
martin.schultz@wiso.uni-hamburg.de

Niels Müller-Wickop
University of Hamburg
Chair for Information Systems
Max-Brauer-Allee 60
D-22765 Hamburg
niels.mueller-wickop@wiso.uni-hamburg.de

Nick Gehrke
Nordakademie
Chair for Information Systems
Köllner Chaussee 11
D-25337 Elmshorn
nick.gehrke@nordakdemie.de

Markus Nüttgens
University of Hamburg
Chair for Information Systems
Max-Brauer-Allee 60
D-22765 Hamburg
markus.nuettgens@wiso.uni-hamburg.de

Abstract

A key objective of implementing business intelligence tools and methods is to analyze voluminous data and to derive information that would otherwise not be available. Although the overall significance of business intelligence has increased with the general growth of processed and available data it is almost absent in the auditing industry. Public accountants face the challenge to provide an opinion on financial statements that are based on the data produced by the automated processing of countless business transactions in ERP systems. Methods for mining and reconstructing financially relevant process instances can be used as a data analysis tool in the specific context of auditing. In this article we introduce and evaluate an algorithm that effectively reduces the complexity of mined process instances. The presented methods provide a part of the foundation for implementing automated analysis and audit procedures that can assist auditors to perform more efficient and effective audits.

Keywords: business process modeling, data mining, process mining, business intelligence, data analysis, enterprise resource planning systems, financial audits

Introduction

Enterprise resource planning (ERP) systems are key components for supporting and automating the processing of business transactions in modern companies. While ERP systems are primarily used for supporting and automating business processes they commonly also provide the functionality to prepare the financial statements that companies are required to publish. The provided information plays a critical role in the economic system. It enables stakeholders to acquire information about the financial situation of the entity they are interested in. Due to their informative significance governments and regulatory institutions have issued laws and regulations that intend to safeguard the correctness of published financial statements. They have entrusted public accountants to carry out audits for ensuring that accounting standards are adhered to and that the published information is free of material misstatements. But when auditors perform their audits they encounter a significant problem. While on the one hand business transactions are processed automatically in ERP systems auditors on the other hand apply mainly manual audit procedures to achieve their audit comfort. With increasing integration of ERP systems and rising numbers of processed transactions manual audit procedures become inefficient and ineffective.

This situation is quite astonishing as the precondition for implementing automated audit procedures is given in every case where the mere number of processed transactions makes manual audit procedures inefficient or even ineffective. Whenever this is the case systems must be involved that support or automate the processing. Otherwise the number of transactions could still be effectively audited with manual audit procedures. While processing ERP systems produce data that is stored in the systems' databases. The stored data includes the journal entries and other logging information that can be used to reconstruct relationships between the stored data entries and their corresponding originally executed transactions. The content of the stored financially relevant data in ERP systems is generally suitable for automated process mining and analysis purposes. It is clearly structured and needs to comply with accounting requirements like completeness and accuracy. It is possible to capitalize on the available structures by using purposeful mining and analysis methods.

Gehrke and Müller-Wickop (2010a, 2010b) present an algorithm for mining business processes from journal entries stored in ERP databases. The approach is similar to mining process models from event logs (van der Aalst 2011) but applied to financially relevant business processes. Werner et al. (2012) show how these process mining methods can be combined with automated testing of controls that are embedded in ERP systems. The combination generally allows implementing system based and automated audit procedures that are efficient and effective to audit highly integrated and automated financially relevant business processes. This way the imbalance between automated processing on the companies' side and manual audit procedures on the auditors' side can be overcome.

The application of these methods can be seen as a business intelligence tool that enables especially auditors to receive information from available mass data that they otherwise do not have access to. Their application would enable the auditor to efficiently gather information for understanding the relationship between the business processes and the financial statements for the audited entity. They would also allow the identification of unusual business transactions that normally inherit higher risk than standard transactions (Werner and Gehrke 2011).

In this paper we focus on the results and issues that arise from analyzing mined process instances. We therefore address a specific sub-problem on the path towards developing automated analysis and audit methods for financial audits. Analyzing mined processes from real life data shows that process instances do exist that contain up to tens of thousands executed transactions leading to very complex graphs consisting of hundreds of thousands elements. The complexity of these graphs does not allow sophisticated interpretation for the purpose of auditing. It is necessary to find mechanisms to reduce their complexity.

In this article we illustrate how the complexity of mined process instances can be reduced by using a graph transformation system. We start with an overview of related work and chosen research methodology. We continue with an illustration of how mined process instances can be represented as Petri nets by using an illustrative example of a mined process instance. Based on this representation we introduce an aggregation algorithm that operates as a graph transformation system by aggregating net elements within a process instance. We graphically show the aggregation results for the used example and provide evaluation results that were derived from applying the aggregation algorithm to test and real life data. The paper closes with a summary and conclusion of the presented methods and derived results.

We focus our attention on financial audits in order to stay in reasonable limits but we like to point out that the application of the discussed process mining, reconstruction and graph transformation methods is not restricted to financial audits but is also relevant for performance and optimization considerations (Werner and Gehrke 2011).

Related Work and Research Methodology

The concept of process mining forms the foundation of the research presented in this paper. Process mining first evolved in the 1990s for investigating process mining in the context of software engineering (Cook and Wolf 1998a, 1998b, 1999). The idea of applying process mining to workflow logs was first introduced by Agrawal et al. (1998). Maxeiner et al. (2001) and Schimm (2000; 2001a; 2001b; 2002) developed mining tools whereas Herbst and Karagiannis (Herbst 2000a; 2000b; 2003; Herbst and Karagiannis 1998; 1999) addressed process mining in the context of workflow management. Substantial research work exists for mining and rediscovering process models from event logs (van der Aalst 1997, 1998, 2005, 2011; van der Aalst et al. 2002a, 2002b; van der Aalst and Dongen 2002; van der Aalst and Weijters 2002; Maruster et al. 2001a, 2001b; Weijters and van der Aalst 2001). Their work covers a variety of aspects like delta analysis, conformance, concurrency, workflow verification and performance. Gehrke and Müller-Wickop (2010a, 2010b) build a bridge between process mining and financial accounting by applying process mining to reconstruct financially relevant business processes.

A traditional application area of business intelligence is the support for decision making processes (Turban et al. 2007; Vercellis 2009). Anandarajan et al. (2004) illustrate business intelligence techniques from an accounting and finance perspective but we are not aware of publications that address business intelligence tools from an explicit auditing perspective. For developing an aggregation method we refer to the theory of graph grammars and transformation (Heckel 2006; Rozenberg 1997).

The research presented in this paper follows a design science approach (March and Smith 1995; Österle et al. 2010; Hevner et al. 2004). For the purpose of developing adequate complexity reduction methods we implemented a software artifact. The artifact allows reconstructing and observing process instances. Based on the observation derived from extensive real life data we engineered formal methods (Brinkkemper 1996) and finally evaluated them against test and real life data.

Automated Reconstruction and Petri Net Representation

The mined processes presented by Gehrke and Müller-Wickop (2010a, 2010b) use a more or less informal presentation that focusses on illustrating essential process components. Müller-Wickop et al. (2011) introduce a Business Process Model and Notation (BPMN) based representation which primarily aims to integrate the process and financial perspective by including graph elements that capture financially relevant aspects of modeled process instances. In order to be able to develop comprehensive methods for complexity reduction we need a formally robust presentation. We chose a Petri net representation due to the fact that Petri nets constitute a formally sound and mathematically powerful modeling language (Valk 2008). Modeling mined process instances as colored Petri nets makes it possible to apply already existing research for event log mining (van der Aalst 2011).

Figure 1 shows a reconstructed instance of a purchasing process that was mined from an ERP system by using the mining algorithm developed by (Gehrke and Müller-Wickop 2010b). This means that a purchasing business transaction was executed and recorded in the examined ERP system. The relevant data was extracted and the process instance was reconstructed. The illustrated example represents an executable colored Petri net that mimics the behavior of the originally processed instance. A colored Petri net can formally be expressed as a tuple $N = (P, T, F, C, cd, W, m_o)$. The specific meaning of each tuple element for the constructed instance is described in Table 1 and can be illustrated by referring to Figure 1. The transitions of the Petri net shown in Figure 1 represent executed transactions that created journal entries in the ERP system. MB01 are transactions that recorded the receipt of ordered material, MR1M processed the received invoices and F110 executed the payment run. FB1S represent automatic clearing transactions. The set of transition T for this Petri net contains all the transitions that represent the executed transactions {MB01_5000004383, MB01_5000004384, ...}. Journal entry items are modeled as places with inscriptions identifying the account the corresponding item was posted to. The different colors illustrate whether the item was a debit or credit posting. The *created*-relationships between items and transactions are illustrated as dotted arrows. The implemented mining algorithm exploits the open-item-accounting structure of journal entries. Journal entries with enabled open-item-accounting are either cleared or not cleared. When a process has been successfully terminated all open items are cleared. Otherwise the process has not been completely executed. A clearing document exists for each cleared item. By following this connection between journal entries the whole process instance can be reconstructed. The *cleared*-relationships between transactions and items are illustrated as dotted arcs. The inscriptions on the arcs represent the amounts that were posted or cleared on the accounts the items were placed on. The modeling of start places marked with tokens representing the document number that was created by the corresponding transaction ensures that each transition in the Petri net model can actually be executed. The flow relation F for this Petri net contains all arcs between the transitions and places. The set of places includes all illustrated places (start places and places for journal entry items). The set of colors includes the set of account numbers {310000, 191100, ...}, indicators for credit or debit posting {credit, debit} and document numbers {5000004383, 5000004384, ...}. The function cd maps the colors to the individual places. The function W maps the arc inscriptions that represent the booking values {13907.17, 10880.29, ...} to the arcs for relations between transitions and places representing journal entry items or the document numbers to the arcs between start places and transitions.

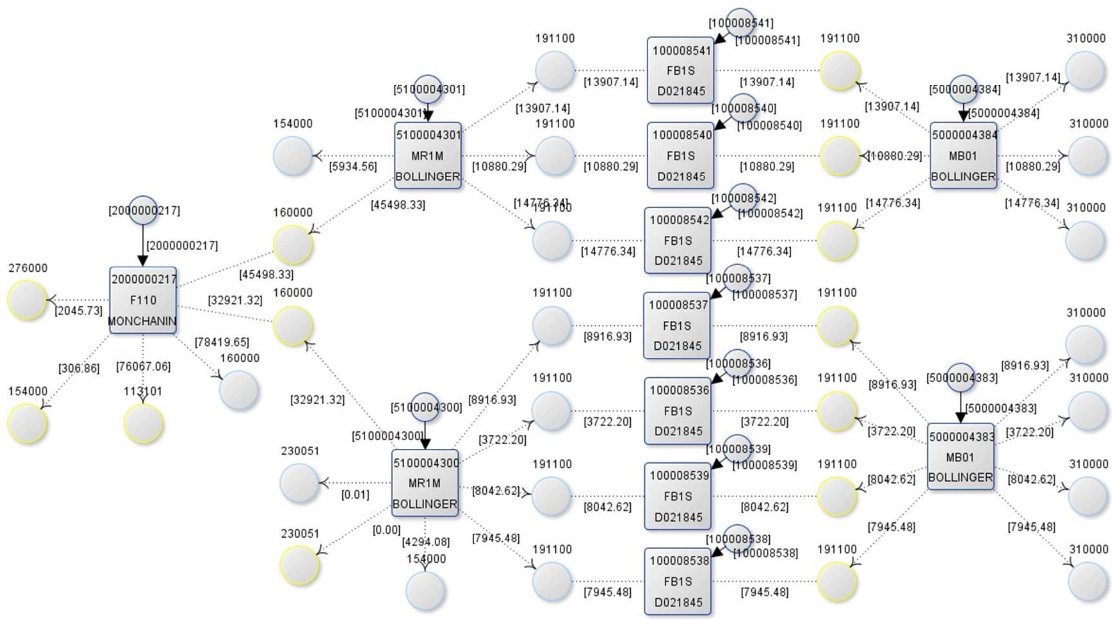


Figure 1. Simple Process Instance A

Table 1. Formal Petri Net Representation		
Tuple elements		**Specific meaning in the representation context**
P	Set of places	Start places and places representing journal entry items
T	Set of transitions	Transactions executed in the ERP system
F	Flow relation	Arcs between the nodes indicating their type of relationship
C	Set of colors	Set of place characteristics, posting values, document numbers
cd	Color domain mapping	Mapping of characteristics to places
W	Arc inscription	Mapping of posting values or document numbers to arcs
m_0	Initial marking	Tokens marking start places with the corresponding document number

Example A contains twelve transitions. The instance is clearly interpretable by observation. It changes for more complex process instances. Example B illustrated in Figure 2 shows a mined process instance like example A and consists of the same Petri net elements as described in Table 1 but it contains 358 transitions. It is still a small instance compared to the most complex instances in our sample database that included up to 27,177 transitions. Table 2 provides an overview of the characteristics of both examples. When looking at Figure 2 it is obvious that the interpretation and analysis of reconstructed instances by simple observation becomes impossible when they include more than a few transactions.

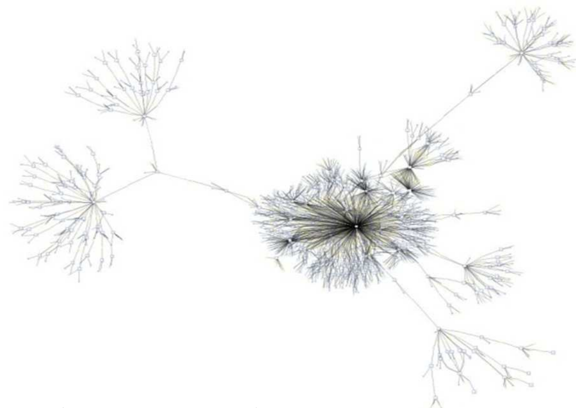


Figure 2. Complex Process Instance B

Table 2. Net Characteristics for Process Instance Examples

Example Instance	A	B
Number of transitions	12	358
Number of places	43	1804
Number of arcs	60	2549
Sum of net elements	115	4711

Graph Transformation for Complexity Reduction

The complexity of a graph can be defined differently, depending on the relevant perspective und purpose (Neel and Orrison 2006). A graph is generally defined as a pair of disjoint sets G = (V,E) where V is the set of vertices and E is the set of edges with $E \subseteq V^2$ (Diestel 2010). For the purpose of this paper we consider the complexity of the graph simply as a function of its number of edges and vertices. The mined process instances are modeled as Petri nets. We therefore use the term net elements for the sum of vertices and edges that determine the complexity of the Petri net under review. A promising approach for reducing complexity of voluminous process instances is to aggregate similar net elements and thereby reducing their total number. A key requirement for any type of transformation is that financially relevant information remains unchanged compared to the originally reconstructed process. In the context of auditing it is crucial to be able to trace any transaction from the point of origin to the final ledger posting. The path that allows tracing a transaction through an information systems is called audit trail (Romney and Steinbart 2008 p. 687). In the context of process mining for audit purposes this means that the analyzed data has to mirror exactly the transactions that were actually executed and that no existing paths in the graph may be deleted or new ones be added. Müller-Wickop et al. (2011) point out that the behavior of the process instances may not be changed by any aggregation method. If an aggregation algorithm changed the behavior, this fundamental requirement would be violated. The behavior of a Petri net can be expressed as the set of possible firing sequences. For designing an aggregation method it is necessary to ensure that the set of firing sequences remains unchanged.

Requirement I: The set of firing sequences has to stay constant

Different arc types in the Petri net mean that transactions interact differently with journal entry items. If the arc types between transitions and places were altered the resulting graph would no longer reflect the original relationship between these elements. The arc types and therefore the type of interaction between places and transitions may not be altered.

Requirement II: Different arc types may not be merged

The methods of financial process mining by Gehrke and Müller-Wickop (2010b) have been developed to exploit the structure of accounting data and to capture financially relevant information. This information primarily concerns the value flow within the processes. Analyzing this information actually allows to identify which amounts have been posted on the different accounts by a process instance and to evaluate how significant they are from a materiality perspective.

Requirement III: The arc inscriptions representing the value of posted journal entries have to be preserved

We reconstructed and analyzed processes instances from different data sets originating from two companies that operate in the retail and manufacturing industries and from the SAP IDES test system (SAP 2012). Figure 3 provides an overview of the frequency distribution of approximately 40,000 mined process instances originating from a corporation in the retail industry with a logarithmic scaling on the x- and y-axes. The mean value of the distribution is 2.30, the standard deviation is 3.37, the median value is 2 and the maximum value is 596. The frequency distributions for the other data sets show the same pattern with slightly different values.

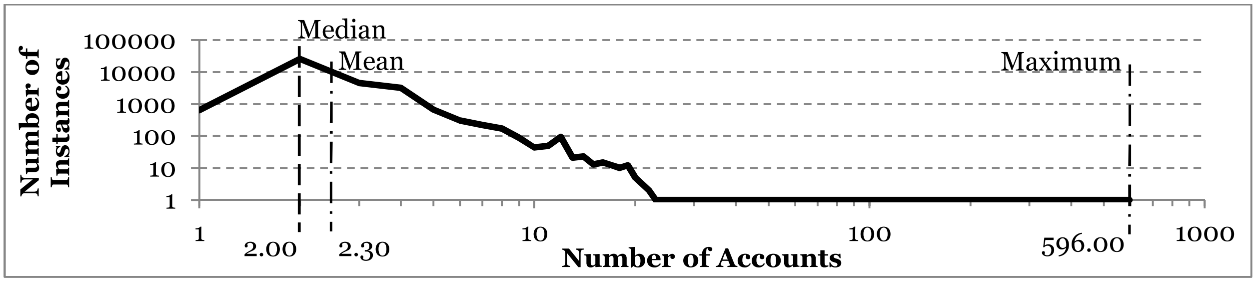
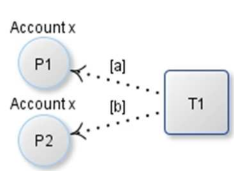
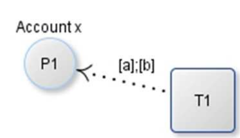


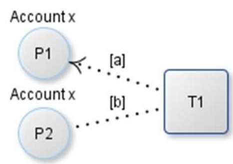
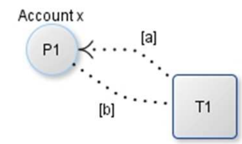
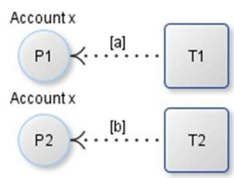
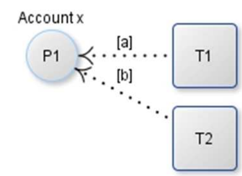
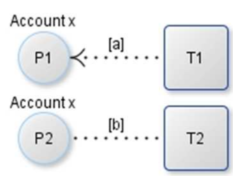
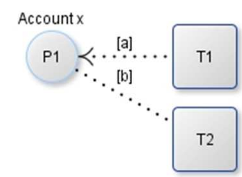
Figure 3. Frequency Distribution of Instances over Accounts

Figure 3 shows that the number of different accounts in process instances is relatively small compared to the instance size. This observation is comprehensible when considering how business processes generally affect accounts. The execution of a certain business process normally produces journal entries only on a very limited subset of the overall available accounts. These are the accounts related to that business process. The execution of a standard procurement process would most likely lead to journal entries on the expense accounts but not on the sales accounts. Although this observation cannot be generalized without further research it is reasonable to assume that it is true also for other industries not covered by the analyzed data sets.

In the Petri net models journal entry items are represented as places carrying an inscription denoting the account they were posted to. Due to the fact that only few accounts are used within the same process instance it is reasonable to assume that the aggregation of places that represent journal entry items on the same accounts might significantly reduce the overall number of net elements. When designing an algorithm for aggregating places requirements I to III have to be considered. The following cases listed in Table 3 illustrate the different constellations that can occur when two places representing journal entry items are to be aggregated. Places are only considered for aggregation if they carry the same account number and credit or debit flag. The case description contains the possible constellations according to the net definition used in this paper and shows the cases for arcs directed from transitions to places (incoming arcs). These constellations also have to be considered for arcs directed from places to transitions (outgoing arcs). Designing an algorithm based on these cases ensures that the set of firing sequences is not changed because the relationship between the transactions is preserved as well as the different arc types connecting the places and transitions. By maintaining inscriptions from merged arcs we ensure that the financially relevant information on posting values does not get lost.

Table 3. Case Distinction for Place Aggregation

	Input	Description	Output
Case 1		P1 and P2 are connected to the same transition T1. The arc types of both arcs are equal. The arcs (T1,P1) and (T1,P2) can be aggregated. The inscriptions [b] is added to [a] resulting in [a];[b]. Place P2 is deleted.	

Case 2		P1 and P2 are connected to the same transition T1. But the arc types are different. The arcs cannot be aggregated. Arc (T1,P2) has to be redirected to P1 resulting in (T1,P1). Place P2 is deleted.	
Case 3		P1 and P2 are connected to different transitions T1 and T2. The arc types of both arcs are equal. The arcs (T1,P1) and (T2,P2) cannot be aggregated in order to preserve the information that the represented items were actually created (or cleared) by different transactions. Arc (T2,P2) has to be redirected to P1 resulting in (T2,P1). Place P2 is deleted.	
Case 4		P1 and P2 are connected to different transitions T1 and T2. The arc types of both arcs are different. The arcs (T1,P1) and (T2,P2) cannot be aggregated. The arc types are different and they originate from different transitions. Arc (T2,P2) has to be redirected to P1 resulting in (T2,P1). Place P2 is deleted.	

The following algorithm in Listing 1 implements the aggregation of places according to the description included in Table 3.

Listing 1. Aggregation Algorithm

```
P_Item ⊆ P        set of all places representing journal entry items in the net
P_ItemAgg= ∅      initially empty set for aggregated places
F_Arcs            set of all arcs in the net

Aggregate Places
While P_Item≠ ∅
      Take p_i ∈ P_Item
      Select all p_j ∈ P_Item with cd(p_i)=cd(p_j)
      Merge arcs for each p_i and p_j
      Add p_i to P_ItemAgg
      Remove p_i and p_j from P_Item
Set P_Item=P_ItemAgg

Merge Arcs
Get incoming arcs F_IncArcsI ⊆ F_Arcs for p_i
Get incoming arcs F_IncArcsJ ⊆ F_Arcs for p_j
For each arc a_i(t_m,p_i) ∈ F_IncArcsI and arc a_j(t_n,p_j) ∈ F_IncArcsJ
      If the arc type of a_i = arc type of a_j
            And if t_m = t_n then add W(a_j) to W(a_i)and         /Case 1
            remove a_j from F_Arcs
      Else set a_j(t_m,p_j)                                       /Case 2,3 and 4

Get outgoing arcs F_OutArcsI ⊆ F_Arcs for p_i
Get outgoing arcs F_OutArcsJ ⊆ F_Arcs for p_j
For each arc a_i(p_i,t_m) ∈ F_OutArcsI and arc a_j(p_j,t_n) ∈ F_OutArcsJ
      If the arc type of a_i = arc type of a_j
```

And if $t_m = t_n$ then add $W(a_j)$ to $W(a_i)$ and /Case 1
remove a_j from F_{Arcs}
Else set $a_j(p_j, t_m)$ /Case 2,3 and 4

The algorithm represents a graph transformation production that iteratively substitutes sub-graphs consisting of two nodes and an arbitrary number of arcs into sub-graphs consisting of one node and the same or smaller set of arcs.

Evaluation

The application of the aggregation algorithm on example instance A leads to the graph illustrated in Figure 4. The number of net elements was reduced from 115 to 79. We used the software Renew (University of Hamburg 2012) for verification purposes. Renew allows to simulate the execution of Petri nets. The testing of samples showed that the algorithm works properly and generates fully reachable Petri nets. We applied the algorithm to two different data sets. Data set 1 originates from the SAP IDES database. The database is available for universities participating in the SAP University Alliance Program (SAP 2012). The database contains over 115,000 journal entries from approximately 81,000 process instances. We chose the SAP IDES database to enable interested readers to reproduce our results. Data set 2 originates from a corporation operating in the retail industry. The used database contains approximately 90,000 journal entries constituting about 40,000 process instances for a period of one year. Table 4 illustrates the characteristics of the distribution of instance frequency over the number of net elements before and after applying the aggregation algorithm. The mean value for the number of net elements per instance is reduced by 23.2 % from 15.31 to 11.76 for the SAP IDES data set and by 23.4 % from 21.18 to 16.22 in data set 2. We achieve an average complexity reduction of approximately a quarter. The complexity reduction is more effective for complex processes with many net elements. We therefore analyzed the effectiveness of the algorithm by selecting a subset from the available data set 2 containing process instances with 100 or more net elements. The mean value of the original subset was 929.48 net elements per instance. The mean value of the aggregated instances was reduced by 44% to 520.

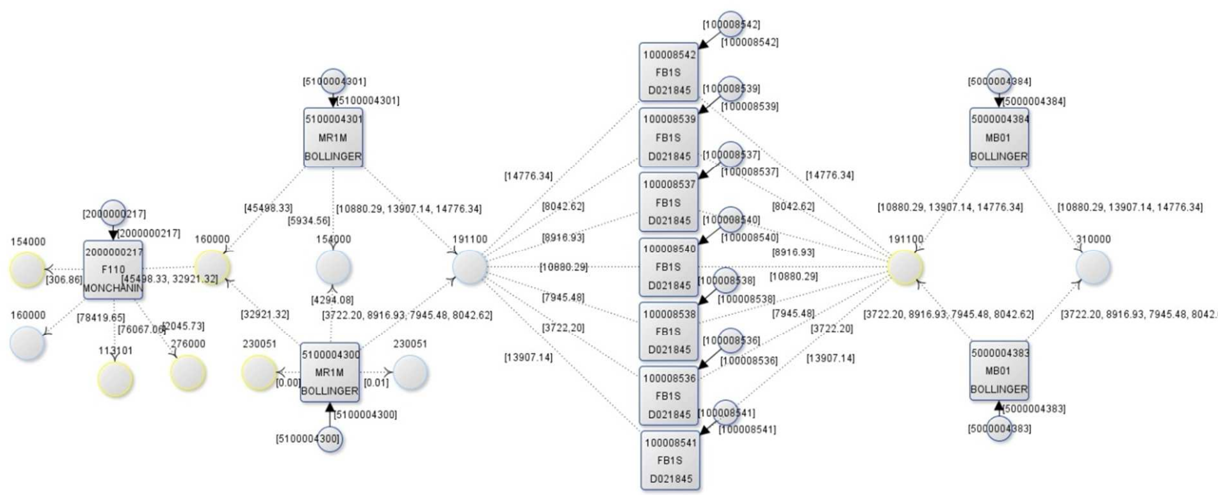


Figure 4. Aggregated Process Instance A

Table 4. Aggregation Results				
	Data Set 1 SAP IDES		Data Set 2 Retail Company	
	original	aggregated	original	aggregated
Mean value of net elements per instance	15.31	11.76	21.18	16.22
Median value of net elements per instance	9.00	7.00	9.00	9.00
Maximum of net elements per instance	32,519	18,602	275,870	146,620
Standard deviation of number of net elements	127.83	73.33	1,380.57	743.81

Conclusion

Public auditors face the challenge of auditing financial statements that base on the data generated by automated transaction processing in ERP systems. While business intelligence techniques have introduced new ways of analyzing data in the general corporate context such techniques are missing in the auditing industry. The mining and reconstruction of financially relevant processes provides the basis for analyzing data from ERP systems for the purpose of financial audits. We have presented an aggregation algorithm in this paper that operates on mined process instances modeled as Petri nets that can be applied to any process instance mined with the used mining algorithm. It aggregates places within the instances and thereby reduces the number of net elements in the graph. The results are less complex models. The evaluation of the algorithm on the basis of test and real life data shows that significant complexity reductions can be achieved. The reduction effect is higher for process instances encompassing many net elements. With the application of the presented mining methods and representation form it is possible to derive and visualize information about executed processes and their effect on the financial accounts that the public accountant has to audit. They provide a means to overcome the imbalance between mainly manual audit procedures on the auditor side and the automated and system based processing on the company side. The usage of these methods could make financial audits more efficient and set free resources for the evaluation of unusual transactions that commonly constitute higher risks than standard transactions.

When using the methods several limitations should be taken into account. The availability of necessary data is the first. The data needs to be extracted from the ERP system, transformed into a data format that can be processed by the mining algorithm and loaded into a database where it can be accessed for mining purposes. This procedure is called ETL (extract, transform, load) process and requires efficient extraction tools that are able to handle voluminous data. When analyzing the data confidentiality and privacy aspects need to be considered. The use of pseudonymization procedures in the ETL process might address this restriction. A second limitation derives from the current scope of application for the mining algorithm. The algorithm for mining financially relevant processes is only applicable for transactions that affect open-item-operated accounts. A purchase requisition sub-process for example does not directly affect open-item-operated accounts but might be relevant for financial audits. Further development for incorporating financially relevant processes and sub-processes that do not affect open-item-operated accounts might significantly enlarge the scope of application.

The presented methods provide a new approach of analyzing financial data in ERP systems. The presented algorithm achieves significant complexity reductions. But although the number of net elements for the most complex process instance on our sampled data was reduced by almost half from 275,870 to 146,620 the aggregated instance is still too complex for meaningful interpretation and evaluation. First observations show that similar structures among sub-graphs within instances might be common. Frequent sub-graph mining is a well-studied data mining problem (Huan et al. 2003; Yan and Han 2002) The development of aggregation procedures that merge sub-graphs within an instance by considering existing sub-graph mining algorithms might be a promising approach for further complexity reduction. Another obstacle to efficient evaluation is the huge amount of process instances that need to be considered. The analysis of mined process instances leads to the assumption that a great amount of process instances are very similar in structure. Further research is needed to investigate if efficient procedures that create clusters or categories across different process instances can be developed.

We have shown how a process mining approach can be used as a data analysis tool especially in the context of financial audits and how the complexity of mined instances can be reduced effectively. Further research is needed to answer open questions and to overcome existing limitations. But the methods discussed in this paper provide a meaningful milestone on the way for designing system based and automated analysis and audit procedures that cannot only be used in the context of auditing but also in the wider business context for example for performance measurement or optimization purposes.

Acknowledgement

The research results presented in this paper were developed in the research project Virtual Accounting Worlds. The project is sponsored by the German Federal Ministry of Education and Research. The authors are responsible for the content of this publication.

References

van der Aalst, W. M. P. 1997. "Verification of Workflow Nets," In *Application and Theory of Petri Nets*, P. Azema and G. Balbo (eds.), Berlin: Springer-Verlag, pp. 407–426.

van der Aalst, W. M. P. 1998. "The Application of Petri Nets to Workflow Management," *Journal of Circuits, Systems and Computers* (8:I), pp. 21–66.

van der Aalst, W. M. P. 2005. "Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing," *Requirements Engineering Journal* (10:3), pp. 198–211.

van der Aalst, W. M. P. 2011. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, (1st ed,)Springer Berlin Heidelberg.

van der Aalst, W. M. P., and Dongen, B. F. 2002. "Discovering Workflow Performance Models from Timed Logs," In *Engineering and Deployment of Cooperative Information Systems*, Y. Han, S. Tai, and D. Wikarski (eds.), (Vol. 2480)Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 45–63.

van der Aalst, W. M. P., van Dongen, B. F., Herbst, J., Maruster, L., Schimm, G., and Weijters, A. J. M. M. 2002. "Workflow Mining: a Survey of Issues and Approaches," *Beta: Research School for Operations Management and Logistics* (Working Paper 74).

van der Aalst, W. M. P., and Weijters, A. 2002. "Rediscovering Workflow Models from Event-Based Data," In *Proceedings of the Third International NAISO Symposium on Engineering of Intelligent Systems (EIS 2002)*Presented at the Third International NAISO Symposium on Engineering of Intelligent Systems (EIS 2002).

van der Aalst, W. M. P., Weijters, A. J. M. M., and Maruster, L. 2002. "Workflow Mining: Which Processes can be Rediscovered?," *BETA Working Paper Series* (Working Paper 74).

Agrawal, R., Gunopulos, D., and Leymann, F. 1998. "Mining Process Models from Workflow Logs," In *Proc. Sixth Int'l Conf. Extending Database Technology*, pp. 469–483.

Anandarajan, M., Anandarajan, A., and Srinivasan, C. A. 2004. *Business intelligence techniques□: a perspective from accounting and finance*, Berlin, Germany; New York: Springer-Verlag.

Brinkkemper, S. 1996. "Method engineering: engineering of information systems development methods and tools," *Information and Software Technology* (38:4), pp. 275–280.

Cook, J. E., and Wolf, A. L. 1998a. "Discovering models of software processes from event-based data," *ACM Trans. Softw. Eng. Methodol.* (7:3), pp. 215–249.

Cook, J. E., and Wolf, A. L. 1998b. "Event-based detection of concurrency," *SIGSOFT Softw. Eng. Notes* (23:6), pp. 35–45.

Cook, J. E., and Wolf, A. L. 1999. "Software process validation: quantitatively measuring the correspondence of a process to a model," *ACM Trans. Softw. Eng. Methodol.* (8:2), pp. 147–176.

Diestel, R. 2010. *Graph theory*, (4th ed,)Heidelberg; New York: Springer.

Gehrke, N., and Müller-Wickop, N. 2010a. "Rekonstruktion von Geschäftsprozessen im Finanzwesen mit Financial Process Mining," In *Lecture Notes in Informatics, Proceedings der Jahrestagung Informatik*Presented at the Jahrestagung Informatik 2010, IT-supported Service Innovation and Service Improvement, Leipzig.

Gehrke, N., and Müller-Wickop, N. 2010b. "Basic Principles of Financial Process Mining A Journey through Financial Data in Accounting Information Systems," In *Proceedings of the 16th Americas Conference on Information Systems*Presented at the Americas Conference on Information Systems, Lima, Peri.

Heckel, R. 2006. "Graph Transformation in a Nutshell," *Electronic Notes in Theoretical Computer Science* (148:1), pp. 187–198.

Herbst, J. 2000a. "A Machine Learning Approach to Workflow Management," In *Proceedings ofMachine Learning: 11th European Conference Machine Learning*, R. López de Mántaras and E. Plaza (eds.), (Vol. 1810)Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 183–194.

Herbst, J. 2000b. "Dealing with concurrency in workflow induction," In *European Concurrent Engineering Conference. SCS Europe*.

Herbst, J. 2003. *Ein induktiver Ansatz zur Akquisition und Adaption von Workflow-Modellen*, Tenea Verlag Ltd.

Herbst, J., and Karagiannis, D. 1998. "Integrating Machine Learning and Workflow Management to Support Acquisition and Adaptation of Workflow Models," In *Proceedings 9th International Workshop on Database and Expert Systems Applications (DEXA'98)*, pp. 745–752.

Herbst, J., and Karagiannis, D. 1999. "An inductive approach to the acquisition and adaptation of workflow models," In *Proceedings of the IJCAI* (Vol. 99), pp. 52–57.

Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. "Design science in information systems research," *Mis Quarterly* (28:1), pp. 75–105.

Huan, J., Wang, W., and Prins, J. 2003. "Efficient mining of frequent subgraphs in the presence of isomorphism," In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pp. 549–552.

March, S. T., and Smith, G. F. 1995. "Design and natural science research on information technology," *Decis. Support Syst.* (15:4), pp. 251–266.

Maruster, L., Van Der Aalst, W. M. P., Weijters, A., van den Bosch, A., and Daelemans, W. 2001. "Automated discovery of workflow models from hospital data," In *Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001)*, pp. 183–190.

Maruster, L., Weijters, A., van der Aalst, W., and van den Bosch, A. 2001. "Process mining: Discovering direct successors in process logs," In *Discovery Science*, pp. 32–36.

Maxeiner, M. K., Küspert, K., and Leymann, F. 2001. "Data Mining von Workflow-Protokollen zur teilautomatisierten Konstruktion von Prozessmodellen," In *Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), 9. GI-Fachtagung,*Springer-Verlag, pp. 75–84.

Müller-Wickop, N., Schultz, M., Gehrke, N., and Nüttgens, M. 2011. "Towards Automated Financial Process Auditing: Aggregation and Visualization of Process Models," In *Proceedings of the Enterprise Modelling and Information Systems Architectures*Presented at the EMISA 2011, Germany.

Neel, D. L., and Orrison, M. E. 2006. "The linear complexity of a graph," *the electronic journal of combinatorics* (13:R9), pp. 1.

Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., and Sinz, E. J. 2010. "Memorandum on design-oriented information systems research," *European Journal of Information Systems* (20:1), pp. 7–10.

Romney, M. B., and Steinbart, P. J. 2008. *Accounting Information Systems*, (11th Revised edition (REV),)Prentice Hall.

Rozenberg, G. 1997. *Handbook of Graph Grammars and Computing by Graph Transformation*, (illustrated ed, , Vols. 1-3, Vol. Volume 1 Foundations)Singapore: World Scientific Pub Co.

SAP. 2012. "SAP-UCC," available at http://www.sap-ucc.com accessed on 29th May 2012.

Schimm, G. 2000. "Generic Linear Business Process Modeling," In *Conceptual Modeling for E-Business and the Web*, S. W. Liddle, H. C. Mayr, and B. Thalheim (eds.), (Vol. 1921)Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 31–39.

Schimm, G. 2001a. "Process Mining Elektronischer Geschäftsprozesse," In *Proceedings Elektronische Geschäftsprozesse*.

Schimm, G. 2001b. "Process Mining linearer Prozessmodelle - Ein Ansatz zur Automatisierten Akquisition von Prozesswissen," In *Proceedings 1. Konferenz Professionelles Wissensmanagement*.

Schimm, G. 2002. "Process Miner — A Tool for Mining Process Schemes from Event-Based Data," In *Logics in Artificial Intelligence*, S. Flesca, S. Greco, G. Ianni, and N. Leone (eds.), (Vol. 2424)Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 525–528.

Turban, E., Aronson, J. E., Liang, T.-P., and Sharda, R. 2007. *Decision support and business intelligence systems*, Upper Saddle River, N.J.; London: Pearson Education International.

University of Hamburg. 2012. "Renew - The Reference Net Workshop," available at http://www.renew.de accessed on 29th May 2012.

Valk, R. 2008. "Lecture Notes: Formale Grundlagen der Informatik II (FGI 2) Modellierung & Analyse paralleler und verteilter Systeme,"Universität Hamburg.

Vercellis, C. 2009. *Business intelligence*, Chichester: Wiley.

Weijters, A., and Van der Aalst, W. M. P. 2001. "Process mining: discovering workflow models from event-based data," In *Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001)*, pp. 283–290.

Werner, M., and Gehrke, N. 2011. "Potentiale und Grenzen automatisierter Prozessprüfungen durch Prozessrekonstruktionen," In *Forschung für die Wirtschaft*Shaker Verlag.

Werner, M., Gehrke, N., and Nüttgens, M. 2012. "Business Process Mining and Reconstruction for Financial Audits," In *Proceedings of the 45th Hawaii International Conference on System Sciences*Presented at the 45th Hawaii International Conference on System Sciences, pp. 5350–5359.

Yan, X., and Han, J. 2002. "gSpan: graph-based substructure pattern mining,"IEEE Comput. Soc, pp. 721–724.