Process Mining: Conformance analysis from a financial audit perspective

Ron Hakvoort
Faculty of Economics and Business Administration, VU University, Amsterdam, The Netherlands
E-mail: ronhakvoort@gmail.com

Alexander Sluiter
Faculty of Economics and Business Administration, VU University, Amsterdam, The Netherlands
E-mail: a.f.sluiter@gmail.com

Abstract: In this paper, we present an application of process mining for the financial audit. We show that conformance analysis can be used as an audit technique in the execution of the financial audit. Using a generalized and simplified process model of an organization's procurement process, and the event log of SAP R/3, we detected a number of deviating process instances and fitting classes of transactions. We discovered numerous pros and cons of using conformance analysis as audit technique.

Keywords: process mining, conformance analysis, financial audit, assurance, risk based audit, internal control, key controls, process exceptions, anomalies, Petri Net, ProM, classes of transactions, transaction logging.


Biographical notes: Ron H.M. Hakvoort received his Master of Science in Business Information Technology, specialization Networked Business from the University of Twente, The Netherlands in 2005. Currently, he is an IT Auditor at an audit firm.

Alexander F. Sluiter received his Master of Science in Economics, specialization Business & ICT from the University of Groningen in 2005. He is currently IT Auditor at a Dutch insurance agency.

1 INTRODUCTION

Current auditing standards emphasize the importance of auditors gaining a broader understanding of an organization’s operations by performing risk assessments (i.e., assess the risks of material misstatement). Auditors’ ability to effectively analyze operations in the form of business processes is by definition a key determinant of their ability to appropriately plan and conduct the audit. (Carnaghan, 2005).

According to van der Aalst et al. (2003), the process mining concept has become a vivid research area. Until recently, the information in event logs of information systems was rarely used to analyze the underlying processes. Process mining aims at improving this by providing techniques and tools for discovering process, control, data, organizational, and social structures from event or transaction logs. The basic idea of process mining is to diagnose processes by mining event logs for knowledge (van der Aalst and De Madeiros, 2005). Process mining could be a useful tool for auditors to gain more knowledge about the actual business processes and enables a better risk assessment. The shortcomings in the financial reporting and auditing system exposed by scandals as Enron and Parmalat have illustrated the importance of effective auditing (Alles et al., 2006). As a consequence, section 404 of the Sarbanes/Oxley Act (SOx) requires both managers and auditors to verify controls over the firm’s financial reporting processes (Alles et al., 2006b). SOx has been a motivating force for the development of Continuous Auditing. This phenomenon could be defined as the process of continuously testing transactions and controls based upon criteria prescribed by the auditor and identification of anomalies (exceptions) for the auditor to perform additional
Many large companies have adopted commercial off-the-shelf Enterprise Resource Planning (ERP) systems to support their inter- and intra-business processes. Furthermore, midsize market firms are now also investing in ERP systems. Due to these facts, process mining can be used increasingly, since event logs become more and more available. (Wu, et al., 2007) (van der Aalst and De Madeiros, 2005).

Ingvaldsen and Guila (2006) applied process mining at the ERP environment of a mid-sized company in order to construct the underlying business flows. They showed that process mining provided new insights that can be used to improve the procurement process. We believe that process mining techniques could also be applied for execution of the audit instead of during the planning phase only. Van der Aalst and De Madeiros (2005) already explored the application of process mining techniques in security auditing. But so far, process mining research has mainly focused on process discovery and process improvement. Application of process mining during the financial audit execution phase is not explored (thoroughly) yet.

Auditors identify processes on basis of a risk based audit approach. Almost all processes serve a dual purpose: supporting the organizational goals, and second, minimizing the risk that certain threats negatively influence the organization. These processes are critical for the audit and have important influence on the procedures and evidence that auditors collect during the audit. (Knechel, 2001). With our research, we intend to correlate process mining to the field of auditing, particularly the audit execution phase. We aim to test our assumption that one of the process mining techniques, conformance checking, lends itself for auditing purposes. Conformance checking means that, based on the recorded events, it is checked whether a process instance matches a certain prescribed process model. A deviation could probably mean an undesired exception on the desired (i.e. controlled) process.

An general approach to interpret process mining results and to assess its practical implications is still lacking. Let alone that such framework exists for the interpretation of conformance analysis results from an audit perspective. This research contribution intends to gain further insight into this area.

## 2 RESEARCH APPROACH

A literature study will be performed on the process mining and conformance analysis concepts. Technical requirements for the log that can be used for process mining will be distinguished. Furthermore, the technical requirements of the prescriptive process model will be investigated.

Our research focuses on the procurement process. The main reason is that expenditures are a key risk area of the financial health of a company. The operating effectiveness of an organization’s procurement process is also one of the focus areas of the financial auditor during the financial statements audit. Another argument is that, given the extent of generality of the procurement processes and often relatively high transaction volumes, this kind of process lends itself particularly well for process mining exploration.

Also the requirements of the procurement process will be distinguished. It is important to know which activities have to be checked from an audit perspective in order to benefit the financial statement audit.

A business case will be performed in order to verify the usability and applicability of conformance analysis for the financial statements audit. The dataset that has been used, the performed steps, and the resulting findings will be described. Then the results will be evaluated and implications for the financial statements will be described. After this, advantages and disadvantages for the financial audit will be aggregated from the evaluation and results.

This paper is organised as follows; In chapter 3 the theoretical framework concerning conformance analysis and the financial statement audit will be discussed. Chapter 4 will describe the requirements of the log file and process model. The approach for conformance analysis will be described in chapter 5. In chapter 6 the procurement process will be explained. The approach has been tested with a business case, which is described in chapter 7. Chapters 8 and 9 contain respectively the reflections on the (dis)advantages and the conclusions about using conformance analysis within the context of the financial statement audit. The last chapter provides the topics for further research.

## 3 THEORETICAL FRAMEWORK

This chapter outlines the theoretical framework as used for our research. The first paragraph briefly explores the process mining and conformance analysis concepts. The last section focuses on aspects of (financial) auditing and combines both areas.

### 3.1 Process Mining and Conformance Analysis

The process mining concept is visualized by figure 1 (adapted from van der Aalst, 2005). Business’ operational processes are more and more supported, and even controlled, by information systems. Today, these information systems store relevant events in some structured form. For example, workflow management systems typically register the start and completion of activities. ERP systems like SAP log all transactions, e.g., users filling out forms, changing documents, etc. These examples show that many systems have some kind of event log often referred to as “history”, “audit trail”, “transaction log” (van der Aalst, 2005).
On basis of the information in the event log, the process mining technique can derive a process model. Depending on the process mining algorithm used, these models can differ. Each algorithm deals differently with duplicate and hidden tasks, process noise, loops, and etc. (Rozinat et al., 2007).

Figure 1: Process mining concept visualized

Rozinat and van der Aalst (2006) state that process models may be of a descriptive or of a prescriptive nature. Descriptive models capture existing processes without being normative. Prescriptive models describe the way that processes should be executed. Nowadays, many organizations implement workflow management systems (WMS) and enterprise resource planning (ERP) systems to enforce a particular way of working. Despite the implementation of prescriptive process models, people may deviate from the information system’s preferred way of process execution.

Auditors will rightly question if the process model and the log conform to each other. Conformance analysis aims at the detection of inconsistencies between a process model and its corresponding execution log. Cases that deviate from the desired process should be subject for further analysis by auditors. Rozinat and van der Aalst (2008) propose an incremental approach, consisting of several conformance dimensions, in order to check the conformance of a process model and an event log. The next paragraphs describe the dimensions fitness and appropriateness.

3.1.1 Fitness

First of all, the so-called fitness between the log and the model can be measured. Fitness means the extent to which the observed process complies with the control flow specified by the prescribed process model.

The fitness concept is demonstrated using a fictitious Petri Net model (see figure 2). The used Petri Net technique is a dynamic structure that consists of a set of transitions, represented by boxes and relate to some task, or action that can be executed, and a set of places, which are indicated by circles (Murata, 1989; Rozinat and van der Aalst, 2008)

Figure 2: Petri net model of a fictitious process

For instance, a workflow has been logged in the following order: A, B, D, E and A (case ABDEA). This trace of logged events can be replayed in the process model. The replay of every logical log trace starts with the marking of the initial place in the model. Then, the transitions that belong to the logged events in the trace are fired one after another (Rozinat and van der Aalst, 2008). It appears that the case can be mapped integrally on the process model (see figure 3).

Figure 3: Case ABDEA fits with the Petri Net model

Case ACHDEA seems not fit with the prescriptive process model. After replaying event A and C, H could not be fired (see figure 4).

Figure 4: Case ACHDEA does not fit with the Petri Net model

3.1.2 Structural and behavioral appropriateness

The appropriateness of the model can be analyzed with respect to the log. Does the model describe the observed process in a suitable way? Appropriateness can be evaluated from both a structural and a behavioral perspective.

Rozinat and van der Aalst (2006) claim that a good process model should somehow be minimal in structure to clearly reflect the described behavior, in other words structural appropriateness. Furthermore that same process model should be minimal in behavior to represent as closely as possible what actually takes place, which they call behavioral appropriateness.

To demonstrate structural and behavioral appropriateness, Rozinat and van der Aalst (2008) created two examples.
Figure 5 shows a process model which is of a too high level of abstraction, i.e. too generic. Both example cases of previous section fit with this process model.

![Process model (too high level of abstraction)](image)

The model represented by figure 5 is much too generic as it covers a lot of extra behavior. It allows for arbitrary sequences containing the activities A, B, C, D, E, F, G, or H.

![Process model (too low level of abstraction)](image)

The process model of figure 6 does not allow for more sequences than those that were observed in the log, but it only lists the possible sequences instead of expressing the specified behavior in a meaningful way. The model is too specific (Rozinat and van der Aalst, 2008).

### 3.2 Auditing and internal controls

The objective of the ordinary audit of financial statement is the expression of an opinion on the fairness with which they present fairly, the financial position, results of operations and its cash flows in conformity with generally accepted accounting principles (Arens et al., 2003).

The risk of misstatement in the financial statements can be reduced if the client has effective controls over computer operations and transaction processing.

An internal control is a process designed to provide reasonable assurance regarding the achievement of management’s objectives in the following categories: reliability of financial reporting, effectiveness and efficiency of operations and compliance with applicable laws and regulations (Arens et al., 2003).

The ability of the client’s internal controls to generate reliable financial information and safeguard assets and records is one of the most important and widely accepted concepts in the theory and practice of auditing. The process of identifying internal controls and evaluating their effectiveness is called assessing control risk. If internal controls are considered effective, planned assessed control risk can be reduced and the amount of audit evidence to be accumulated can be significantly less than when internal controls are not adequate. To justify this, the auditor must test the effectiveness of the internal controls. The procedures involved are called test of controls. For example, assume that an internal control requires the authorization of a manager when a purchase order exceeds $1,000. A possible test of effectiveness is to check whether all orders above $1,000 have been approved by the manager after checking the price and goods or services. Next to this substantive approach, it is also possible to audit if the system authorization and workflow is configured in such a way to guarantee that all orders above $1,000 must be approved by management.

When the results of these tests of controls support the control risk assessment below maximum, the auditor is able to reduce planned substantive testing for related accounts. Substantive tests are those activities performed by the auditor to gather evidence as to the completeness, validity and/or accuracy of account balances and underlying classes of transactions. For example, testing whether the ordered amount and price are the same on the purchase order and the invoice.

Conformance checking is useful for testing internal controls, since logfiles can be analyzed and checked whether all necessary steps have been taken in the right order.

### 4 REQUIREMENTS

In this section requirements will be provided in order to be able to use logs and transaction data for conformance analysis purposes. Furthermore it is important to use an adequate process model for conformance analysis. Requirements of process models will be given in section 4.2.

#### 4.1 Log requirements

In this section the requirements for event logs generated by ERP systems or workflow systems are provided. It is also possible to use transaction data in stead of event logs. The requirements for transaction data are the same.

The event log typically contains information about events referring to an activity and a case (van Dongen and van der Aalst, 2005). The case (also named process instance) is the matter which is being handled, e.g. a purchase order. The activity (also named task, operation, action, or work-item) is some operation on the case. Typically, events have a
timestamp indicating the time of occurrence. Moreover, when people are involved, event logs will typically contain information on the person executing or initiating the event, i.e., the originator. Other data about the case and/or task, i.e. attributes can be logged. Examples are price and amounts. The attributes which will be needed for the business case that will be introduced in paragraph 6. Based on this information several tools and techniques for process mining have been developed e.g. ProM, Aris PPM and the HP Business Cockpit.

For process mining, log files of such systems are needed as a starting point. When events are logged in some information system, we need them to meet the following requirements in order to be useful in the context of process mining (van Dongen and van der Aalst, 2005):

1. Each audit trail entry should be an event that happened at a given point in time. It should not refer to a period of time. For example, starting to work on some work item in a workflow system would be an event, as well as finishing the work-item. The process of working on the work-item itself is not.

2. Each audit trail entry should refer to one activity only, and activities should be uniquely identifiable.

3. Each audit trail entry should contain a description of the event that happened with respect to the activity. For example, the activity was started or completed.

4. Each audit trail entry should refer to a specific process instance (case). We need to know, for example, for which invoice the payment activity was started.

5. Each process instance should belong to a specific process.

On basis of these requirements, van Dongen en van der Aalst (2005) created a meta model for the information that should be used for process mining. With this meta model they introduced a formal XML definition for event logs, called MXML (Mining eXtensible Markup Language), to support the meta model. See figure 7 for the MXML mining format.

For the technical requirements of this format we refer to Günther and van der Aalst (2006).

The time stamp in a log (see point 1 of the above mentioned requirements) is very important because we are interested in the relation between attributes of the case and the actual route followed by a particular case. The sequence of the taken steps is important in the purchasing process, see section 6.1. The used log has to be sorted per case and all log entries have to appear in the order in which they took place (van der Aalst et al., 2003).

4.2 Process model requirements

As described in the former paragraph, the event log should meet a number of requirements. The process model that is used for checking the conformance of the event log should also meet some requirements.

It is important that the process model consists of all possible and permitted paths. If only the ideal process has been modelled, this possibly results in a high number of cases that do not fit the process model. These exceptions could be explainable and of minor importance for financial auditors.

In order to get a process model which can be used for conformance analysis with available modern software, the model must be in some Petri Net format (a .tpn or .pnml file). The software which can be used for conformance analysis is described in chapter 7.

5 CONFORMANCE ANALYSIS APPROACH

Based on our literature study, we propose a conformance analysis approach that could be applied in the field of auditing of procurement processes.

Probably only in the ideal world, a process model and a log have both 100% fitness, and behavioral and structural appropriateness. Rozinat and van der Aalst (2008) expect that in a practical setting the fitness dimension is typically more dominant. Therefore, they recommend carrying out the conformance analysis in two phases; (1) the analysis of fitness, and subsequently (2) the appropriateness of the model.

From an audit perspective, fitness is the most important conformance metric since auditors are particularly interested in the process instances that deviate from the ‘controlled’ and desired process.
In our approach, we propose the following steps for the conformance analysis:

A. Petri Net modeling
1. Defining the process model. Define and model the allowed flow of processes, resulting in a general prescribed process model;
2. Adding alternative paths. Add to this model the allowed loops and paths that skip certain process tasks. Apply an appropriate level of abstraction;
3. Extending the model. By application of the descriptive approach of process mining, the process model can be extended with newly discovered paths. In case these new variances are allowed from an internal control perspective, these paths could be added to the process model;

B. Pre-processing the raw log file
4. Renaming of logged events. In order to automatically link the events as modelled in the Petri Net to the events in the log file, it is necessary to align the naming of the events. In order to increase the comprehensibility of the results, it is also recommended to rename the event names of system-based process executions;
5. Removal of duplicate events. From a conformance analysis perspective it is not relevant to determine that for example five PO items are created in a row, as it does not impact the sequence of events with respect to the allowed process model. This pre-processing activity removes all the duplicate events. Only the events that are executed multiple times in a row will be aggregated to a single event;
6. Start/end event filtering. In order to maintain only the whole process instances (from purchase order creation to payment), the log file has to be filtered on process instances that start and end with a particular event. Instances that are split because of the cut off, will be ignored during the conformance analysis;
7. Removal of non-modeled events. In order to not to distort the conformance analysis results, non-modeled events could be removed from the log file. This has to be done with care, since it is not the purpose to. The steps that are not modeled, but that are present in the log file, should be evaluated on its impact. Since it is not the purpose to influence the results from an audit perspective. We assume that important events are added to the generic model in step 3.
8. Classes of transaction grouping. A rather technical pre-processing activity is to group all similar process sequences to one process instance. This has to be done in order to enable the log for conformance analysis.

C. Performing Conformance Analysis
9. Importing the Petri Net model. The Petri Net model that resulted from step A3 has to be imported in ProM.
10. Running the Conformance Checker. Now the conformance analysis can be initiated. The retrieved and pre-processed transaction log file has to be compared to the prescriptive process model, using the conformance checker.

D. Analysis of results
11. Analyzing the results. Having executed the conformance checker, the results, i.e. conform and non-conform process instances, have be to analyzed.

The abovementioned steps of phases B and C, have been described in more detail in the addendum.

6 PROCUREMENT PROCESS

6.1 Procurement process and controls
The overall objective in the audit of the procurement process (acquisition and payment cycle) is to evaluate whether the acquisitions of goods and services and the cash disbursements for those acquisitions are fairly presented in the accounts in accordance with generally accepted accounting principles (Arens et al., 2003). Within the cycles are three classes of transactions:
- Acquisitions of goods and services;
- Cash disbursements;
- Purchase returns and allowances and purchase discounts.

According to Arens et al. (2003), there are four business functions which occur in every business in the recording of the three classes of transactions in the cycles. These are:
- Processing purchase orders (the request for goods and services and the required approval for purchasing);
- Receiving goods and services (receipt of goods and services which after adequate control normally leads to recognizing the liability for an acquisition);
- Recognizing the liability (the prompt and accurate recording of the liability for the receipt of goods and services);
- Processing and recording cash disbursements (the payment including authorization and the recording of the payment).

For the business functions key controls have been identified.

Authorization of purchases. Authorization for acquisitions ensures that the goods and services acquired are for authorized company purposes and it avoids the acquisition
of excessive or unnecessary items. After the purchase requisition has been approved, a purchase order to acquire the goods or services must be initiated.

Separation of asset custody from other functions. When goods are received a receiving report should be issued from independent employees (other than acquisition). The goods should be controlled physically.

Timely recording and independent review of transactions. The propriety of acquisitions should be verified, this is typically the responsibility of the Accounts Payable department. Details of the purchase order are compared with the receiving report and the vendors invoice to determine that descriptions, prices, quantities, terms and freight on the vendor’s invoice are correct (3-way matching). Matching of the documents is nowadays often done by information systems. It is important that personnel who record the acquisitions have no access to cash and other assets.

Authorizations of payments. Most important controls for cash disbursements are authorization of payment by individual with proper authority, separation of responsibilities for authorizing and performing Accounts Payable function, examination of supporting documents by the one who authorizes at the time of authorization.

In Figure 8 the procurement process is visualized. Line \((a)\) represents the receipt of split orders. It is possible that a part of the order can not be delivered. The original order should be adjusted (line \((b)\)).

![Figure 8: Procurement process](image)

6.2 Procurement process and conformance analysis

Some of the abovementioned controls could be checked with conformance checking of event or transaction log. The following controls inhibit characteristics that could possibly lend itself for conformance analysis:

- The sequence of documents or registrations in the process. For example, there must be a purchase requisition before a purchase order. Goods or services are received after the creation of a purchase order etc.
- acquisitions are approved at the proper level;
- payments are approved at the proper level.

7 BUSINESS CASE

In order to verify the statements and observed opportunities of process mining in the field of auditing, a business case has been executed.

In this particular empirical study, the authors have chosen to run a pilot using real data from a SAP ERP system. Since SAP is a mainstream and widely implemented ERP system, this system has been picked for demonstrating the application of conformance analysis.

This chapter describes briefly the organization, from which the SAP data has been extracted, the tools that have been used for extraction and process mining activities, the process of defining the required data set to the final execution of the conformance analysis.

7.1 Data definition and resulting data set

The data has been extracted from a SAP system of a multi-national manufacturing company. The procurement process as implemented in SAP is used for the purchasing of both materials and services. In order to gather mineable data from the target organization that provides the necessary information on the procure-to-pay process, it is required to define a complete and extensive data definition.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Description</th>
<th>Field Name</th>
<th>Field description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBAN</td>
<td>Purchase Requisition</td>
<td>BANFN</td>
<td>Purchase requisition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BNFPO</td>
<td>Item of requisition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BSART</td>
<td>Document type</td>
</tr>
</tbody>
</table>

Table 1: Example of a SAP data definition

Because of the complexity of the SAP data model, it was necessary to download a total of 31 tables and 351 corresponding data fields.

Without defining constraints, the resulting data file would be tremendously large. In order to limit the file size and thus limit the number of records that will be included in the download, a condition has been added. In the example below, only the records between January 2nd 2007 and the January (10th) 2008 are extracted from the target SAP system.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Description</th>
<th>Field Name</th>
<th>Field description</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BKPF</td>
<td>Accounting Document Header</td>
<td>GLFHR</td>
<td>Fiscal year</td>
<td>2007/01/02-2008/01/10</td>
</tr>
</tbody>
</table>

Table 2: Example of a data condition

Limiting the fiscal year would cause unfinished or incomplete process instances, e.g. purchase orders registered in December 2007 and orders paid in February
2008. These unfinished and incomplete instances will be filtered and discarded at a later moment, in order to prevent noise during the analysis phase (step B6 of our approach).

7.2 Tooling

Several tools have been used in this research to extract and process the data, create the Petri Net, and perform the actual process mining and conformance analysis activities.

7.2.1 Data Extraction Tool

Based on a predefined set of table names, table fields, and optionally a set of additional conditions, the Data Extraction Tool is able to extract data from the target SAP system. The data extraction tool can be run from any desktop computer that is able to locate the target SAP system on the network. Any computer that also has an SAP GUI interface installed capable of connecting to the target system is suitable for this.

On the server side, no software needs to be installed. The only requirement on the server side is a user account which has been setup for the tool to use. This has to be a user profile with full read access but no modify or deletion privileges. This user can be a normal dialog user but a system user is preferred.

Using the read-only user profile, and given the data definition, the required data will be downloaded from the target SAP system. The Data Extraction Tool automatically encrypts the resulting output data, so the data could not be tampered before the data analysis is carried out by the auditor.

7.2.2 MXML translation tool

The SAP table transaction log extracted from the target organization is not suitable for process mining without performing any post-processing. A MXML log has been generated using scripts based on PHP and Java scripting language, and the correlation between the different tables using unique identifiers.

7.2.3 ProM

ProM is a pluggable environment for process mining offering a wide variety of plug-ins for process discovery, conformance checking, model extension, model transformation, etc. (Van der Aalst et al., 2007).

ProM will be used for performing the conformance analysis. The post-processing activities, described in chapter 5, can also be done with ProM. Next to grouping MXML logs, filtering and aggregating repeated processes, ProM can also convert a process model in a modeling paradigm other than Petri Net to Petri Net.

7.3 Procurement process model

In order to discover what possibilities conformance analysis has for the financial statement audit a simplified Petri Net model of the procurement process has been made. See figure 8.

Figure 9: ProM 5.0

7.2.4 Yasper

The procurement process has been modelled with Yasper (Yet Another Smart Process Editor, www.Yasper.org). Yasper is a tool for modeling and simulating stepwise processes.

Figure 10: Yasper

A Yasper process model shows the steps of a process and the order dependencies between them in one or more diagrams. The diagram technique supports alternative and parallel paths, repetitions of steps, and contention for resources between steps. Yasper uses extended Petri Nets as its modeling technique.

When the modeled Petri Net is saved as a PNML-file, it can be imported in ProM for conformance analysis purposes.

7.3 Procurement process model

In order to discover what possibilities conformance analysis has for the financial statement audit a simplified Petri Net model of the procurement process has been made. See figure 8.

Figure 11: Simplified procurement process

The process model contains two scenarios. The first one is the standard procurement process starting with creating a purchase order. The order is placed and next the ordered
goods are received, followed by receiving the invoice. This is the end of the process. The other modeled path consists of creating a purchase order and then receiving an invoice. This is a permitted path for two reasons. First, it can be a purchase order for services. In this case their will not be a ‘goods received’. The invoice has to be authorized by the person who received the services. Unfortunately, the available data in the event log misses this step.

Second, it is possible to have a contract which has to be paid immediately, but the goods will be delivered later on. The invoicing is done before delivering the goods or services.

The simplified model that is used in year X can be extended in year X+1 with more permitted paths when more is known about the procurement process to be analyzed.

7.4 Findings

In this paragraph the findings based on the practical experiences using a real life transaction log will be presented. The addendum that is enclosed to this article, provides a more detailed insight into the executed steps and intermediate results.

7.4.1 Pre-processing the raw log file

Table 3 provides an overview of the development of the different metrics during the described pre-processing activities, in order to prepare the log file for conformance analysis purposes.

<table>
<thead>
<tr>
<th>Metrics:</th>
<th>Filter/action</th>
<th>Raw log</th>
<th>Renamed events</th>
<th>Log w/o duplicates</th>
<th>Start/end event filter</th>
<th>Model items only filter</th>
<th>Grouped log</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processes</td>
<td>7,893</td>
<td>7,893</td>
<td>7,893</td>
<td>1.395</td>
<td>1.395</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>Cases</td>
<td>52,799</td>
<td>52,799</td>
<td>20,254</td>
<td>6,065</td>
<td>5,932</td>
<td>442</td>
<td></td>
</tr>
<tr>
<td>Events</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>12</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Event classes</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Event types</td>
<td>48</td>
<td>48</td>
<td>46</td>
<td>36</td>
<td>33</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Originators</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Metric overview of pre-processing activities

The unfiltered raw log file consists of 1 process, which is executed 7,893 times (represented by the number of cases). These cases consist of 52,799 events (activities). Furthermore, ProM reports that the log file consists of 13 different event classes. This means that there are 13 unique events executed (e.g. create purchase order, invoice receipt, good receipts, and etcetera). In this case one event type is covered in the log file, namely the completion of events (not for instance the cancellation, etcetera). It also appears that 48 originators (user or system accounts) are involved in the execution of the cases.

Out of 7,893 cases that were covered in the raw unprocessed log file, 1,395 cases (representing 38 classes of transaction) were relevant for conformance analysis. Running the allowed process model against the pre-processed log file, it appeared that 1,202 (86%) of these
cases were conform to the Petri Net model. The remaining 14% were reported as exceptions.

The majority of these so-called conformance exceptions tend not to be real exceptions. These exceptions were caused by the rather simplified Petri Net. Loops of various recurring goods and invoice receipts in a row were not modelled. The recurring events within one process instance, were reported as an exception.

Also more remarkable process instances have been found that need to be further investigated using the target SAP system:

- After creation of the purchase order (without a line item), directly an invoice is received. No other events occurred. This might be the purchasing of a service;
- After the creation of a purchase order of a line item, three invoices have been received. No goods receipt took place.

7.4.3 Analysis of remaining cases (known exceptions)

After the removal of the duplicate events, the start/end event filter has been applied on the log file. When this activity was not performed before running the conformance analysis, there would be a few known exceptions. This would distort the conformance analysis findings. An overview of the known exceptions:

- Because of the data extraction constraint, only the transactions in the log between 2-1-2007 and 10-1-2008 have been extracted from the target SAP environment. This implies that a number of incomplete cases are included in the current log file. The purchase order has been created in the previous period, but the goods receipt and/or invoice receipt takes place in the selected period;
- Another class of transaction that will arise as a known exception, is the purchase order that is created in the selected period, and the goods and/or invoice receipt takes place in the next period. This known exception is called the period-end cut off. 313 of these cases were identified;
- It also could happen that purchase orders have been created and cancelled before goods receipt and invoice receipt take place. A conceivable scenario is that the purchase order has been cancelled because of a mistake. Furthermore, SAP can create purchase orders automatically based on the MRP-run (MRP stands for Materials Requirements Planning). This means that when certain stock levels are running below critical values. SAP automatically creates purchase orders in order to increase the stock amount to the desired level. 5,089 cases were identified as a result of this known exception;
- Framework agreements. In case that the company concluded framework agreements with a number of its suppliers, not every delivery will have a separate sales order. The sales order (contract) will be created once, whereupon the goods and invoice receipt will take place. In 810 cases this scenario was identified.

<table>
<thead>
<tr>
<th>Known exceptions</th>
<th>Applied Start/End event filter</th>
<th>Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancelled or non-finished PO</td>
<td>Start Create PO - End Create PO / Create PO item</td>
<td>5,089 +</td>
</tr>
<tr>
<td>Cut off (period end)</td>
<td>Start Create PO - End All (except Create PO, PO item, H)</td>
<td>313 +</td>
</tr>
<tr>
<td>Framework agreements</td>
<td>Start &amp; End invoice Receipt / Goods Receipt</td>
<td>810 +</td>
</tr>
<tr>
<td>Cancellation of Goods Receipt</td>
<td>Start All (except Create PO) - End Goods Receipt cancellation</td>
<td>3 +</td>
</tr>
<tr>
<td>Return for PO</td>
<td>Start All (except Create PO) - Return for PO</td>
<td>11 +</td>
</tr>
<tr>
<td>Other reasons</td>
<td>n/a</td>
<td>272 +</td>
</tr>
<tr>
<td>Cut off (period beginning, etc.)</td>
<td>Start All (except Create PO) - End All</td>
<td>1,096 +</td>
</tr>
</tbody>
</table>

Sub total known exceptions: 6,498

<table>
<thead>
<tr>
<th>Input Conformance Analysis</th>
<th>Start Create PO - End Invoice Receipt</th>
<th>Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conformance</td>
<td>n/a</td>
<td>1,202 +</td>
</tr>
<tr>
<td>Non-conformance</td>
<td>n/a</td>
<td>493 +</td>
</tr>
</tbody>
</table>

Sub total input Conformance Analysis: 1,695

Total of all cases (Known exceptions + input Conformance Analysis): 7,693

Figure 13: Exception analysis

Out of 6,498 cases, 272 could not be traced back to a certain known exception.

7.5 Evaluation of the conformance analysis

In this paragraph the use of conformance analysis with ProM will be evaluated.

From this study it appeared that it is possible to:

- Filter the acquired raw log file on finished process instances only;
- Grouping of comparable process sequences to classes of transactions;
- Mapping of the activities of the Petri Net to the activities in the log file;
- Identify process sequences (classes of transaction) that did (not) fit with the desired process model.

Other practical experiences:

- Renaming of the process activities in the log file is necessary to get a better understanding of the process. By default, the process activities are quite technical. Another advantage of renaming these activities is easier mapping of the process activities with the activities in the Petri Net model. When the process activities are equally named, ProM provides a suggestion for the mapping;
- The way of creating (transaction) logs is inherent to the ERP system. Because of significant differences between systems like SAP, Oracle and JD Edwards, it is not possible to model a generic process that can be applied for conformance analysis;
- The Petri Net modeling technique using Yasper appeared to be difficult for modeling the allowed process model. Especially the modeling of loops seems to have drawbacks. Creating loops is necessary; otherwise every allowed path should be modelled separately. In theory you have to model A-B-C-D-E, A-
8 CONFORMANCE ANALYSIS AND FINANCIAL AUDIT

The assurance that conformance analysis provides for the financial statement audit will be described in this chapter. Advantages and disadvantages of conformance analysis with ProM will also be described.

8.1 Assurance

With the results of the conformance analysis in itself the auditor does not get the necessary assurance. A test of control is necessary for the part of the log which is conform to the modeled allowed process. The part of the log which is not conform has to be tested substantively. For instance the properties of the events can be tested using Linear Temporal Logic (LTL). Figure 13 visualizes the breakdown including the follow-up activities.

![Figure 14: Exception analysis](image)

A selection of the cases can be made based on significance for the financial statement audit. This is the level of materiality, the magnitude of an omission or misstatement of accounting data that mislead financial statement readers. The combination of the dollar amount and frequency determines which cases have to be considered by the financial auditor.

Although by the grouping of the MXML log (as described in chapter 5) the attribute values will be lost, individual cases can be tracked to the original log. The traceability is maintained, which is an important aspect for auditing.

8.2 (Dis)advantages

In this paragraph, the (dis)advantages of the conformance analysis as technique used in the financial statement audit are described.

**Advantages:**

- When substantive testing is necessary to get assurance for the financial statement, the part of the event log which has to undergo substantive testing can be reduced significantly with conformance analysis;
- Using the option grouped MXML (same sequences) in ProM, cases with the same sequence of activities in the log can be grouped automatically. Conformance analysis provides an interesting insight into the number of process instances that are present in the log within a class of transaction. During the study appeared that the two fitting classes of transaction were together responsible for 86% of the cases;
- ProM offers interesting possibilities for insight in log files. For example the statistics on the frequency of events (per instance), most common starting and ending events, involved originators, and etcetera provides an interesting overview of the extracted transaction log.

**Disadvantages:**

- Conformance analysis has been used to identify and filter the cases which did not follow the allowed paths. The part of the cases that did follow the allowed paths was discarded. When there is no 3-way matching in the system the auditor does not have assurance whether these discarded instances were correct. For example the amounts on the purchase order and the invoice might not be the same;
- The conformance analysis concept does not evaluate the attributes of the process. Although traceability of classes of transaction to the underlying process instances is possible, there is no straightforward way to analyse the attributes after performing conformance analysis. No dollar amount impairs the practical applicability of the conformance analysis approach from an audit perspective;
- Whether acquisitions or payments are approved at the proper level can not be tested with conformance analysis (paragraph 6.2).
9 CONCLUSION

Process mining is not a magic bullet: audit work with respect to e.g. authorization setup and segregation of duties is still necessary in order to get a sufficient level of assurance. It could be the case there is no segregation of duties and one person performs all activities within the purchasing process. The risk may exist that an unauthorized invoice is settled for payment. This could potentially result into fraudulent payments.

Conformance analysis provides a tool for a rather quick differentiation between conform and deviating process instances. The processes that tend to be conforming to the desired process model can be analyzed using other (statistical) techniques or the necessary assurance could be gathered using additional controls testing.

Furthermore, the conformance analysis implementation in ProM has a number of shortcomings to use it effectively for assurance in financial statement audits. Additional visualization and analysis functionality could make the process of analyzing the results more efficient.

10 FURTHER RESEARCH

In order to avoid false positives, i.e. processes that are reported as non-fitting but tend to be allowed, it is necessary to extend the Petri Net model with additional allowed paths and loops. Further research in creating extended process models in Petri Net which will work in ProM is necessary.

Behavioral appropriateness might be useful to analyze the classes of transactions. ProM shows additional information on relationships between the different process activities. For example which activities never follows or always follows another activity (C never follows D, and D always follows C). From control perspective, this dimension of conformance analysis provides interesting insights.

Also from the ERP perspective, additional research on transaction logs is required, in order to include also events like payment settlement, credit notes, and etcetera.

ACKNOWLEDGEMENT

We are grateful for the support and true interests of our coaches B. van Kuijck and M. Verdonk.

REFERENCES


WEBSITES

http://www.processmining.org

http://www.yasper.org
A. ADDENDUM: FROM A RAW MXML LOG FILE TO CONFORMANCE ANALYSIS

This addendum covers a rather detailed description of the activities that were performed in order to prepare the raw MXML log file for the performance of the conformance analysis. The activities are illustrated with clarifying screen dumps. The addendum is divided in the following sections:

1. Overview of the raw MXML log file
2. Pre-processing activities
3. Conformance Analysis
4. Results Breakdown

The steps refer to our proposed approach to conformance analysis for auditing, sections B and C.

1. Overview of the raw MXML log file

Importing the raw unprocessed MXML log file into ProM, results into the following summary overview:

![Dashboard Image]

This unfiltered log file consists of 1 process, which is executed 7,893 times (represented by the number of cases). These cases consist of 52,799 events (or activities). Furthermore, ProM reports that the log file consists of 13 different event classes. This means that there are 13 unique events executed (e.g. create purchase order, invoice receipt and good receipts). In this case one event type is covered in the log file, i.e. the completion of events (not for instance the cancellation or other events). It also appears that 48 originators (user or system accounts) are involved in the execution of the cases.

The center of the dashboard shows the number of events per case and the number of event classes per case. These distributions are visualized in a histogram. In provides an insight into the key characteristics of the log.
2. **Pre-processing of the MXML log file**

   **STEP B4: Renaming of event names**

   In order to make the log file more comprehensive, the original SAP movement names in the log file are renamed to human readable and understandable event names.

<table>
<thead>
<tr>
<th>Original event name</th>
<th>New event name</th>
</tr>
</thead>
<tbody>
<tr>
<td>101 - movement</td>
<td>101 Goods Receipt for PO</td>
</tr>
<tr>
<td>102 - movement</td>
<td>102 Cancellation Goods receipt for PO</td>
</tr>
<tr>
<td>103 - movement</td>
<td>103 Goods receipt for PO into GR blocked stock</td>
</tr>
<tr>
<td>104 - movement</td>
<td>104 Cancellation of 103</td>
</tr>
<tr>
<td>105 - movement</td>
<td>105 Release from GR blocked stock for PO</td>
</tr>
<tr>
<td>122 - movement</td>
<td>122 Return delivery to supplier (or to production)</td>
</tr>
<tr>
<td>161 - movement</td>
<td>161 Return for PO</td>
</tr>
</tbody>
</table>

   This has been done using the ‘Remap Element Log Filter’. This is one of the available default filter features in ProM:
The above settings result into the mapping below:

Renaming of the event names results into the following log summary:
It is obvious that the renaming of events does not affect the metrics. In the filter overview, it is visible that the event names are more comprehensive:

**STEP B5: Removal of duplicate events**

Looking at one of the process instances, it appears that repeated events occur sequentially. In the example below, the ‘Create PO item’ event is executed five times in a row. From a conformance analysis perspective, it is not relevant to determine that five PO items are created in a row, as it does not impact the sequence of events with respect to the allowed process model. It is sufficient to determine that one or more PO items are created. But from an audit perspective, it could be relevant to know the background of this repeated execution. For instance, it could be interesting to look into a case where 10 different invoices for one purchase order are received in a row. As conformance analysis cannot evaluate the attribute values, the auditor should apply other techniques to analyze the impact of this. An alternative is to add loops for each event in the process model. In that case, repetitive events will not be reported as deviations during the conformance analysis. The authors chose to reduce the complexity from a log point of view instead of increasing the complexity of the process model, by adding loops.
The ‘duplicate task filter’ supports the auditor to aggregate duplicate tasks. The filter is demonstrated below:
Once the ‘duplicate task filter’ has been applied, the number of process instances (cases) and the number of event classes remain the same. Only the number of events and the number of originators has been reduced:

The decreasing number of originators is caused by the removal of duplicate tasks that were executed by originators, that are not involved in the rest of the executed cases. In a sequence of multiple consecutive events, only the first event, including its attribute data (e.g. the originator) remains.

The process instance that originally had duplicate tasks as shown above, has been aggregated to the process instances as shown below:
**STEP B6: Start/end event filtering**

In order to remove all processes that do not start with the ‘Create PO’ event (Purchase Order) and all non-finished process instances (i.e. not ending with ‘invoice receipt’), the start/end event filter options as shown below have been configured:

The results in the following new filtered mxml log file:

The number of cases has dropped from 7,893 to 1,395 cases. Apparently, the other cases do not start with ‘Create PO’ and end with ‘Invoice receipt’. As a result, also the number of executed events within this population dropped to 6,065 events. One event class has been removed from the log. The only possible explanation is that this particular event type (‘104 Cancellation of 103’) was part of one or more cases that did not fulfill the start/end event filter requirements. Also less originators are involved in the population of remaining cases.
**STEP B7: Removal of non modelled events**

For the purpose of this empirical study, the complexity of the log has been reduced in accordance to the desired process model. Only the most trivial events are included in this analysis. This means that all events, except the events ‘Create PO’, ‘Create PO item’, ‘101 Goods receipt’ and ‘invoice receipt’ will be removed from the MXML log file. Only the cases that have one or more events that are also present in the Petri Net, remain in the log. Cases that have also other events, besides one or more of the four modeled events, also remain in the log file. However, the non-modeled events will be removed from the case.

This is done using the filter settings as shown below:

![Filter settings](image)

This results into the following log summary:

![Log summary](image)

The number of cases has not been reduced, since all the cases consist of a minimum of a ‘Create PO’ and ‘Invoice receipt’. This was a result of the previous filter activity. Obviously, the number of executed events has slightly dropped. Furthermore, the number of case events is now four. There are the same events as modeled in...
the Petri Net model. Apparently, there were also three originators who only executed cases that are not in scope anymore.

**STEP B8: Grouping of classes of transactions**

In order to be able to perform conformance analysis, it is required to group the instances of the same class of transaction. This can also be done using standard functionality, i.e. export the log file as a ‘Grouped MXML log (same sequences)’:

Grouping the same sequences, i.e. classes of transactions, results into the following log summary:

Each class of transaction is now represented as one case, since all cases with the same sequence have been aggregated to one case. This means that this log file inhibits 38 different classes of transactions. The number of originators has been reduced to zero, since attribute information has been aggregated for each class of transaction. This is also applicable for the timestamps (‘no timestamp information’). From an audit perspective, it is important to maintain traceability of the individual cases. ProM safeguards the traceability through keeping track of the individual case ID per class of transaction, as shown below (right upper corner):
3. Conformance analysis

**STEP C9: Importing the Petri Net model**

The simplified process model below, which has been designed using YaspeR, contains the sequence of events which are allowed. It is the “soll” position. With this model the conformance analysis will be performed.

The process model is imported into ProM:
**STEP C10: Running the Conformance Checker**

After running the conformance checker, the following figure appears:

![Conformance Checker Diagram](image1)

Using the ‘Select Fitting’ functionality, the tool determines automatically which instances do conform to the allowed process model. This results in the next figure:

![Conformance Checker Diagram](image2)

It appears that two classes of transactions, together responsible for 1,202 process instances (86.2%) match with the prescribed process model. The 193 process instances (13.8%) that do not fit with the process model are interesting from an audit perspective. These instances can be viewed using the ‘Invert Selection’ functionality:
Instead of a process model view with all the deviations, ProM has also the possibility to show the different log traces and problems that have been detected:
The resulting fitting or non-fitting selection of the log can be exported to a separate MXML-file. Performing the conformance checker on the two instances that do conform the process model, ProM shows a trivial fitting of 1.0.