VRIJE UNIVERSITEIT AMSTERDAM

INTERNSHIP THESIS

---

# Process Conformance in the Audit

**Identifying Bottlenecks in the Purchasing Process**

---

*Author:*
Koen Koopen

*Supervisors:*
Mark Hoogendoorn
Christiaan Dommerholt
Eduard Belitser

*Abridged Version*

*A thesis submitted in fulfillment of the requirements*
*for the degree of Masters in Business Analytics*

*at*

Vrije Universiteit Amsterdam
Grant Thornton - IT Audit

July 24, 2017

# Abstract

Process Mining is upcoming in the field of accountancy. More and more are accountants willing to explore alternative techniques in performing an audit. Improving the efficiency and decreasing the risk on errors are potentials Process Mining could contribute to. Process Mining can be described as techniques to explore, check and improve actual business processes. These business processes can be clarified through process discovery and checked using process conformance checking.

The purchasing process of 3 clients of Grant Thornton is considered. The goal is to combine both process discovery as well as conformance checking to create a model that can identify bottlenecks in the purchasing process. A purchasing process consists of the creation of a purchase order, possibly goods coming in and invoices being received. Next to that orders can be changed. For process discovery to be applied, an event log needs to be formed where each event at least contains an activity, timestamp and person responsible for the event. Conformance checking requires a discovery model representing the accepted process flow and an event log to be tested on this discovery model. The event log on which the discovery model needs to be build must be cleaned of bottlenecks, after which the conformance check is applied.

A discovery model must represent the underlying event log as best as possible. All traces in this log should come back in the model built. In addition, evaluation metrics are needed to determine which model represents the underlying event log the best. The fuzzy miner and the $\alpha$-algorithm are commonly known discovery methods, however the first technique neglects infrequent behavior where the other cannot handle real life data. This study requires a discovery technique taking all real life data into account, the Adapted Genetic Algorithm has been found most suitable. This technique is based on evolution, beginning with an initial population of models using operators of evolution like variation, selection and mutation to evolve the population towards a suitable discovery model for the underlying event log. This algorithm provides the fitness value, behavioral appropriateness as well as the structural appropriateness of the model. These 3 metrics together are used to evaluate the resulting models and determine which models can be used in the conformance check.

The conformance check is based on token replay and solely focusses on the sequences of events. Checks on additional event based data is also required in the audit, techniques to implement these checks into a conformance check are lacking. Nevertheless, both discovery as well as conformance results indicate clear differences between the purchasing processes of not only different clients, but also different companies of a client. This makes a conformance check over multiple clients very difficult besides the lack of applicable conformance techniques for auditing purposes. However, this does not mean Process Mining is useless for accountants. Discovery analysis can be very usefull for visualizing a process as well as for applying

one by one filters on this process to determine which orders contain bottlenecks.

# Preface

As the final part of the Masters Business Analytics at the VU Amsterdam, students must complete a graduation project in the form of an internship. This internship differs from all other parts of the Masters program, the student is required to spend 6 months doing full time research within an organization. Here the student will work on a predefined project which will result in a product of value for the host organization. This internship report is the end product of the internship as well as the graduation thesis.

The subject of this graduation project is Process Mining applied on the purchasing process of clients of Grant Thornton with respect to the annual audit. Grant Thornton is an accountancy & advisory firm of a size just below the traditional 'big four'. Many thanks to the people who guided me through this project. That is, Mark Hoogendoorn for his time and advice as VU supervisor on this graduation project, Eduard Belitser for being involved as second reader and Hajo Reijers for his time and advice during the research. In addition, Christiaan Dommerholt for being closely involved with the project as supervisor from Grant Thornton and thanks to all the other members of the IT Audit team for helping out when needed.

Amsterdam, Juli 2017

# Contents

# Chapter 1

# Introduction

## 1.1 Setting the stage

Nowadays, everything that happens within a company can be brought back to processes. Producing products, helping customers in a service line or developing a new product for a client are all processes going on in a business aimed to be as efficient as possible. In some processes it might be very interesting to visualize the actual process or to check up to which point the norm of that process is met. At the end, every company wants their business process to be as optimal as can be. All of this can be captured by Process Mining, which is the subject of this thesis.

Process Mining stands for techniques to analyze business process models and their execution traces (event logs). Process Mining provides methods for reconstructing process models from logs (so-called process discovery), checking the conformance of an existing or reconstructed model and a log (conformance checking), and enhancing process models based on the results of analysis (process enhancement) as stated in Accorsi and Stocker (2012).

Process Mining could be a useful tool in various business processes. That is, discovery can clearly visualize the actual business process within a company. For example an insurance claim handling procedure in an insurance company. When each event in the procedure is logged by the insurance company, process discovery techniques can be applied to visualize the actual insurance claim handling process(Rozinat et al., 2007). Conformance checking again can be used to highlight bottlenecks, such as violations of the 'four eyes' principle in the audit. What makes an event log the unique and potentially invaluable resource according to Jans, Alles, and Vasarhelyi (2010), is that it not only provides the auditor data on an event to analyze but at the same time additional data which is recorded automatically and independently of the person whose behavior is subject of the audit. Meaning that this data is free of noise due to human behavior. Furthermore, process enhancement could for example be used to avoid miscellaneous claims in case of the insurance company by optimizing their process model.

Challenges for this research are the creation of the event logs, the specification of the norm log out of the created log and the choice of suitable modeling techniques to built the eventual process conformance model. The norm log will be the final log where all bottleneck traces are filtered out. This log will solely consist of accepted traces which can be used to build a discovery model on. Such a discovery model can than be used as input for the conformance check. The creation of an event log which is applicable in the audit and can be process mined is no straightforward action. Future information systems may anticipate on the value of process mining and

facilitate the extraction of event logs from ERP databases better, but at this moment this still requires a lot of manual effort (Jans, Alles, and Vasarhelyi, 2010). Events logged in information systems did happen at a given point in time, refer to only 1 activity and a process, contain a description and refer to a specific case in order to be usefull for process mining (Hakvoort and Sluiter, 2008). The compliance rules in the audit need to be checked, however as discussed in (Ramezani, 2017) it can be a challenging task to implement a business process compliance check with various rules.

Applying process mining in auditing has been subject of various researches done [Jans, Alles, and Vasarhelyi (2010), Accorsi and Stocker (2012), Jans, Alles, and Vasarhelyi (2014), Dommerholt (2015) and Rozinat and Aalst (2008)]. Here, Dommerholt (2015) scouts the applicability of process mining in the audit without putting it to practice. However, most of these researches mentioned focus on applying process discovery in a case study, where Rozinat and Aalst (2008) is focusing on matching recorded events with a model. This research contributes to this field of research by combining all these aspects in a case study in order to provide a useful audit tool. That is, applying process discovery on created event logs to form a norm log and model, in order to apply process conformance checking aimed at identifying bottelnecks in the actual process.

## 1.2 Business Context

### 1.2.1 Setting the Stage

Accountancy firms aim at finding inconstancies in a clients books while performing an audit on that client. Here the accountants check the paper trail of a clients process such as the purchasing or sales process. Process Mining can be a useful tool to perform this check in the audit. Within Grant Thornton there is more and more willingness to apply Process Mining in business. A special interest group on Process Mining has been formed which indicates that the applicability and added value of Process Mining is acknowledged within the company. However, there is not enough technical(modeling & programming) knowledge to apply its techniques (Dommerholt, 2015). Meaning that this research project is scouting this subject within Grant Thornton Netherlands when it comes to applying Process Mining techniques.

Process Mining can be used as a tool in the audit. It aims to discover, monitor and improve real processes by extracting knowledge from event logs as stated in Aalst (2014). Being able to clarify and check these actual business processes of clients can be very helpful for accountants in the audit. In the current situation accountants use very simple compliance rules and check deviating traces of event logs manually by checking a random sample out of the pool with deviating traces. This comes with a factor of uncertainty but also inefficiency. Because not every trace is being checked individually, bottlenecks can slip through. Process Mining looks for more complex relations in the data as well, this can be of significant added value. That is because this could lead to results that would not have been found by the accountants in the usual manual way. However, the event log to be used here needs to be created and must have certain characteristics to be applicable. Critical steps here are the identification of activities and the selection of a process instance (Jans, Alles, and Vasarhelyi, 2010).

### 1.2.2 Business Problem

The process in the scope of this thesis is the purchasing process of clients of Grant Thornton. Where depending on the results, this could be a start of applying Process Mining on other business processes within the department. The purchasing process is an often used example process in the literature on Process Mining. Next to that, this process is of significant importance in the annual audit where it is generalizable up to some point for different clients. In this research project the data considered is from the clients Client A, Client C and Client B. Each of these clients uses SAP as ERP system to store their business process related data in a database. This research aims at getting a better view of their actual purchasing processes and correctly detecting bottlenecks within the process. Here we are especially interested in the traces which differ from the intended norm protocol and the reason for this to happen.

A goal of this thesis is to form an activity log dataset for SAP clients which again can be used to perform process discovery. This discovery analysis can be seen as preparation for building a conformance checking model. Although, the analysis by itself could also be very useful for the accountants because it provides a helping hand towards the visualization of the purchasing process. However, to identify the bottlenecks in the process conformance checking has to be applied. Introducing another challange for this research, building a model that can identify bottlenecks in traces. That is, when the purchasing process is considered an auditor should be able to use the model to be built in combination with the discovery analysis to visualize the actual purchasing process of a client and perform a conformance check. Expected is that this could result in risk based audit evidence and more efficiency. Especially considering the current data analysis being done by the IT-Audit team within Grant Thornton on the purchasing process. Here the focus mainly lies only on checking segregation of duties, quantities and order values, meaning that there is room for improvement.

There has not been much focus yet on converting data into useful information contributing to insights on the purchasing process for clients of Grant Thornton. So for example, how does a trace of actual event logs look like and how often is it according to the standard of that process. This research project will look into these issues where the goal is to get more insight in Process Mining as a tool to check the purchasing processes of clients. Here we are especially interested in the traces which differ from the intended protocol and the reason for this to happen. So the global question that needs to be answered is:

*"Can we build a Process Mining model which supports the accountants of Grant Thornton, by correctly detecting bottlenecks through a conformance check on the purchasing process of a client?"*

To answer this research question, it is split up in 3 sub-questions which will be touched one by one during this research. First problem encountered is the creation of the event log from the SAP data, leading to the first sub-question:

- *"Which specific SAP data is needed from the clients to create an event log suitable for process mining usage and in which way has this data to be processed to form this log?"*

When the event logs can be formed the process discovery stage is next. The activity log as a whole needs to be analyzed to get a better understanding of the actual process and the filters that need to be applied to end up with the norm log. Meaning that the next problem in this research is:

- *"Which process discovery techniques are suitable to clearly vizualize the actual purchasing process of the clients in order to built a normative model?"*

After answering this sub-question a discovery norm model can be built from the norm log which could serve as input for a process conformance model. In order to compare the built model with 'new' data that needs to be checked on bottlenecks, a conformance check model is needed that is as precise as possible. Leading to the final sub-question in this research:

- *"What process conformance technique is most applicable with respect to the constructed discovery model for creating a conformance check model in terms of correctly identifying bottlenecks in the process?"*

## 1.3 Outline

First focus of this research is the creation of an event log containing the right properties for being used as process discovery input. That is, data needs to be acquired, cleaned and combined into the resulting event log for each client. After that, discovery techniques and filters to be used need to be determined to create a norm log and model serving as input for the process conformance. The modeling techniques chosen here for both discovery and conformance determine the outcome of this thesis study, meaning that both need to be optimized for the best results.

First, this report will focus on the theory of Process Mining in section 2.1. After which previous research done is discussed in section 2.2 followed by background on the purchasing process in SAP(2.3). After that, chapter 3 will go into the process discovery and conformance methods and models used in this reseach. Next, the data used in this thesis will be described(4.1) including how it is preprocessed(**??**). In addition, the use and results of interviews taken with accountants is described in sections 4.3 and 5.2. The discovery results over the datasets are presented in 5.1 after which this report will focus on the models that have been built. These results are finally used in chapter 6 to draw conclusions on the project and give recommendations for further research.

# Chapter 2

# Literature Review

## 2.1 Process Mining

Process Mining can be split into 3 different spectra, that is: discovery, conformance checking and process enhancement or so called model extension (Aalst et al., 2010). These spectra are visualized in figure 2.1. The names of the spectra already indicate their use, where the figure gives an initial indication as well. However, in this section we will go into each of the 3 spectra and give some examples of algorithms that can be applied for some of them.

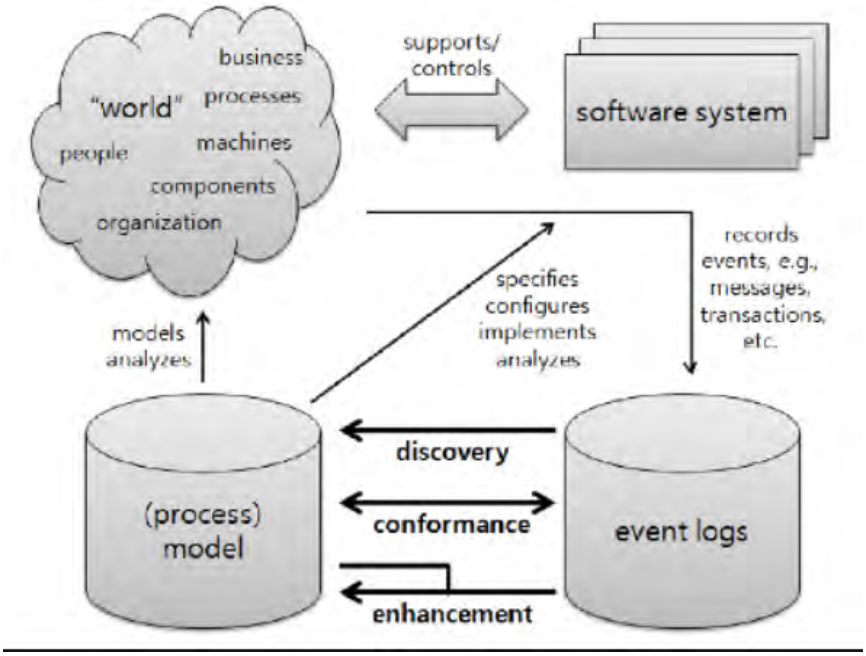



FIGURE 2.1: Process Mining spectra (Aalst, 2016)

### 2.1.1 Process Discovery

A process discovery method is defined as a method for reconstructing process models from an event log consisting of a collection of traces(Accorsi and Stocker, 2012). This model is produced without any a-priori information. This way of Process Mining can be used to get a clear image of the actual process without having a predefined model. Data from practice is translated into a model giving systematical view of the process. A common example of a discovery technique is the $\alpha$-algorithm, which takes an event log and produces a Petri net out of it. A Petri net is a set of transitions indicated by boxes related to some task or action that is executed. Furthermore, a

Petri net contains circles called places which can hold tokens, directed arcs connect the transitions and places. For example consider figure 2.2, here the transitions are '(EKKO-EKPO)Purchase Order created' and '(CDHDR) Insert: Create Purchase Order'. The circle inbetween is a place which will hold a token after a purchase order is created, this token will then be consumed when the order is inserted. The arrows between the transitions and the place are directed arcs expressing the connections. In this section we will go into the theory of 2 discovery algorithms and give an example.

For a discovery problem we can define $L$ as being an event log which is multi set of traces over $\mathscr{A}$, where $\mathscr{A}$ is a set of activities (Aalst, 2014). A process discovery algorithm can be seen as "a function that maps an event log $L$ onto a process model such that the model is 'representative' for the behavior seen in the event log". For a discovered model to be representative, it should be able to replay all behavior in the log. This can be defined as the fitness of the discovered model, i.e. the model should allow for the behavior seen in the event log. However, there are 4 quality metrics for a mined model and there is a tension between these metrics in a way that they contradict making it hard to balance them out. According to Accorsi and Stocker (2012) the quality of a mined model varies along the following 4 quality metrics:

- *Fitness*: The fitness measure determines the amount of log behavior considered in the model.
- *Precision*: The resulting model can allow deviating paths from the process behavior. Precision reflects the over all degree of deviance between log and model.
- *Generalization*: On the other hand generalization measures how close a model stays to the behavior seen in the log, i.e., up to which amount the log is generalized by the discovered model.
- *Simplicity*: The simplicity of the discovered model reflects if the model does not have an unnecessary complicated structure.

It is a challenging task to build a discovery model that covers all 4 criteria evenly well. For example, when a model is oversimplified it is likely to have a lack of precision or low fitness. Aim is to look for a process model which is observed minimal in structure, structural appropriateness, so that it clearly reflects the behavior in the log. But at the same time the model should be minimal in behavior as well to represent as closely as possible what actually takes places, called behavioral appropriateness(Hakvoort and Sluiter, 2008).

The $\alpha$-algorithm takes event log $L$ over activity set $\mathscr{A}$ as input. These activities are corresponding to transitions in the discovered Petri net which is the output of the algorithm. Here we focus on so called 'work flow nets'(WF-nets) having a unique sink and source place. The algorithm goes through the event log looking for patterns or so called 'Log-based ordering relations'. Here Aalst (2014) defines 4 relations which can be found in log $L$:

- $a >_L b$ if and only if there is a trace $\sigma = <t_1, t_2, t_3, ..., t_n>$ in log $L$ where event $a$ is followed by event $b$

- $a \rightarrow_L b$ if and only if $a >_L b$ and $b \not>_L a$
- $a \#_L b$ if and only if $a \not>_L b$ and $b \not>_L a$
- $a ||_L b$ if and only if $a >_L b$ and $b >_L a$

An example of how these relations patterns can be visualized is given in figures 2.2, 2.3 and 2.4. For example in figure 2.3 'Purchase Order created' and 'Debit: Invoice receipt' are both followed by 'Debit: Goods receipt' but never the other way around and they also never follow each other. This means that:

Purchase Order created $>$ Debit: Goods receipt
Debit: Invoice receipt $>$ Debit: Goods receipt
Purchase Order created $\not>$ Debit: Invoice receipt
Debit: Goods receipt $\not>$ Debit: Invoice receipt
Debit: Goods receipt $\not>$ Purchase Order created
Debit: Invoice receipt $\not>$ Purchase Order created

Which can be expressed by:

Purchase Order created$\rightarrow$ Debit: Goods receipt
Purchase Order created$\rightarrow$ Debit: Invoice receipt
Debit: Goods receipt# Debit: Invoice receipt

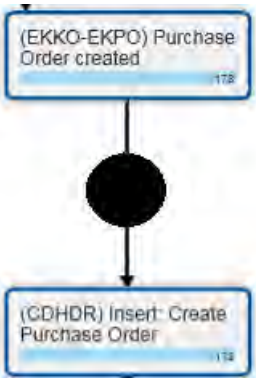
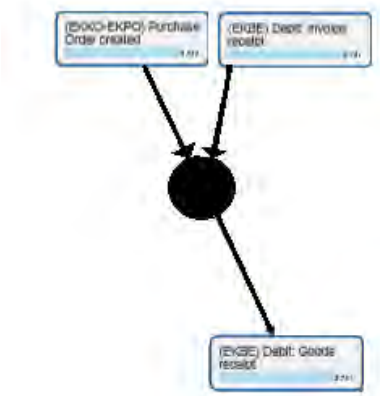
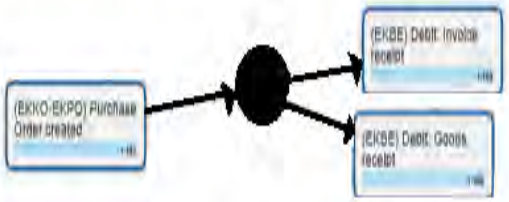



FIGURE 2.2: Sequence pattern: PO created $\rightarrow$ Insert: Create PO

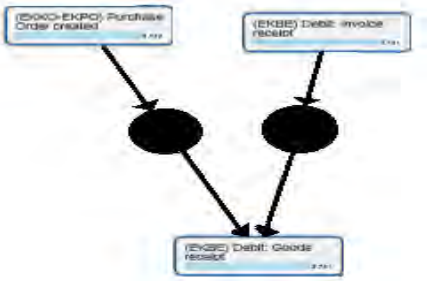



(A) XOR-join pattern:
PO created→Goods Receipt,
PO created→Invoice Receipt,
and Goods Receipt#Invoice Receipt

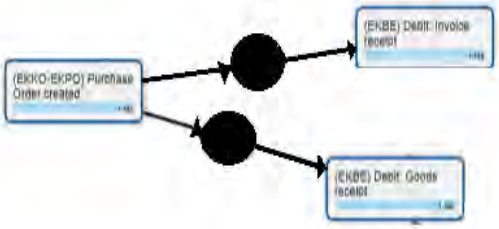



(B) XOR-split pattern:
PO created→Goods Receipt,
Invoice Receipt→Goods Receipt,
and PO created#Invoice Receipt

FIGURE 2.3: Log based XOR relations




(A) AND-Join pattern:
PO created→Goods Receipt,
Invoice Receipt→Goods Receipt,
and PO created||Invoice Receipt




(B) AND-split pattern:
PO created→Goods Receipt,
PO created→Invoice Receipt,
and Goods Receipt||Invoice Receipt

FIGURE 2.4: Log based AND relations

Now, Aalst (2016) describes the $\alpha$-algorithm by 8 rules. The algorithm checks in step 1 which activities appear in the log that correspond to the transitions of the generated work flow net. In its next 2 steps $T_I$ is formed, being the set of start activities, as well as $T_O$ being the set of end activities. That is all activities which respectively appear first and last in some trace. The next 2 steps form the heart of the $\alpha$-algorithm. Here the places $p_{(A,B)}$ are constructed with $A$ being the set of input transitions and B the set output transitions. The algorithm needs to determine $p_{(A,B)}$ here because all elements of A should have causal dependencies with all elements of B. In addition, elements of A should never follow each other which also holds for B stated in step 4. Where in step 5 all non-maximal pairs are removed. In the next step, all places $p_{(A,B)}$ plus the unique source and sink places are gathered in a set $P_L$. In step 7 all arcs are generated between the places in $P_L$, where finally the resulting Petri net is formed (Aalst, 2016).

An example using the following event log:
$L = [<$PO created, Insert: PO created, PO released, Invoice receipt$>^2$,
$<$PO created, Insert: PO created, Update: PO changed, PO released, Goods receipt,

Invoice receipt$\rangle^3$,
$\langle$PO created, Insert: PO created, Update: PO changed, Goods receipt, Invoice receipt$\rangle^2$,
$\langle$PO created, Insert: PO created,Update: PO changed,Invoice receipt, Goods receipt$\rangle^4$].
Here the numbers on the right top of each trace indicate how often this trace occurs in the log. Given the event log $L$ the footprint matrix can be computed which is shown in 2.1, this is done in the same way as previously with the example on figure 2.3 according to the criteria mentioned on log-based ordering relations. Something that stands out here, and which holds for all footprint matrices, is that the matrix is symmetric except for the arrows which turn direction. This property of a footprint matrix can be drawn back to the final 3 out of the 4 log-based ordering relations, when a and b are swapped the same relations will be found only with the arrow relation pointing in the opposite direction.

| | PO created | Insert PO | PO changed | PO released | Goods rec. | Invoice rec. |
|---|---|---|---|---|---|---|
| PO created | # | $\rightarrow$ | # | # | # | # |
| Insert PO | $\leftarrow$ | # | $\rightarrow$ | $\rightarrow$ | # | # |
| PO changed | # | $\leftarrow$ | # | $\rightarrow$ | $\rightarrow$ | $\rightarrow$ |
| PO released | # | $\leftarrow$ | $\leftarrow$ | # | $\rightarrow$ | $\rightarrow$ |
| Goods rec. | # | # | $\leftarrow$ | $\leftarrow$ | # | \|\| |
| Invoice rec. | # | # | $\leftarrow$ | $\leftarrow$ | \|\| | # |

TABLE 2.1: Footprint of L

Using this event log the footprints shown in table 2.1 the 8 rules of the $\alpha$-algorithm can be applied which eventually leads to the model shown in figure 2.5.

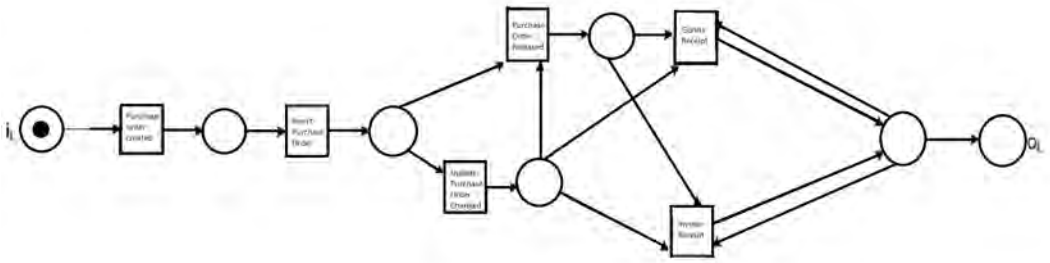


FIGURE 2.5: Work flow net derived from L using the $\alpha$-algorithm

Final example of a discovery algorithm is the Genetic Algorithm. Genetic Algorithms are based on the process of evolution which is also known as Evolutionary Computing. Evolutionary computing is a research area within computer science being a special flavor of computing, which draws inspiration from the process of natural evolution. The power of evolution in nature is evident in the diverse species that make up our world. Evolutionary computing relates this powerful natural evolution to a particular style of problem solving, that of trial-and-error(Eiben and Smith, 2003). In a given environment a population of individuals strive for survival and reproduction. Their fitness is determined by the environment, relating to how well they succeed in achieving their goals. Variation operators create new individuals from old ones through crossover and/or mutation. Mutation is a unary variation operator applied to a genotype resulting in a slightly different individual. Crossover or recombination combines information of multiple individuals to create a new individual. The common underlying idea behind all Evolutionary

Computing techniques is the same: given a population of individuals within some environment that has limited resources, competition for those resources causes natural selection(survival of the fittest)(Eiben and Smith, 2003). The most important components of an Evolutionary based algorithm are:

- Representation
- Evaluation function
- Population
- Parent selection
- Variation operators
- Survivor selection

A Genetic Algorithm(GA) is the most widely known Evolutionary algorithm and can be summarized by the standards in table 2.2. This means that a Genetic Algorithm is represented by strings of bits. These bits could be boolean values where 1 represents true and 0 false resulting in a natural genotype-fenotype mapping. Another example could be bit-strings encoding other nonbinary information, such as a bit-string of 80 bits representing 10 integers(Eiben and Smith, 2003). Recombination in a GA is applied through 1-point crossover. Here a random number r is chosen in the range 1 to l-1 (with l the length of the string), after which both parents are split at this point to create two children by exchanging the tails. Meaning that the first child receives the first r bits from the first parent and the bits r+1 until l from the second parent (Eiben and Smith, 2003). Parent selection in a GA is proportional to the fitness of the individual. That is, the fitness of 1 individual relative to the sum of all fitness values of all individuals is the chance for that individual to be selected as parent. Genetic algorithms use generational survivor selection, meaning that all individuals only live 1 generations after which they will be replaced by their offspring. Alternative option here is to combine generational survival with an elitism rate, this means that a top percentage of the population in terms of fitness is automatically copied into the next generation(Medeiros, Weijters, and Aalst, 2007).

| | |
|---|---|
| Representation | Bit-strings |
| Recombination | 1-Point crossover |
| Mutation | Bit flip |
| Parent Selection | Fitness proportional - implemented by Roulette Wheel |
| Survival Selection | Generational |

TABLE 2.2: Sketch of a Genetic Algorithm (Eiben and Smith, 2003).

The bigger a population is in an Evolutionary Algorithm, the more diversity will be present within the population. This means that the mating pool for creating an eventual best individual is larger, or in other words the search space is bigger. Examples of building an initial population are randomly, using existing solutions or by seeding a random population with one or more known good solutions (Eiben and Smith, 2003).

An example of such an algorithm, described in Medeiros, Weijters, and Aalst (2007), is what they call the Genetic Algorithm. However, this technique does not

comply with a Genetic Algorithm as defined in section 3.1. That is, the representation is no bit-string but a causal matrix where each task has a set of input and output tasks describing which task enable the execution of other tasks. The causal matrix of the example case in figure 2.5 is shown in table 2.3.

| Activity | Input | Output |
|---|---|---|
| Purchase Order Created | {} | {Insert Purchase Order} |
| Insert Purchase Order | {Purchase Order Created} | {Purchase Order Changed, Purchase Order Released} |
| Purchase Order Changed | {Insert Purchase Order} | {Goods Receipt, Invoice Receipt, Purchase Order Released} |
| Purchase Order Released | {Insert Purchase Order, Purchase Order Changed} | {Goods Receipt, Invoice Receipt} |
| Goods Receipt | {Insert Purchase Order, Purchase Order Changed, Invoice Receipt} | {Invoice Receipt} |
| Invoice Receipt | {Insert Purchase Order, Purchase Order Changed, Goods Receipt} | {Goods Receipt} |

TABLE 2.3: Causal matrix example

Their crossover technique is derived from 1-point crossover only adapted to be applied on a causal representation. The split is done on each task its input and output set with a given crossover probability. Mutation is done by randomly changing, removing or adapting the causal set of a task with a given mutation probability. Finally, Medeiros, Weijters, and Aalst (2007) their algorithm uses tournaments for selection. That is a tournament of respectively 2 or 5 randomly drawn individuals determine who will go into the next generation or form the group of parents that can produce offspring. In such a tournament all individuals will compete against each other based on fitness after which the individual with the most wins will be selected as parent/survivor (Medeiros, Weijters, and Aalst, 2007). This means that the algorithm of Medeiros, Weijters, and Aalst (2007) is not a Genetic Algorithm as described in Eiben and Smith (2003), for that reason will from here on be refered to the 'Adapted Genetic Algorithm'(AGA).

### 2.1.2 Conformance Checking

The second spectrum in Process Mining is conformance checking. Conformance check problems are problems where based on recorded events there needs to be checked whether a process instance matches a certain prescribed process model. A deviation here could mean an undesired exception on the desired process (Hakvoort and Sluiter, 2008). Conformance checking techniques can be used to check if reality, which is recorded in the log, is conform to the model or the other way around. Figure 2.6 illustrates the main idea of conformance checking as it is just described.

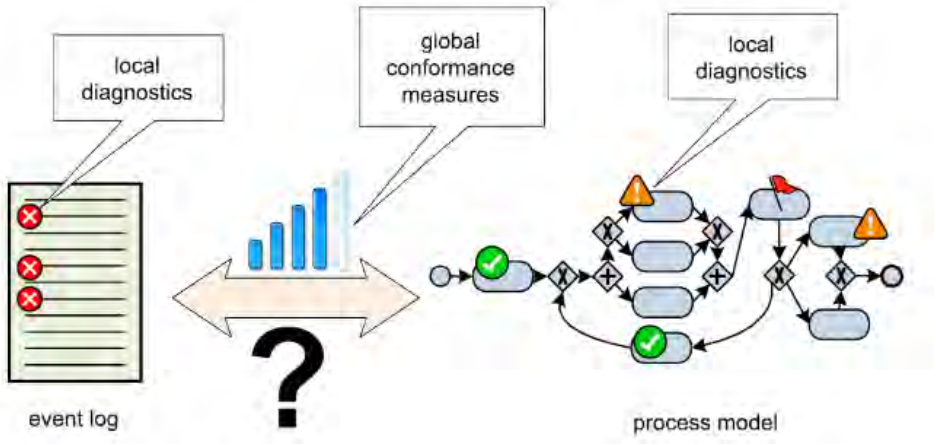



FIGURE 2.6: Conformance checking: comparing observed and modeled behavior(Aalst, 2016)

Where process discovery focuses on the control-flow perspective, in conformance checking other perspectives like time, case and organizational perspectives are considered. An example of this is checking the 'four-eyes' principle. This states that certain activities should not be executed by one person completely on his own (Aalst, 2016). A way to measure the fit between the event log and the process model is to replay the log in the model and show mismatches (Rozinat and Aalst, 2008).

A naive simple approach for conformance checking is to simply count the fraction of traces that can be replayed perfectly. Here we could measure the fitness by

$$\frac{\text{Total \# replayed correctly}}{\text{Total \# traces}}$$

However, this approach does not make a difference between almost perfectly replayed traces and traces that do not fit at all. Therefore, Aalst (2014) suggests to use a fitness notion defined at the level of events rather than full traces. While replaying, the number of tokens needed to be created artificially and the number of tokens left in the model are respectively counted as missing tokens $m$ and remaining tokens $r$. Next to this, the produced tokens $p$ and consumed tokens $c$ are being counted (Rozinat and Aalst, 2008). At the start of the replay $p = c = 0$ and the places are empty. Next, a token is produced for the starting place meaning that $p = 1$. After which all events of a trace being fired which are line with the work flow net will lead to the production and consumption of tokens. When a token is missing or remaining in the work flow net, this is captured by $m$ and $r$. Here, Rozinat and Aalst (2008) defines the token based fitness metric, with $n_i$ the number of process instances combined into the current trace and $k$ being the number of different traces, as follows:

$$f = \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}) + \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i})$$

Although this approach is easily understood and implemented, plus the fact that it can differentiate between fitting and non-fitting cases, it has some drawbacks. Fitness values will be too high for extreme event logs. Furthermore, if a case does not fit the approach will not create a corresponding path through the model. To avoid

these limitations alignments can be used (Aalst, 2016), which will be explained using an example.

Having the trace $\sigma = <$PO created, Insert: PO created, PO released, Invoice receipt$>$ and the model in figure 2.5, it shows that $\sigma$ perfectly fits in the model. Aalst, Adriansyah, and Dongen (2012) define a so-called optimal alignment as an alignment which 'distance' is smaller or equal as all other alignments, being the best match given a trace and a model. Meaning that for $\sigma$ and the model in figure 2.5 there is 1 optimal alignment namely:

$$\gamma_1 = \begin{array}{|c|c|c|c|} \hline \text{PO created} & \text{Insert:PO created} & \text{PO released} & \text{Invoice receipt} \\ \hline \text{PO created} & \text{Insert:PO created} & \text{PO released} & \text{Invoice receipt} \\ \hline \end{array}$$

Here, the top row indicates $\sigma$ where the bottom row relates to the path of the model from initial to final marking. In case of
$\sigma = <$PO created, PO changed, Insert: PO created, PO released, Invoice receipt$>$ and the model in figure 2.5 there is no optimal alignment:

$$\gamma_2 = \begin{array}{|c|c|c|c|c|} \hline \text{PO created} & \text{PO changed} & \text{Insert:PO created} & \text{PO released} & \text{Invoice rec.} \\ \hline \text{PO created} & \gg & \text{Insert:PO created} & \text{PO released} & \text{Invoice rec.} \\ \hline \end{array}$$

In this case the symbol $\gg$ is shown, indicating a misalignment. These misalignments indicate that an activity is skipped or occurs when it is not supposed to. Looking at figure 2.5 it can be seen that 'PO changed' can not occur after 'PO created', indicating a misalignment which is recorded. Next, we can continue where left in the trace. Aalst (2016) defines the fitness function of a trace for alignments as follows: $fitness(\sigma, N) = 1 - \frac{\delta(\lambda^N_{opt}(\sigma))}{\delta(\lambda^N_{worst}(\sigma))}$, where $\delta(\lambda)$ is the cost function indicating the total number of $\gg$'s in the alignment.

A final conformance checking technique is that of comparing footprints(as the one shown in figure 2.1). These footprints characterize the corresponding event log. So when the footprint of an event log and a model are derived, these can be compared on dissimilarities (Aalst, 2016). For example, the footprint matrix of figure 2.1 can be compared with a footprint of a model for that event log. Imagine that the result of this would be that 12 out of the 36 cells of the footprint matrices differ, the conformance of the footprints is equal to $1 - \frac{12}{36} = 0.8$

### 2.1.3 Enhancement

Process enhancement aims to extend or improve the a-priori model using the observed events. This can for example be done by using timestamps in the event log to extend the model and show bottlenecks, service levels, throughput times and frequencies (Aalst, Adriansyah, and Dongen, 2012). Types of process enhancement are repair and extension. When repairing the model, it is being aligned with reality. That is, paths that are not taken can be removed from the model and further reparation can be done by using the $r$ and $m$ tags in the model resulting from token replay. In case of extension, a new perspective is added to the process model by cross-correlating with the log. This can for example be done with data mining

techniques like decision trees, after which the different perspectives are merged into a single integrated process model(Aalst, 2016).

## 2.2 Previous Research

As mentioned, there is not much knowledge within the Grant Thornton Assurance-IT Audit department when it comes to applying Process Mining techniques to practice. However, this does not mean that there is no knowledge at all. The theory is well known within the department, however the skills of being able to put it to practice is missing. Some very basic discovery techniques have been applied on a small basis which was mostly by manually assigning relations between activities on a small scale. The results of this were not very significant for worth mentioning. However, the possibilities of successfully applying Process Mining within the audit are acknowledged. This shows by the formation of a special interest group on Process Mining within the Assurance department of Grant Thornton.

Dommerholt (2015) focuses on the relevance of Process Mining in the annual audit and the key ingredients of successfully putting it to practice. Research question here was: "Which points of attention are significantly recognizable when it comes to putting conformance checking to practice as an audit technique for the auditor in an interim check on the purchasing process?". Conclusion of this thesis is that the 3 mayor points of attention are: Data quality of the event logs, consistency of the process model and event log and finally IT general controls. Furthermore he concluded that back then applying Process Mining was a bridge too far. This was because there was not much experience with applying data analysis as a tool of control in the audit. Here the main challenge for the future was to produce an event log dataset with a suitable structure for being used in a data analysis. At the time this thesis was written, support for taking this step was missing which was marked as a challenge for the upcoming years.

Other researches that have been done on applying process mining in the audit are Jans, Alles, and Vasarhelyi (2014), Accorsi and Stocker (2012), Hakvoort and Sluiter (2008), Heijden (2012) and Jans, Alles, and Vasarhelyi (2010). In Heijden (2012) the different phases and activities in business process mining projects are described in order to be applied in practice. Main findings in the other papers are that the difficulties in applying Process mining most commonly involves the construction of the event logs and a lack of efficient tools for implementation. Here Accorsi and Stocker (2012) and Jans, Alles, and Vasarhelyi (2010) both explicitly mention that the creation of the event logs is the most important and at the same time most difficult step to take. With specifically the identification of activities and the selection of a process instance as the most crucial steps to take(Jans, Alles, and Vasarhelyi, 2010). Both these papers name 3 fundamental process mining perspectives: process perspective, organizational perspective and the case perspective. The process perspective can be used to compare process against a norm, the organizational perspective uses the 'Originator' field to perform checks on for example segregation of duties and the case perspective focuses on a single case in the log such as the size of a single order.

Another interesting conclusion made by Hakvoort and Sluiter (2008) is that the way of creating 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. This conclusion made is not directly related to this thesis using only SAP data, even tough this is interesting for the business when it comes to implementing the results of this research.

Next, Accorsi and Stocker (2012) name several methods for applying process discovery. They conclude that heuristic based approaches favor frequent behavior but neglect infrequent behavior. Besides that, another finding in their research is that fuzzy models can drop certain transitions with an occurrence below a certain threshold, making them disappear in the model, this is confirmed in Jans, Alles, and Vasarhelyi (2014). When it comes to process conformance, Jans, Alles, and Vasarhelyi (2010) name descriptive and prescriptive models. Descriptive models are preferred models where prescriptive describe how a process should be executed. They name the conformance checker algorithm providing 3 analysis methods: state space, structural and log replay. Lack of this algorithm is that there is no option to make the model recognize whether a certain event is missing or not based on an attribute value, as for example an indicator that an order includes goods. Hakvoort and Sluiter (2008) focus on fitness and appropriateness of the conformance check model, that is because auditors are most interested in instances deviating from the desired process.

## 2.3 Purchasing Data Flow

All data considered in this research originates from SAP, the way this purchasing data relates to each other is generalized and visualized in figure 2.7. However this is a very basic representation of a purchasing data flow according to SAP, although such an overview can provide an informative view on the raw data. However, this is only the data flow in SAP and is not a model on the actual purchasing process.

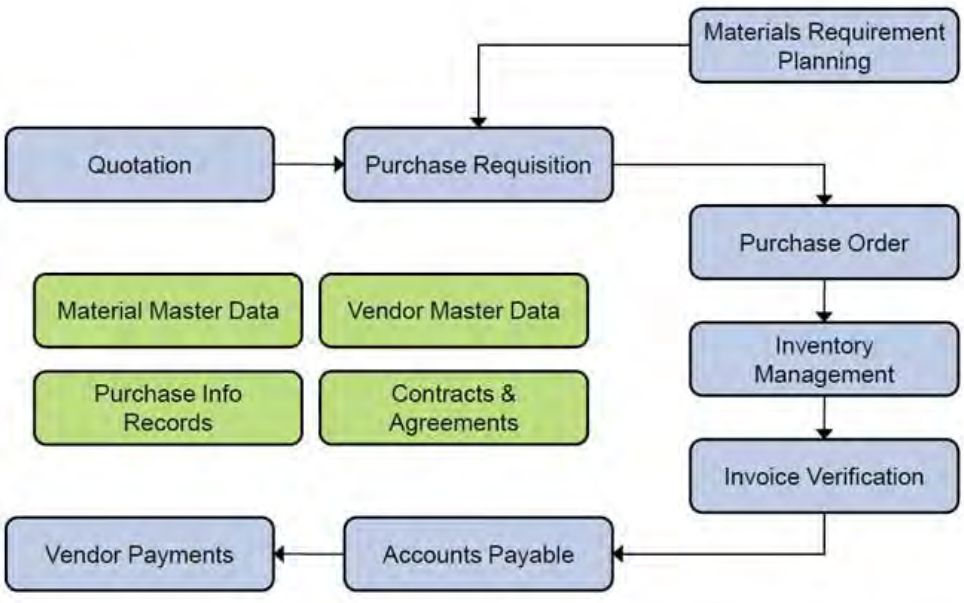



FIGURE 2.7: Purchasing Process Flow Diagram in SAP(*SAP Purchasing Data Flow*)

This diagram shows that given a quotation and the requirement of certain materials, a purchase requisition can be done. This can lead to a purchase order, which goes through inventory management who receive the materials before the invoice is verified to be correct. When this is all handled the vendor can be payed for the purchase order through accounts payable. All activities considered in this thesis should in some way fit into this process flow diagram. Although this diagram is generalized it gives a good indication of the process that is considered in this research. Meaning that it gives an indication of what is expected to be found in the data up to a certain point.

# Chapter 3

# Methods and Models

There are many different techniques for process discovery as well as process conformance. The goal is to discover a good process model reflecting the data well. However, all approaches have particular challenges to face with, such as loops and duplicate tasks. In addition, not every event log is similar besides that different quality measures are being used. This all illustrates that no standard process mining algorithm is available that works best for all problems(Rozinat et al., 2007). As for process conformance, one has to make a decision with respect to the 4 quality measures, discussed in 2.1.1, and how to deal with the tension between those(Rozinat and Aalst, 2008). How to handle this depends on the data considered and the aims for applying process mining. What, why and how to filter the event log so that a norm log remains forms the gap between process discovery and conformance in this case study. Filtering out bottlenecks resulting in a norm log causes the process model to go from an unstructered model containing an infinite number of relations to a more structured model containing clear sub components (Heijden, 2012).

So the workflow of this study after creating an event log is as follows. All 'correct' traces will be identified and all bottlenecks will be filtered out. These bottlenecks are identified based on interviews with experts in combination with the analysis done on the log. This norm log will be translated into a model using process discovery. Suitable algorithms to do so are discussed in section 3.1. After which these models will be used as input for conformance checking in order to find the best conformance model for identifying bottlenecks in the purchasing process.

## 3.1 Discovery techniques

The process discovery methods to be used on an event log depend on the goals of the project. As done in Accorsi and Stocker (2012), process discovery can be done with various types of discovery algorithms each having their own specifications. Depending on the interests and goals of the analysis as well as the data to be used, different discovery techniques can be applied. Each of them having their own functionalities and disadvantages which will be discussed next to end up with the discovery techniques which could be applicable in this research. Aspects of the data that can influence the appropriateness of a model are the number of duplicates, loops and noise. In addition, when the goal of a process discovery project is to model all frequent behavior then a model is needed that neglects all low-frequency behavior. In contrast, such a model is not appropriate for auditing purposes where low-frequency behavior is very interesting.

Abstraction based algorithms construct a model based on ordering relations amongst process activities. The $\alpha$-algorithm, as described in 2.1.1 is the classical example of

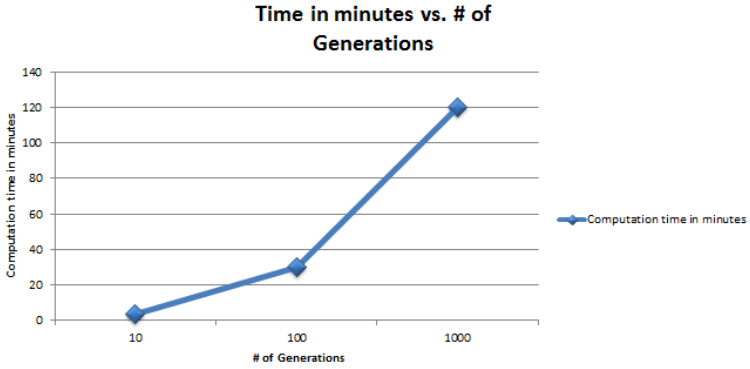



FIGURE 3.1: Computation time Genetic Algorithm on a dataset of 660 traces

an abstraction based algorithm where other algorithms are built on. However, abstraction based algorithms require the event log to be complete when it comes to ordering relations and absence of noise. Because this study uses real life data, these requirements cannot be met(Accorsi and Stocker, 2012). Meaning that abstraction based techniques are not appropriate for this study.

A common discovery model is the fuzzy model, as for example used in the process mining tool Disco (Jans, Alles, and Vasarhelyi, 2014). The fuzzy miner is an example of a heuristic based algorithm which considers the frequency of ordering relations(Accorsi and Stocker, 2012). The fuzzy miner algorithm filters for typical issues encountered with large real-life datasets which is then simplified an visualized(Jans, Alles, and Vasarhelyi, 2014). Again, these type of algorithms are not appropriate for use in this study. That is because low-frequency behavior is neglected by these algorithms, where these traces on the other hand are interesting for auditing purposes. In addition, fuzzy models do not have clear evaluation semantics to compare them with each other.

Search based algorithms can be compared with the process of evolution. These algorithms are able to handle duplicates and perform better than the heuristic based algorithms when initialized with knowledge (Accorsi and Stocker, 2012). Evolutionary algorithms have been discussed in section 2.1.1. These type of algorithms do not have the disadvantages of the heuristic and abstraction based algorithms, however drawback is that when the population size or number of generations grows, the computation time grows rapidly as shown in figure 3.1. In addition, this algorithm could get stuck in local optima when the diversity within a population is low and the variation operators cannot change this fact (Eiben and Smith, 2003). An overview including the (dis)advantages of the discovery techniques discussed is given in table 3.1. Here it shows that abstraction based algorithms are not applicable in this research because they cannot be used on real life event logs. Looking at other algorithms, heuristic based techniques do not have the option of providing evaluation metrics as output alongside the model. The Adapted Genetic Algorithm of Medeiros, Weijters, and Aalst (2007) does not have these issues, which is the advantage of this modeling technique above the others. For this reason this technique will be used for building the discovery models that will serve as input for the conformance checking models.

| Algorithm | Example | Evaluation metrics | + | - |
|---|---|---|---|---|
| Abstraction | $\alpha$-algorithm | Fitness, precision, generalization, simplicity | Easily implemented | Require complete noice free logs |
| Heuristic | Fuzzy model | - | Applicable on real life data | Neglects behavior and duplicates |
| Search | Evolutonary algorithm | Fitness, precision | Handle duplicates & real life data | Computation time, local optima |

TABLE 3.1: Summary of Discovery algorithms

## 3.2 Filtering

Through the discovery analysis in this study the filters for the norm log will be identified next to the fact that the actual processes of the clients will be examined. In Heijden (2012), 3 perspectives of process mining an event log are mentioned. That is, the process perspective, which can be split into the control flow perspective and the organizational perspective, and the case perspective. This study will use the first 2 perspectives in the process discovery phase to get a better understanding of the actual process and an indication of possible filters at the same time. The case perspective is the verification phase of this study where the filters will be determined.

First the control flow analysis where frequent and infrequent traces are identified. This gives a first insight in the actual process and could already give an indication of possible filters to apply. The control flow perspective focusses on the ordering of activities (Heijden, 2012). For example which tasks precede which other ones, which tasks are the starting tasks and which are the ending tasks in the event log. In case of this study, this can be used to filter on all traces starting with the creation of a purchase order, as will be discussed in chapter 4.

Second, the organizational perspective where through the performance analysis the length and throughput times of traces are looked into. Very long or short traces in terms of duration or length should be looked into, after which can be determined whether these cases should be filtered out as bottlenecks or not. This will be determined in section 5.2. The final analysis of the process perspective is the role analysis which is interesting for audit purposes. Example of which is checking that the same person does not have control over a certain combination critical tasks (Jans et al., 2011). In both Aalst et al. (2009) and (Jans et al., 2011) the '(Semantic) LTL Checker Plugin' in ProM is mentioned as a useful tool to perform these checks on an event log.

The case perspective concerns the verification of certain properties and actually includes filtering the event log. Jans et al. (2011) divides the checks into checks on role conflicts, case specific checks and other internal control checks. The first group of checks is known as 'segregation of duties', which is the case when a single person is involved with 2 or more critical tasks in a trace. In case of this study the tasks on which should be filtered are determined in section 4.3. Case specific checks focus on the absence or presence of certain events. For example cases needing a release event before anything can happen with a purchase order such as receiving goods or an invoice. Other internal control checks could be checks on the presence of a goods receipt event except for those cases which have no goods involved according to the goods receipt indicator. Other examples of such an internal control checks are checks on changes made after a purchase order was released as mentioned in Jans et al. (2011) or checking if the quantities on the invoice matches the quantities on

the purchase order made. Again, for this study the exact filters to be applied will be determined in chapter 5.

## 3.3 Process Conformance

When a norm log is created and modelled so that it reflects the norm log best, it can be used as input for process conformance. This again indicates the importance of a process discovery technique. A bad process model has a big influence on the process conformance model, as it will qualify traces incorrectly due to a bad discovery norm model. As well as for process discovery, process conformance can be split into different kind of perspectives. That is, a process conformance model can be of a descriptive nature or of a prescriptive nature as mentioned in section 2.2. In this study the assumption is made that all valid traces are captured in the datasets of these clients looking at the considerably large size. Meaning that the process conformance model to be built will be of prescriptive nature. So, the discovery model built from the norm log aims to represent the preferred model describing the way in which the process should be executed (Rozinat and Aalst, 2008).

First conformance checking technique considered is 'The Compliance Checker' discussed in Ramezani (2017). Here the input of the function is the event log to be checked and a Petri net describing the norm log. The alignment that is shown as output of this process conformance function is computed through the optimal alignment technique explained in section 2.1.2. However, drawback of this method which came to light during the modeling phase of this study, is that only complaince rules on the sequence relations can be formalized using this method. This means that rules on additional data next to the timestamp, activity and resource cannot be made. For this reason these compliance checking models have not been used in this research.

Next conformance checking method considered is 'The Conformance Checker' described in Aalst et al. (2007) and Aalst et al. (2009). Here an event log is replayed within a Petri net model making use of the token based fitness metric, as discussed in section 2.1.1. This results in a fitness value as well as the so called behavioral- and structural appropriateness(Rozinat and Aalst, 2008). The behavioral appropriateness expresses the precision of the model where the structural appropriateness expresses how compact and understandable the process model is making this metric a combination of generalization and simplicity discussed in section 2.1.1. Drawback of this method is that checks comparing values of events within the same trace cannot be implemented, this method only consideres the sequences of events.

More extended types of conformance checkers are the data-aware conformance checkers described in Mannhardt, Leoni, and Reijers (2015) and De Leoni, Aalst, and Dongen (2012) . These techniques take data and resources into account where most conformance checking techniques focus on the control flow, i.e. the order of activities(De Leoni, Aalst, and Dongen, 2012). In this study the person('resource') executing an event is important as well. Because this study focusses on more than only the basic control-flow and organizational perspective attributes, being case, activity, time and resource (Mannhardt, Leoni, and Reijers, 2015) these data-aware checkers are considered. Meaning that checks on order amounts can only be included in the conformance check model when not only the order of events but also resource and data are considered. This makes the data-aware conformance checkers interesting

for this research. Drawback of all of these techniques is that the only rules that can be built into the model are checks like: Order amount $\neq 100$, but not: Order amount $\neq$ Invoice amount (Mannhardt, Leoni, and Reijers, 2015). Table 3.2 captures the overall positive and negative characteristics of the conformance techniques described in this section.

| **Method** | **+** | **-** |
|---|---|---|
| Compliance checker | Checks on additional rules | Rules comparing 2 events impossible |
| Conformance checker | Covers all evaluation metrics | No data aware checks |
| Data-aware | Checks on 'extra' data | Cannot compare 'extra' data of 2 events |

TABLE 3.2: Summary of Conformance Methods

All methods are inable to compare additional data of events within a single trace, this makes the positives of the compliance checker and the data-aware checker irrelevant. Next to that, the Conformance checker takes all evaluation metrics, discussed in section 3.1, into account. For these reasons the Conformance checker will be used in the confromance phase of this research eventhough not all checks can be done using this method.

# Chapter 4

# Data

## 4.1 The Datasets

## 4.2 Analysis

## 4.3 Exploratory Interviews

# Chapter 5

# Results

## 5.1 Discovery Analysis

### 5.1.1 Client A

### 5.1.2 Client B

### 5.1.3 Client C

## 5.2 Explanatory Interviews

## 5.3 The Benchmark Model

## 5.4 Optimized Models

# Chapter 6

# Conclusion

At the start of this research, the following research question has been formulated:

*"Can we build a Process Mining model which supports the accountants of Grant Thornton, by correctly detecting bottlenecks through a conformance check on the purchasing process of a client?"*

Expected for this study was to create a conformance check model that could identify bottlenecks in the purchasing process of any client using SAP, based on several initial checks holding for all clients. Unfortunately, achieving all that was formulated in this research question could not be done. Differences between the clients such as all having their own small exceptions on certain rules makes it hard to define a solid standard rule model. A lack of conformance techniques all together makes this a bridge to far for now. However, this does not mean Process Mining cannot be of added value in the audit. The analytical application of process discovery can lead to a lot of interesting information for the audit as well as advisory projects. Furthermore, bottlenecks can also be identified using step by step filters per client like done in this study. Drawback of this method is that it is based on predefined rules instead of historical data.

When all of the clients and companies use the same ERP system, the fact that different clients use different functionallities of a single ERP system and have different regulations in their process cause differences in their event logs on this process. Next to which, this research resulted in more possibilities for further research. First, it is possible to program Process Mining algorithms on your own. At this moment there is a lack of process mining algorithms being able to apply conformance checking on both sequence and event data level, specifically on comparing multiple events within a trace. A research on this matter would be interesting to explore the possibilities of such an algorithm. If succesfull this would be of big added value for applying Process Mining in the audit, let alone Process Mining in field of application.

What has been considered in this study as well is using Data Mining and Machine learning to identify bottlenecks in the process. In that case the event log needed to be transformed into a dataset were data mining/machine learning would be applicable to, that is a dataset on trace level. Meaning that each line in the dataset represents a trace, where each trace in the train dataset is marked as bottleneck or not being the response variable. This data should be created manually througout the filtering done in this study and the interviews being held. However, the event logs contain traces of variable size in terms of events. In addition, these events all contain attributes as well, where each of these event attributes is (part of) a predictor

variable in the dataset to be made.

Looking at the event logs consisting of thousands of traces containing hundreds of different sequences and millions of events, this option has been put out of the scope of this research. Translating these event logs with variable traces of variable length into a consistent and usefull dataset would be very time consuming, and if not impossible a research itself could be done on this matter. However, it would be interesting for future research to scout the possibilities of combining Process Mining and Data Mining/Machine learning in such a way, especially because nothing can be found on matters like these. Studies on combining both process mining and machine learning/data mining, like Nikolov (2015), use data mining in process discovery. In contrast, applying these techniques in process conformance is still an undiscovered path.

# Chapter 7

# Bibliography

Aalst, Wil Van der (2014). "Process mining in the large: a tutorial". In: *Business Intelligence*. Springer, pp. 33–76.

Aalst, Wil Van der, Arya Adriansyah, and Boudewijn van Dongen (2012). "Replaying history on process models for conformance checking and performance analysis". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2.2, pp. 182–192.

Aalst, Wil MP van et al. (2010). "Auditing 2.0: Using process mining to support tomorrow's auditor". In: *Computer* 43.3.

Aalst, Wil MP van der et al. (2007). "ProM 4.0: comprehensive support for real process analysis". In: *International Conference on Application and Theory of Petri Nets*. Springer, pp. 484–494.

Aalst, Wil MP Van der et al. (2009). "ProM: The process mining toolkit." In: *BPM (Demos)* 489.31, p. 2.

Aalst, Wil van der (2016). *Process Mining: Data Science in Action*. Heidelberg: Springer.

Accorsi, Rafael and Thomas Stocker (2012). "On the exploitation of process mining for security audits: the conformance checking case". In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. ACM, pp. 1709–1716.

De Leoni, Massimiliano, Wil MP van der Aalst, and Boudewijn F van Dongen (2012). "Data-and resource-aware conformance checking of business processes". In: *International Conference on Business Information Systems*. Springer, pp. 48–59.

Dommerholt, Christiaan (2015). "Process Mining, Het gebruik van conformance checking bij de jaarrekeningcontrole". MA thesis. Amsterdam, The Netherlands: UvA.

Eiben, Agoston E, James E Smith, et al. (2003). *Introduction to evolutionary computing*. Vol. 53. Springer.

Hakvoort, Ron and Alexander Sluiter (2008). "Process Mining: Conformance analysis from a financial audit perspective". In: *Int. J. Business Process Integration and Management*.

Heijden, THC van der (2012). "Process mining project methodology: Developing a general approach to apply process mining in practice". In: *Master of Science in Operations Management and Logistics. Netherlands: TUE. School of Industrial Engineering*.

Jans, Mieke, Michael G Alles, and Miklos A Vasarhelyi (2014). "A field study on the use of process mining of event logs as an analytical procedure in auditing". In: *The Accounting Review* 89.5, pp. 1751–1773.

Jans, Mieke et al. (2011). "A business process mining application for internal transaction fraud mitigation". In: *Expert Systems with Applications* 38.10, pp. 13351–13359.

Jans, Mieke Julie, Michael Alles, and Miklos A Vasarhelyi (2010). "Process mining of event logs in auditing: Opportunities and challenges". In:

Mannhardt, Felix, Massimiliano de Leoni, and Hajo A Reijers (2015). "The Multi-perspective Process Explorer." In: *BPM (Demos)*, pp. 130–134.

Medeiros, A KA de, A JMM Weijters, and W MP van der Aalst (2007). "Genetic process mining: an experimental evaluation". In: *Data Mining and Knowledge Discovery* 14.2, pp. 245–304.

Nikolov, Boris Svetozarov (2015). "Combining Data mining and Process mining for analyzing food safety processes". In:

Ramezani, E. (2017). "Understanding non-compliance". PhD thesis. Eindhoven: Technische Universiteit Eindhoven.

Rozinat, Anne and Wil MP van der Aalst (2008). "Conformance checking of processes based on monitoring real behavior". In: *Information Systems* 33.1, pp. 64–95.

Rozinat, Anne et al. (2007). "The need for a process mining evaluation framework in research and practice". In: *International Conference on Business Process Management*. Springer, pp. 84–89.

*SAP Purchasing Data Flow*. https://www.sap.com. Accessed: 2017-04-03.

# Appendix A