Delft University of Technology

MSc Thesis

Workflow Mining: A stepwise Approach for extracting Event logs from Corporate Network Data

January 9, 2011

Author:
Guido Demmenie

Chairman:
Dr.ir. J. van den Berg

1^{st} *Supervisor:*
Dr. M.V. Dignum

2^{st} *Supervisor:*
Dr. J.L.M.Vrancken

External Supervisor:
Menno Israël

Preface

This research has been performed as the final stage of the masters program System Engineering, Policy Analysis and Management (SEPAM) with the Information Architecture track. The research has been performed in the period between March 2010 and January 2011 at the Netherlands Forensic Institute (NFI) with the department 'Knowledge Center of Intelligent Data Analysis' (Kecida).

During the research support has been provided by the ICT and the System Engineering section of the faculty Technology, Policy and Management of the TU Delft and from department Kecida from the Netherlands Forensic Institute.

Therefor I would like to thank Menno Israël (Kecida) for providing me with a problem to solve. Virginia Dignum, Jan van den Berg and Jos Vrancken (TU Delft) for their effort, insight, enthusiasm and positive criticism. I also want to thank André Hoogstrate from Kecida for all the feedback and sparring that shaped many ideas to solve many encountered problems. I want to thank to the whole Kecida team for the fun times I experienced during my stay. And I want to thank the Xiraf team (NFI), all the effort they put into all the questions and problems I posed. I want to thank everyone who has been interested in (the progress of) the research I have done. And my parents for making it possible to have been following this study.

And finally I want to express special thanks to my Nicolle for being as patient as she has been.

SEPAM

The Master's Program Systems Engineering, Policy Analysis and Management (SEPAM) is a program that learns students how to integrate technology, engineering and management knowledge. The curriculum is built on the three pillars of systems engineering, multi-actor network stakeholder theory and technology specialization.

It is imperative for enterprises to achieve and maintain an optimal balance with ICT developments. An optimal balance means that an enterprise should always exploit the possibilities of ICT optimally, such that its organization operates efficiently while achieving constant innovation in its business (the enabling role of ICT). To find a good balance an integral architecture is needed to guide the (re)design and the (re-)engineering of both the organisation and the business. The Information Architecture (IA) specialization addresses both topics.

This research needs knowledge of at least two of the SEPAM pillars, the systems engineering for developing a systematic approach and the technology

specialization, in this case ICT, to be able to extract information from computer systems. The knowledge gained from the Information Architecture track helps with re(verse) engineering organisations and thus being able to extract the correct information from all the data available.

Abstract

For (forensic) auditing purposes it is useful to have a wise view of how processes are performed in an organisation. Workflow mining can help in creating such a view by mining workflow models that give a visual representation of the processes. But not all organisations use workflow management systems that produce the needed event logs that are used as input for the workflow mining techniques. What this research is set to accomplish is to design a stepwise approach that extracts these event logs from corporate network data that is available in any organisation To accomplish this the stepwise approach must extract caseIDs, activities and the order in which the activities occur. The focus of the approach is on the process around invoices. The approach that have been designed has four steps.

1. Indexing of the Corporate Network Data
2. Mining the invoice numbers to be used as caseIDs
3. Document discovery of all the documents that are related to the invoice process.
4. Extracting the activities those documents support.

The case study shows that the event logs can indeed be created and a workflow model is mined from those event logs. Thought the validation of the quality of the mined workflow model is very difficult. The approach does look promising for extracting event logs.

Contents

Chapter 1

Introduction

Gigabytes of digital documents reside on servers and computers within organisations. Digital documents usually support a task, making it easier to perform, or store data needed to perform the task, quickly calculate values, etcetera. The complete set of all files found on the storage devices from an organisation we call *Corporate Network Data*. This can be harddrives of computers or servers, but also USB-drives and other removable storage devices.

With the increasing amount of data and information to search through, it becomes more and more difficult to audit larger organisations with human labour. Automation of the auditing process could help creating less time consuming process to confirm or deny whether the organisation complies with all legislation or its own rules and guidelines. For internal auditing, but also for forensic auditing, there is a gap between having the Corporate Network Data available and performing workflow modeling to extract the workflows for use in the auditing process.

To bridge the gap between having the Corporate Network Data of an organisation available and being able to do an automated audit workflow using modelling can be used. The workflow is a formal description of routine aspects of work activities. Resulting in a description of well-defined tasks, roles, rules and procedures prescribing the work in manufacturing or in the office [12]. Workflow modelling helps visualizing those workflows. A workflow model shows all possible paths to achieve a certain goal. Whether that is producing a football or buying a football only differs in the tasks to be performed and the possible exceptions to be included in the model. The first task of an auditor usually is to check whether the prescribed steps are actually taken, which is exactly what workflow models can help with.

There are two kinds of workflow models, *designed* model and the *emerged* model. The designed models are the workflow models as they were designed before they were actually executed. The emerged workflow models are models of the actual performed workflows. The emerged model will very likely differ from the designed workflows because people tend to take shortcuts, make mistakes et cetera.

As described by Cook and Wolf in 1999 the most important function of workflow models is to check the differences between the two models [6]. This is exactly what is done during an audit, the designed workflow models are known and the auditor wants to know whether the emerged workflow models comply

with the designed models. To find the emerged workflow models information about the performed workflows is needed. Using that information, workflow mining techniques can discover the emerged workflow model. The techniques make it possible to discover workflow models from large amount of data. The data does have to be in a specific format though. Most techniques for workflow mining such as the α-algorithm [23] and its variations [8, 28, 29, 18] and Heuristics Miner [27] use so called event logs as their input.

These event logs are not always available. Though digital documents found in Corporate Network Data harbour a large amount of information resulting from working processes. The challenge is to convert the Corporate Network Data into event logs. Corporate Network Data is very unstructured data, it includes Operating System files, configuration files, chat logs, emails or any other file stored on a computer.

In order to search for the right information it is needed to have a specific workflow in mind to extract. Because finding new ways to do forensic auditing, the search for fraudulent activities, is the motivation of this research the focus is on the workflows around transactions and thus invoices.

1.1 Goal

The Netherlands Forensic Institute (NFI) initiated this research in order to find a way to extract fraudulent transaction documents from Corporate Network Data. The current process to find such documents is to index the Corporate Network Data and then manually search for already known keywords and names and discover new ones and use those again. This means that only a limited view of the whole transaction landscape is available. The goal for the NFI is to have a method that shows the whole transaction landscape that is available in the Corporate Network Data and then search for anomalies.

There is a large gap between having the Corporate Network Data and the possibility to perform Workflow mining. There is no literature that describes how to process the wildly unstructured data like Corporate Network Data such that it can be used in the available workflow mining techniques. Because of this the focus for this research has been narrowed to focus on extracting the information from the Corporate Network Data that is needed to perform workflow mining.

In this research we want to create the link between the unstructured Corporate Network Data and the already existing workflow mining techniques, which use a special formatted list of events called Event Logs to perform workflow mining.

> *The goal of this research is to design a stepwise approach to extract Event logs from Corporate Network Data*

When this goal is reached it will open up possibilities to use workflow mining on a broader field, since also unstructured data like Corporate Network Data can then be used as input for the existing workflow mining techniques and tools. This research does not provide new workflow mining techniques, or finding new ways to perform workflow mining.

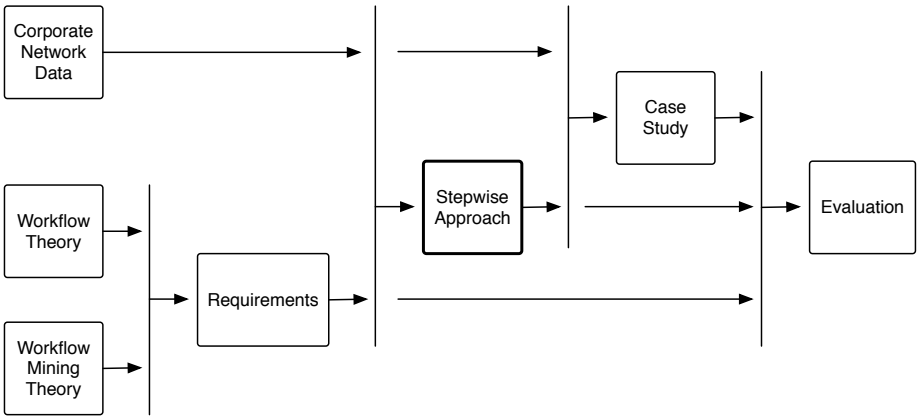

Figure 1.1: Visual representation of the research plan.

1.2 Research

In this chapter the relevance and the goal of this research have been described. In Figure 1.1 the research plan is visually represented. A *literature study* on the topics of Workflows and Workflow Mining outlined in Chapter 2 forms the input for devising the requirements to which the designed approach has to comply. These requirements are described in Chapter 3 and form together with the Corporate Network Data the input for *designing* the Stepwise Approach which is detailed in Chapter 4. Using the newly designed Stepwise Approach together with the Corporate Network Data a *case study* is performed as described in Chapter 5. The results of the case study are then combined with the requirements and the Stepwise Approach to *evaluate* the designed Stepwise Approach in Chapter 6. Then a discussion will follow, Chapter 7 after which the final conclusions are drawn in Chapter 8.

Chapter 2

Workflow Theory

Workflows can help to check compliance of organisations. To extract workflows from information sources workflow mining techniques are applied. Those techniques need specific information as input which has to be in a specific format. In this chapter we explain the concept of workflows and the input, methods and output of workflow mining techniques as being described in literature.

2.1 Workflows

When someone needs to execute a specific task he usually has to work through different steps to come to that what he wants to accomplish, whether that is something tangible or not. These different steps that need to be performed together we call a workflow. The notion of workflows comes from the idea of processes in manufacturing and the office. Workflows are focussed on the often repeated chained activities of work being done and typically separate those chained activities into well-defined tasks, roles, rules and procedures which regulate those chained activities [12].

As an example to demonstrate the idea of workflows we show the workflow of making coffee. There are eight activities to be executed in order to end up with a cup of coffee.

A Use coffee can to pour water in coffee machine

B Put filter in coffee machine

C Put ground coffee in coffee machine

D Put can under the coffee machine

E Switch coffee machine on

F Wait for the machine to finish

G Switch coffee machine off

H Take coffee can from coffee machine

I Throw away used filter.

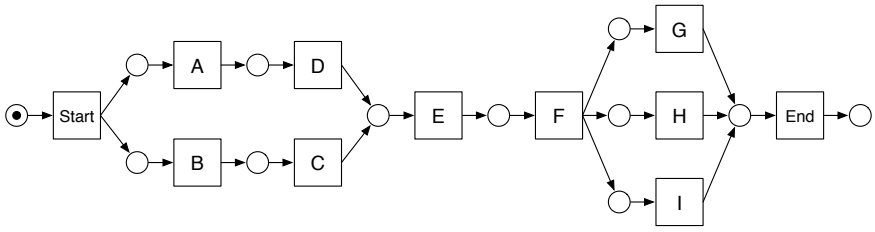

Figure 2.1: A Workflow Net representation of the workflow model of making coffee.

One possible chain of activities would be $\{ADBCEFHIG\}$, but another possible chain $\{BCADEFGIH\}$ would produce the same result. This is possible because not all tasks are depended on all other tasks. Some tasks have a causal relation, for instance performing C before B would end up in a mess. A visual representation of the workflow is modelled in Figure 2.1.

The model implies that before switching on the coffee machine (E) both ground coffee has to have been put in the machine (C) and water has to have been put in the machine (D). And after switching the machine on (E) the task of waiting for the machine to finish (F) always follows.

This is a very simple example of a workflow, but about this workflow there will be no recording of digital information. Though with the workflow about invoices there is a lot of information recorded. Some of the information is recorded in information systems like financial systems, but a lot of information is also just captured in digital documents. For invoices information about transactions, the amount of money to be transferred, the data it is transferred, by linking invoice numbers to payment requests, dates and client names. The invoice numbers are also important to be able to trace the transaction steps. Late payed orders are usually collected in overviews so that the clients that ordered them can be send a reminder. This all belongs to the process of selling products the company produces and thus shows that at least part of the process steps are re-discoverable from the corporate network data.

2.2 Workflow Mining

Since the second half of the nineties different groups have been working on techniques for workflow mining. Workflow mining is meant to collect data during runtime of a process and aid in the workflow design and analysis [21]. From the data recorded while the workflow is being executed during normal operations a workflow model is mined. Then the model can be compared to prior designs of the workflow and analyzed. This aids in making the workflows more efficient and help steering of an organisation.

Different workflow mining techniques differentiate on three basic notions. First, there is the input, or data, used to discover the workflows from. Second, the method to actually discover the workflows. Third, the outcome, or more specifically the representation of the outcome.

The input data can be split up in two different groups of data, structured data and unstructured data. Structured data, with as example so-called event

logs, can be provided by all sorts of systems, in the first place WorkFlow Management systems, but other, transactional, systems also provide for these sort of logs. Sometimes including even more information than initially needed. The other sort is unstructured data, like emails where the activities and tasks are not explicitly defined yet. In order to be able to still apply the same sort of methods to discover workflows, they are transformed so that they are structured more in the likeliness of event logs.

The methods for workflow mining can be differentiated into, pure algorithmic, pure statistic and a hybrid between the two. Cook and Wolf conclude on early experience that the algorithmic and hybrid form show most promise and that the purely statistical method used was not sufficiently mature [5].

The output of the various techniques are most often represented in Petri nets, Heuristic nets, or a simple graph.

Here will will go into more detail of these three basic notions.

2.2.1 Input data

Due to the availability of workflow management systems like Staffware, IBM MQSeries, COSA and others [22] many research groups started of using the resulting event logs that those systems generated.

Many research is uses these event logs to extract workflow models from [5, 6, 7, 9, 10, 13, 15, 16, 21, 22, 23]. Others like Ang et al. offer a method to mine the organisational structure from those event logs [3]. Berlingerio et al. use the event logs and enhance the workflow model derived by using temporal mining [4]. Greco et al. goes beyond the model itself and describes a framework to mine the model taxonomies [14].

Only a few also try to create these event logs from unstructured data like emails so that they can use other input than just the event logs [11, 17].

Event logs

When a system logs all tasks that have been executed at the time the task has finished one could speak of a log. Event logs though have a few characteristics that make them useful for workflow mining. The building blocks of an event log are the events and the notion of cases.

Definition Let an *Event* be an executed task.

Definition Let a *Case* be a list of events with causal relations, hence the order in which they are recorded in the case is important.

Now that the building blocks for an event log are defined the actual event log can be defined.

Definition Let an *Event Log* be a set of cases where the cases can be intertwined, but the order of events within a case needs to be preserved.

So added to the simple log is the notion of cases, which represents a onetime execution of a workflow, hence the executions of the various tasks that should be performed for that workflow. A workflow also prescribes the order in which tasks should be executed, therefore within a case the order of execution of the

CaseID	Task
1	A
2	A
3	A
3	B
1	B
1	C
2	C
4	A
2	B
2	D
5	A
4	C
1	D
3	C
3	D
4	B
5	E
5	D
4	D

Table 2.1: An event log where case 2 is highlighted and the order within the case is important

tasks should also be recorded. This does not have to be with timestamps, it is enough to be able to assume that a task recorded earlier in the log also was preceded tasks recorded later in the log. When several cases are recorded in one log then that log is called an event log. It is not necessary that the cases are strictly separated in the log, they may be intertwined as long as the order within the cases are preserved. This means that if two cases are executed at the same time that the execution of the events of the two cases can be recorded in serial and thus get mixed the order still stays intact.

In Table 2.1 an example of an event log is shown where case 2 is being highlighted. It is clearly visible that various cases have mingled, but the assumption is made that the order within the cases are preserved. The example of case two can be described as $Case2 = \{ABCD\}$, the tasks performed where $CaseID = 2$, while the shortest form to represent the complete event log, or all its contained cases, would be $\{ABCD,\ ACBD,\ AED\}$. Which are all list of tasks performed within the five cases while maintaining the order in which they appear in the log.

Many other attributes can be added. For instance actors performing the task, this makes it possible to mine organisational structures [3, 25]. Timestamps make it possible to omit the explicit order in which the execution of the tasks is performed as it is then possible to reconstruct the order using the timestamps. Timestamps also add information in the form of throughput times of a workflow and make it possible to use different mining algorithms [4]. If there is a start and end timestamp even the throughput of the task itself can be traced.

2.2.2 Methods

Most recent research has been done in the class of purely algorithmic approach. The development of pure algorithmic approaches includes the α-algorithm [23] and the heuristic methods LittleThumb [26] and HeuristicsMiner [27]. The α-algorithm by van der Aalst et al. has been further developed since its inception to various forms that all tackle one or more difficulties that the α-algorithm could not cope with. HeuristicsMiner first described in 2006 by Weiters at al. [27] can be used to express the main behaviour found in event logs.

Challenges

In previous research many challenges to rediscover a process model have been identified. Rediscovering several constructions in the model are difficult to do. Parallelism for instance can also be modelled as sequence of varying tasks. Some of those challenges have been addressed by extending existing algorithms or applying different methods.

There is the problem whether there is a need for *prior knowledge*. Some claim that this is needed to evaluate the resulting model [13, 20]. Mostly in the form of Domain Knowledge about mutual exclusive tasks and parallelism. Others show that using different algorithms and clustering there is no need for a base model of the process [7].

Another challenge is finding *Hidden tasks*. During the mining process it is often assumed that all tasks are present as an event in the event log files. This is not always the case, for instance when there has to be made a choice, for instance that in some cases task A should be followed by B and other times by C that choice is not present in the model, while the choosing is a task to be performed. And while the α-algorithm can discover AND/OR split and joins, prime invisible tasks are not properly discovered. Prime invisible tasks are defined in [29] where also the $\alpha^{\#}$-algorithm is proposed that can mine these prime invisible tasks.

The next defined one is *duplicate tasks*, tasks labelled with the same name used in different contexts within one process. This leads to problems as to which tasks are preceding each other. Focused on this is [18]

Then some processes have *loops* of tasks within their process. Either one task that is repeatedly executed, basic loops, or possibly one task that puts the whole process several steps back, arbitrary loops. The former manifests itself in equally named tasks preceding each other, the latter are more difficult to find if they accommodate a jump further back into the process. Basic loops are made discoverable by the α^{+}-algorithm [8], while the arbitrary loops still are a difficulty.

Constructs

Some of the difficulties described above have been converted in constructs that can show up in workflow models. Some of the difficulties in rediscovering workflows can be described in constructs. These constructs are specific patterns in the workflow model that are difficult to rediscover. Table 2.2 names several of these constructs and the method from the various α-algorithm variations and HeuristicsMiner and the constructs those methods can cope with.

		Constructs				
Algorithm		parallelism	loops	non-free choice	hidden tasks	duplicate tasks
α	[23]	+	±	-	-	-
α^+	[8]	+	+	-	-	-
α^{++}	[28]	+	+	+	-	-
$\alpha^\#$	[29]	+	+	-	+	-
α^*	[18]	+	±	-	-	±
HeuristicsMiner	[27]	+	+	±	+	-

Table 2.2: Overview of described methods and their ability to cope with constructs adapted from [24]

Noise

Mining algorithms like the α-algorithm assume the event log data to be correct. But reality does learn us that this is often not the case and data can be logged with errors. Errors often cause low frequency behaviour in the event logs. It is unwanted to mine the low frequency behaviour resulting from the noisy data. In other words, mining algorithms must not fit this noise. Research on the impact of noise on the resulting models has been done [19] and various steps to reduce the effect of noise have been proposed using an adapted version of the van der Aalsts α-algorithm [23]. Heuristic mining techniques can much better cope with noise, these techniques look not only at whether events are happening after each other, but also record the frequency, and use this information to estimate whether it is an actual causal relation or if it is a possible noisy record.

2.2.3 Output

Petri nets provide all routing needs that is supported by various information systems currently in use. (CRM, WFM, BPM, ERP etc)[29] Workflow nets (WF-nets), often used in the area of workflow mining, are a sub-class of Petri nets with a single source place, that is the start of a process, and a single sink place, that corresponds to the end of the process [23]. Such a model describes the possible cases of the workflow.

Another possible output is an Heuristics Net. The heuristics net does not only show the model it also shows information about the event log it is mined from. It shows the number of executions of the activities, how dependent linked activities are and how many links they found during the mining process.

Chapter 3

Workflow Mining in Corporate Network Data

In this chapter we describe why we need workflow mining. We describe also the required output of the to-be designed Stepwise Approach. These requirements should make sure that the output of the stepwise approach can be used to apply existing workflow mining techniques.

3.1 Workflow mining

Workflow mining is the means to be able to say something about the actual processes being executed within an organisation. To perform workflow mining an event log is needed. We have Corporate Network Data, a set of data that is captured from the various storage devices available in an organisation. An exact copy of the storage devices is often called an image. So Corporate Network Data is a collection of all the files that can be found on a computer.

We will mostly focus on finding information about transactions and convert that information to event logs. Workflow mining on corporate network data means that we first have to perform several steps of preprocessing and ways to extract information about the process we want to mine from the data. The workflows we will be after in this research are the transactions. These are identifiable by the invoice numbers used during the transactions and showing up in invoices and other documents regarding the transactions. These appearances of invoice numbers should help to find all the various steps made in the process of a transaction.

We have no named identifiers for the various tasks, the most we have is filenames, which vary largely due to for instance including dates. Due to the fact that we only see the last time a file is saved it is also impossible to gather historic data from the file itself. Even when it is a common used step in a process to add a transaction, or its assigned invoice number, to for instance a spreadsheet, we will only be able to identify the last time this has happened, even if there are many transactions listed in that document.

The files we encounter are for instance trail balances, lists of outstanding orders and invoices. An example of an invoice is given in Figure 3.1 where it is

shown what we are searching for within the file and what we also can find but that we classify as noise.

We first need to discover these case identifiers from the dataset itself and then use them to gather all the information about these transactions.

We find that literature currently lacks the information and methodology to find the case identifiers for transactions as well as good methods to extract the tasks from unstructured data. Hence our in-dept research on how to find these identifiers and tasks and designing a methodology that can be used for future research or analysis on this kind of data.

3.2 Requirements

In order to be able to provide workflow mining on corporate network data we first need to address what requirements we have so that we can actually apply any existing workflow mining tools to our data.

The unstructured data found on harddisks of various computer systems is not fit to directly do any workflow mining on. What in the end we want to find is several executions of a workflow. So our task is to search for a way to find events that can be linked to the workflow, but we still need to be able to link them to one execution of the workflow. These links should be in the form of a case identifier, and activity name and the possibility to order them in time [21].

3.2.1 Case Identifier

In order to be able to find all the events belonging to one execution of the whole workflow we need to know which events belong to which case. A case here is one full execution of the workflow. When we have a case identifier we can link all the various events to one workflow execution.

3.2.2 Activity

We need to know what activity is performed at each event, for otherwise it would not be possible to differentiate between different steps in the workflow. Without named activities it would also be impossible to map different cases onto each other.

3.2.3 Order

A workflow is a sequence of steps and the order of them is very important. If we do not know the order it will be impossible to make a model that can represent the actual workflow. The mining algorithm needs the order so that it can see what tasks depend on each other and generate the model of the workflows.

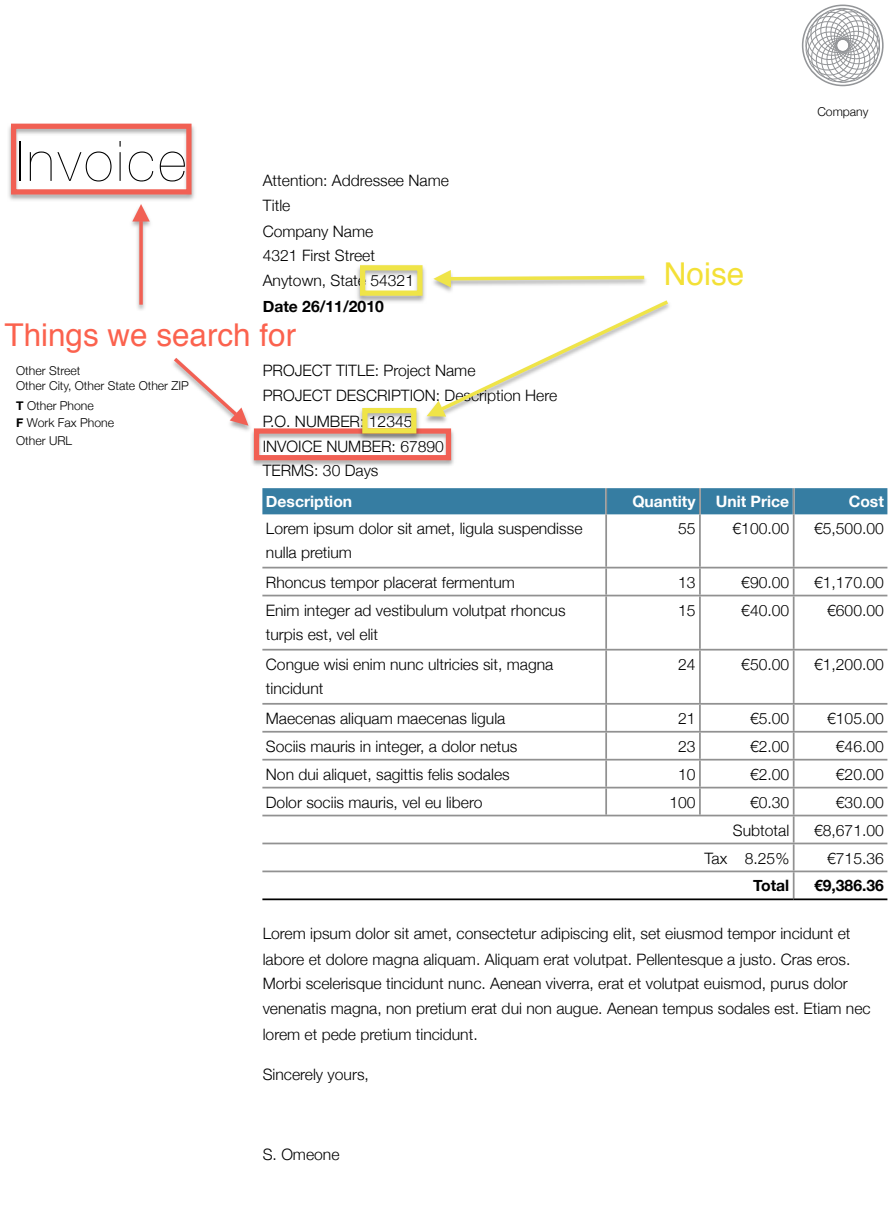

Figure 3.1: Method Model

Chapter 4

The Stepwise Approach

In this chapter we will describe the designed Stepwise Approach as depicted in Figure 4.1. The firgure shows the four distinct steps needed to extract the information from the Corporate Network Data and to put it in the format of an event log. In between the steps the various intermediate deliverables are shown in the form of sets that need to be created. The approach assumes the availability of the Corporate Networkk Data and ens where all information for the event logs is available.

4.1 Corporate Network Data

The starting point of this research is that Corporate Network Data is available. This encompasses, depending on how the organisation works, images of all the data stored on workstations, servers and possibly removable storage devices. How this is acquired is outside the scope of this research.

4.2 STEP 1: Indexing

Indexing does not directly contribute to comply with the requirements set for this approach, but it reduces the search times needed to search for various files in preceding steps immensely. The indexing information needed per file is defined below.

Definition Let the index of a file be $i_f = (N_f, W_f, id_f, fn_f, t_f)$

Where,

- i_f is the index of a file f.
- N_f is the set of numbers available in f,
- W_f is the set words available in f,
- id_f is an identifier to uniquely identify f,
- fn_f is the filename of f,
- t_f is the timestamp associated with f,

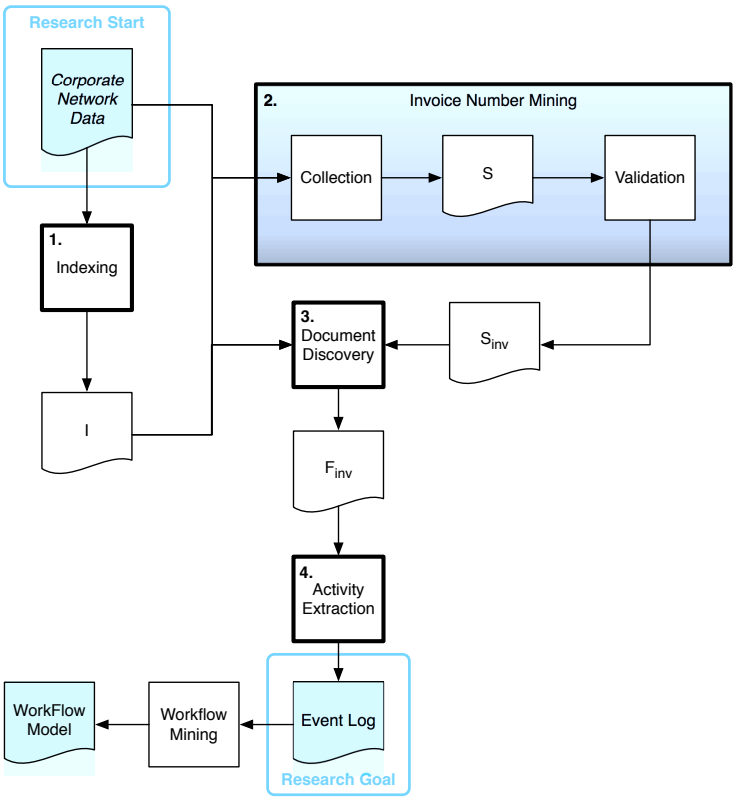

Figure 4.1: Visual representation of the stepwise approach

For each file an index is created and all the indices combined is defined below.

Definition Let I be the set of indices such that $\forall f : i_f \in I$

4.3 STEP 2: Invoice Number Mining

Finding the invoice numbers is needed to comply to the first requirement, finding a case identifier. The invoice numbers is the identifier for the transactions we are after, hence we need to find the invoice numbers.

Trying to find invoice numbers is a two step approach. First, it is needed to collect numbers that might possibly be invoice numbers. Second, the numbers found need to be verified.

4.3.1 Invoice number collection

Collecting the possible invoice numbers means that a search has to be done in the locations where invoice numbers are expected to be found. Limiting the search to the files containing the word 'invoice', or a synonym in the language the company works in, is already effectively limiting the set of files to search through. Also only including document type files that are used to support the expected tasks helps limiting to converge the search space toward a more specific environment where numbers are more find to be invoice numbers. Usually the invoice numbers do have a specific pattern, it thus pays to examine a few files thoroughly and search for hints to be able to limit the scope of possible invoice numbers again. When this all has been done the limited set needs to be searched for the possible invoice numbers. The set of numbers that results from this step contains only natural numbers and is called the set S.

4.3.2 Invoice number validation

Verification of the invoice numbers follows a two-step approach. First we assume that invoice numbers always are a range. Secondly we assume that the first occurrence of a lower number should be before higher numbers. Besides the two previous assumptions a sample of the found invoice numbers should be sought for in the files they originate from and be checked to be in the context of being an invoice number. The set of numbers for which this is true is the set of invoice which we call S_{inv}.

The assumption that invoice numbers should be a range of numbers follows from Dutch regulation that demands that invoice numbers are a range without missing numbers [1]. This can also be represented as follows:

Definition $\forall x \in \mathbb{N} : min(S_{inv}) \leq x \leq max(S_{inv}) => x \in S_{inv}$

Due to the above assumption we can also argue that numbers are not only in a range, but also increase over time. This leads to the concept that lower numbers generally should appear earlier in time than higher numbers. So we assume the following:

Definition Let I_x be the set of indices such that $\forall x \in S_{inv} : I_x = \{i_f | x \in N_f\}$

Resulting from that we can say that if we have an invoice number x, we collect all indices i_f for which $x \in N_f$ this results in a set of indices we call I_x. To find out when x was recorded for the first time we look at all timestamps t_f that are in I_x and we select the earliest one. If we would do the same for y and $x < y$ then the earliest timestamp for x should be, in time, before that one of y.

When it is found that both assumptions have been met for a set of numbers found earlier and at least a sample of the numbers are found in the right context, it is safe to say that it is highly probable that the numbers represent invoice numbers.

4.4 STEP 3: Document Discovery

Document discovery is finding all the documents that contain information about invoices. To be able to find those documents, a search for the invoice numbers (S_{inv}) in all the files of the Corporate Network Data needs to be performed. If a search is done through the complete set of files, using the index, a lot of noise is generated. So limiting the dataset to only the subset where invoice numbers are expected to be found helps decreasing the amount of noise. Using the limitation that the documents need to contain the word 'invoice' or a synonymous word in the language of the documents reduces searching in documents that are unrelated to the transaction processes.

To be able to have the appropriate data available in a later stage we use the following definition for a document. The mentioned attributes need to be captured for later use.

Definition Let a *document* be $d = \{s_{inv}, fn_d, l_d, t_d, u_d\}$

Where

- $s_{inv} \in S_{inv}$ and is the invoice number found in d
- fn_d is the filename of d
- l_d is the stored location of d
- t_d is timestamp associated with d
- u_d is the username associated with d

The information extracted from the found documents makes it possible to comply partly to the second and fully to the third requirement. The filename recorded as fn_d provides the base for finding the activity, this will be refined in the last step. The timestamp t_d provides the possibility to create the order of the activities within the case that we need as the third requirement.

4.5 STEP 4: Activity Extraction

We assume that the extracted documents support a specific task in the workflow. Extracting the activity is the last part that we need to comply with the three requirements as defined in Chapter 3.

The most rudimentary way to extract the activities is to look at the naming of the files. Most organisations have a naming scheme for the products or supporting documents of their processes. But this might not always be enough to determine a sensible name for the activity it supports. In that case in-depth inspection of the files is needed to extract the supported task.

4.6 Event Logs

To align this work with workflow mining research and to be able to use the tools resulting from that research there is a need to create so-called event logs.

We have fulfilled the the requirements from Chapter 3 that read as follows.

1. The case identifier, an identifier that is unique for each time the workflow is executed.
2. The activity, the task being performed in this event.
3. The order of the events can be deduced.

The identifier is found in the second step where the invoice numbers are extracted and verified. The activity can partly be deduced from the filename as described in step four, but in some cases it means that the content of the files needs to be examined thoroughly to find the task being supported by the file. The order of the activities can be deduced from the time associated with the file extracted in step three.

Because all the information to build an event log is available, it is now trivial to create that event log and produce the input for the workflow mining techniques available. At this point we have reached the point where the existing literature can be used again.

4.7 Workflow mining

Using the event logs it is possible to start mining workflows. Either by using PRoM framework by vd Aalst et al., or possibly other tools. The resulting workflow models can then be examined in detail to see what are workflows are the norm and which deviate from them. They can also be compared with the original design of the workflow to see whether it conforms the actual workflow or not.

Chapter 5

Case Study

For this case study the in Chapter 4 designed Stepwise Approach as depicted in Figure 5.1 is used to extract the information from Corporate Network Data to create event logs. The input for the approach, the Corporate Network Data, is the captured data from an international company.

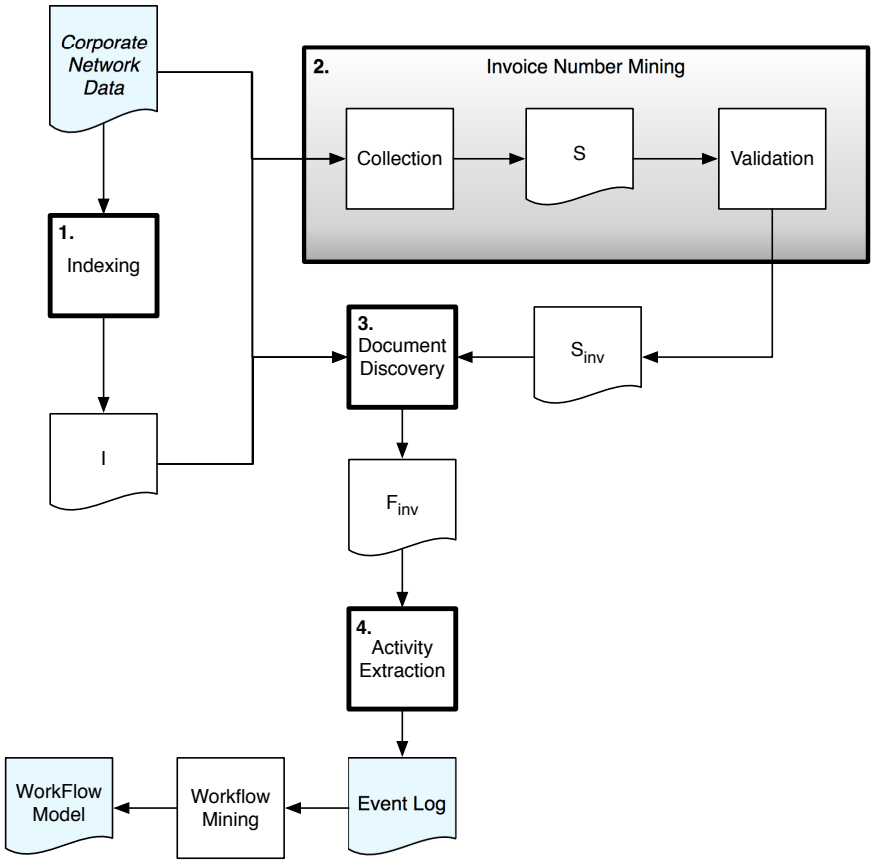

Figure 5.1: The stepwise approach

5.1 Corporate Network Data

The Corporate Network Data consists of a exact copy of various harddisks that came from the computer systems of a international company. This encompasses 300 Gigabyte of data in total. The systems from which the harddisks were copied include servers, workstations and laptops. The total number of systems included in the set are twelve physical separated systems.

The corporate network data is taken from those twelve systems. The systems captured are three fileservers and the rest are workstations. The complete set of data consists of a exact copy of various harddisks that came from the computer systems of a international company. Also a lot of paper documents are scanned and using OCR[1] these documents are digitised and made searchable using computerised techniques. The total encompasses 300 Gigabyte of data with about 3.6 million files. This whole set of files is called the F_{cnd}

We do not always need the complete F_{cnd} hence in the following steps we will create three subsets that are useful for that step as depicted in Figure 5.2. In step two we create a set of files to extract the invoice numbers from F_n such that $F_n \subset F_{cnd}$. In step three we create two sets, on is used to search for documents (F_e) such that $F_e \subset F_{cnd}$ and the other set is the result of that step three and contains all the found documents that have something to do with invoices (F_{inv}) such that $F_{inv} \subset F_e$.

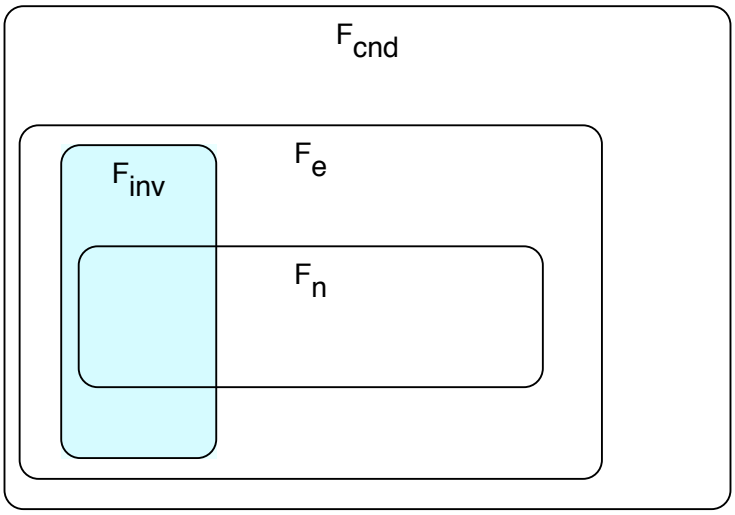

Figure 5.2: File Datasets and their relations

5.2 STEP 1: Indexing

This step was done for us by the Xiraf team that operates within the NFI. Xiraf is a forensic processing and analysis system that makes it possible to search digital evidence [2]. Xiraf extracts information like file metadata, document

[1] Optical Character Recognition

properties, email, chat log records, browser history records, etc. We only used the file metadata and document properties options of this. The indexing being performed by Xiraf makes it possible for us to quickly find documents with specific keywords and extract the metadata of those files, or search in the content of the files for other clues. The total number of objects to search through now encompasses 9.6 million objects that are quickly searchable for content or metadata.

5.3 STEP 2: Invoice Number Mining

As the approach prescribes this part takes two steps. First we will need to collect as many invoice numbers as possible. We assume have no access to the financial administration, thus the numbers should be found in the data available to us in the Corporate Network Data. Second, we will need to verify the found invoice numbers so we know with some certainty that we actually are modelling the right process and not a lot of noise.

During these two sub steps three different sets of numbers will be created, their relations are depicted in Figure 5.3. In the collection step a set of natural numbers S is created. Then during the validation step first the set of S'_{inv} is created such that $S'_{inv} \subset S$, this set is not completely the set of invoice numbers needed. The S'_{inv} is transformed into the S_{inv} in the final stages of this step.

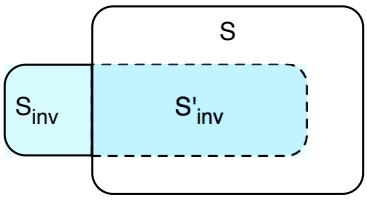

Figure 5.3: Reference numbers

5.3.1 Invoice number collection

While in search of as many invoice numbers as possible, a common problem arises. Although we want to retrieve all invoice numbers, (recall) we also want that what numbers we find are indeed all invoice numbers (precision). From a data-analysis perspective it is more important that the numbers found are really invoice numbers as noise will complicate further steps in the process. Due to the structure of the dataset and the composition of the invoice numbers it is hard to have a high recall and at the same time a high precision. Numbers are overly available in the available dataset and thus if high recall is requested, then there will be a lot of noise in the found invoice numbers. This leads to complications in further steps in the research and add unnecessary time consuming queries in a later stage of the process.

Reducing the size of the dataset

Due to the composition of the numbers we are after it is known that the dataset will contain a large amount of noise. Numbers originate in many files and attributes of files on computers, thus it is needed to reduce the dataset to only relevant set of data. Numbers have many different meanings depending on the context they are found in. They can be sizes of files, amounts of money transferred in various currencies, IDs used by the computer, telephone numbers, dates and of course invoice numbers. By trying to reduce the dataset to known contexts it is possible to curb the number of meanings a number can have.

Several contextual demarcations are made. First, only MS Office documents are included in the dataset. Initial exploration of the data showed that many processes include entering invoice information in spreadsheets and word processing files. Second, the set of documents is reduced to only the documents including the word 'invoice'. This sets the context of where the numbers 'live' in. Invoices still contain various numbers, including amounts of money, VAT numbers, invoice numbers, shipment numbers, but it is more focused on the numbers we want to extract. This means that the possibility that the number we found is actually an invoice number is much larger than that we would be searching in the full dataset. It was found in the initial exploration of the data that most of the files used for keeping track of invoices and orders are usually not over 1MB size.

So in order to reduce the possible contexts a number is found in the initial dataset needs to be reduced to only files with the attributes mentioned above. The resulting set of files are called F_n.

Defining the invoice number

Within the previous reduced set of files the plan is to extract numbers, but numbers can have various length and might include special characters like dashes, commas or full stops. Based on inspection of the invoice numbers in several files in F_n we have decided to reduce our search to numbers following the structure of the next regular expression:

```
'\b[1-9][0-9]{5,8}\b'
```

To explain the regular expression the following definitions:

`\b`

: Matches an empty string, but only at the beginning or end of a word. A word is defined as a sequence of alphanumeric or underscore characters, so the end of a word is indicated by whitespace or a non-alphanumeric, non-underscore character.

`[1-9]`

: Matches a string of 1 numeric characters of the value between 1 and 9 inclusive.

`[0-9]{5,8}`

: Matches a string of 5 to 8 numeric characters with each a value between 0 and 9 inclusive.

This means that the expression includes numbers of length 6 to 9 characters if they appear as a 'word', broadly meaning that they should have a non-alphanumeric, non-underscore character before and after the sequence of numbers. The sequence of numeric characters must not be separated by any non-numeric characters. Also the number must not start with a 0.

F_n is a subset of the *Corporate Network Data* (F_{cnd}). It reduces the full set to only files that contain digital documents that contain the word 'invoice' and are smaller than 1 Megabyte in size. This set is meant to find the reference numbers we need to find the transactions.

Mining the invoice number

In the actual invoice mining the above two concepts are used. Using the API to connect to Xiraf all documents from F_n are retrieved and the content searched for matches of the regular expression as described above.

Then we come to our first set that does not consist of files, but merely out of numbers, S. This set holds the numbers mined from the F_n. It holds around the 1.5 Million numbers and after deleting duplicates S still contains 207 Thousand unique numbers.

5.3.2 Invoice number validation

As been described earlier numbers can represent a variety of different concepts. While the context has been curbed already, still many, explanations for the numbers found can come up. Thus it is important to find out what the possibility is that a found number indeed is what we hope it is, an invoice number. The method described in Chapter 4 prescribes that the numbers should confirm the following three hypotheses.

1. We expect them to be ranges given that by (Dutch) law they should be given out in a range without missing values.

2. We expect higher numbers to first appear at a later time than the lower numbers of a range.

3. We should be able to find these numbers in the context of invoices when actually looking inside some of the documents.

Ranges

To identify ranges we have decided to look at the gaps between every two successive numbers in S. When the numbers are ordered ascending we looked at the distance between successive numbers. If this is 1 it means that there is no gap between this number and the next, if it is bigger then there is such a gap. To represent the results we have plotted this distance between numbers in Figure 5.4. Because the high variance between the different distances we have shown the distances on a logarithmic scale, where the scale in this picture is the color of the points. The scale is represented in heatmap colors, where in this case dark blue colors mean that the distance of the next value is rather low, while lighter blue up till red represent larger distances between consecutive numbers.

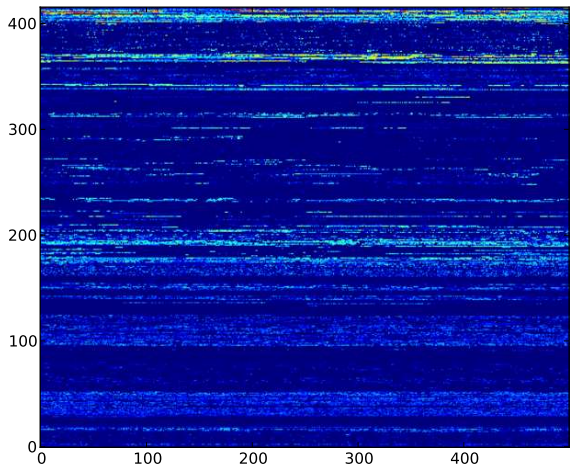

Figure 5.4: Distance between next number

The axis of the figure are just the enumeration of the values. We have found 207,000 numbers and the colour of the points in the heatmap show the gap between one number and the next starting at the left bottom and the highest found number is in the right top. It is clearly visible that there are dark blue 'bands' that represent low gaps between successive values between $y = 140$ and $y = 150$.

Using this figures we have identified a candidate successive range (S'_{inv}) and zoomed in on it. Due to the collection method we know we probably will not find a consecutive range of numbers in S'_{inv}. We also have seen that although we do see something that is very much a successive range it still is not consecutive. Since we know that, in the Netherlands, by law an organisation is forced to use a complete consecutive range, we decided that instead of only using the numbers we found, we inject the missing numbers in our range S'_{inv} and thus create our invoice number set S_{inv}. Due to confidentiality we decided to represent our range with values between 0 and 7000.

Increasing over time

Our second hypothesis about invoice numbers is that if they are to be consecutive that then lower number should appear earlier in time than higher numbers. In order to check that we have plotted the appearances of numbers in time in Figure 5.5.

The figure plots all occurrences of the numbers against the time of the file in which they are found. So if a number is found in more files it can be that there are more dots on the horizontal axis. If one file or more files with the same date, harbour more than one number there will be multiple dots along the vertical axis. When looking at the leftmost observation of an invoice number, which is the first appearance of that number, it is clearly visible that the red

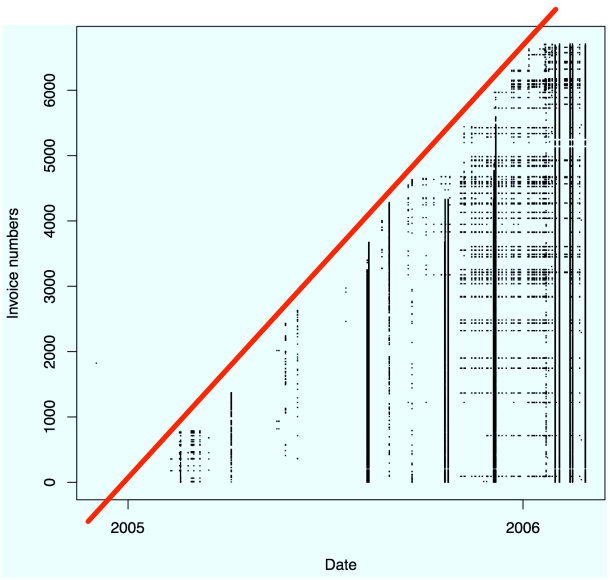

Figure 5.5: Numbers plotted in time with a linear line showing that higher numbers appear later in time than lower numbers

linear line increases over time and all observations but one are under that line. Hence higher numbers appear later in time than lower numbers.

There is only one noisy observation, around the invoice number 1900 we observed one early appearance of a number. Close inspection of the actual file it showed that the number in that context was not an invoice number, but a telephone number.

In-dept file analysis

The third hypothesis was that close inspection of a sample of the files numbers are found in and interpreting the context of the numbers should show that the numbers are indeed invoice numbers. We did inspect various files in which we did find numbers and the context all showed that the numbers found were indeed invoice numbers.

5.4 STEP 3: Document Discovery

To find the documents that will represent our activities later in the research we are now going to use the S_{inv} to find all documents that harbour one of our invoice numbers. We want to find all the files that can be linked to numbers from the S_{inv}, but we also want that if we find such a number in a file that again the number is an invoice number. To make sure that we find invoice numbers again we decided to apply approximately the same strategy as we did earlier during the invoice mining step. This time we did not restrict on filesize, so the

only restrictions are, first that it must be an MS Office document, since those are used to support some of their tasks related to financial transactions and second that it must include the word 'invoice'. The resulting set is called F_e.

The extraction of the workflow information means that we need to extract all the files or documents that contain information about the transactions we are looking for.

A file containing a invoice numbers is seen as an event, because we assume that when a file is saved some event or activity linked to a transaction has taken place.

So for each invoice number found in the previous step of extracting invoice numbers, we now try to extract all the events linked to them. For this we use S_{inv} as input and F_e to search in for files. Using the API from Xiraf we can easily extract all files containing numbers from the S_{inv}.

The resulting set we call F_{inv} which is a subset of F_e and it overlaps partly with F_n. It contains all files that contain one or more of the invoice numbers from S_inv. The number of documents discovered is 618 and they contain between 1 and 6100 numbers each.

Because building this dataset takes quite a long time the information is extracted from the Xiraf system and the metadata is stored in a database.

The stepwise approach prescribes the format of the found documents to be defined as:

Definition Let a *document* be $d = \{s_{inv}, fn_d, l_d, t_d, u_d\}$ (see Section 4.4)

The mapping of the actual data found in our dataset to format as prescribed in the definition of a document is shown in Table 5.1.

Attribute	File Attribute Used
s_{inv}	Invoice Number
fn_d	Path (partly)
l_d	Path (partly)
t_d	lastSavedOn
u_d	lastAuthor

Table 5.1: The mapping from the definition of documents to the available information.

This information is directly stored as raw data in a database, then we transformed it into a datawarehouse. The facts are then describing the fact that an event has taken place. The dimensions are *time*, *user*, *location* and *files*.

Following steps will use as time the lastSavedOn value, as that is the best predictor of when the process has taken place. For the user information it is chosen to use the lastAuthor, as that is probably the one that performed the task. From the path of the files two different pieces of information is retrieved. First, it can tell us on which system the file is found. Second, the filename is later used to categorize the files into similar files and used to name the activity the file support.

5.5 STEP 4: Activity extraction

We assumed that we would find quite a few files that are alike, not only in format but also in name. Most organisations apply some sort of a naming scheme for the files that are vital for their processes. When a list of all the files found is made they should be grouped in similar files. We did this simply by looking at similar named files. From the 618 files found we could group them in about 100 groups, of which some contained only one file and others up to sixty. Of some groups files were examined and labelled by an expert. Some of the groups were discarded as being noise, for instance where a file contained few numbers, that while looking at the context turned out to be a phone numbers.

Confidentiality issues make it impossible to provide the full list of activities, but we want to explain a few examples. First there is the activity we called 'Outstanding', this represents the listing of all outstanding, thus not yet payed, orders. We have seen that this activity was performed on a weekly basis for quite some time during the used time window. Also the activity 'Profit and Loss' is one of the activities that caught our eyes. The profit and loss balance has been created often in the time window we were looking at.

After this labelling of all the files to activities all the information to create event logs is available.

Chapter 6

Results

6.1 Event Logs

The resulting data from the four steps taken comprises all information needed to create the event logs we are after. Per event, which is a document, we have the invoice number to represent the case ID. The activities is extracted from the name and the content of the document. And the order have been deduced from the timestamps available per document. If this information is stored in a simple text file we have the event log. In our case this is an event log with 6367 cases and total of 194524 events in a timeframe of just over a year.

6.2 Workflow mining

Using the ProM tool from vdAalst et al. the event log is mined and the resulting workflow model is shown in Figure 6.1. The mining has been done using the HeuristicsMiner algorithm [27]. The choice for HeuristicsMiner is because it can cope with the possibility of noise and the possibility to show or hide the noise. As this noise is probably exactly where possible fraud will be visible.

The resulting model is quite complex with many connections, loops and shortcuts. We have not fully interpreted the resulting model, but we have seen some interesting behaviour during a simulation of the model. Using the ProM tool we have created a simulation of the mined model. This simulation visualizes all the cases in the event log and shows how they travel through the model. It appears to be the case that when for instance an invoice comes into the activity 'outstanding', which is the activity in which all outstanding invoices are collected, it often loops back to that same activity for a considerate amount of times.

It is also quite interesting that there are indeed a few paths within the model that are taken far more often than others.

6.3 Validation

Validating the actual workflow model is not possible since there is no access to the organisation the data originates from. And even if it was possible to

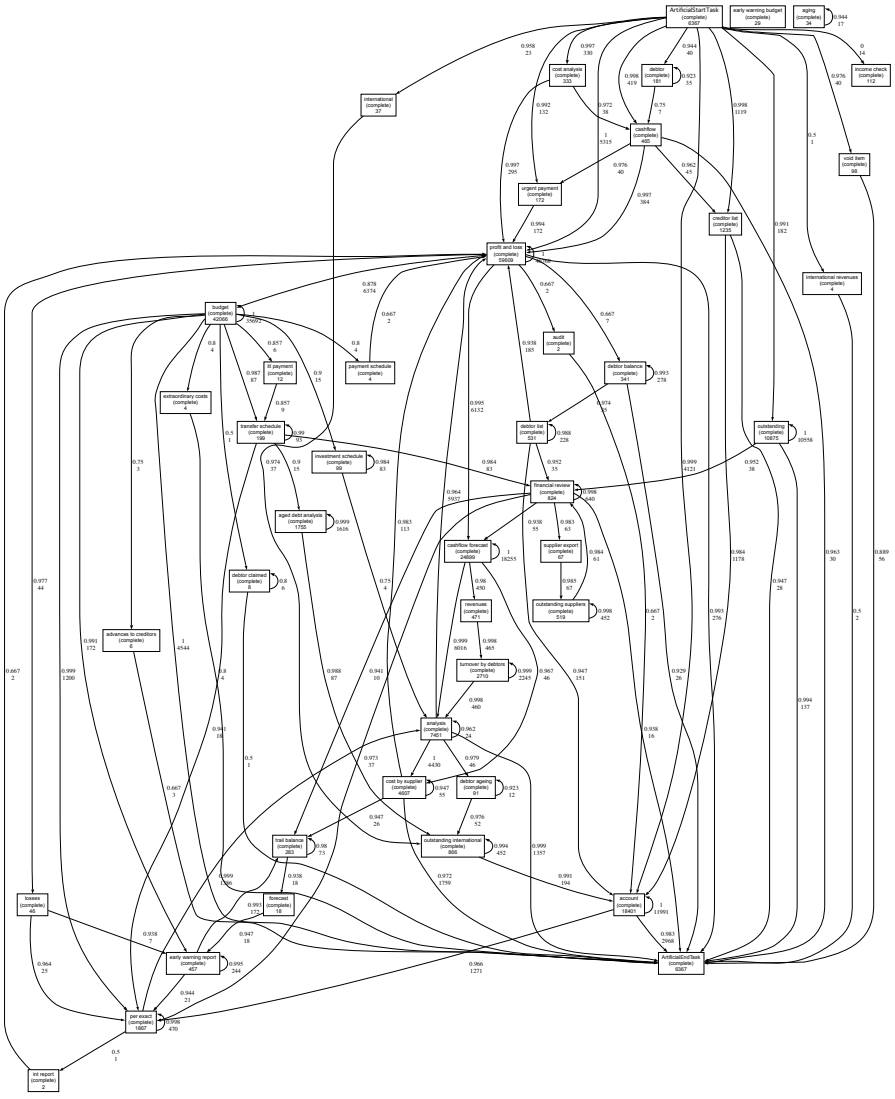

Figure 6.1: The resulting Workflow model.

interview the employees it would be rather difficult to extract the model by that means. This is one of the reasons research in mining workflows has started.

Further more the invoice numbers themselves have been validated to actually be invoice numbers using the steps defined in the stepwise approach. Also the files in F_n have been validated to indeed contain invoice numbers.

6.4 Evaluation

We have shown in this case study that the requirements set in Chapter 3 have been met. We have extracted the case ID from the Corporate Network Data. We have also extracted the activities from the filenames and the content of files. And we have been able to extract the order in which the activities take place in the cases using the timestamps of files.

And we have shown that we with this stepwise approach we have bridged the gap between the availability of Corporate Network Data and the possibility to apply workflow mining techniques. We have done that by applying one of those techniques on the output of our stepwise approach which resulted in a workflow model. What we cannot say is how good the workflow model represents the real processes within the organisation the Corporate Network Data originates from as discussed in Section 6.3.

6.5 Conclusion

With this case study we wanted to show the applicability of the designed stepwise approach. And we wanted to show that it is conform the requirements as posed in Chapter 3. The Evaluation shows that the requirements have been met. Also has the bridge between the availability of Corporation Network Data and the possibility of workflow mining been demonstrated.

Chapter 7

Discussion

The ambition level of the initial problem definition turned out to be too high within the time available to perform the research. The initial idea was to find fraudulent invoices using the assumption that cases that follow a workflow that happens infrequent compared to other workflows is suspicious. The time needed to prepare the data for workflow mining turned out to much longer and the road to there non-existent. This lead to the research being ended before the initial goal could be reached. Still it did lay a new road as to how to prepare the poorly structured data of Corporate Network Data for workflow mining.

The most important feature of this research is that it makes it possible, with some reservations, to mine workflows regarding financial transactions from a vast amount of poorly structured data. We have shown how to structure the data into a known structure called event logs, from there proven methods can be used to extract workflows that give insight in the paths a specific item, in this case the invoice, travels.

Although we have taken the invoice as a means to demonstrate this principle, it could also be argued that for instance tracking the workflows around contracts could be possible. If there is a possibility to find all the documents surrounding a contract and if there is an identifier that shows which documents belong to a specific case the same approach can be taken in order to find the workflows around contracts. This approach can be used in e-discovery where fraudulent activities are tried to be found.

The preprocessing of the data to make it fit the input format for software that can do the workflow mining was difficult. We have developed some scripts that help, but those are not generic enough t be used on other datasets that the one used in this research.

Chapter 8

Conclusions

8.1 Conclusions

The problem was to bridge the gap between the availability of Corporate Network Data and the possibility to apply workflow mining techniques in order to get a wider view on processes within an organisation. the format and information needed for input in the existing worflow mining techniques, event logs, can not directly be taken form the Corporate Netowrk Data. Our stepwise approach is designed to make it possible to extract all information needed to create those event logs. Those event logs can then be used as input for the workflow mining techniques.

We have shown in Chapter 4 that we developed a stepwise approach to distill event logs from Corporate Network Data by applying four distinct steps.

First, indexing of the Corporate Network Data. Second, finding the invoice numbers hidden amongst the data. Third, finding the documents that contain the found invoice numbers. Fourth, extracting the activities supported by the documents found.

This gives the input to create the event logs needed so we can apply proven methods to extract the workflow model around the invoices.

In Chapter 5 we have shown how to apply the stepwise approach to construct an event log from Corporate Network Data. And how the resulting event logs can actually be used as input for existing and proven methods to develop a workflow model. The case study shows that the approach can be applied to corporate network data and that it results in event logs which produce a workflow model.

The resulting workflow model though is very hard to verify, as we did not have access to the employees of the organisation where the data originated and because it is inherently difficult to validate the real workflows even if access to the employees would have been possible. Hence the resulting workflow model is not verified.

Although the validation is not rock solid, the designed approach does look promising to be further researched and adapted.

8.2 Further research

As already noted, the research has not reached it initial goal. Further research could be done, both towards enriching the method itself and towards applications of the method to everyday problems.

8.2.1 Enriching

Value of invoices

If one could also add the monetary value represented by invoice to the invoice numbers, one could then search for fraud by looking if and when the value of the invoice changed.

Organisational models

If the users are mapped to real persons it can very well show that one case is for instance handled by a different user. Which in case of a search for fraudulent invoices could point to an anomaly in the usual workflow. It could also give more insight in the roles users fulfill in the organisation as Ang et al. show in their research [3].

8.2.2 Application

Fraud Detection

If one can really identify the 'norm' in the process it would be interesting to identify deviating patterns, being patterns that happen only sporadically. Assuming that often occurring patterns are legitimate one could then focus on the deviating patterns that are found. That would give more insight in whether that transaction was legitimate or not.

e-discovery

Fraud detection is actually one of the possibilities of e-discovery, but using this stepwise approach not only the workflows around invoices can be found. If the right identifier is found for other processes the same approach can be used to for instance find the workflows around contracts.

Bibliography

[1] Wet op de omzetbelasting 1968 hoofdstuk vi afdeling 4 artikel 35a 1.b. Accessed 05-10-2010.

[2] W. Alink, R. Bhoedjang, P. Boncz, and A. de Vries. Xiraf - xml-based indexing and querying for digital forensics. *Digital Investigation*, 3(Supplement 1):50 – 58, 2006. The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06).

[3] G. Ang, Y. Yang, Z. Ming, J.-L. Zhang, and Y.-W. Wang. Organizational structure mining based on workflow logs. pages 455–459, 2009.

[4] M. Berlingerio, F. Pinelli, M. Nanni, and F. Giannotti. Temporal mining for interactive workflow data analysis. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–118, New York, NY, USA, 2009. ACM.

[5] J. Cook and A. Wolf. Discovering models of software processes from event-based data. *ACM Transactions on Software Engineering and Methodology*, 7(3):215–249, 1998.

[6] J. E. Cook and A. L. Wolf. Software process validation: quantitatively measuring the correspondence of a process to a model. *ACM Trans. Softw. Eng. Methodol.*, 8:147–176, April 1999.

[7] A. de Medeiros, A. Guzzo, G. Greco, W. van der Aalst, A. Weijters, B. Van Dongen, and D. Sacca. Process mining based on clustering: A quest for precision. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4928 LNCS:17–29, 2008.

[8] A. de Medeiros, B. van Dongen, W. van der Aalst, and A. Weijters. Process mining for ubiquitous mobile systems: An overview and a concrete algorithm. In L. Baresi, S. Dustdar, H. Gall, and M. Matera, editors, *Ubiquitous Mobile Information and Collaboration Systems*, volume 3272 of *Lecture Notes in Computer Science*, pages 151–165. Springer Berlin / Heidelberg, 2005.

[9] F. S. Esfahani, M. A. A. Murad, M. N. Sulaiman, and N. I. Udzir. Using process mining to business process distribution. In *SAC '09: Proceedings of the 2009 ACM symposium on Applied Computing*, pages 2140–2145, New York, NY, USA, 2009. ACM.

[10] W. Gaaloul, K. Gaaloul, S. Bhiri, A. Haller, and M. Hauswirth. Log-based transactional workflow mining. *Distrib. Parallel Databases*, 25:193–240, June 2009.

[11] L. Geng, S. Buffett, B. Hamilton, X. Wang, L. Korba, H. Liu, and Y. Wang. Discovering structured event logs from unstructured audit trails for workflow mining. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5722 LNAI:442–452, 2009.

[12] D. Georgakopoulos, M. Hornick, and A. Sheth. An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3:119–153, 1995.

[13] S. Goedertier, J. de Weerdt, D. Martens, J. Vanthienen, and B. Baesens. Process discovery in event logs: An application in the telecom industry. *Applied Soft Computing*, 2010.

[14] G. Greco, A. Guzzo, and L. Pontieri. Mining taxonomies of process models. *Data & Knowledge Engineering*, 67(1):74 – 102, 2008.

[15] M. Hammori, J. Herbst, and N. Kleiner. Interactive workflow mining–requirements, concepts and implementation. *Data & Knowledge Engineering*, 56(1):41 – 63, 2006. Business Process Management.

[16] S. He, T. Lv, and B. Huang. A new process mining algorithm of workflow. pages 83–85, Haikou, 2009. cited By (since 1996) 0; Conference of 2009 International Conference on Industrial and Information Systems, IIS 2009; Conference Date: 24 April 2009 through 25 April 2009; Conference Code: 77555.

[17] N. Kushmerick and T. Lau. Automated email activity management: An unsupervised learning approach. pages 67–74, 2005.

[18] J. Li, D. Liu, and B. Yang. Process mining: Extending α-algorithm to mine duplicate tasks in process logs. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4537 LNCS:396–407, 2007.

[19] L. Maruster, A. Weijters, W. Van Der Aalst, and A. Van Den Bosch. A rule-based approach for process discovery: Dealing with noise and imbalance in process logs. *Data Mining and Knowledge Discovery*, 13:67–87, 2006. 10.1007/s10618-005-0029-z.

[20] W. van der Aalst. Process-aware information systems: Lessons to be learned from process mining. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5460 LNCS:1–26, 2009.

[21] W. van der Aalst, B. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. Weijters. Workflow mining: a survey of issues and approaches. *Data Knowl. Eng.*, 47:237–267, November 2003.

[22] W. van der Aalst and A. Weijters. Process mining: a research agenda. *Computers in Industry*, 53(3):231 – 244, 2004.

[23] W. van der Aalst, T. Weijters, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.

[24] B. van Dongen, A. Alves de Medeiros, and L. Wen. Process mining: Overview and outlook of petri net discovery algorithms. In *Transactions on Petri Nets and Other Models of Concurrency II*, volume 5460 of *Lecture Notes in Computer Science*, pages 225–242. Springer Berlin / Heidelberg, 2009.

[25] Z. Weidong, D. Weihui, W. Anhua, and F. Xiaochun. Role-activity diagrams modeling based on workflow mining. volume 4, pages 301–305, 2009.

[26] A. Weijters and W. Van der Aalst. Rediscovering workflow models from event-based data using little thumb. *Integrated Computer-Aided Engineering*, 10(2):151–162, 2003.

[27] A. Weijters, W. van der Aalst, and A. A. de Medeiros. Process mining with the heuristics miner-algorithm. BETA Working Paper Series, WP 166, Eindhoven University of Technology, Eindhoven, 2006., 2006.

[28] L. Wen, W. van der Aalst, J. Wang, and J. Sun. Mining process models with non-free-choice constructs. *Data Mining and Knowledge Discovery*, 15:145–180, 2007. 10.1007/s10618-007-0065-y.

[29] L. Wen, J. Wang, W. M. van der Aalst, B. Huang, and J. Sun. Mining process models with prime invisible tasks. *Data & Knowledge Engineering*, 2010.