CERTIFICATE

This is to certify the project synopsis entitled **"Big data quality: Early Detection of Errors in Process flow using Petri Nets with Data",** submitted by Melody W. Murakwani **(Roll No: 2K13/SWE/26)** as the record of the work carried out by her, is accepted as the Major 2 Project Synopsis submitted in partial fulfillment of the requirements for the award of **Master of Technology Degree** in **Software Engineering** in the Department of Computer Engineering, Delhi Technological University, Delhi.

Mr. Manoj Sethi
Project Guide
Professor
Dept of Computer Engineering
DTU, Delhi

Date:----------

ACKNOWLEDGEMENTS

This thesis owes its existence to the help, support and inspiration of several people. Firstly, I would like to express my sincere appreciation and gratitude to Mr. Manoj Sethi for his wisdom, and guidance during my research.

My greatest gratitude goes to my family and friends, particularly to my husband for his unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

Finally, I would like to thank the Lord God Almighty from whom all blessings flow.

Melody Wadzanayi Murakwani
Roll No: 2K13/SWE/26
M.Tech Software Engineering
Department of Computer Engineering
Delhi Technological University

ABSTRACT

In our Big Data era, data is being produced at scale, in motion, and in heterogeneous forms. Uncertainty is another significant attribute exhibited by this data and hence there is need to comprehend and (perhaps) repair erroneous data timely. Due to heterogeneity of data source and usage, data quality rules are contextual; hence we require data management solutions that acknowledge these varied uses and incorporate them to determine the level of quality and standardization required. Today, there is a wide range of process mining techniques that are able to uncover the reality of processes through a systemic analysis of event data. These techniques are being applied in this work with the aim to isolate the source of the introduction of data flaws to fix the process instead of correcting the data. This paper employs the Heuristic Miner algorithm for process discovery, Petri nets with data (DPN nets) and conformance checking using alignments and compliance rules. We showed that alignments between event logs and the discovered Petri Net from process discovery algorithms reveal frequent occurring deviations and compliance rules are an effective data management solution. Insights into these deviations are then exploited to repair and enhance the original process models. Our novel diagnostic data-aware process discovery technique is applied on a real-life event log and evaluated for its success in providing new and valuable insights and failure in other areas of performance.

Table of Contents

List of figures

List of Tables

Introduction

This chapter gives introduces this study with a brief discussion of the objective and purpose of the research. The chapter concludes with a brief description of how this thesis is organised.

While the potential and promise of Big Data is real, for instance, it is estimated that Google alone contributed 54 billion dollars to the US economy in 2009 [1], recent studies show that poor quality data issues widen the gap between its potential and its realization. Erroneous data is estimated to cost US businesses 600 billion dollars annually [2]. The level of inaccurate data is a cause for concern when one considers how much research and industry are relying on information for analytics and business. There is an increasing demand to improve data quality so as to add accuracy and value to research and industry alike.

The thrust for this work is to isolate the source of the introduction of data flaws to fix the process instead of correcting the data, thereby reducing data cleaning costs. Big data characteristics include heterogeneity and veracity, and veracity is a big data characteristic which directly refers to inconsistency and data quality problems [3]-[6]. Process mining as a new and promising research field has shown strong capabilities to mine knowledge from data as it applies both process modelling and data mining techniques to discover models from the event logs. By leveraging IT footprints, process mining attempts to create a realistic picture of the process as it actually takes place, and as a consequence enables targeted adjustments to improve the performance or compliance of the process. The gained transparency of what is actually going on is a huge value in itself. Moreover, knowledge of the current status is also a prerequisite for any improvement actions, because one can only improve what they can measure.

The greater chunk of research in the field of process mining focuses on the control-flow aspects of a process and the data-flow perspective is not awarded much attention. The control-flow perspective presents the order of process' activities, but ignores data movements within the process. This research adopts data-aware process mining introduced in [14] to

show that the data-flow perspective coupled with resource network analysis and conformance checking using compliance rules may bring improvements to the process.

This data driven fault detection enabled by use of process mining techniques will enable us to do diagnostic and enhancement of existing processes towards data quality improvement. This work is inspired by Massimiliano de Leoni et al in [14] where they use recent advances in conformance checking using alignments, and they also employed the use of Petri Nets with Data, in their paper: Data-Aware Processing Mining: Discovering Decisions in Processes Using Alignments.

Figure 1 offers an overview of the thesis' structure. It focuses on the main aspects of each chapter. First three chapters provide an analysis of the research done in the domains of interest for this work which include Big Data, Big Data Quality as a challenge, Process Mining and the Literature Review

First chapter gives an overview of big data quality and the Process Mining technology. Firstly, we give a discussion that includes big data quality as a challenge, its dimensions and its impact on analytics. Then, we go through Process Mining, in order to familiarise with the technique and its underlying technologies. Mining algorithms that extract the data-flow perspective of a process are defined here.

Chapter 2 presents the existing methods employed to discover and check for conformance of the data-flow perspective in processes. The literature shows the existence of mixture techniques that bring together data-flow perspective with control-flow with a bias towards analysis of the control-flow perspective. This chapter also gives the problem definition as well as the aim for this work.

The third, fourth, fifth and sixth chapters emphasize the contributions of the thesis. Chapter 3 proposes the implemented methodology for analysis and its validation on both real life and synthetic event logs.

Chapter 3 is the main formal contribution of this thesis, and focuses on the step by step methodology that first defines conformance rules for data-flow in a process, then, discovers the data-flow and resource-aware perspective, before finally applying conformance rules of the data-flow and resource aware perspective to diagnose deviations.

Chapter 4 shows the implementation work which started with acquiring the event log. The steps defined in the methodology are executed on real life event logs.

Chapter 5 details our analysis of the observed results.

Chapter 6 validates the proposed early data-aware fault detection in process flow.

Finally, chapter 7 concludes the thesis and proposes the future work.

Chapter 8 provides our publication from this work.

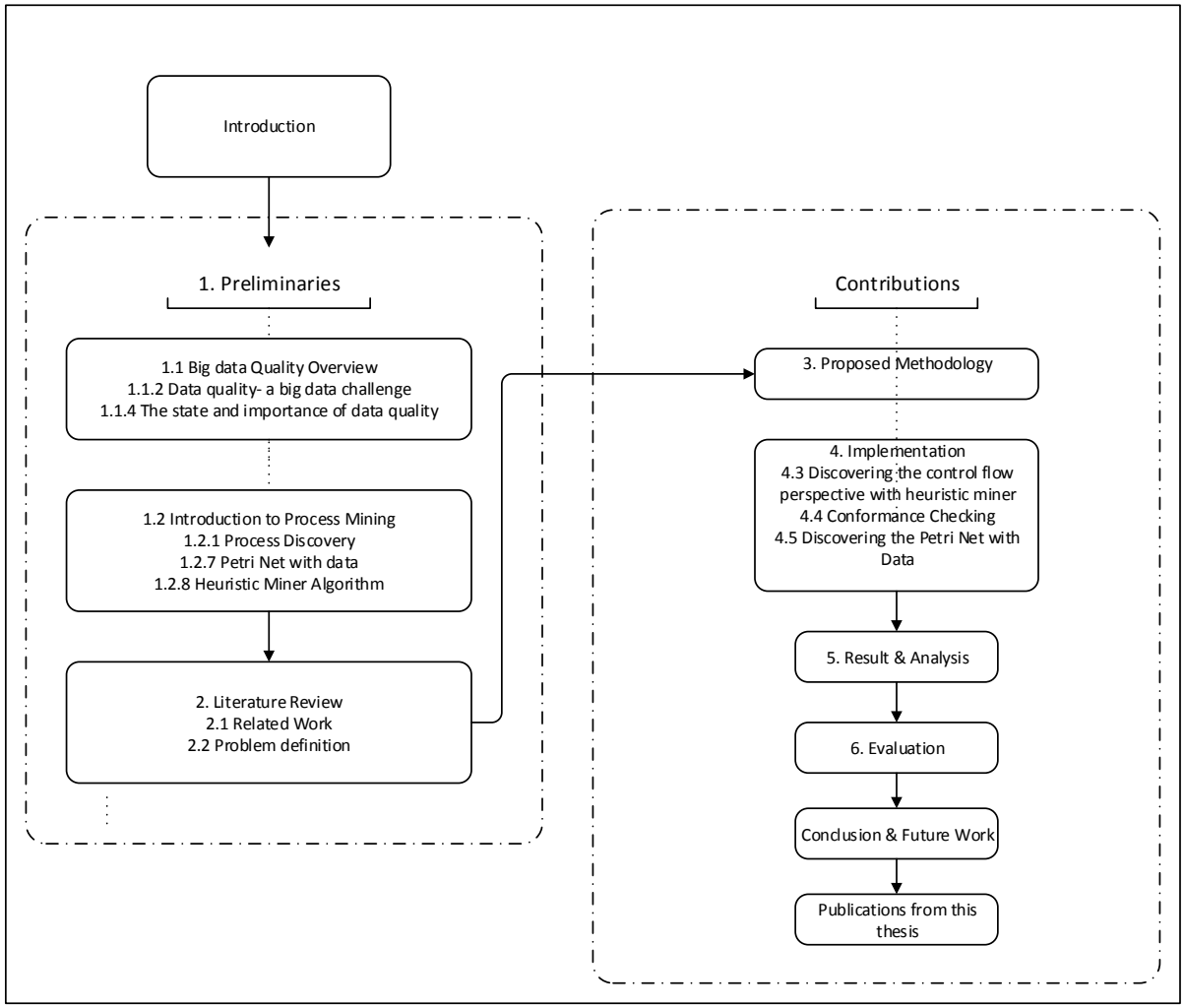


Figure 1-Overview of the thesis

Chapter One: Preliminaries

In this chapter we introduce key concepts of process mining that will be employed in proceeding chapters and discuss the state of data quality. First, we look at what the term Big Data entails, then we give an overview of Big Data challenges and go on to discuss at length the quality issues associated with Big Data environments. Then, we introduce process mining, and two aspects of process mining which are process discovery and conformance checking.

1.1 Big Data Quality Overview

1.1.1 Big Data Overview

It is often said that data is the new oil; to illustrate some economic, research, societal, and legal issues concerning big data. Like oil, data is a multifaceted product derived from several refinement and processing steps and a whole economic ecosystem involving drilling activities, refineries and distribution network plans, which include fuel stations. Likewise, "Big Data is data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract value and hidden knowledge from it…" [7],[8].

Big data has invoked the growth of an entire economy of distribution networks with data marketplaces that sell raw, semantically enhanced, and further improved forms of data.

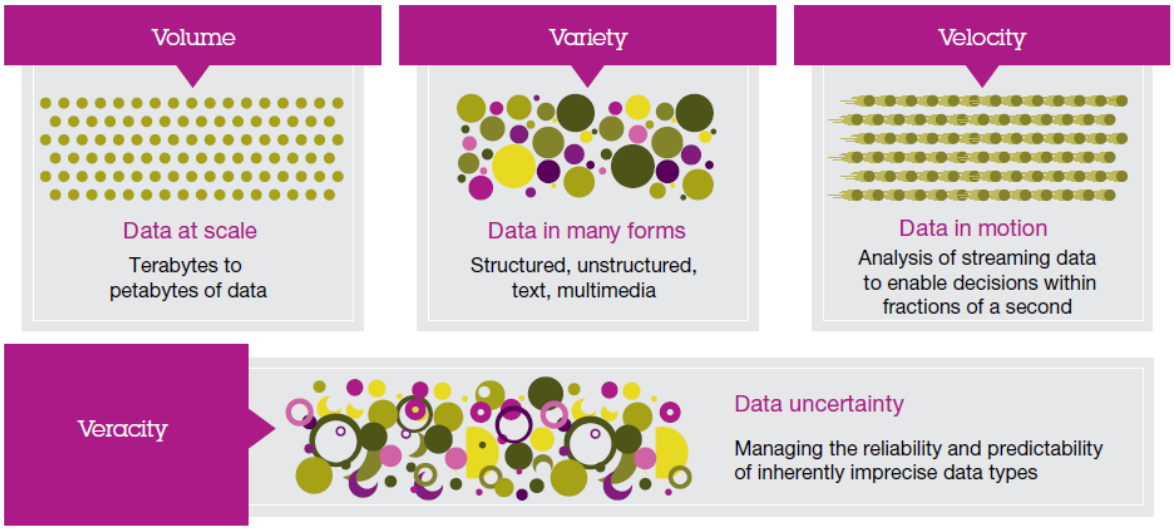


Figure 2: The 4Vs that define Big Data [6]

1.1.2 Data Quality- a big data challenge

Veracity (data uncertainty) is the fourth characteristic of big data which refers to the level of reliability associated with certain types of data. To harness the full potential of big data, high data quality is an important big data requirement and challenge.

1.1.3 Data quality and its dimensions

Data quality is a perception of data to be fit for its purpose. Fitness for use has many aspects which include:

Accuracy-the degree to which the data correctly reflects a verifiable source

Completeness- the degree to which it can be verified that all necessary data is present

Relevance-the degree to which data is applicable to the needs of users.

Accessibility-the ease with which the information is obtained

Timeliness-the degree to which the data is available when needed

Consistency- the degree to which data is consistent between different systems

In the white paper by Thomas Schutz, [9], as shown in the figure 3, they performed a research to identify the reason for poor quality in data which reported that information collected across various channels is frequently exposed to human error as consumers and individual employees enter information manually. Collectively, 78 percent of companies have problems with the quality of data they collect from various channels. Globally, call centres produce the poorest data quality, followed by websites. The level of inaccurate data relates to a lack of a sophisticated data management strategy [9].

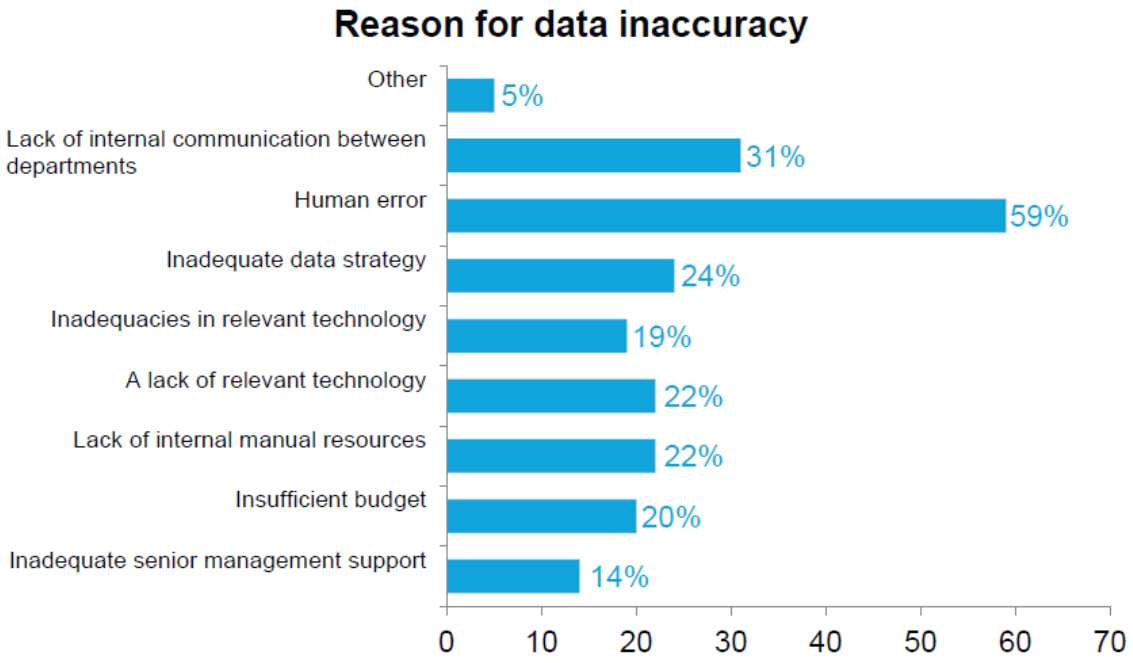


Figure 3- Reason for data inaccuracy [9]

1.1.4 The state and importance of data quality

Data quality continues to be a challenge for many organizations as they look to improve efficiency through data insight. 91 percent of companies suffer from common data errors [9]. The most common data errors are incomplete or missing data, outdated information and inaccurate data. Because of the prevalence of these common errors, the vast majority of companies suspect their contact data might be inaccurate in some way. Globally, the average amount of inaccurate data has risen to 22 percent in 2013 from 17 percent in 2014. U.S. organizations actually believe they have the highest percentage of inaccurate data at 25 percent [9].

The level of inaccurate data is staggering when one considers how much research and industry are relying on information for analytics and business. The main cause of inaccurate data remains to be human error, which has consistently been the main cause of errors in the past. While all other causes clearly lagged behind the front runner, they include a lack of communication between departments and technical limitations as shown in figure 3.

Enterprises typically find data error rate of approximately 1-5%, and for some companies, it is above 30% [10], [11]. In most data warehousing projects, data cleaning accounts for 30-

80% of the development time and budget for improving the quality of the data rather than building the system. On the web, 58% of the available documents are XML, among which only one third of XML documents with accompanying XSD/DTD are valid [12]. 14% of the documents lack well-formedness, a simple error of mismatching tags and missing tags that renders the entire XML-technology useless over these documents. These all highlight the pressing need of data quality management to ensure data in our databases represent the real world entities to which they refer in a consistent, accurate, complete, timely and unique way. There has been increasing demand in industries for developing data quality management systems, aiming to effectively detect and correct errors in the data, and thus to add accuracy and value to business processes. Indeed the market for data quality tools is growing at 16% annually, way over the 7% average forecast for other IT segments [13].

With the advent of big data, data quality management has become more important than ever. Veracity directly refers to inconsistency and data quality problems. As [14] states, one of the biggest problems with big data is the tendency for errors to snowball. Without proper data quality management, even minor errors can accumulate resulting in revenue loss, process inefficiency and failure to comply with industry and government regulations (the butterfly effect [15]).

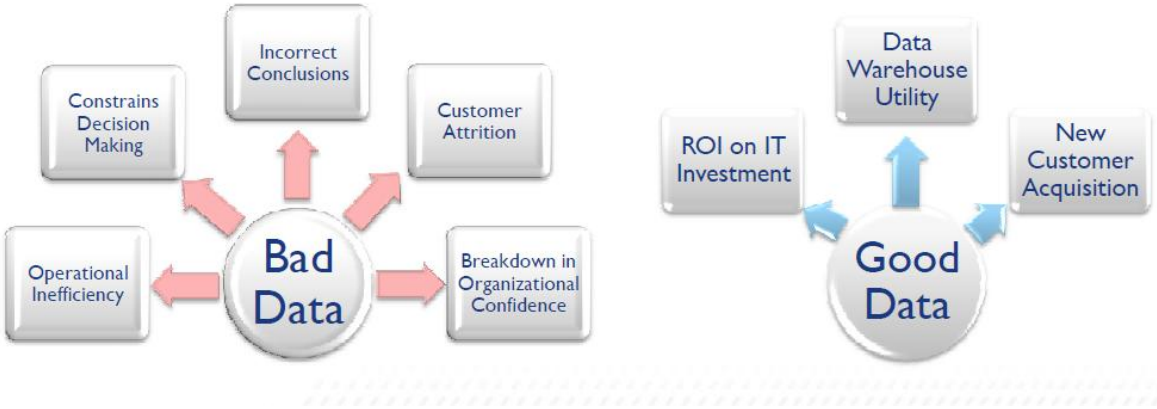


Figure 4- The importance of data quality [15]

1.1.5 Big Data quality challenges

The challenges of quality in big data are tied to the rapidly increasing volumes of data, the speed at which data is created, the heterogeneity and uncertainty of data because data rules cannot be specified apriori.

i. Heterogeneity

Data is received from multi-structured sources; hence there is no control of the created data, such that determining the semantics of data and understanding correlations between attributes is an intimidating task.

ii. Incompleteness

Data may be incomplete due to user entry error or its heterogeneity. In some cases even after data cleaning and error correction, some incompleteness and some errors in data are likely to remain. This incompleteness and these errors must be managed during data analysis. Doing this correctly is a challenge. Recent work on managing probabilistic data suggests one way to make progress.

iii. Scale

Due to its rapid growth in size there is a huge demand on better processing capabilities which requires new ways of determining how to run and execute data processing jobs so that we can meet the goals of each workload cost-effectively, and to deal with system failures, which occur more frequently as we operate on larger and larger.

iv. Timeliness

Larger data sets take longer to be processed for analysis. For example, if a client applying for a financial loan is being verified, at the first indication of previous bad debt record it should ideally be marked before the loan is approved. Unfortunately, a thorough analysis of a client's debt history is not likely to be practical in real-time.

v. Inconsistency

Errors in big data have the tendency to escalate. Data capture errors, redundancy, and corruption all affect the value of data, as minor errors can accrue resulting in direct and indirect costs, and process proficiency (the butterfly effect [15]).

1.1.6 Big Data Quality Management

1.1.6.1 Quality is contextual

The over-arching principle of data quality in a data warehouse is a flat earth model. There is one data model and one set of integrated data. Therefore there is one set of rules for the cleaning and standardizing of data before it is loaded. These rules ensure that the data is clean enough for the most stringent use case. This is good for the transactions that data warehouses were designed to manage. You don't want financial data with material impacts to be incorrect because it's expensive or slow to clean and load.

The problem with this approach is that it doesn't factor in other uses. Quality is a measure of the fitness for a particular use. The data may not be good enough to provide public financial forecasts, but it is good enough to estimate how much inventory to have on hand to avoid stores running out of stock. Transaction data, being core to the business, has the strictest requirements. The data warehouse has its origins managing this type of data, and the associated data requirements.

Big data applications, in contrast, use interaction and transaction data, and use that data for different purposes. These purposes are most often directional in nature, rather than providing an absolute answer to a question like "what was the profit on this product line last quarter?" At other times, they may be very specific to the action to be taken for a specific customer. The varied uses should dictate the required level of standardization and quality required.

1.1.6.2 Quality is expensive

The interaction data that is created by people is usually text rather than discrete values in a form. Much of this data is noisy and contains gaps, however analyzing the patterns over periods of time or in large collections of events provides useful information. This is a form of data extraction, involving analytic techniques rather than queries or the data processing techniques used for warehouse extraction and loading. Analytics require different cleaning, standardizing and formatting than the data in data warehouses; you are preparing data for machine consumption rather than human-readable reports. The type of data cleansing required is heavily dependent on the analytic techniques to be applied. This makes it hard to anticipate in advance exactly how to clean, standardize and store the data.

Cleansing the data ahead of time means it will be optimal for some analytics, but not others, and it still will not be in a human-usable format for querying. Because of these restrictions, the event streams and interaction data are often better left in a raw or lightly processed form. To do otherwise would incur significant costs given the complexity and volume of data.
The output of analytic processing is delivered to people or into operational systems. In many cases it's more important to manage this information than the large collection of source data fed into the processes that created it. There is no quality concern in the usual sense since this data is the output of an analytic process, but the lineage and other aspects are still important.

The data management nuances and tradeoffs are important to consider when working with big data. By storing the source data you are moving the time and effort it will take to process the data into a usable form for consumption. In a data warehouse you expend this effort up front, the benefit being quick response times to user queries. In a big data platform you often wait to do this until you're sure it's an ongoing requirement. This approach trades the time to access and use data for the speed one can make data available for use. The net effect is more rapid projects and lower development effort, and therefore cost. When processes become repeatable and reusable, the data can then be cleaned and transformed on load rather than on use.

1.1.6.3 Tracing data quality through the process

The primary use cases for big data are storage for large volumes of detailed data that would otherwise be archived from a conventional data warehouse, large scale processing of event streams, processing of textual or non-tabular data, and both batch and ad-hoc analytic processing. This difference in use doesn't exempt a big data platform from the data governance that has been implemented in the BI world, although how and where it is implemented may change. In particular, links between master data and the processes that create event. The logs must be put in place. Without these, it can be impossible to tie interaction data back to the reference data that gives it meaning.

Data quality is vital when it comes to the management of unique identifiers for the core elements of the business: products, customers and channels. As a result, lack of quality means that data from heterogeneous sources cannot be linked. Data standardization has less

emphasis because a big data platform isn't used for daily operational reporting. When quality is important, the scalable processing capabilities can be used to clean the data at the time of use rather than cleaning it all before storage.

Big data offers many new capabilities that extend what can be done with information beyond BI and data warehousing, but only with proper attention to the data being produced and used. We state some of the questions that can be answered by employing process mining which motivate this work to employ process mining techniques for early fault detection in data.

1.2 Introduction to Process Mining

Classical data mining techniques such as association rule learning, clustering, classification, regression, and sequence/episode mining are often only used to analyze a specific step in the overall process and generally do not analyse business process models. Process mining thrusts on end-to-end processes and is made possible owing to the growing availability of IT footprints and new model enhancement, conformance checking and process discovery techniques.

Process models are used for study and enactment of workflow systems. In the past, process models were classically made by hand without the use of event data, such as done is simulation. Nowadays, activities executed by human resources, machines, and software leave event trails in event logs. Process mining techniques make use of these event trails to mine, analyze, and improve business processes. Process mining aims to discover, monitor and improve real processes by mining knowledge from event logs available in computer systems [17]. Over the last decade there has been a spectacular growth of event data and process mining techniques have matured significantly.

There are three major types of process mining: which are process discovery, conformance checking and model enhancement as shown in Figure 5 below:

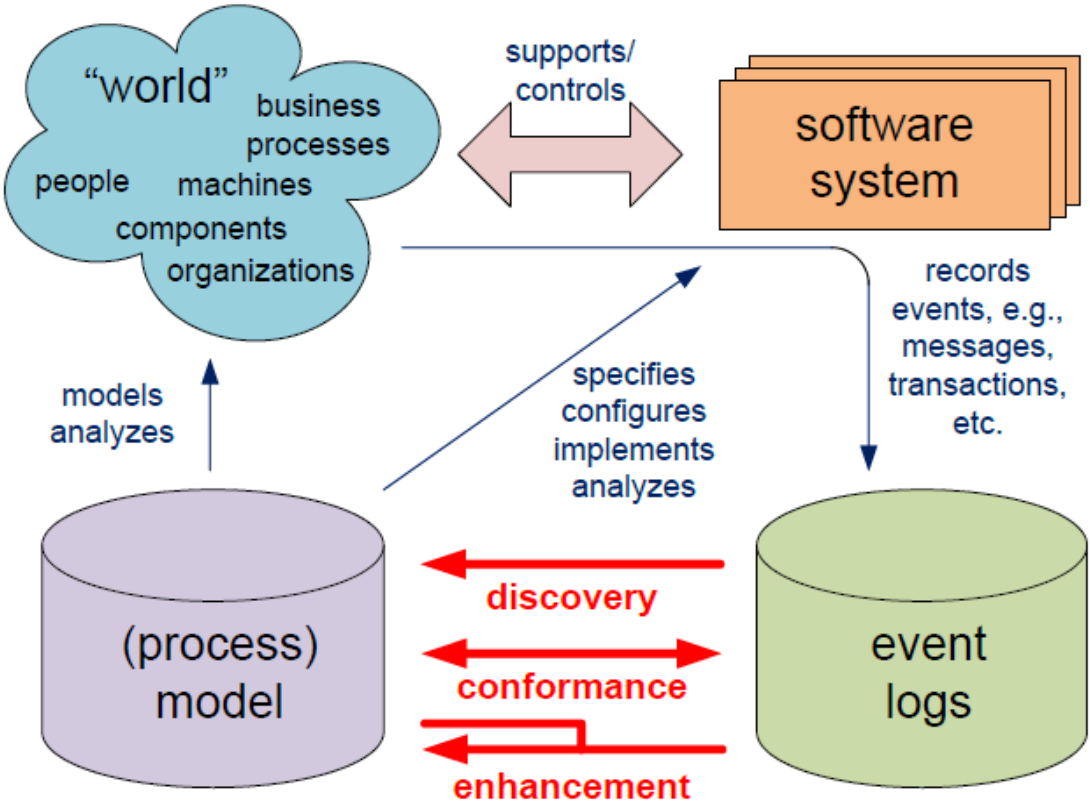


Figure 5- The three main types of Process Mining [17]

Process mining begins with acquiring an event log. Process mining aims to draw information (e.g., process models) from these logs, i.e., process mining describes a family of a-posteriori analysis techniques exploiting the information recorded in the event logs. Typically, these techniques assume that it is realizable to sequentially record actions such that each event/action refers to a move in the process and is connected to a process instance. Furthermore, some mining techniques use additional information such as the resource performing or initiating the activity, the event timestamp or other data elements such as (e.g., loan amount).

Process mining addresses the problem that most "process/system owners" have limited information about what is actually happening. In practice, there is often a significant gap between what is prescribed or supposed to happen, and what actually happens. Only a concise assessment of reality, which process mining strives to deliver, can help in verifying process models, and ultimately be used in system or process redesign efforts [17].

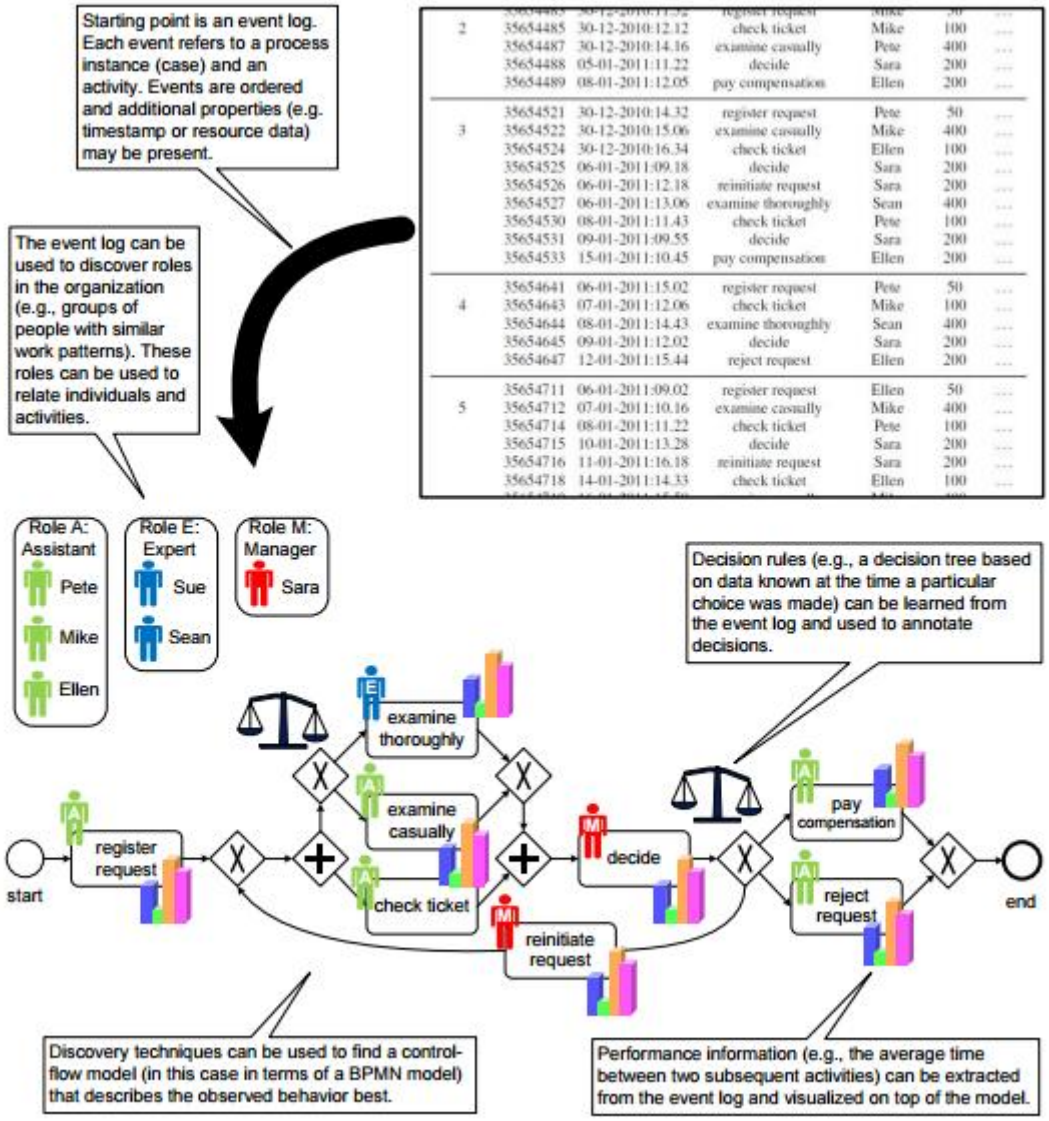


Figure 6-Overview of process mining [17]

1.2.1 Process Discovery

Process discovery takes an event log as input and produces a model without using any other apriori information. There exists dozens of approaches to extract a process model from raw event logs. In this work we use the Heuristic Miner Algorithm to discover a Petri net by modelling dependencies existing in process tasks. This section discusses the terms in process discovery. The starting point of process discovery is its input: event data.

Events are related to particular process instances, e.g., a urine test in a hospital is related to a patient being treated and a payment transaction in a sales process is related to a particular

customer order. All events related to a particular process instance (i.e., case) can be ordered. Since these events also refer to activities, we can describe each case as a trace of activity names.

For a process with activities A, B, C, D, and E, we may find the following traces: ABCE, ACBE, and ADE. For example, an event log may contain information about 238 cases following trace ABCE, 56 cases following trace ACBE, and 88 cases following trace ADE. Process discovery algorithms can transform such an event log into a process model that adequately describes the observed behavior. For this simple example it is easy to construct a process that always starts with activity A, ends with activity E, and in between either B and C occur (in any order) or just D occurs. For processes consisting of dozens or even hundreds of activities, this is much more challenging.

Figure 2 above shows a fragment of a larger event log from a claim handling process. Based on such an event log we can learn the control-flow model showing the ordering of activities. The discovered model depicted in Figure 2 is expressed in the so-called BPMN notation and illustrates that concurrency, choices, loops, and other control flow constructs can be learned from example traces. As shown, the process always starts with activity register request and ends with pay compensation or reject request. The examination and the checking of the ticket can be done concurrently. There are two types of examinations. The decision to pay or to reject is based on the examination and check activities. It is also possible that no decision can be made yet. Activity reinitiate request restarts the examination and check activities thus modelling a loop construct.

In recent years, dozens of process discovery algorithms have been proposed. They are able to extract models from a wide variety of events logs in different domains (banking, insurance, manufacturing, high-tech systems, e-government, e-learning, logistics, and healthcare).

1.2.2 Event data

An event log is defined as “an ordered trace of computer system activities that are stored to a file on the system under investigation” [17]. That data consists of both user entry information and transaction meta-information such as time stamps and user identity. Importantly, it is the

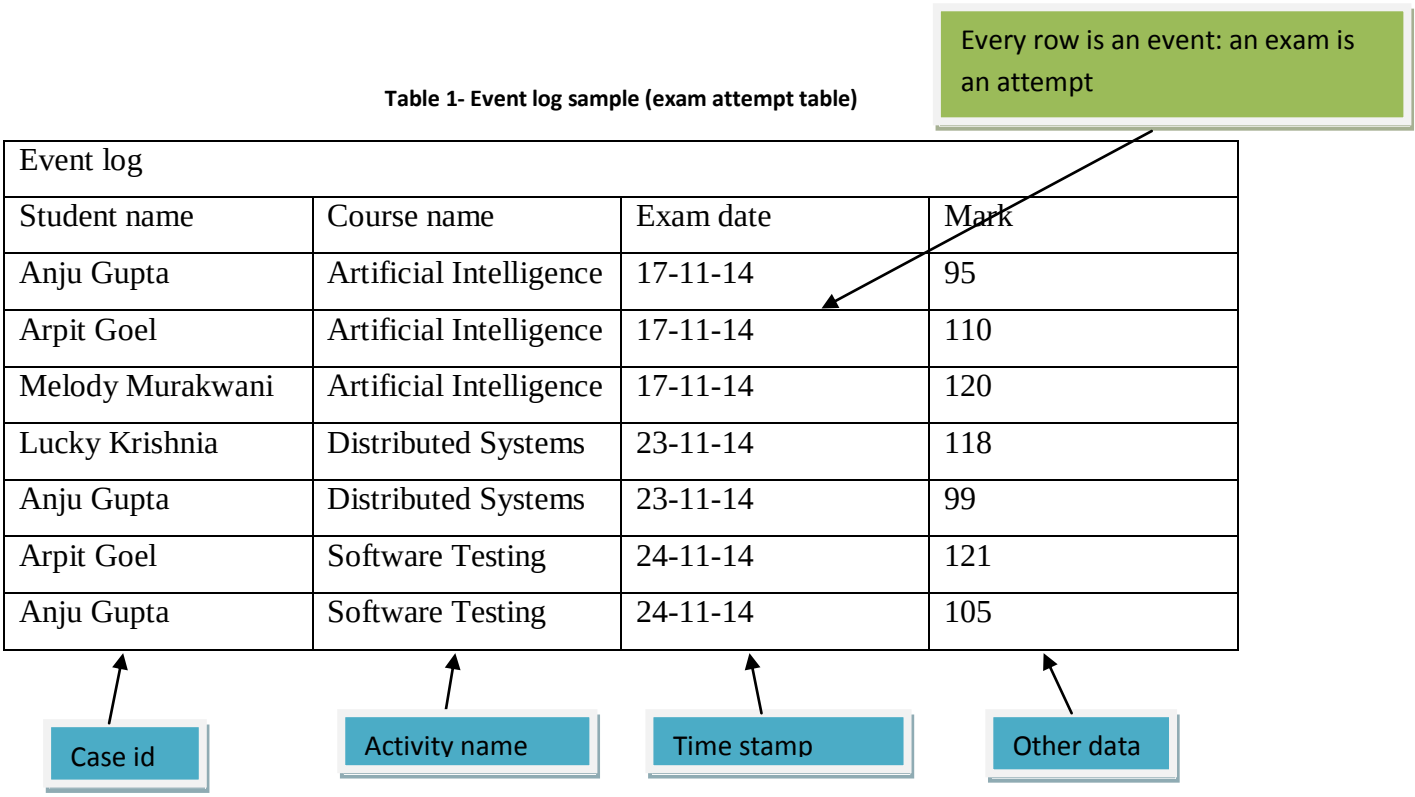
system which records meta-information, without user manipulation, and that makes event logs valuable tools for control and audit [17]. Event logs are readily available as they are stored by various computer based systems, including smart phones, corporate ERP systems, the digital Cloud, and wireless sensors.

The logs provide a basis for what is actually happening in processes. Furthermore, event log data are so wealthy that different kinds of analysis can be performed on it, resulting in many different insights into core business processes.

The events in an event log belong to a case. They are ordered and describe one "run" of the process. Whenever possible, process mining techniques employ additional information such as the resource performing the activity, the event timestamp, and other data attributes (e.g., student mark).

Table 1- Event log sample (exam attempt table)

Event log			
Student name	Course name	Exam date	Mark
Anju Gupta	Artificial Intelligence	17-11-14	95
Arpit Goel	Artificial Intelligence	17-11-14	110
Melody Murakwani	Artificial Intelligence	17-11-14	120
Lucky Krishnia	Distributed Systems	23-11-14	118
Anju Gupta	Distributed Systems	23-11-14	99
Arpit Goel	Software Testing	24-11-14	121
Anju Gupta	Software Testing	24-11-14	105



An event log shows processes, and each process consists of process instances in the form of events. Each event comprises a case id, activity name, timestamp, and other data.

The first step in process mining is to mine event logs from data pools such as transaction logs, databases, and audit trails. Process mining tools only work on specific formats which are XES (eXtensible Event Stream) and Mining eXtensible Markup Language (MXML). The IEEE Task Force on Process Mining selected XES as the standard format for logging events [12]. It is important to select relevant event data for the analysis work at hand.

1.2.3 Process

The underlying focus of process mining is a process, which is a defined set of steps executed to achieve a given objective.

Actual business processes are quite complex, with multiple interactions taking place either concurrently or with differing time lags. The actual business processes bear no resemblance to the ideal design portrayed by the process designer. Hence, the thrust of process mining is that it discovers the reality of the business process through a methodical analysis of the data stored in the event log. The accuracy of the discovered model is therefore closely related to how complete the logs are, and the researcher's access to the total log.

1.2.4 Process Discovery Notation

Let L be an event log as defined in section 4.1.1.1 of this report or as specified by the XES standard. A process discovery algorithm is a function that maps an event log E onto a model such that the process model is "indicative" for the behaviour recorded in the event log. This definition does not stipulate the kind of process model to be used for representation, e.g., a YAWL (Yet Another Workflow Language), BPMN (Business Process Modelling Language), EPC ('Event-driven process chain), or Petri net model. In fact, the notation is less relevant. The one requirement is that the behaviour of the model is a good representation of the event data [19].

1.2.5 Process Model Quality

The requirement for the model to be "representative" means that it should be possible to replay all behaviour in the log, that is, a trace in the event log is a map out of the Petri Net. This defines the "fitness" requirement. As a general practice, there is usually a trade-off between the four quality criteria below:

1. *Fitness:* the discovered model should represent the sequences in the event log.
2. *Precision:* the discovered model should not permit for behaviour completely dissimilar to what is captured in the event log.
3. *Generalization:* the discovered model should allow for a broad example of the records the event log.
4. *Simplicity:* the model should be simple to understand and analyse [19].

A model with good fitness should replay most of the process instances. Precision is related to the notion of under-fitting presented in the context of data mining. A model having a poor precision is under-fitting, i.e., it models behaviour that is unrelated to the recorded events. Generalization is related to the notion of over-fitting. An over-fitting model is not broad enough, i.e., it is too precise and strictly sticks to examples in the event log. Simplicity is related to Occam's Razor which states that "one should not increase, beyond what is necessary, i.e beyond the number of entities required to explain anything" [19]. Using this Occam's Razor principle, we attempt to find the "simplest process model" that can capture the recorded events. It is a challenging task to balance the four quality criterion. For an example, a model that is over-simplified is likely to have a lack of precision or low fitness. It is imperative then, that more often than not researchers have to settle for a trade-off between over-fitting and under-fitting.

In this work we employ the use of Petri Nets for process discovery. Our choice for using Petri nets is because they are simple and graphical and yet they still model concurrency, choices and loops. Figure 7 below illustrates a Petri Net drawn, where activities b and c are concurrent. Petri nets are as a result briefly introduced and discussed below as our discovery notation of choice.

1.2.6 Petri Nets

A Petri net is a formal, graphical, and executable technique used for the specification and analysis of concurrent, discrete-event dynamic systems [20].

A Petri net is a triple $N = (P, T, F)$where:

1. P and T are disjoint finite sets of places and transitions, respectively.
2. $F \subset (P \times T) \cup (T \times P)$ is a set of arcs (or flow relations)[10].

Where places represent possible states of the system, transitions are events or actions which cause the change of state and every arc simply connects a place with a transition or a transition with a place [20].

A change of state is denoted by a movement of token(s) (black dots) from place(s) to place(s); and is caused by the firing of a transition. The firing represents an occurrence of the event or an action taken. The firing is subject to the input conditions, denoted by token availability. A transition is enabled when there are sufficient tokens in its input places. After firing, tokens will be transferred from the input places (old state) to the output places, denoting the new state [20].

The figure below gives an example of modelling a vending machine using a Petri Net. The specifications of the vending machine are:

i. The machine dispenses two kinds of snack bars – 20c and 15c.
ii. Only two types of coins can be used which are 10c coins and 5c coins.
iii. The machine does not return any change.

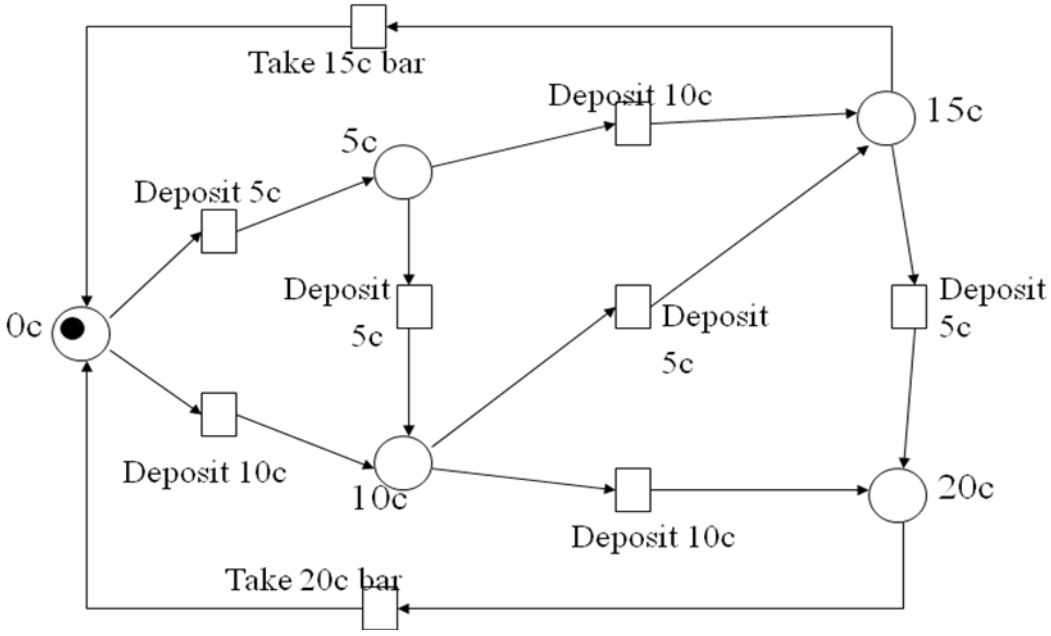


Figure 7 - Vending machine net model

1.2.7 Petri net with data ((DPN-net)

A DPN-net is a Petri net in which transitions (modelling activities) can read and write variables. A DPN-net may use a finite set of process variables V and a function U that determines the domain of each variable. A transition modelling an activity is allowed to write (or update) a predefined subset of the process variables. A transition can have a data-dependent guard that blocks when it evaluates to false. Only if the guard evaluates to true and all input places are marked, a transition can fire. A guard can be any Boolean expression over V using logical operators such as conjunction (^), negation (¬) and disjunction (v) [20].

A Petri net with data N = (P; T; F; V; U; R; W; G) consists of:

- a Petri net (P; T; F);
- a set V of variables;
- a function U that defines the values admissible for each variable v ∈ V , i.e. if U(v) = Dv, Dv is the domain of variable v;
- a read function $R \in T \to 2^V$ that labels each transition with the set of variables that it must read;
- a write function $W \in T \to 2^V$ that labels each transition with the set of variables that it must write;
- a guard function $G \in T \to \mathcal{G}_V$ that associates a guard with each transition[20].

In order to provide an operational semantics, we introduce the concept of the state of a DPN-net:

State of a Petri Net with Data

Given a DPN-net N = (P; S T; F; V;U;R;W;G) and let D = $\bigcup_{v \in V} U(v)$ the state of N is a pair (M;A) consisting of

i. A marking M for Petri net (P; T; F)

ii. A function A that associates a value with each variable, i.e $A : V \rightarrow D \cup \{\perp\}$, with $A(v) \in U(v) \cup \{\perp\}$. If a variable v is not given a value, we use the special symbol $\perp$, i.e. A(v) = $\perp$.

In the initial state, there is only one token in a so-called start place $p_0 \in P$. A process instance is considered as concluded when a token is produced in a so-called end place $p_e \in P$ [10].

The initial state is (M_0, A_0) where $M_0(p_0) = 1$, $M_0(p) = 0$ for any other place p, and for all $v \in V: A_0(v) = \perp$. In the remainder, dom(*f*) denotes the domain of some function *f*. The operational semantics can be introduced in term of valid transition ring and state transition:

1.2.7 Process Discovery Algorithm

Let L be an event log as defined in section 4.1.1.1 of this report or as specified by the eXtensible Event Stream standard. A process discovery algorithm takes an event log L and maps it onto a process model so that the model is "illustrative" of the observed behaviour in the event log [19]. In this work the target model is a Petri net model.

To illustrate process discovery we use a simple data trail D which is a set of activities.

D = [(a, e, d), (a, c, b, d) [2], (a, b, c, d) [3]]

D is a simplified log that describes a record of six cases. The aim is to mine a Petri net that "replays" our event log D as shown in figure 8 below.

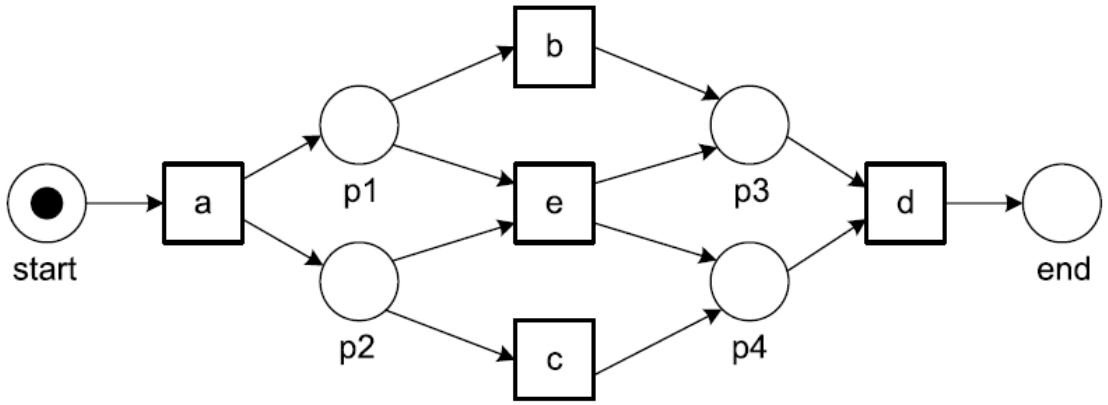


Figure 8 - a Petri Net discovered for D = [(a, e, d), (a, c, b, d) ², (a, b, c, d) ³] [19]

1.2.8 Heuristic Miner Algorithm

The HM algorithm focuses on the control flow perspective and produces a process model from a given event log. Other approaches such as the alpha algorithm i.e. an algorithm for mining event logs and producing a process model, presupposes that the mined log must be complete and there should not be any noise in the log [19].

Other process discovery algorithms do not take the frequency of ordering relations into account when inferring the advanced ordering relations to build the process model. Therefore, in this section we present an algorithm that considers the frequency of tasks when building process models: the Heuristics Miner (HM) [19].

The HM algorithm can deal with noise and can be used to express the main behaviour (i.e., not all the details and exceptions) registered in an event log. It supports the mining of all common constructs (such as sequence, loops, choice, invisible tasks, and parallelism) in process models except for duplicate tasks. The HM algorithm involves two main steps.

i. First, it builds a dependency graph. The dependency graph contains the causal dependencies that will be adhered to when building the Petri net model. In comparison to abstraction-based algorithms, the HM uses the frequencies of the basic ordering relations, to compute of the strength of the causal relations. By default, the algorithm creates one dependency to the best causal successor and predecessor of a given task.

ii. In the second step, the split/join points are set in the dependency graph. The type of each split/join is decided based on frequencies of the dependencies as noted in step 1. The decision whether there should be a choice or a parallel construct in the model is made using threshold values.

1.2.9 Conformance Checking

Process models may be descriptive (showing what really happens) or normative (defining what should happen) and can be made by hand or discovered through process mining. In all cases it is interesting to compare model and reality (as recorded in the event log). Conformance checking techniques can be used to discover discrepancies between the modelled behaviour and the observed behaviour. These techniques provide metrics for the degree of conformance and diagnostic information explaining the observed differences. Moreover, it is possible to drill down and apply process discovery techniques to the non-conforming cases.

Conformance checking can be used to judge the quality of discovered process models. However, more important, it can also be employed as an enabling technology for auditing, six sigma, and compliance checking.

For conformance checking we use the "oracle" λM that relates traces in the event log to paths in the model. In this section, we consider the standard four quality dimensions for comparing model and log: (a) fitness, (b) simplicity, (c) precision, and (d) generalization [21]. However, only fitness is discussed within the scope of this work.

Fitness: Can the Model Generate The Observed Behaviour?

A model with good fitness allows for most of the behaviour seen in the event log. A model has a perfect fitness if all traces in the log can be replayed by the model from beginning to end. To quantify fitness, we can directly apply the distance function δ on the optimal alignments provided by the "oracle" λM. The total alignment cost for event log L and model M is

$$fcost(L, M) = \sum_{\sigma_L \in L} \delta(\lambda_M(\sigma_L)) \quad [21]$$

Fitness is expressed as a value between 0 (very poor fitness) and 1 (perfect fitness). There are several ways to normalize the alignment costs. One approach is to divide *fcost(L, M)* by the maximal possible cost. Let us assume that for x ∈ AL and y ∈ AM: δ(x, y) = ∞ if x ≠ y, i.e., we only allow a move on both model and log (x, y) if x = y. In this case, the worst-case scenario is that there are just moves in the log and moves in the model (and never both) in the optimal alignment. $move_L(L) = \sum_{\sigma_L \in L} \sum_{x \in \sigma_L} \delta(x, \bot)$ is the total cost of moving through the whole log without ever moving together with the model. $move_M(M) = \min_{\sigma_M \in \beta(M)} \sum_{y \in \sigma_M} \delta(\bot, y)$ is the total cost of making moves on model only. Note that we consider the "least expensive path" because an optimal alignment will try to minimize costs. This results in the following definition of fitness:

$$fitness(L, M) = 1 - \frac{fcost(L, M)}{move_L(L) + |L| \times move_M(M)}$$ [21]

1.2.10 Model Enhancement

The third type of process mining also uses a model and an event log as input. However, now the model is improved or extended. For example, Figure 2 illustrates how a process model can be extended using timestamp information in the event log. Timestamps of causally related events can be used to measure durations between two subsequent activities. For example, analysis may show that it takes on average 21 days to make a decision after checking the ticket. This information can be used to show bottlenecks and predict remaining flow times for running cases.

If the event log contains information about resources, it is also possible to discover roles, work distribution mechanisms, and resource characteristics. Additional event and case attributes can also be used to learn decision rules explaining the choices made in the process. For example, one may learn that cases that are thoroughly checked by Sue tend to be rejected.

1.2.11 Common Questions that are answered by Process Mining

1. What is the most frequent path for every process model?
2. How is the distribution of all cases over the different paths through the process?

3. How compliant are the cases (i.e. process instances) with the deployed process models? Where are the problems? How frequent is the non-compliance?
4. What are the routing probabilities for each split task (XOR or OR split/join points)?
5. What is the average/minimum/maximum throughput time of cases?
6. Which paths take too much time on average? How many cases follow these routings? What are the critical sub-paths for these paths?
7. What is the average service time for each task?
8. How much time was spent between any two tasks in the process model?
9. How are the cases actually being executed?
10. What are the business rules in the process model?
11. Are the rules indeed being obeyed?
12. How many people are involved in a case?
13. What is the communication structure and dependencies among people?
14. How many transfers happen from one role to another role?
15. Who are important people in the communication flow? (the most frequent flow)
16. Who subcontract work to whom?
17. Who work on the same tasks?
18. What are the most frequent deviations?
19. Why do these deviations take place?
20. Can we predict a deviation?
21. Is the model or the event log wrong?
22. Using time taken between activities measure where bottlenecks are experienced
23. Which roles are performing which activities?

Chapter Two: Literature Review

In this chapter we conduct a literature survey on the existing literature about the different ways to capture the data-flow perspective of a process for data verification proposed by researchers in the past few years. We follow the survey up with the problem definition as well as the aim for this study.

2.1 Related Work

The literature offers different ways to model the dataflow perspective of a process, but: a) most of them employ a data modelling approach such as ERDs as opposed to our process data perspective or b) not all approaches refer on discovering data model from event logs. This section reviews all previous research that approached the process data perspective.

The most research done in business process modelling area focused on the control flow perspective. It analyses the sequence of the tasks: which task must be executed and when. A task represents a unit of work which may be executed for many cases, for example “make payment”. Tasks are executed by resources with certain roles and organized in groups. The resources are actors able to execute activities and they can be human beings or not.

[22] Proposes a data-centric approach in order to find deadlocks from a business process. They consider the business process being modelled as a Petri Net. The reachability graph of the Petri Net helps to detect the deadlocks in the control flow. One of the causes of deadlocks is using inappropriate guards. Adding a guard or replacing a guard by a more restrictive one may remove deadlocks. But this approach does not offer a proper model of data involved in a workflow execution; it only provides another method to verify deadlocks from a workflow using the data from the workflow.

The data-flow perspective was almost never considered in the research done in this dimension of process analysis. In [10] there were identified a series of potential data validation problems that can occur in data-flow view of a process: redundant data, lost data, missing data, mismatched data, inconsistent data, misdirected data and insufficient data. Moreover three data flow implementation models were defined: explicit data flow, implicit data flow through control flow and implicit data flow through process data store. In the first

model, the data flow transitions are defined as part of the workflow model. Basically the data flow transitions model the data changes from one activity to another. The second model calls the control flow to cross data from one activity to another. In the last model all the inputs and outputs of the activities are recorded in the process data store.

Our review of related work focuses on the approaches directly relevant to this paper, particularly those in which both the control-flow and data-flow are considered for analysis. The significance of data-flow verification in process workflows was introduced in [23], [24]. [23], identifies a number of possible errors in the data-flow, such as the missing and redundant data error, but the means to check these errors is not provided. Afterward, [25] gave a technique to check the errors identified in [23] using UML diagrams, and gave supporting testing algorithms. None of these approaches consider control-flow properties.

In [26], a model that uses dual workflow nets is given, their model could capture describe the control-flow and the data-flow. However, no explicit data correctness properties are considered. In [27], model checking is used to verify business workflows, tackling both the control-flow and data-flow perspective using UML diagrams as opposed to the Petri net modelling method taken in this paper. [27], gives some data correctness attributes without providing a systematic classification. [28], presented an approach to detect data errors in workflow using a systematic graph traversal approach. The work closest to this work is in [29], and [20]. In [29], Nikola Trocka et al presented an analysis approach that uses so-called “anti-patterns” expressed in terms of a temporal logic to discover data-flow errors in workflows. In [20], Massimiliano de Leoni et al use recent advances in conformance checking using alignments, and they also employed the use of Petri Nets with Data.

Model checking is a known successful method used to discover program bugs during software verification [30]. In this light, we take this successful approach and apply it in the field of data verification and data quality.

2.2 Problem definition

Today data is being integrated from heterogeneous sources and used for the data analytics vary depending on purpose. The varied uses should dictate the required level of standardization and quality required, because in big data, quality is contextual unlike in

traditional warehouses. The scope of this work ties the problems in data quality to 2 factors which are:

- Quality in big data in contextual
- Quality is expensive

These factors are discussed at length below.

2.2.1 Quality is contextual

The over-arching principle of data quality in a data warehouse is a flat earth model. There is one data model and one set of integrated data. Therefore there is one set of rules for the cleaning and standardizing of data before it is loaded. These rules ensure that the data is clean enough for the most stringent use case. This is good for the transactions that data warehouses were designed to manage. You don't want financial data with material impacts to be incorrect because it's expensive or slow to clean and load.

The problem with this approach is that it doesn't factor in other uses. Quality is a measure of the fitness for a particular use. The data may not be good enough to provide public financial forecasts, but it is good enough to estimate how much inventory to have on hand to avoid stores running out of stock. Transaction data, being core to the business, has the strictest requirements. The data warehouse has its origins managing this type of data, and the associated data requirements.

Big data applications, in contrast, use interaction and transaction data both, and use that data for different purposes. These purposes are most often directional in nature, rather than providing an absolute answer to a question like "what was the profit on this product line last quarter?" At other times, they may be very specific to the action to be taken for a specific customer; hence we require data management solutions that acknowledge these varied uses and incorporate them to dictate the required level of standardization and quality required.

2.2.2 Quality is expensive

The interaction data that is created by people is usually text rather than discrete values in a form. Much of this data is noisy and contains gaps, however analyzing the patterns over periods of time or in large collections of events provides useful information. This is a form of data extraction, involving analytic techniques rather than queries or the data processing

techniques used for warehouse extraction and loading. Analytics require different cleaning, standardizing and formatting than the data in data warehouses; you are preparing data for machine consumption rather than human-readable reports. The type of data cleansing required is heavily dependent on the analytic techniques to be applied. This makes it hard to anticipate in advance exactly how to clean, standardize and store the data.

Cleansing the data ahead of time means it will be optimal for some analytics, but not others, and it still will not be in a human-usable format for querying. Because of these restrictions, the event streams and interaction data are often better left in a raw or lightly processed form. To do otherwise would incur significant costs given the complexity and volume of data.

2.3 Aim of the work

Quality is a measure of fitness of purpose, and purpose is significantly tied to the underlying process, hence when processes become repeatable and reusable, the data can then be guarded from dirt on load rather than on use.

In this light, the aim of this work is to answer the question:

- How do you isolate the source of the introduction of data flaws to fix the process instead of correcting the data?
- In short, how can we detect errors early in the life cycle of a system so as to produce quality data, and reduce data cleaning costs?

Data-driven fault detection is based on historical observations of process data. Data mining techniques either cover only a few sections of the processes or elicit abstract patterns in terms of rules or decision trees. It is not possible to infer global optimal functioning of an entire process by summing up inspection results of many local sections. Process mining combines data mining and process modelling to generate models from IT log data. Process mining is a young research area which has demonstrated capabilities to overcome the difficulties arising with large amounts of unstructured data and events produced by heterogeneous resources associated with information systems at all levels.

Hidden patterns are discoverable in the immense amount of data information systems are producing at an unprecedented rate. Inferences are done based on historical process

information. This might include time, energy, workload, social network, dependency of activities, resources, states of components, embedded rules, flow of activities, etc. Based on the discovered information, process models can be improved in real time. Comparison of real data against initial process models in conformance checking may lead to conclusions regarding possible faults and failures in the systems analysed, and in some cases leads to enactment of further processes.

2.4 Motivation of the work

The greater chunk of process mining research focuses on control-flow while ignoring data movements within the process. The motivation for this thesis is to offer more insights about the data perspective of process models to yield the following aspects:

Firstly, the data-flow perspective may bring improvements to the process (e.g. if the data-flow model shows us an operation with low frequency we may verify if that operation is really executed a few times or it represents a deviation of the intended behaviour of the process). Secondly, assuming we are executing a process and we are in a particular state of its execution we can identify based on the data known until the current state of the process which operations we can perform further. Moreover, a data-flow model of the process shows us what additional data we need in order to execute a particular operation. Usually, researchers considered the data-flow already created by experts or pulled out by analyzing the semantics.

Process mining addresses the problem that most process/system owners have limited information about what is actually happening. In practice, there is often a significant gap between what is prescribed or supposed to happen, and what actually happens, that is how error is introduced during system use. Only a concise assessment of reality, which data-aware process mining strives to deliver, can help in verifying process models, and ultimately be used in system or process redesign efforts.

Chapter Three: Proposed Methodology

This chapter describes our method for detecting data errors in process flow. We briefly introduce ProM, our tool for analysis and the ProM digestible event log format.

All analysis in this paper is performed using existing and dedicated plug-ins within the open-source process mining toolkit ProM. The starting point for the process mining activities is getting event data in the required MXML or XES format.

3.1 The ProM Framework

The ProM framework is a flexible framework in which different algorithms for each of the process mining perspectives can be plugged in. Figure 9 below shows an overview of the ProM framework [31]. It explains the relations between the framework, the process log format, and the plug-ins. As Figure 9 shows, the ProM framework can read files in the XES format through the Log filter component. This component is able to deal with large data sets and sorts the events within a case on their timestamps before the actual mining starts. In the remainder of this section, we describe both the process log format and the plug-ins.

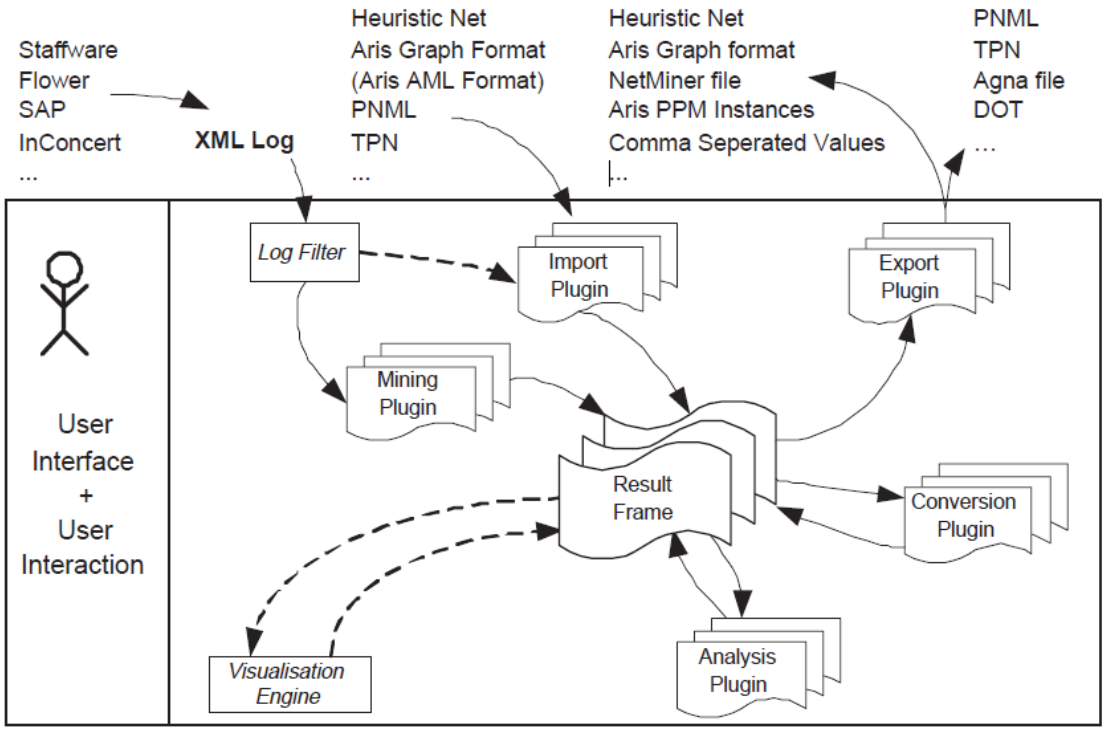


Figure 9-an overview of the ProM framework [31]

3.1.1 Process Log Format (XES)

Since each information system has its own format for storing log files, a generic XES format has been developed for the ProM framework to store a log in. XES (eXtensible Event Stream) was selected by the IEEE Task Force on Process Mining as the standard format for logging events. This format was based on a thorough comparison of the input needs of various existing (ad-hoc) process mining tools and the information typically contained in an audit trail or transaction log of some complex information system (e.g., an ERP or a WFM system).

The root element is a process log element. The process log element contains a ProcessInstance which is an instance of the process, i.e., a case, an activity,an eventtype (Eventtype), a timestamp (Timestamp), and a person that executed the activity (Resource).

3.1.2 Plug-ins

The framework allows plug-ins to operate on each others' results in a standardized way. Through the Import plug-ins a wide variety of models can be loaded ranging from a Petri net to logical formulas. The Mining plug-ins do the actual mining and the result is stored in memory, and in a window on the ProM desktop. Typically, the mining results contain some kind of visualization, e.g., displaying a Social network, a Petri net, an Event-Process driven Chain or further analysis or conversion. The Analysis plug-ins take a mining result an analyze it, e.g., calculating a place invariant for a resulting Petri net. The Conversion plug-ins takes a mining result and transforms it into another format, e.g., transforming an Event-Process driven Chain into a Petri net.

3.2 Proposed Approach

Our approach aims to use Petri nets with data (DPN-net) for modelling workflows, followed by conformance checking using compliance rules and alignments. Non conformance from a data perspective could assist us to observe deviations in consistency or redundancy in data to mention a few possible data errors. The methodology involves the following steps:

1. Filter the log by removing redundancies to simplify and improve analysis.
2. Discover the process control-flow from the log, using the Heuristic Miner algorithm.
3. Define data-aware and resource-aware compliance rules

4. Then we align the event log and control-flow, enriched with diagnostics from the previous step. Once the alignment is computed, then,
5. Data-flow perspective can be discovered, i.e., the read and write operations in each activity. This is modelled in a Petri net with data. The Petri net with data is used to check if the data related compliance rules are being adhered to.
6. Perform analytics on the data-flow perspective to:
   a. Identify where data/resource compliance rules are broken,
   b. Identify process fragments where most deviations occur,
   c. Identify roles in the system and the subsequent activities, with the data produced at that activity.
   d. Identify where bottlenecks are prevalent in the system, as this may cause data corruption through buffer overflows.

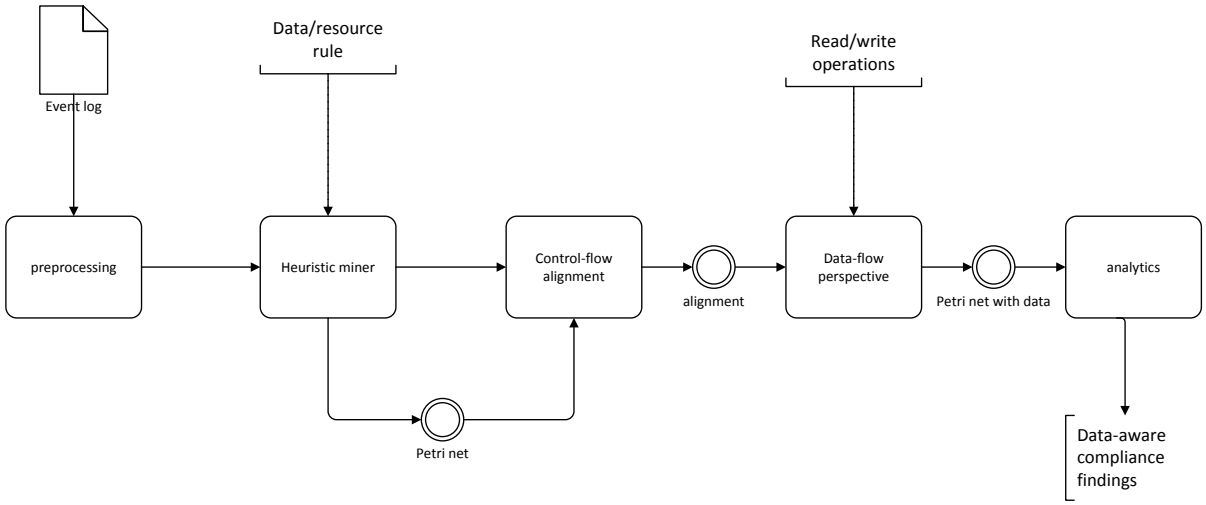


Figure 10- Our diagnostic data-aware fault detection methodology

Chapter Four: Implementation

This chapter discusses the implementation of the proposed methodology in detail. The implementation work is done on a real life event log.

The work done in this research begins with acquiring data that is in ProM framework digestible format, which is XES or MXML format. Upon data acquisition, we performed a manual analysis to understand the data to assist in our understanding of the problem scenario and the process represented by the data.

4.1 Dataset

4.1.2 Data Source

This work is performed on both synthetic and real life event logs in XES format obtained from 3TU.Datacentrum. 3TU.Datacentrum is an educational repository for archiving scientific data at the University of Twente, Netherlands. The BPI Challenge 2012 dataset was sourced at [33].

Table 2 - Metadata for event log [33]

Metadata for event log	
Doi	10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f
name	BPI Challenge 2012.xes.gz
Description	Event log of a loan application process
Language	Dutch
log type	Real-life
process-type	Explicitly structured
source institute	Eindhoven University of Technology
rights type	Public
number of traces	13087
number of events	262200

The real life event log is a process log provided for the Business Process Intelligence Challenge 2012[33]. The log reflects a loan application process in a global financing organisation. The process starts with a customer applying for a loan, followed by some automatic checks and processing of the application by the resources working in the financial organisation. Examples of process steps include the submission of the application, declining, cancelling or accepting the application, etc.

The data reflects process events for 13,087 loan applications over a six month period starting in October 2011 and ending in March 2012. In the event log is a total of 262,200 events in 13,087 cases, that start with a client submitting an application and ends with eventual conclusion for that application which can be an Approval, Rejection (Declined) or Cancellation. Each case contains a single case level attribute, AMOUNT_REQ, which indicates the amount requested by the applicant. For each event, the extract shows the type of event, life cycle stage (Schedule, Start, and Complete), a resource indicator and the time of event completion.

The events describe steps along the approvals process and fall into three categories which are:-

1) "A_" (Application Events) - Refers to the different states of application. The customer initiates an application. The bank does the necessary checks and completes the application through a decision.

2) "O_" (Offer Events) - Refers to states of an offer communicated to the customer

3) "W_" (Work item events) - Refers to states of work items (manual) that occur during the approval process as executed by bank resources.

Table 3 below shows the event types and our understanding of what the events mean.

Table 3 - Names and Descriptions of Events

Event	Event Description
A_SUBMITTED/A_PARTLY SUBMITTED	Initial application submission
A_PREACCEPTED	Application received but requires additional information
A_ACCEPTED	Application accepted and pending screening for completeness
A_FINALIZED	Application has passed screen for completeness
A_APPROVED/A_REGISTERED/A_ACTIVATED	End state of successful or approved applications
A_CANCELLED/A_DECLINED	End state of unsuccessful applications
O_SELECTED	Application selected to receive offer
O_PREPARED/O_SENT	Offer prepared and transmitted to applicant
O_SENT BACK	Offer response received from applicant
O_ACCEPTED	End state of successful offer
O_CANCELLED/O_DECLINED	End state of unsuccessful offers
W_Afhandelen leads	Following up on incomplete initial submissions
W_Completeren aanvraag	Completing pre-accepted applications
W_Nabellen offertes	Follow up after transmitting offers to qualified applicants
W_Valideren aanvraag	Assessing the application
W_Nabellen incomplete dossiers	Seeking additional information during assessment phase
W_Beoordelen fraude	Investigating suspect fraud cases
W_Wijzigen contractgegevens	Modifying approved contracts

Work items that take place within the approvals process (denoted by "W_" in the event log) are themselves associated with three transitions each of which occur at distinct stages of the item's life cycle (Table 4 below)

Table 4- Description of Transitions in the Work Item Life Cycle

Transition	Description
SCHEDULE	Indicates a work item has been scheduled to occur in future
START	Indicates the opening / commencement of a work item
COMPLETE	Indicates the closing / conclusion of a work item

Figure 11 below shows a dashboard overview of the event log in ProM 6.4. The dashboard overview highlights a summary of important overall statistics such as the number of cases, events, resources, event types in the process among others.

Figure 11- A dashboard overview of the log

4.3 Discovering the control-flow perspective with Heuristic Miner Algorithm

The control-flow perspective of a process summarizes the sequence followed by most/all cases in the log. Figure 12 below shows us a screen shot of the control flow discovery plug-in used for this step.

Figure 12- Discovering the control flow perspective

Hence, in this phase a process model is observed from our event log. This process model is represented as a Petri net. A Petri net consists of places represented using circles with a start and an end point and transitions represented using rectangles .Transitions may be connected to places and places may be connected to transitions, as shown in the Petri net for our data set in Figure 13 below

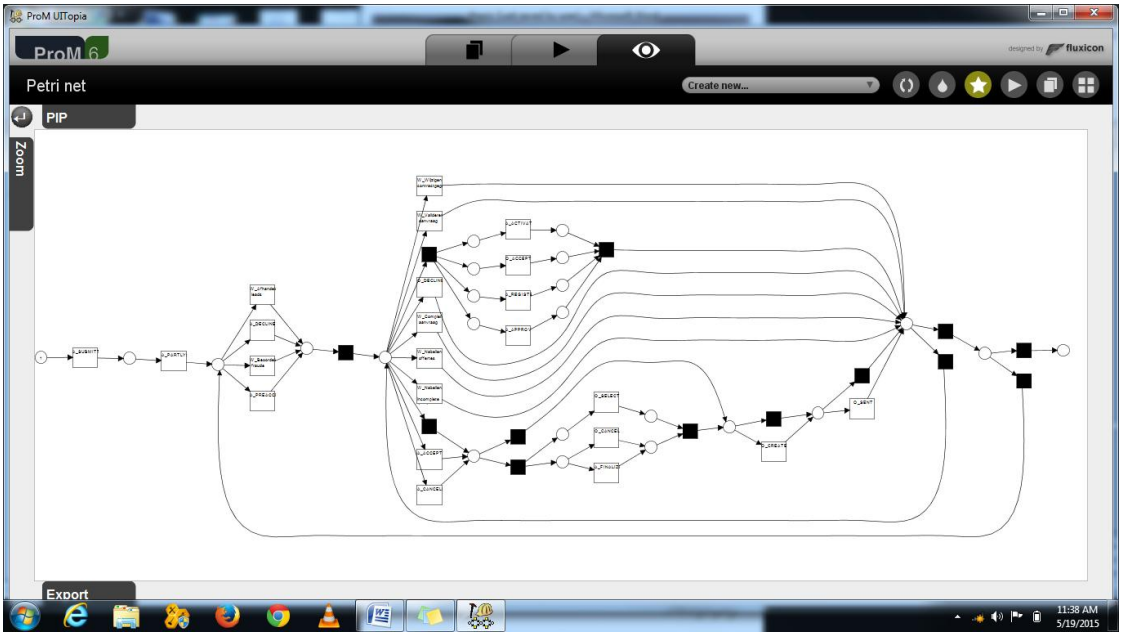


Figure 13- Petri net observed from event log

4.4 Compliance requirements

Compliance requirements are employed to ensure that all necessary data governance requirements are met without making the system vulnerable through unnecessary duplication of activities and effort from resources. They control one or several process perspectives such as the data flow, process time, control flow, or organizational aspects. These restrictions can be applied on both individual cases and groups of cases; they can impose process execution flow [32].

A resource-aware compliance requirement for our scenario might state that, "Only a supervisor can approve/deny a loan".

4.4.1 Data-aware and Resource-aware compliance rules

Elham Ramezani et al in their work [32] compiled a list of compliance rules, also adopted for conformance checking. Below is a collection of compliance rules employed in this work from the compilation in [32].

Table 5 - Data-aware and resource-aware compliance rules collection [32]

Compliance requirement	Adopted example
Four-eye principle: A security principle that segregates privileges and associates the execution of critical tasks to groups of users.	The person dealing with offer processing of a loan should not be the one who approves it.
Authorization (Access control): A security principle that ensures only authorised individuals perform certain activities or access certain data objects.	Only a 2nd level Administrator or supervisor can approve a loan.
Two (three)-way match: An accounting rule that requires the value of two different related data objects to match.	All customer invoices for purchases should be matched with respective purchase order lines
Activity L may only be executed if attribute P has the value greater than or equal to v	An application must not be approved for processing in case risk is high.

4.4 Conformance Checking (Aligning event logs and process models)

In this step we align the event log and control-flow process model, i.e., events in the log need to be related to transition executions.

Figure 14 below is screenshot of the conformance analysis plug-in employed for this step. Then next, we discuss how the concept of alignment can be extended to incorporate data. First, we introduce the basic alignment concept.

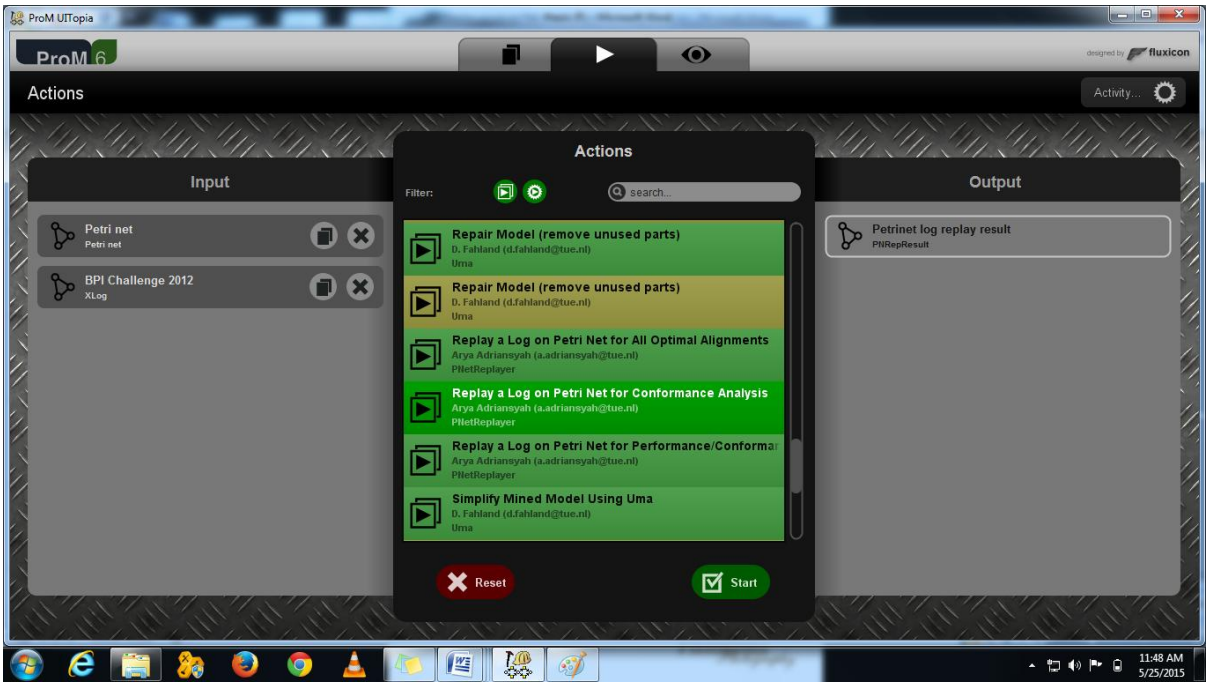


Figure 14 - Replay a log on Petri Net for conformance analysis screenshot

4.4.1 Basic Alignment concept

A Petri net P = *(T, P, F)* that describes a process relies on constructs such as parallel split nodes, synchronization nodes, decision/choice nodes, conditions, merge nodes, etc. Although we did not formalize this, we assume Petri nets with a defined initial and final state and consider all traces D ⊆ T (i.e., firing sequences) that start in the initial marking and end in the final marking. D ⊆ T fully describes the behaviour of the process model, i.e., $\sigma_M \in \mathcal{D}$ is a complete trace. An event log contains events associated with cases, i.e., process instances. Hence, a case can be described in terms of a trace σ_L, i.e., a sequence of events. Each event describes a log execution step and can be represented by a pair (a, ϕ) consisting of an

activity a pair and a value assignment ϕ. Here we assume that activities directly correspond to transitions, i.e., $a \in T$. However, this can be relaxed if needed, e.g., multiple transitions referring to the same activity or activities that are described by multiple transitions, e.g., to denote the start and completion of the activity. $\phi \in V \nrightarrow U$ is a function that assigns a value to some of the variables in V. $\Phi = V \nrightarrow U$ is the set of all such functions. An event log is a multi-set of steps where each trace consists of events of the form (a; -). In other words, $\mathcal{L} \in \mathcal{B}((T \times \Phi)^*)$. A no move is denoted by $\gg$.

Control-flow Alignment

Let us denote $S_L = (T \times \Phi) \cup \{\gg\}$ and $S_M = T \cup \{\gg\}$. A pair is $(s_L, s_M) \in (S_L \times S_M) \setminus \{(\gg, \gg)\}$

- a move in log $s_L \in (T \times \Phi)\ and\ s_M = \gg$,
- a move in model $if\ s_L = \gg\ and\ s_M \in T$,
- a move in both if $if\ s_L = (a_L, \phi) \in (T \times \Phi),\ s_M \in T$, and $a_L = a_M$.

$\Sigma = (S_L \times S_M) \setminus \{(\gg, \gg)\}$ is the set of the legal moves.

Alignment of a log trace $\sigma_L \in (T \times \Phi)$ and a model trace $\sigma_M \in T^*$ is a sequence $\gamma \in \Sigma^*$. If γ an alignment of log trace σ_L and model trace σ_M, and if $\sigma_M \in \mathcal{D}$, it is called a complete control-flow alignment of σ_L and $\mathcal{D}$. An alignment of the process model D and the event log L is a multi-set $\mathcal{A} \in \mathcal{B}(\Sigma^*)$ of alignments such that, for each log trace σ_L, there exists an alignment $\gamma \in \mathcal{A}$ of σ_L and D. A is a multi-set because an event log may contain the same log trace σ_L multiple times, potentially resulting in multiple identical alignments. In order to measure the impact of outliers, end users need to configure a cost function on the legal moves $\kappa \in \Sigma \rightarrow \mathbb{R}_0^+$. It may be defined differently for individual processes, since, generally speaking, the costs depend on the specific characteristics of the process [20].

4.4.2 Conformance Analysis Result

The conformance analysis plug-in is employed for this purpose, the plug-in takes two inputs:- the discovered model which is a Petri Net in this work and the event log and returns a Petri Net showing statistics of

i. synchronous moves between the log and the model,
ii. moves on the model only
iii. moves that appear in the log only
iv. frequency of moves in particular events
v. frequency of moves between particular events

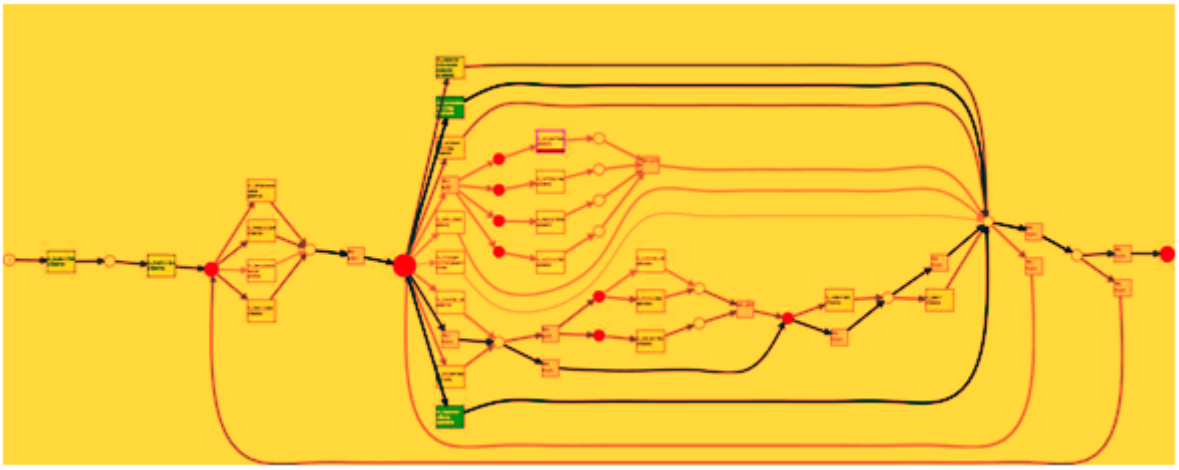


Figure 15- Conformance analysis result for log and Petri Net

4.5 Discovering the data-flow perspective

The data flow perspective takes a Petri net (P, T, F) and applies read/write variables captured at each activity using the Data-flow Discovery plug-in in ProM. The plug-in takes two objects as input: a Petri net, the read/write variables captured by each activity. It returns a Petri net with data where the read and write operations show the data flowing through the process. The Data-flow Discovery plug-in is captured in the figure below

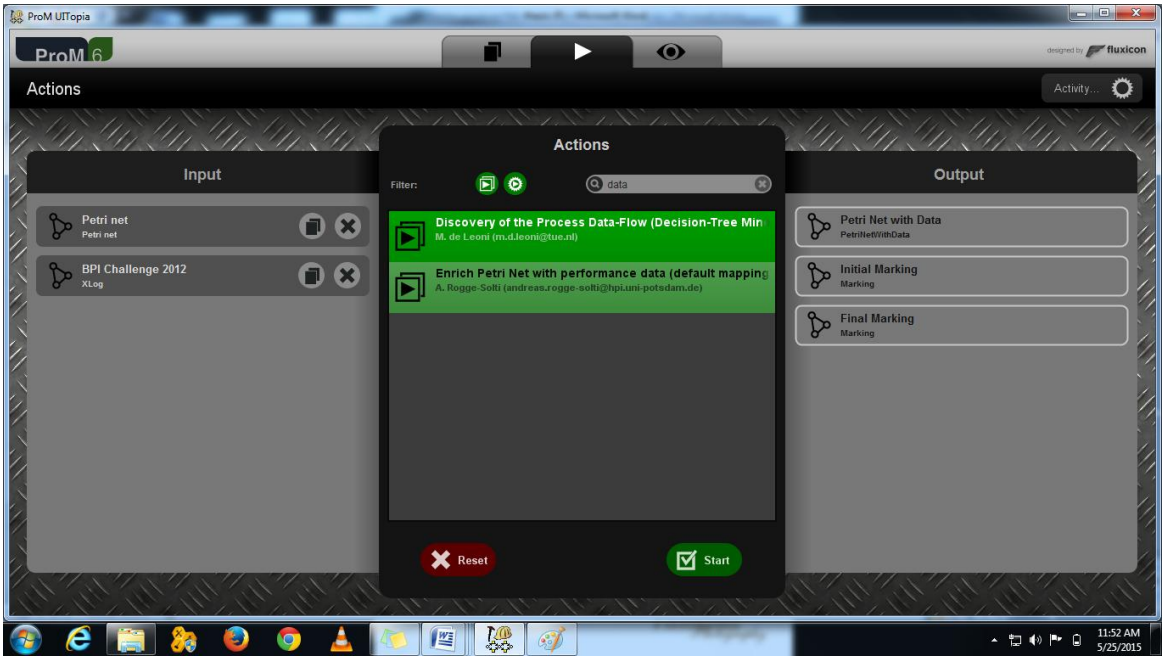


Figure 16 - Discovery of the Process Data-Flow

The outcome is a Petri net with data N = (P; T; F; V; U; R; W; G), where our technique mines V; U; R; W and G. In the remainder, we say an event $(t, \phi) \in \mathcal{L}$, if there exists a trace $\sigma \in \mathcal{L}$ such that $(t, \phi) \in \sigma$.We reasonably assume the set of variables of N are the set of variables defined in the event logs. For each event we assumed write and read operations given the data objects captured by the system.

We will start by defining the variables through detailing the data objects captured by the system at various events,

4.5.1 Defining variables captured by the system

To effectively derive variables captured in the system, we first analyse the overall communication steps in the process. A summary of events begins with a customer applying for a loan through a clerk. The initial step is to open the credit request by providing the information about the requester, i.e. the loan applicant, and the amount. Afterwards, the credit institute verifies the validity of the information provided. If the provided information is not valid, then the application is immediately rejected and stored in system along with informing the applicant. If the information is valid, a loan opening is made, hence a positive decision. If the decision is negative, the applicant is informed about the decision. The data objects extracted from the description of events is noted below.

4.5.1.1 Definition of variables

Applicant {
Applicant id
Personal details
}
Loan Application {
Application_id
Applicant_id
Amount
Status
Decision
}
Offer {
Offer_id
Applicant_id
Status
Valide_Until
}

The scope of this work is restricted to showing error as it is introduced into the process. The variables indicated in Table 6 are actively read or written to within the scope of this paper.

Table 6 - Definition of variables

Variable	***Type***
Amount	Non-negative number
Status	Boolean
valid_Until	Date
Decision	Boolean

4.5.1.2 Read/Write operations

Table 7 below captures read/write operations at various Application and Offer events.

Table 7 - Read/write operations

Transition	Variable Read	Variables Written
A_ACCEPTED	amount, valid_Until	Status
A_APPROVED	valid_Until, status, amount	Decision
A_CANCELLED	valid_Until, status, amount	Decision
A_DECLINED	valid_Until, status, amount	Decision
O_CANCELLED	status, amount	Decision
O_ACCEPTED	Amount	valid_Until, status

4.5.2 Petri Net with Data

Figure 17 below shows a screenshot of the Data-Flow Discovery plug-in in ProM. The output of the plug-in is a Petri net with data. The white and black rectangles identify the visible and invisible transitions. The yellow \rounded rectangles" represent the variables defined in the process data-flow. The dotted arrows going in and out the yellow rectangles describe the write and read operations, respectively.

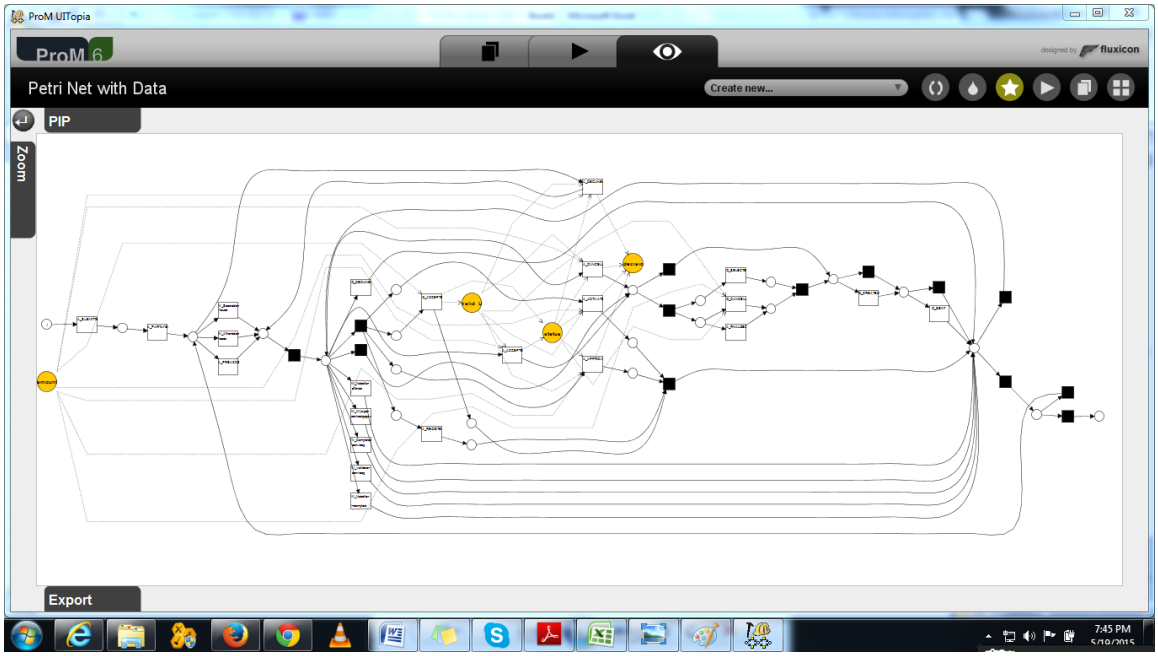


Figure 17- Petri net with Data

Experimental analysis begins now that the data-flow model has been discovered. In the next chapter we evaluate and analyse our Experimental results to show whether the defined

conformance rules defined are followed or not. Using conformance checking to discover, trace and clarify deviations, and to quantify the impact of these deviations, we will show how these deviations have a ripple effect on data error.

Chapter Five: Experimental Results and Analysis

This chapter details our analysis of the results observed from the Implementation phase. The analysis is two-fold, analysing the resource and data-flow perspective.

The focus of the work is on the resource and data-flow perspective. However, a brief analysis of the control-flow perspective is given to cement our illustration of the existence of deviations in data and how the process is very central in this endeavour.

5.1.1 Observations

1. Using figure 19 below, 6.8% of the activities are performed by an unknown resource.
2. Resource named 112 is an automated resource because
   i. Only resource 112 performs initial activities such as A_SUBMITTED and A_PARTLYSUBMITTED in all 13,087 cases.
   ii. And this resource does not handle work or offer (manual) activities outside of initiating application activities.
3. Resources named 10138, 10609, 10809, 10972 and 10629 seem as though they are super user resources, since they are active in approval and/or cancellation of applications.
4. Global statistics that indicate the longest case and the shortest case in the log calculated in terms of number of activities performed and total time taken to complete the case.

Table 8 - Longest and shortest case statistics

case id	events	start time	end time	Duration
173688	26	1/10/2011 4:08	13/10/2011 14:07:37	12 days 9hours
173691	39	1/10/2011 11:38	10/10/2011 17:47	9 days 6hours
173694	59	1/10/2011 11:40	15/02/2012 16:59:26	137 days 5hours

5.2 Analysing the resource perspective

This involves the analysis of the resource perspective against the compliance rules defined in the previous chapter. The event log contains 69 resources. 68 resources have a resource id while the 69th resource is an unknown resource.

Table 9 below shows an extract from applying the Social Miner plug-in on our data to extract information relating to individual resources and their tasks.

Table 9 - An extract from the Social network miner result

Resource	A_ACCEPTED	A_ACTIVATED	A_APPROVED	A_CANCELLED	A_DECLINED	A_FINALIZED
10124	0	0	0	0	0	0
10125	0	0	0	0	0	0
10138	8	681	681	5	156	8
10188	0	0	0	0	56	0
10228	5	0	0	2	48	3
10609	11	335	335	5	206	11
10629	28	359	359	1	119	27
10779	2	2	2	11	3	2
10789	0	0	0	2	0	0
10809	6	271	271	1	87	6
10821	0	0	0	0	0	0
10859	39	0	0	8	52	39
10861	198	0	0	85	137	200
10862	51	0	0	26	35	50
10863	87	0	0	28	82	84
10880	79	0	0	14	72	77
10881	85	0	0	60	114	84
10889	99	0	0	33	112	101
10899	0	0	0	9	0	0
10909	165	0	0	76	116	159
10910	164	0	0	23	244	164
10912	122	0	0	24	52	121

10913	166	0	0	82	155	167
10914	10	0	0	8	4	10

5.2.2 Resource-aware compliance checking

Result and observation - Rule 1: Four-eye principle: *A security principle that segregates privileges and associates the execution of critical tasks to groups of users.*

Resource 10138, 10609, 10809, 10972 and 10629 are violating the four-eye principle, i.e these resources are involved in almost all loan handling activities from request to approval in the process. Figure 19 below gives an overview of resource against activity with some of the resources violating the Four-eye principle highlighted.

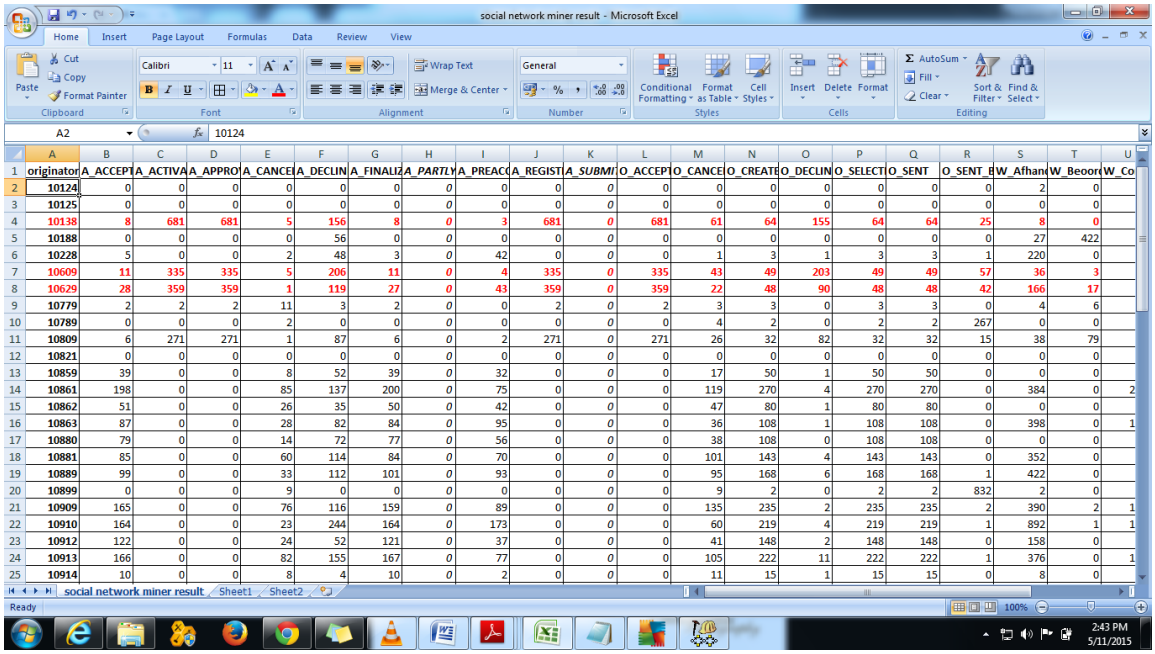


Figure 18 - Resource/activity overview

Analysis – *Rule 1: Violation of the Four-eye principle*

This is a strong indication that there is no strict separation of roles and consistent adherence to the four-eye principle to ensure compliance. The four-eye principle is a core perspective of Information governance. Information Governance is a key internal control measures to mitigate introduction of error.

Result and Observation: ***Rule 2- Authorization (Access control):*** *A security principle that ensures only authorised individuals perform certain activities or access certain data objects.*

Observation 1: Resource 112 violates Rule 2 as it is an automated response or a system but it has approved 3 loans, as shown by Table 10 below

Table 10 - Resource against frequency of their decision on applications

Resource	A_APPROVED	A_CANCELLED	A_DECLINED
10138	681	5	156
10609	335	5	206
10629	359	1	119
10809	271	1	87
10972	518	3	106
112	**3**	1004	3429
11289	68	3	55

Observation 2: Missing resource information. An unknown resource performs several work related items in a number of cases. Case 173688 has been pulled out for illustrative purposes in the table below.

Table 11 - Case 173688 shows missing resource information

caseID	Task	resource	Eventtype	Timestamp	AMOUNT_REQ
173688	A_SUBMITTED	112	Complete	1/10/2011 4:08	20000
173688	A_PARTLYSUBMITTED	112	Complete	1/10/2011 4:08	20000
173688	A_PREACCEPTED	112	Complete	1/10/2011 4:09	20000
173688	W_Completeren aanvraag	112	Schedule	1/10/2011 4:09	20000
173688	W_Completeren aanvraag		Start	1/10/2011 15:06	20000
173688	A_ACCEPTED	10862	Complete	1/10/2011 15:12	20000
173688	O_SELECTED	10862	Complete	1/10/2011 15:15	20000
173688	A_FINALIZED	10862	Complete	1/10/2011 15:15	20000
173688	O_CREATED	10862	Complete	1/10/2011 15:15	20000
173688	O_SENT	10862	Complete	1/10/2011 15:15	20000

173688	W_Nabellen offertes		Schedule	1/10/2011 15:15	20000
173688	W_Completeren aanvraag		Complete	1/10/2011 15:15	20000
173688	W_Nabellen offertes		Start	1/10/2011 15:45	20000
173688	W_Nabellen offertes		Complete	1/10/2011 15:47	20000
173688	W_Nabellen offertes	10913	Start	8/10/2011 19:56	20000
173688	W_Nabellen offertes	10913	Complete	8/10/2011 20:02	20000
173688	W_Nabellen offertes	11049	Start	10/10/2011 15:02	20000
173688	O_SENT_BACK	11049	Complete	10/10/2011 15:03	20000

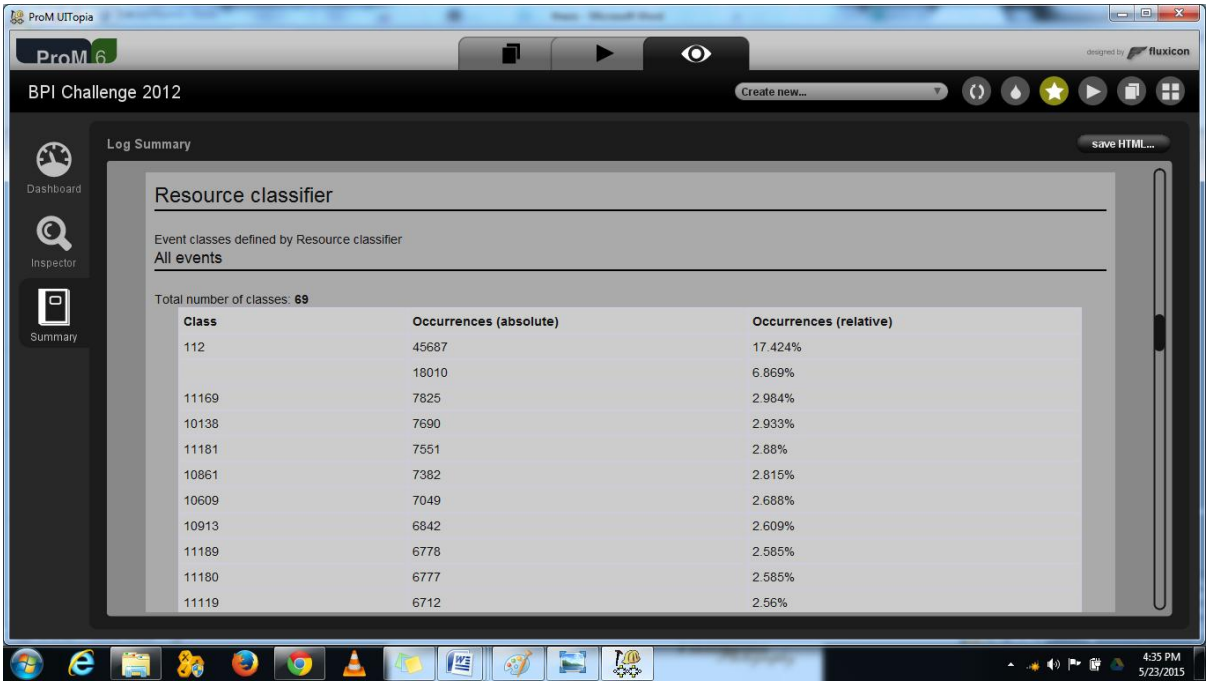


Figure 19- Overall log statistics indicate an unknown resource

Analysis: ***Rule 2- Authorization (Access control):*** *A security principle that ensures only authorised individuals perform certain activities or access certain data objects.*

In both Observation 1 and Observation 2 resources violate the principle of least privilege in access control policy. The principle of least privilege dictates refers to the practice of granting subjects access only to what they need to perform their jobs and no more. Security violations that threaten access control such as the one exhibited by resource 112 and the unknown resource may result in Trojan Horses or viruses being implanted whose activity may not be triggered long after the original event.

5.3 Analysing the data-flow perspective

Result

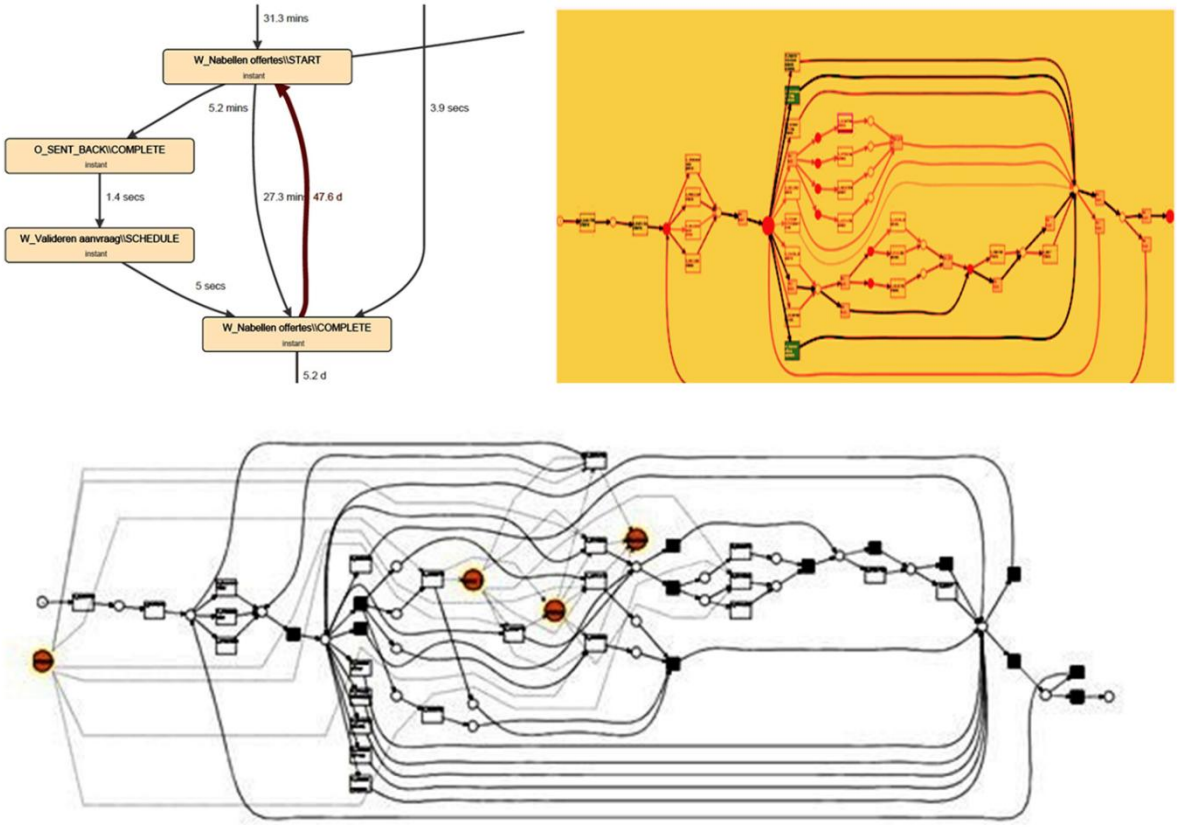


Figure 20- compilation of figures - Conformance Analysis result, Petri Net with Data, and snapshot of Performance Analysis

The result in figure 20 above, illustrates the milestones in our data-aware diagnostic model. It is triple feature initiative which incorporates:-

i. the performance analysis result (whole performance analysis result in appendix),
ii. the alignment result from conformance analysis (in the top right corner refer to figure 15).
iii. the Petri net with data (our data-aware discovery model) (bottom diagram refer to figure 17)

Observation 1 – Transition frequency:- In all results captured in figure 22 below, event *W_Nabellen offertes*(start and complete) which involves following up after transmitting offers to qualified applicants is the most frequently traversed transition.

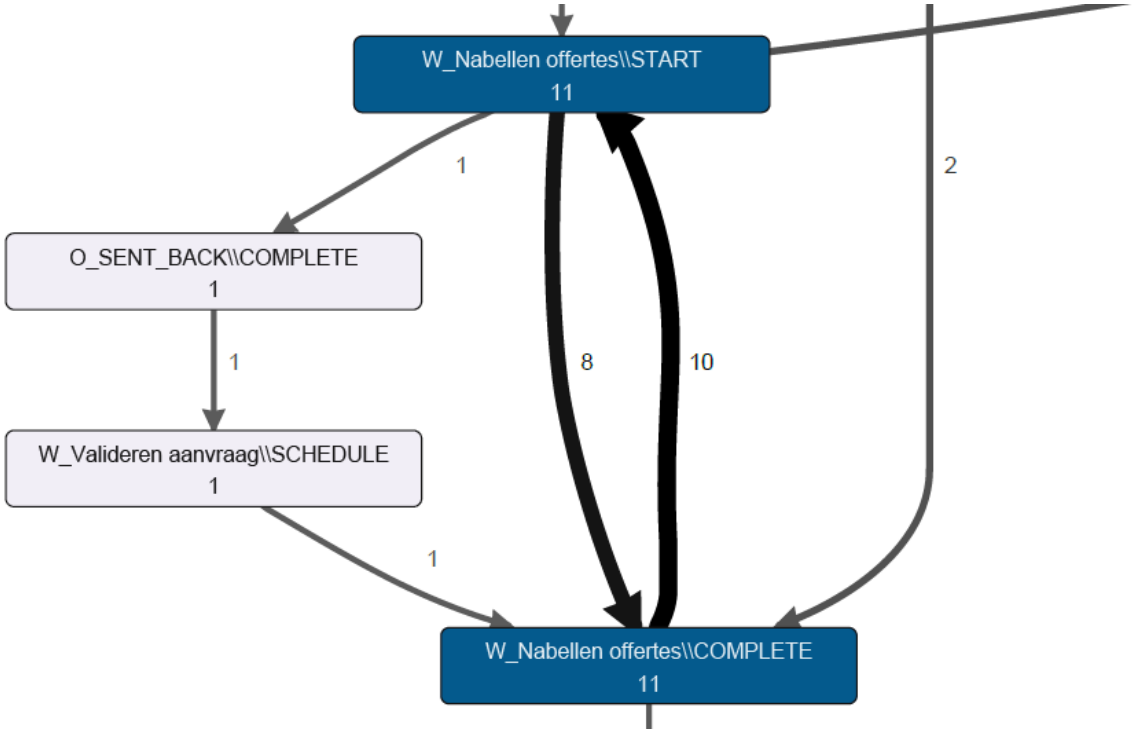


Figure 21-Activity performance snapshot diagram

Analysis- Transition frequency

The high traffic between W_Nabellen offertes start and W_Nabellen offertes complete might have a negative impact as resources may evade decision-support defences that the system provides during eventful periods. If the existing process is not flexible for offer processing for example it often causes adjustments in workflows. These alternate workflows introduce challenges because they increase deviation from routine sequences.

Observation 2 – Bottleneck: - In case number 173694 - the longest case in the log; transition between W_Nabellen offertes start and W_Nabellen offertes complete took 47.6days.

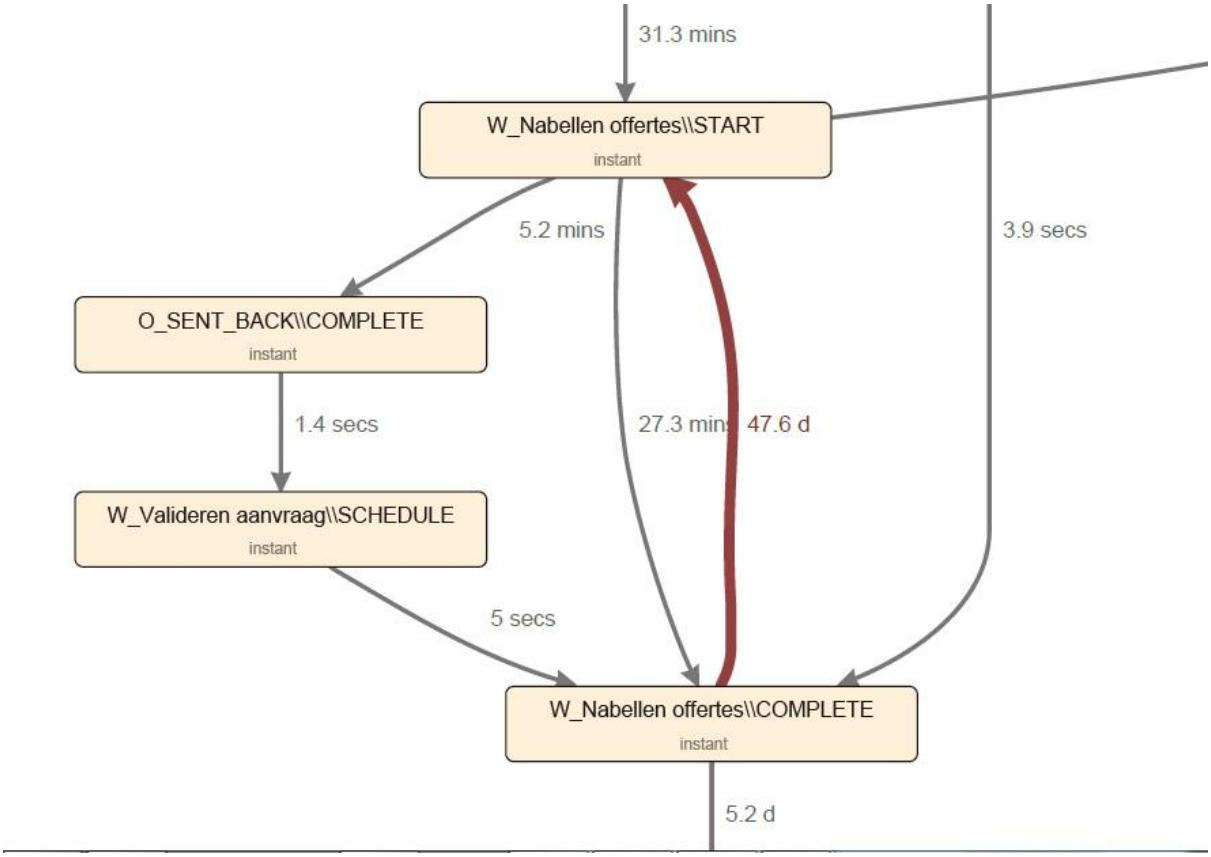


Figure 22-snapshot of event frequency diagram

Analysis- Bottleneck

Delays observed between W_Nabellen offertes start and W_Nabellen offertes completes, show that there is a bottleneck between the events at that place. Bottlenecks directly affect performance and scalability by limiting the amount of data throughput or restricting the number of application connections. Bottlenecks can cause buffer overflows which could result in corrupt data.

Observation 3- Non-alignment of events – In the alignment result zoomed in figure 23 below we observe non-conformance at event O_ACCEPTED, the red border on O_ACCEPTED indicates that out of the 2248 moves recorded in that event, 3 moves that were seen in the model were not present in the log.

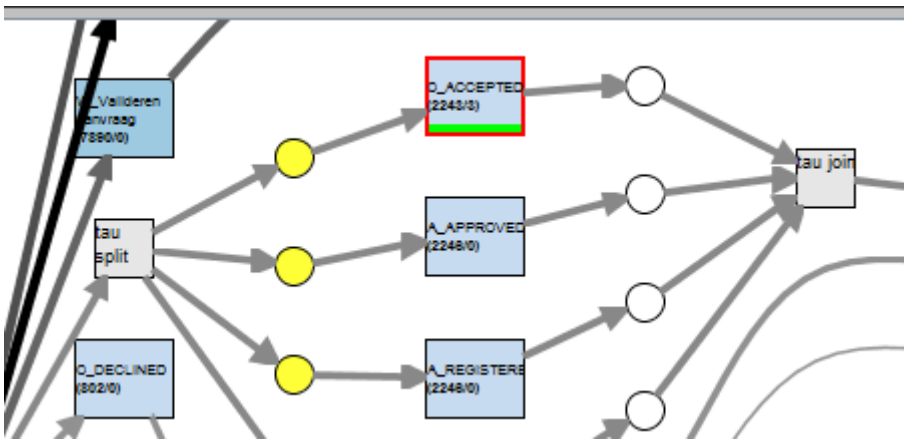


Figure 23-snapshot of Conformance analysis result

Analysis – Non-Conformance

Non conformance is failure to adhere to requirements which directly impacts internal and external costs of a system. The system data used in this work shows only internal costs. Internal failure shows through deficiencies which may be caused by inefficiencies in processes. The internal costs of non-conformance include repeating work, delays, re-doing designs, shortages, failure investigation, verification, inflexibility and malleability.

Chapter Six: Validation of our Data-Aware Diagnostic Fault Detection method

This chapter validates the proposed early data-aware fault detection in process flow approach implemented in previous chapters.

Using our data-aware diagnostic method for early fault detection we have successfully noted entry points of dirt in data as influenced by the workflow. Next, we evaluate the approach using performance and conformance analysis statistics before applying the method and after applying the method to the real-life data log.

To validate our method we used information observed in the results above to clean, filter and improve the data and we compared the result to the initial raw data result.

6.1 Comparing overall statistics

Observations

First we compare the dashboard for our data as shown in Figure 24 below. There is very little change observed from the dashboard overview diagram.

Figure 24 - Raw dashboard of log vs Filtered dashboard of log

Figure 25 and figure 26 below shows supporting statistics for the overall statistical change observed in events per case as well as events classes per case.

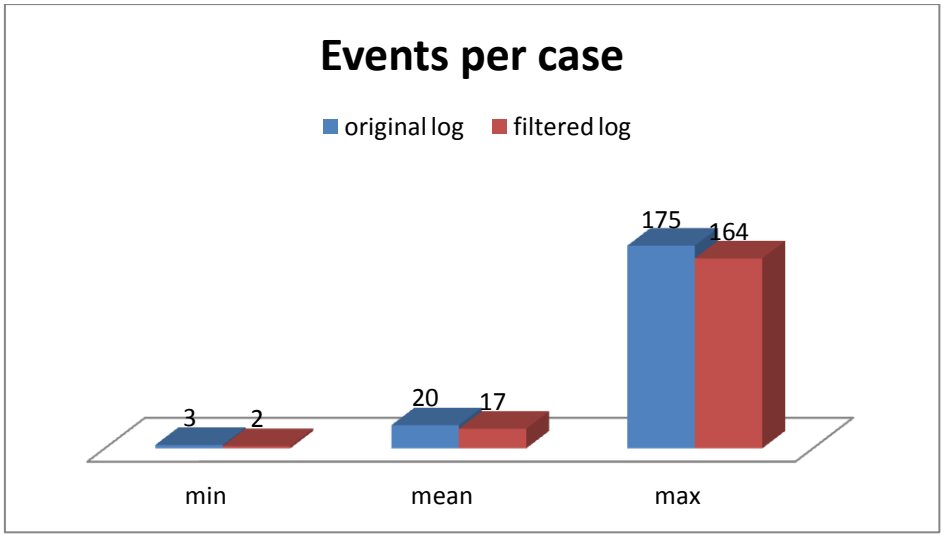


Figure 25- comparison of events per case statistics

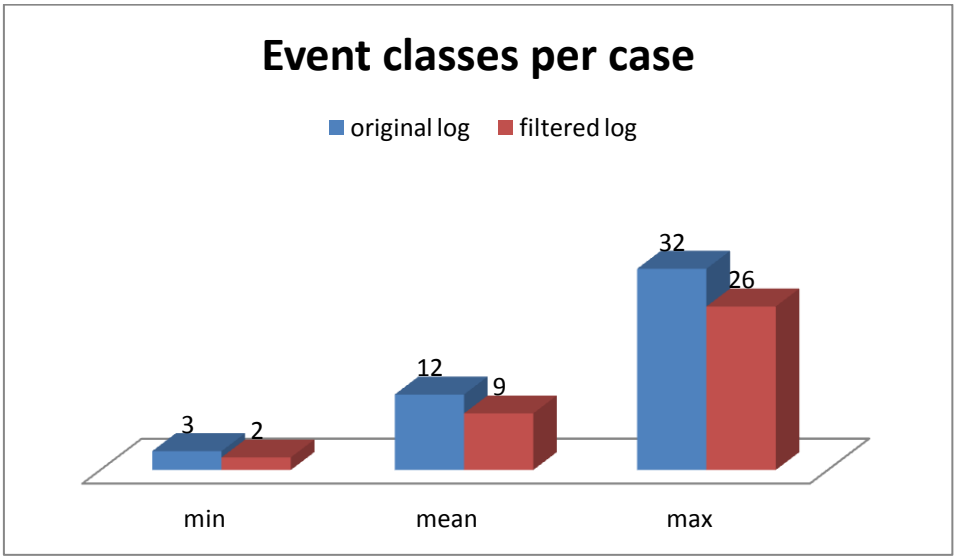


Figure 26- comparison of event classes per case statistics

Evaluation- Overall statistics

The statistics tabulated in figure 25 and 26 indicate a decline in the number of events per case as well as the events classes per case. The removal of redundant cases in the log simplifies and improves analysis.

6.2 Conformance analysis statistics

The statistics obtained from conformance analysis furnish us with details regarding the raw fitness cost of the alignment, calculation time in milliseconds, move-model fitness, trace

fitness, move-log fitness, and trace length. Below is a comparison of these results followed by an evaluation of a filtered log against the raw log.

1. Raw fitness cost statistics

A model with good fitness allows for most of the behavior seen in the event log. Fitness is expressed as a value between 0 (very poor fitness) and 1 (ideal fitness) as defined in Chapter 1 of this work. From the figure below, the average fitness/case has increased and standard deviation has also increased to show an increase in the variation of cases in the log.

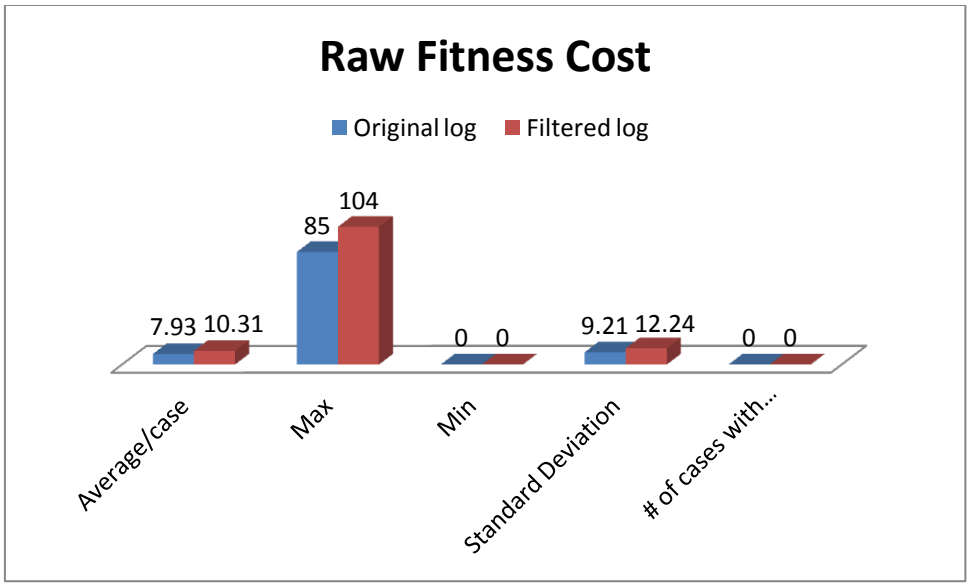


Figure 27- Raw fitness cost statistics (Original log vs Filtered log)

2. Calculation time statistics

These statistics capture the performance of the conformance checker algorithm in terms of time taken to replay the log on the model. In general, calculation time for the filtered log is lower than calculation time for the original log, however the Average Calculation time per case is significantly lower for the original log. It is possible that our filtering has brought about complications in the control flow that make Average Calculation time per case high in the filtered log.

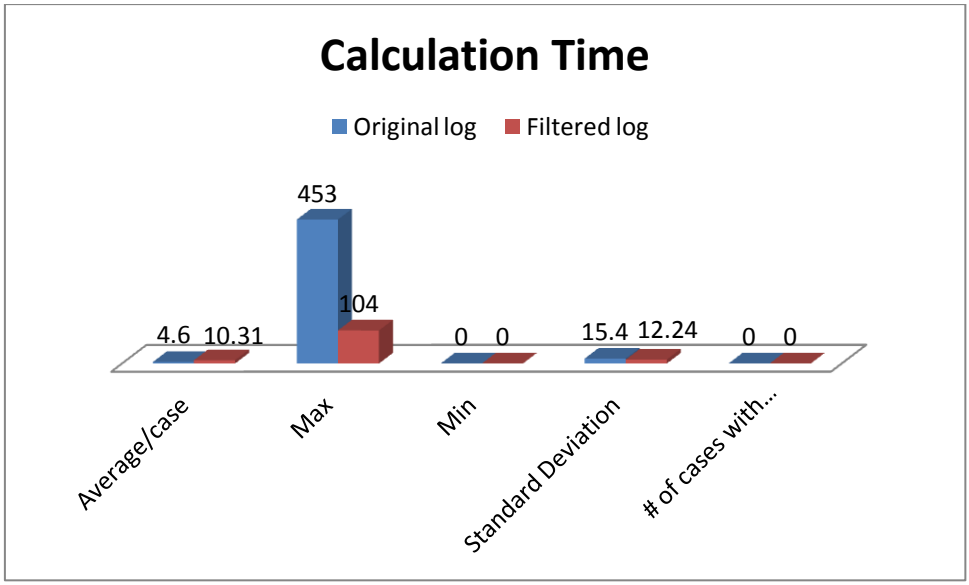


Figure 28 - Calculation Time statistics

3. Move-model fitness

On average the move-model fitness is near perfect for both the original log and the filtered log. However, for the total 13087 cases in the log, the original log observes a misfit for 3 cases.

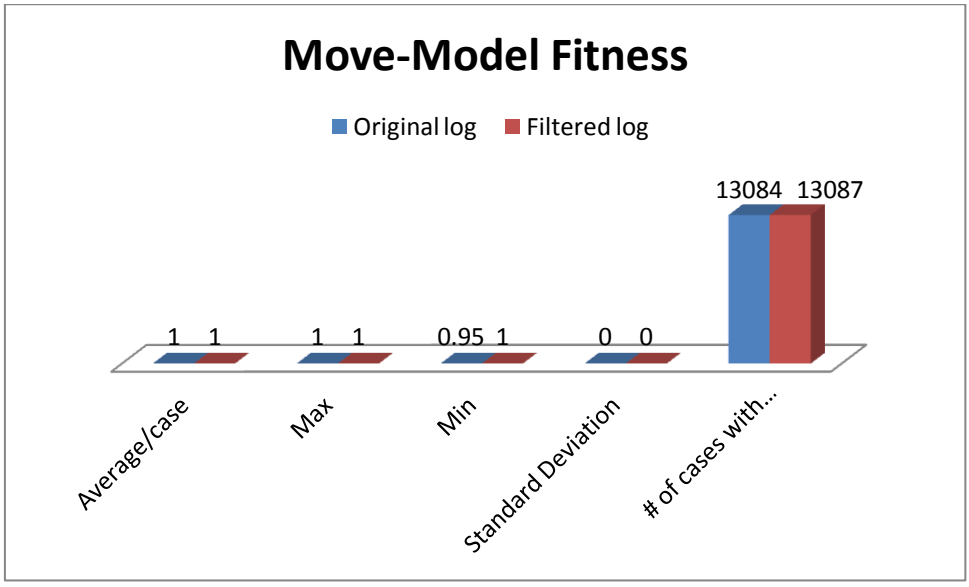


Figure 29- original log vs filtered log Move-Model fitness

4. Trace fitness

Trace fitness is investigating whether a process model is able to reproduce all execution sequences that are in the log at the case level, i.e., the fraction of traces in the log which can be replayed fully.

The filtered log shows lower trace fitness compared to the original log for all 3,429 traces, it is possible that the changes made to the log had a negative impact on the control flow of events.

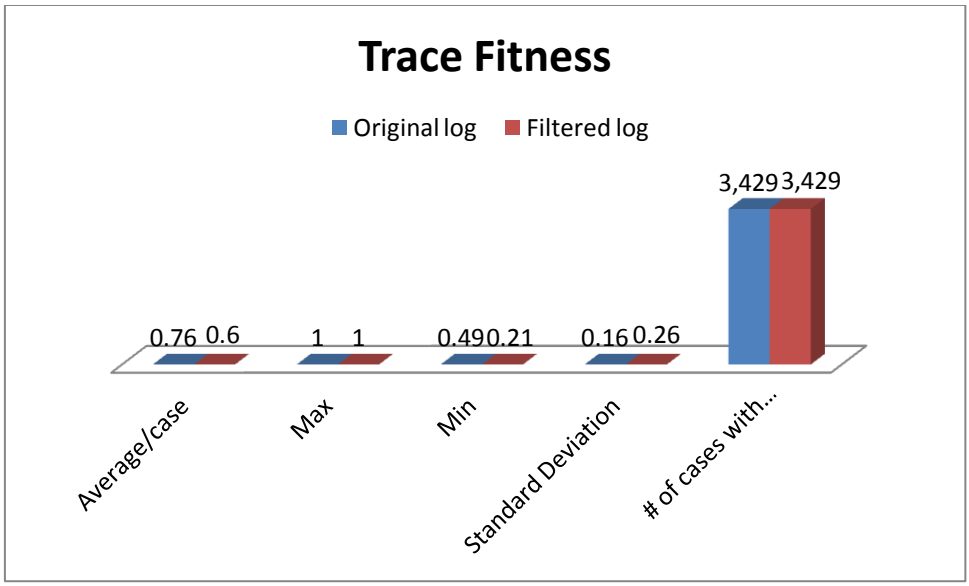


Figure 30- Trace fitness

5. Move-log fitness

In move-log fitness the algorithm moves a step in the log while the model remains in position and shows if there is disparity between the experiential activities in the traces and the achievable activities in the log.

For the 3,429 cases, the original log has higher fitness compared to the filtered log.

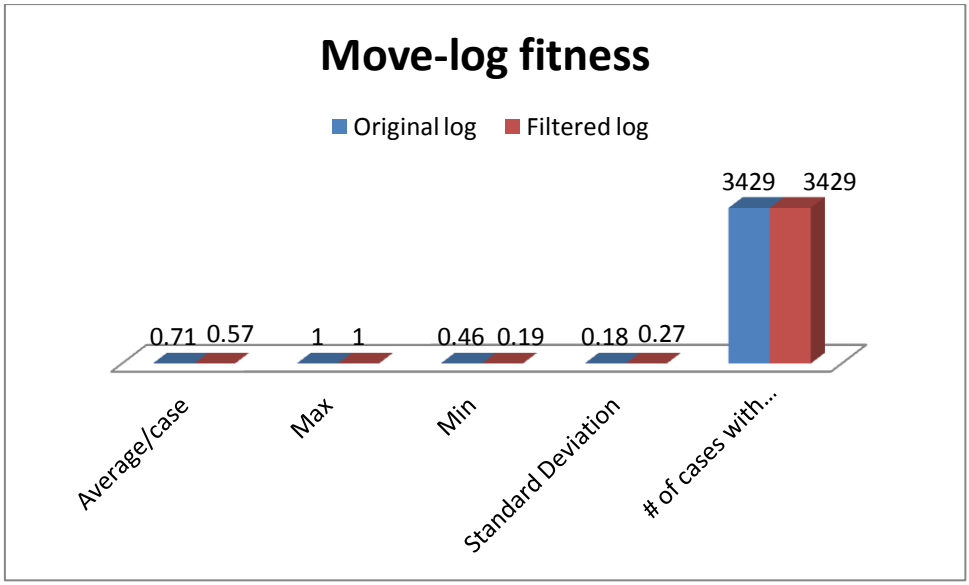


Figure 31- move - log fitness

6. Trace Length

The length of the traces (number of activities in a case) can indicate potential outliers, as observed in the first section of this chapter. For example, a process instance such as case 173694 which is the longest case with 59 events in the log is deviant from the average trace length.

Figure 32 below indicates that the average trace length reduced for the filtered log in comparison to the original log. Given that an alignment is at least equal to the maximum trace length, having anomaly traces in the log may lead to an alignment with several break symbols. As such, a lower trace length indicates the removal of deviant cases.

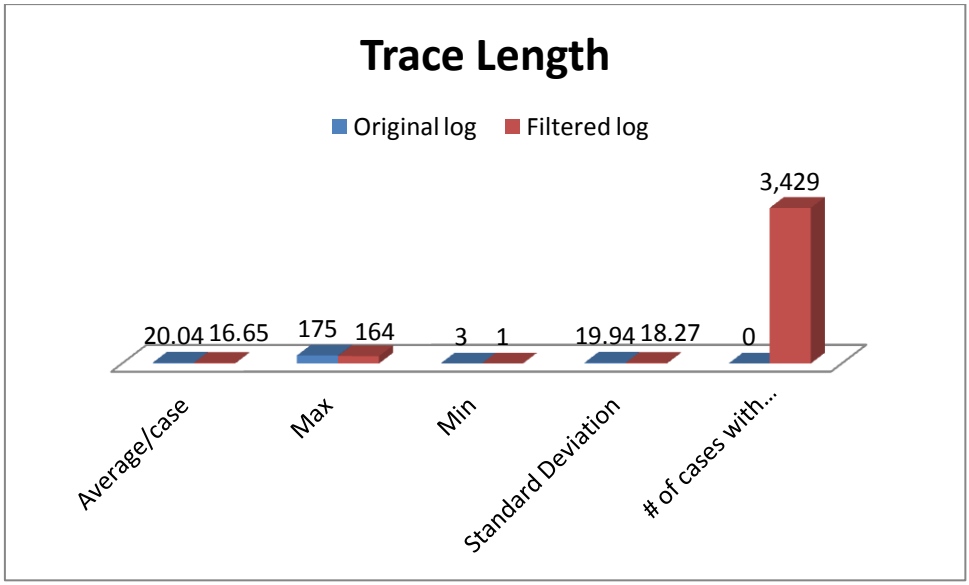


Figure 32- Trace length

7. Overall Conformance Analysis statistics

i. *Synchronous event class (log + model)* -this means both an activity in the model as well as an activity in the current trace can be 'moved' without misalignment.

   a. For the 13,087 cases replayed for both the original log and the filtered log, the original log has a higher synchronous event class (log+model) fitness value compared to our filtered log which shows that the overall alignment of the original log is better compared to the filtered log.

ii. *Calculation time* – the overall calculation time for the filtered log is 43.4% higher than that of the original log. The algorithm performed poorly on the filtered log compared to the original log.

iii. *Skipped events classes* – the original log recorded a total of 3 event classes seen in the model but not captured in the log; in the filtered log no event classes were skipped.

iv. *Violating synchronous event class* – no log+ model activities were noted in both logs that occur in the log without taking any obligations (i.e. tokens) in the model.

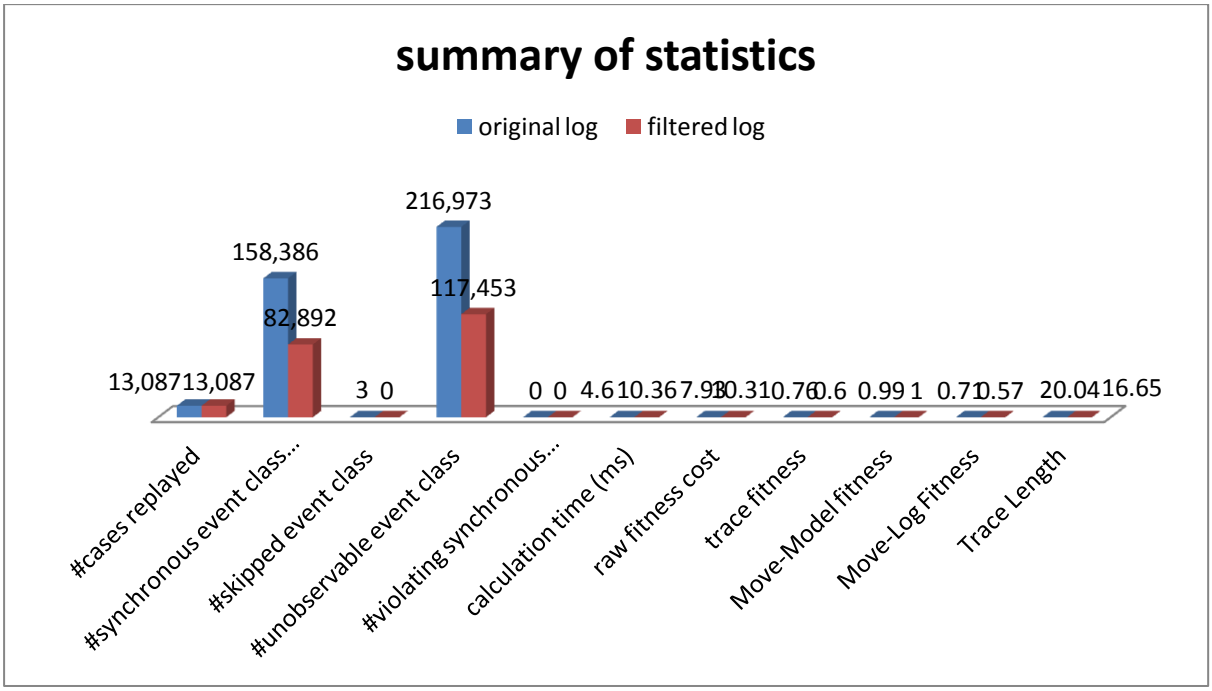


Figure 33- summary of conformance analysis statistics

Chapter Seven: Conclusion and Future Work

This chapter presents the conclusion derived from this research and exhibits the possibilities of expansion of this work in future.

In this paper, we applied data aware-process mining as a process-data centric approach to improve data quality early in the life cycle of big data applications by remodelling the process to capture data more accurately. Data-aware process mining is an approach presented by M. de Leoni at al. in [20]. We applied their approach and improved it by adding compliance rules for conformance checking.

The proposed method has been implemented using existing plug-ins in ProM and evaluated on a real-life event log obtained from the Dutch Financial Institute. The evaluation of our method for early detection of errors in process flow using Petri nets with data was successful in providing new and valuable insights in detecting introduction points of errors. However, our method also failed in other areas of performance as indicated by calculation time statistics, and synchronous event classes' statistics.

In future work, this team would like to implement the suggested the diagnostic data-aware detection algorithm as a Prom plug-in which is an integrated algorithm for checking the alignment of a Petri Net with data and finally generating a verification report for the workflow designer.

Chapter Eight: Publications from this thesis

This chapter briefly states the publication that has been communicated during this research work.

8.1 Published paper

This section briefs about the published research paper during the research.

1. Melody Wadzanayi Murakwani, Manoj Sethi, "Big data quality: Early Detection of errors in process flow using alignments and compliance requirements", in International Journal of Science and Research, Volume 4, Issue 6, June 2015.

References

[1] Alexandros Labrinidis, H.V Jagadish, "Challenges and Opportunities with Big Data", Proceedings of the VLDB Endowment, Istanbul, Turkey, 2012.

[2] Eckerson W, "Data quality and the bottom line: Achieving business success through a commitment to high quality data", The Data Warehousing Institute, 2002.

[3] Barna Saha, Divesh Srivastava, "Data Quality: The other Face of Big Data", ICDE Conference, 2014.

[4] Mark Madsen, "The Challenges of Big Data & Approaches to Data Quality", Third Nature Inc, 2013.

[5] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Byers, "Big Data: The Next Frontier for Innovation, Competition, and Productivity", McKinsey Global Institute, 2011.

[6] Michael Schroeck, et al, "Analytics: The real world use of big data", 2012. Available: http://www-ibm.com/systems/hu/resources/the_real_word_use_of_big_data.pdf [Accessed: January 21, 2015].

[7] Smitha T, V. Suresh Kumar, "Application of Big Data in Data Mining", International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 7, July 2013.

[8] James Boyd, "BIG DATA: National data linkage infrastructure", Available: https://www.ecu.edu.au/__data/assets/pdf_file/0003/513516/James-Boyd-National-Data-Linkage-Infrastructure.pdf [Accessed: February 5, 2015].

[9] Thomas Schutz, SVP, "The state of data quality", 2014, Available: www.qas.com/the state of data quality [Accessed: February 5 2015].

[10] Saha, Divesh Srivastava, "Data Quality: The other Face of Big Data", ICDE Conference, 2014.

[11] W. Fan and F. Geerts, "Foundations of data quality management," Synthesis Lectures on Data Management, vol. 4, no. 5, pp. 1–217, 2012.

[12] S. Grijzenhout, M. Marx, "The quality of the XML web", CIKM: 1719-1724, 2011.

[13] Gartner, "Forecast: Data quality tools", Available: http://www.gartner.com/it-glossary/data-quality-tools.

[14] J. Tee, "Handling the four 'V's of big data: volume, velocity, variety, and veracity", Available: http://www.theserverside.com/feature/Handling-the-four-Vs-of-big-data-volume-velocity-variety-and-veracity, 2013.

[15] S. Sarsfield, "The butterfly effect of data quality", the Fifth MIT: Information Quality Industry Symposium, 2011.

[16] Rajkumar Selvaraj, Gaurav Luthra, "Data Quality in Big Data Testing – A New Paradigm", STeP-IN SUMMIT, 2013.

[17] W. van der Aalst, "Process Mining: Discovery, Conformance and Enhancement of Business Processes", Springer-Verlag, 2011.

[18] Mieke Jansa, Michael Allesb, Miklos Vasarhelyi, "The case for process mining in auditing: Sources of value added and areas of application", International Journal of Accounting Information Systems Volume 14, Issue 1, Pages 1–20, March 2013.

[19] W. van der Aalst, "Process Mining: Discovery, Conformance and Enhancement of Business Processes", Springer-Verlag, 2011.

[20] Massimiliano de Leoni and Wil M.P. van der Aalst, "Data-Aware Process Mining: Discovering Decisions in Processes Using Alignments", ACM 978-1-4503-1656-9, 2013.

[21] Wil van der Aalst, Arya Adriansyah, and Boudewijn van Dongen, "Replaying History on Process Models for Conformance Checking and Performance Analysis", Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, The Netherlands.

[22] C. Wagner, "A Data-Centric Approach to Deadlock Elimination in Business Processes", In Daniel Eichhorn, Agnes Koschmider, and Huayu Zhang, editors, Proceedings of the 3rd Central European Workshop on Services and their Composition, ZEUS 2011, Karlsruhe, Germany, February 21-22, volume 705 of CEUR Work, 2011.

[23] S.W. Sadiq, M.E. Orlowska, W.Sadiq, and C. Foulger, "Data Flow and Validation in Workflow Modelling", In Fifteenth Australasian Database Conference (ADC), Dunedin, New Zealand, volume 27 of CRPIT, pp 207–214, Australian Computer Society, 2004.

[24] S.X. Sun, J.L. Zhao, J.F. Nunamaker, and O.R. Liu Sheng, "Formulating the Data Flow Perspective for Business Process Management", Information Systems Research, 17(4):374–391, 2006.

[25] S.W. Sadiq, M.E. Orlowska, W.Sadiq, and C. Foulger, "Data Flow and Validation in Workflow Modelling", In Fifteenth Australasian Database Conference (ADC), Dunedin, New Zealand, volume 27 of CRPIT, pp 207–214, Australian Computer Society, 2004.

[26] S. Fan, W.C. Dou, and J. Chen, "Dual Workflow Nets: Mixed Control/Data-Flow Representation for Workflow Modeling and Verification", In Advances in Web and Network Technologies, and Information Management (APWeb/WAIM 2007 Workshops), volume 4537 of Lecture Notes in Computer Science, pp 433–444, Springer-Verlag, Berlin, 2007.

[27] W.M.P. van der Aalst, "The Application of Petri Nets to Workflow Management", The Journal of Circuits, Systems and Computers, 8(1):21–66, 1998.

[28] M.H. Sundari, A.K. Sen, and A. Bagchi, "Detecting Data Flow Errors in Work-flows: A Systematic Graph Traversal Approach", In 17th Workshop on Information Technology & Systems (WITS-2007), Montreal, 2007.

[29] Nikola Trocka, Wil M.P. van der Aalst, and Natalia Sidorova, "Data-Flow Anti-Patterns: Discovering Dataflow Errors in Workflows", LNCS 5565, 2009.

[30] D.A. Schmidt, "Data Flow Analysis is Model Checking of Abstract Interpretations", In Proceedings of the 25th ACM SIGPLAN-SIGACT symposium on Principles of programming languages (POPL'98), pages 38-48, ACM,1998.

[31] B.F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, W.M.P. van der Aalst, "The ProM Framework: A New Era in Process Mining Tool Support", ICATPN LNCS 3536, pp. 444–454, Springer-Verlag Berlin Heidelberg 2005.

[32] Elham Ramezani, Vladimir Gromov, Dirk Fahland, and Wil M. P. van der Aalst, "Compliance Checking of Data-Aware and Resource-Aware Compliance Requirements", On the Move to Meaningful Internet Systems: OTM 2014 Conferences Lecture Notes in Computer Science Volume 8841, pp 237-257, 2014.

[33] Available: http://data.3tu.nl/repository/uuid:3926db30-f712-4394-aebc-75976070e91f [Accessed: March, 2015].

Appendix A Conformance analysis result with all alignments and statistics

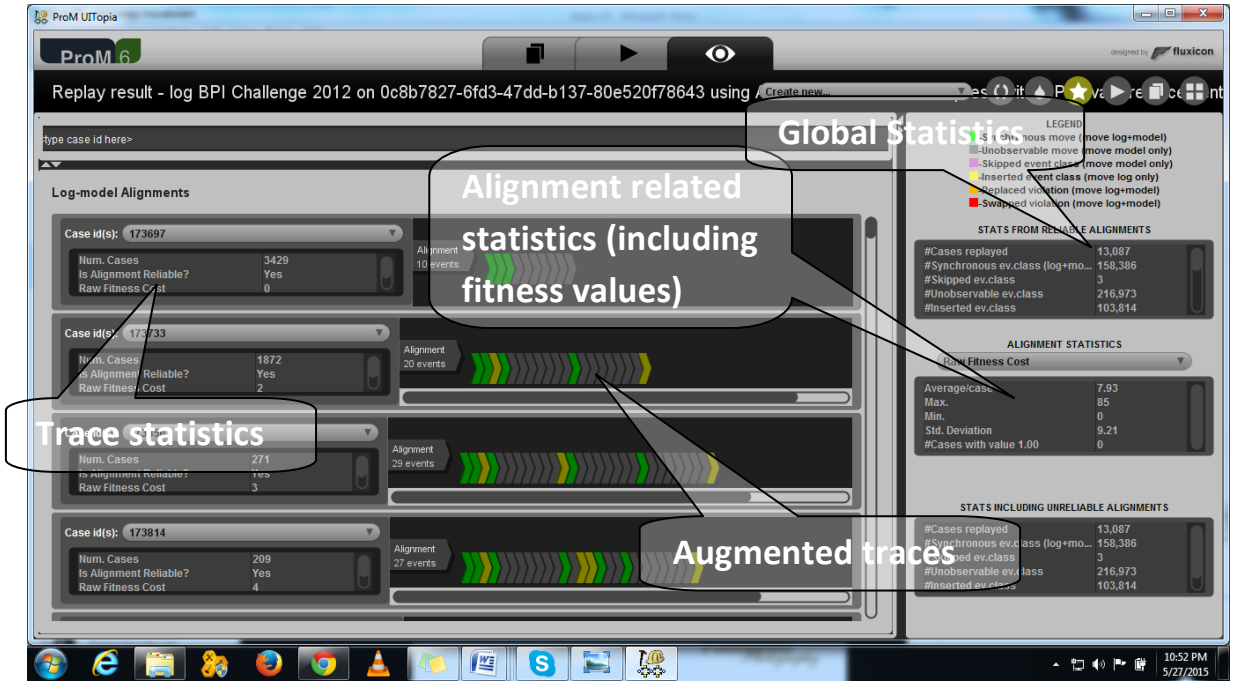


Figure 34 – Conformance Analysis result with all Alignments

Appendix B Event frequency diagram

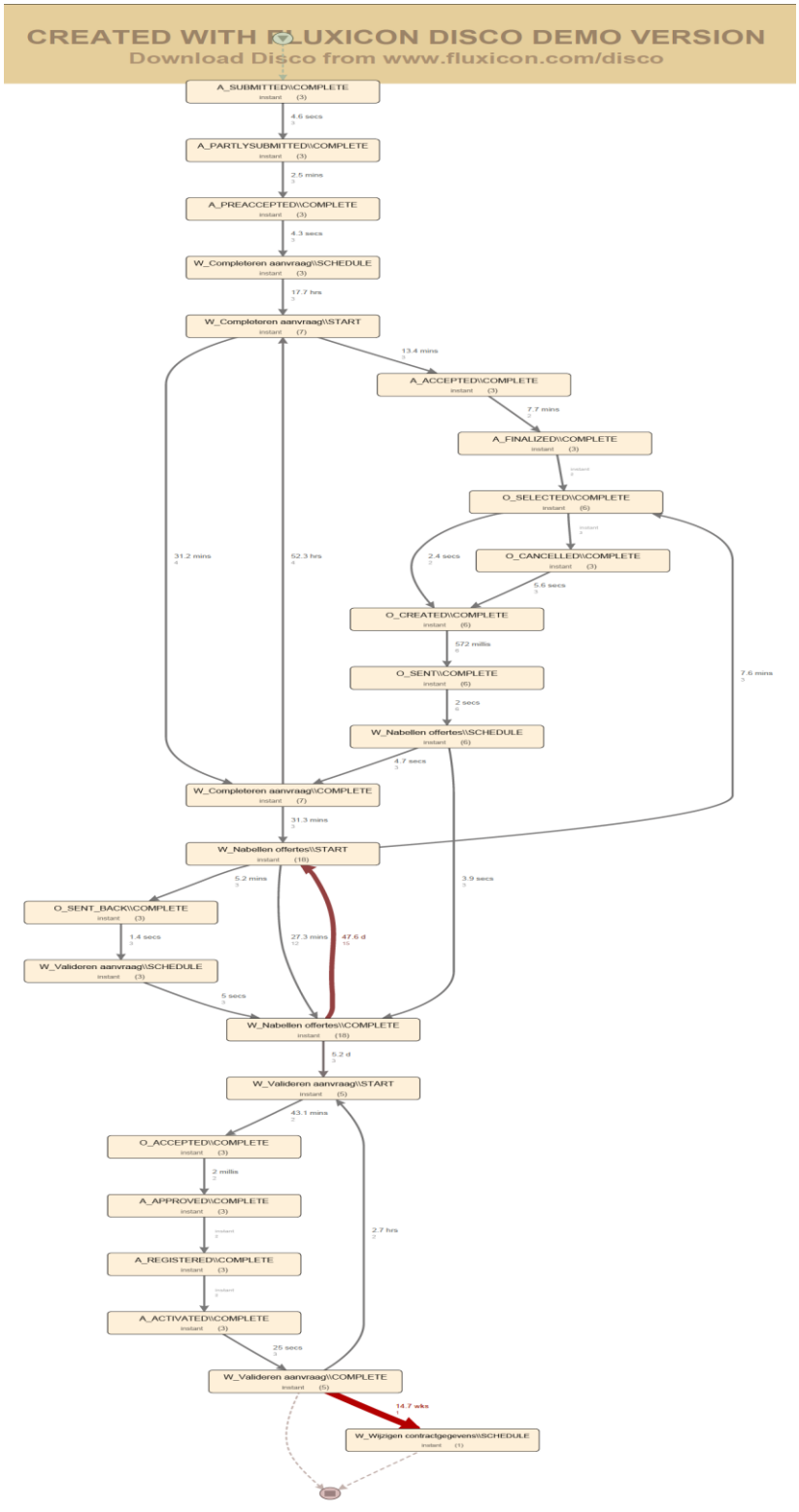


Figure 35- Event frequency diagram

Appendix C Activity Performance diagram

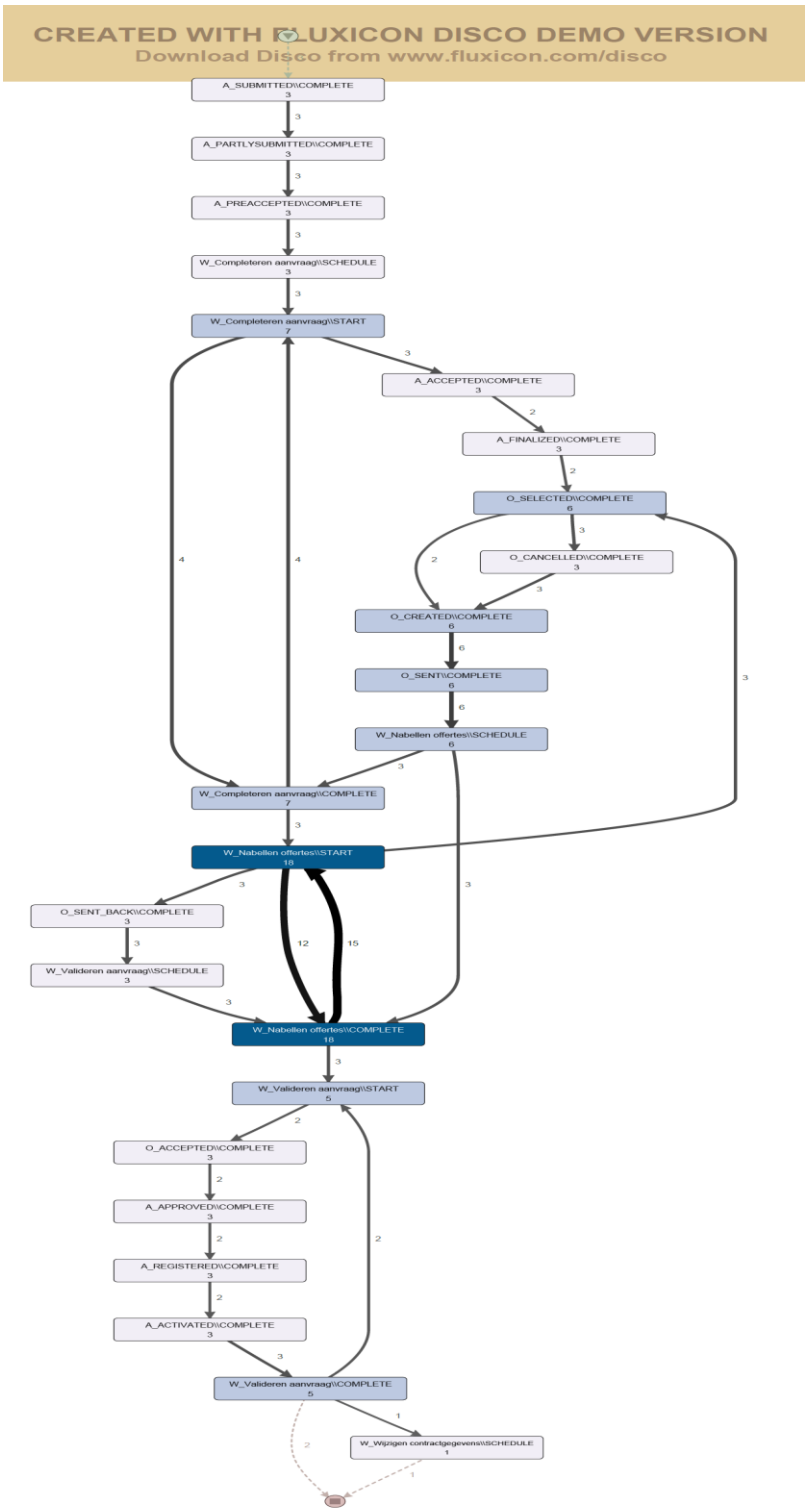


Figure 36: Performance by activity diagram