Wanna Improve Process Mining Results?
It's High Time We Consider Data Quality Issues Seriously

R.P. Jagadeesh Chandra Bose, Ronny S. Mans and Wil M.P. van der Aalst

Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands.
{j.c.b.rantham.prabhakara,r.s.mans,w.m.p.v.d.aalst}@tue.nl

Abstract. The growing interest in process mining is fueled by the increasing availability of event data. Process mining techniques use event logs to automatically discover process models, check conformance, identify bottlenecks and deviations, suggest improvements, and predict processing times. Lion's share of process mining research has been devoted to analysis techniques. However, the proper handling of problems and challenges arising in analyzing event logs used as input is critical for the success of any process mining effort. In this paper, we identify four categories of process characteristics issues that may manifest in an event log (e.g. process problems related to event granularity and case heterogeneity) and 27 classes of event log quality issues (e.g., problems related to timestamps in event logs, imprecise activity names, and missing events). The systematic identification and analysis of these issues calls for a consolidated effort from the process mining community. Five real-life event logs are analyzed to illustrate the omnipresence of process and event log issues. We hope that these findings will encourage systematic logging approaches (to prevent event log issues), repair techniques (to alleviate event log issues) and analysis techniques (to deal with the manifestation of process characteristics in event logs).

Key words: Process Mining, Data Quality, Event Log, Preprocessing, Data Cleansing, Outliers

1 Introduction

Business processes leave trails in a variety of data sources (e.g., audit trails, databases, and transaction logs). Process mining is a relatively young research discipline aimed at discovering, monitoring and improving real processes by extracting knowledge from event logs readily available in today's information systems [1]. Remarkable success stories have been reported on the applicability of process mining based on event logs from real-life workflow management/information systems. In recent years, the scope of process mining broadened from the analysis of workflow logs to the analysis of event data recorded by physical devices, web services, ERP (Enterprise Resource Planning) systems, and transportation systems. Process mining has been applied to the logs of high-tech systems (e.g., medical devices such as X-ray machines and CT scanners),

copiers and printers, and mission-critical defense systems. The insights obtained through process mining are used to optimize business processes and improve customer service. Organizations expect process mining to produce *accurate insights regarding their processes while depicting only the desired traits and removing all irrelevant details*. In addition, they expect the results to be *comprehensible and context-sensitive*.

While the success stories reported on using process mining are certainly convincing, it is not easy to reproduce these best practices in many settings due to the quality of event logs and the nature of processes. For example, *contemporary process discovery approaches have problems in dealing with fine-grained event logs and less structured processes*. The resulting *spaghetti-like process models* are often hard to comprehend [1].

We have applied process mining techniques in over 100 organizations. These practical experiences revealed that real-life logs are often far from ideal and their quality leaves much to be desired. Most real-life logs tend to be *incomplete*, *noisy*, and *imprecise*. Furthermore, contemporary processes tend to be complex and subject to a wide range of variations. This results into event logs that are *fine-granular*, *heterogeneous*, and *voluminous*. Some of the more advanced process discovery techniques try to address these problems. However, as the saying "garbage in – garbage out" suggests, more attention should be paid to the quality of event logs and the manifestation of process characteristics in event logs before applying process mining algorithms. The strongest contributions addressing the 'Business Process Intelligence Challenge' event logs illustrate the significance of log preprocessing [2–5].

The process mining manifesto [6] also stresses the need for high-quality event logs. The manifesto lists five maturity levels ranging from one star ($\star$) to five stars ($\star\star\star\star\star$). At the lowest maturity level, event logs are of poor quality, i.e., recorded events may not correspond to reality and events may be missing. A typical example is an event log where events are recorded manually. At the highest maturity level, event logs are of excellent quality (i.e., trustworthy and complete) and events are well-defined. In this case, the events (and all of their attributes) are recorded automatically and have clear semantics. For example, the events may refer to a commonly agreed upon ontology.

In this paper, drawing from our experiences, we elicit a list of common process characteristics issues and data quality issues that we encounter in event logs and their impact on process mining. We describe several classes of process characteristics problems and data quality problems. For example, regarding data quality problems, three timestamp related issues are identified: (1) missing timestamps (e.g. events with no timestamps), (2) imprecise timestamps (e.g., logs having just date information such that ordering of events on the same day is unknown), and (3) incorrect timestamps (e.g., events referring to dates that do not exist or where timestamps and ordering information are conflicting). Interestingly these problems appear in all application domains. For example, when looking at hospital data we often encounter the problem that only dates are recorded. When looking at X-ray machines, the events are recorded with millisecond precision, but due

to buffering and distributed clocks these timestamps may be incorrect. In this paper we systematically identify the different problems and suggest approaches to address them. We also evaluate several real-life event logs to demonstrate that the classification can be used to identify problems. *The goal of this paper is not to provide solutions on how to address these data quality issues but to highlight the issues and their impact on process mining applications.* We hope that these findings will encourage process mining researchers to think about systematic logging approaches and repair/analysis techniques towards alleviating these quality issues.

The rest of the paper is organized as follows. Section 2 provides the preliminaries which are needed for the remainder of the paper. Section 3 provides a high level classification of process characteristics problems and event log quality problems. In Section 4, the high level problems that arise in event logs are classified in further detail. We evaluate the prevalence of the process related issues and event log quality issues highlighted in this paper in five real-life logs and discuss their results in Section 5. Related work is presented in Section 6. Finally, conclusions are presented in Section 7.

2 Preliminaries

The starting point for process mining is the notion of an *event* and an *event log*. An event log captures the manifestation of events pertaining to the instances of a single process. A *process instance* is also referred to as a *case*. Each event in the log corresponds to a single case and can be related to an *activity* or a *task*. Events within a case need to be *ordered* (an optional *position* attribute can specify the index of an event in a case). An event may also carry optional additional information like *time*, *transaction type*, *resource*, *costs*, etc. Timing information such as date and timestamp of when an event occurred is required to analyze the performance related aspects of the process. Resource information such as the person executing the activity is useful when analyzing the organizational perspective. We refer to these additional properties as *attributes*. To summarize,

- an event log captures the execution of a process.
- an event log contains process instances or cases.
- each case consists of an ordered list of events.
- a case can have attributes such as case id, etc.
- each event can be associated exactly to a single case.
- events can have attributes such as activity, time, resource, etc.

Fig. 1 depicts the characteristics of events and cases and their relationship. For analysis, we need a function that maps any event e into its class $\overline{e}$. For example, for control-flow process discovery, we assume that each event is classified based on its activity. Consider a *case* for which n events have been recorded: $e_1, e_2, \ldots e_n$. This process instance has a *trace* $\overline{e_1}, \overline{e_2}, \ldots \overline{e_n}$ associated to it. If we use activity names as a classifier, the trace corresponds to a sequence of activities. However, if we would classify events based on resource information, a trace

could correspond to a sequence of person names. We use $\mathcal{A}$ to denote the set of activities (event classes) in an event log.

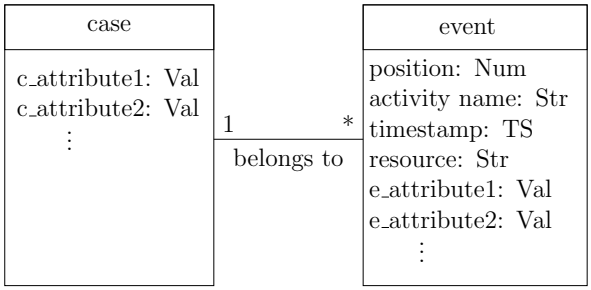

Fig. 1. Cases and events can have attributes and each event can be associated to a single case.

3 High Level Classification of Problems

The problems and challenges arising in analyzing event logs in process mining can be broadly classified into two categories:

1. *Process characteristics:* This class deals with challenges emanating from characteristics such as fine-granular activities, process heterogeneity/variability, process evolution, and high volume processes.
2. *Quality of event log:* This class deals with problems that stem from issues related to the quality of logging manifested in event logs.

We discuss the two classes of problems in detail in the following subsections.

3.1 Process Characteristics

Given an event log, we can extract a few basic metrics such as the *number of cases* in the event log, the *average number of events per case*, the *number of unique activities*, and the *number of distinct traces* as illustrated in Fig. 2. Depending on how these metrics manifest in an event log, one can face several challenges such as dealing with fine-granular events, case heterogeneity, voluminous data and, concept drifts.

Voluminous Data–Large Number of Cases or Events Today, we see an unprecedented growth of data from a wide variety of sources and systems across many domains and applications. For example, high-tech systems such as medical systems and wafer scanners produce large amounts of data, because they typically capture very low-level events such as the events executed by the system components, application level events, network/communication events, and

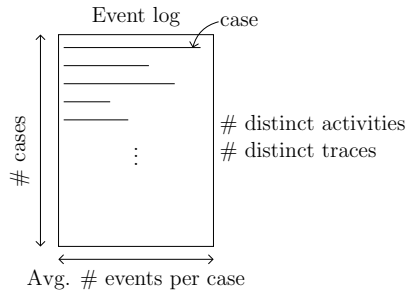

Fig. 2. Process characteristics manifested in event logs.

sensor readings (indicating status of components etc.). Each atomic event in these environments has a short life-time and hundreds of events can be triggered within a short time span (even within a second) (i.e., *the average number of events or the number of cases can be very high of the order of a few millions (thousands) of events (cases)*).

Boeing jet engines can produce 20 terabytes (TB) of operational information per hour. In just one Atlantic crossing, a four-engine jumbo jet can generate 640 terabytes of data [7]. Such voluminous data is emanating from many areas such as banking, insurance, finance, retail, healthcare, and telecommunications. For example, Walmart is logging one million customer transactions per hour and feeding information into databases estimated at 2.5 petabytes in size [7], a global Customer Relationship Management (CRM) company is handling around 10 million calls a day with 10–12 customer interaction events associated with each call [8]. The term "big data" has emerged to refer to the spectacular growth of digitally recorded data [9].

Process mining has become all the more relevant in this era of "big data" than ever before. The complexity of data demands powerful tools to mine useful information and discover hidden knowledge. Contemporary process mining techniques/tools are unable to cope with massive event logs. There is a need for research in both the algorithmic as well as deployment aspects of process mining. For example, we should move towards developing efficient, scalable, and distributed algorithms in process mining.

Case Heterogeneity–Large Number of Distinct Traces Many of today's processes are designed to be *flexible*. This results in event logs containing a heterogeneous mix of usage scenarios with diverse and unstructured behaviors (i.e., *the number of distinct traces is too high*). Another source of heterogeneity stems from operational processes that change over time to adapt to changing circumstances, e.g., new legislation, extreme variations in supply and demand, seasonal effects, etc. Although it is desirable to also record the scenario chosen for a particular case, it is infeasible to define all possible variants.

There are several scenarios where such processes exist. There is a growing interest in analyzing event logs of high-tech systems such as X-ray machines, wafer scanners, and copiers and printers. These systems are complex large scale

systems supporting a wide range of functionality. For example, medical systems support medical procedures that have hundreds of potential variations. These variations create heterogeneity in event logs.

Process mining techniques have problems when dealing with heterogeneity in event logs. For example, process discovery algorithms produce spaghetti-like incomprehensible process models. Analyzing the whole event log in the presence of heterogeneity fails to achieve this objective. Moreover, users would be interested in learning any variations in process behavior and have their insights on the process put in perspective to those variations.

Trace clustering has been shown to be an effective way of dealing with such heterogeneity [10–15]. The basic idea of trace clustering is to partition an event log into homogenous subsets of cases. Analyzing homogenous subsets of cases is expected to improve the comprehensibility of process mining results. In spite of its success, trace clustering still remains a subjective technique. A desired goal would be to introduce some objectivity in partitioning the log into homogenous cases.

Event Granularity–Large Number of Distinct Activities Processes that are defined over a huge number of activities generate event logs that are fine-granular. Fine-granularity is more pronounced in event logs of high-tech systems and in event logs of information systems where events typically correspond to automated statements in software supporting the information system. One can also see such phenomena in event logs of healthcare processes, e.g., one can see events related to fine-grained tests performed at a laboratory in conjunction with coarse-grained surgical procedures.

Process mining techniques have difficulties in dealing with fine-granular event logs. For example, the discovered process models are often spaghetti-like and hard to comprehend. For a log with $|\mathcal{A}|$ event classes (activities), a flat process model can be viewed as a graph containing $|\mathcal{A}|$ nodes with edges corresponding to the causality defined by the execution behavior in the log. Graphs become quickly overwhelming and unsuitable for human perception and cognitive systems even if there are more than a few dozens of nodes [16]. This problem is compounded if the graph is dense (which is often the case in unstructured processes) thereby compromising the comprehensibility of models.

Analysts and end users often prefer higher levels of abstraction without being confronted with low level events stored in raw event logs. One of the major challenges in process mining is to bridge the gap between the higher level conceptual view of the process and the low level event logs. Several attempts have been reported in the literature on grouping events to create higher levels of abstraction ranging from the use of semantic ontologies [17, 18] to the grouping of events based on correlating activities [19, 20] or the use of common patterns of execution manifested in the log [21, 22]. However, most of these techniques are tedious (e.g., semantic ontologies), only partly automated (e.g., abstractions based on patterns), lack domain significance (e.g., correlation of activities), or result in discarding relevant information (e.g., abstraction).

Process Flexibility and Concept Drifts Often business processes are executed in a dynamic environment which means that they are subject to a wide range of variations (*variations lead to diversity in traces and thereby lead to a high number of distinct traces*). Process changes manifest latently in event logs. Analyzing such changes is of the utmost importance to get an accurate insight on process executions at any instant of time. Based on the duration for which a change is active, one can classify changes into *momentary* and *evolutionary*.

– **Evolutionary change:** Evolutionary changes are changes that are persistent. These changes manifest in such a way that for the group of traces subsequent to the point of change there are substantial differences in comparison with earlier performed traces with regard to the activities performed, the ordering of activities the data involved and/or the people performing the activities. Fig. 3 depicts an example manifestation of an evolutionary change. For a given process, events "A", "B", and "C" are always recorded sequentially as part of the normal process flow. Due to an evolutionary change, the activities "B" and "C" are replaced (substituted) with the activities "D" and "E" respectively. Thus events "D" and "E" are recorded sequentially instead of the events "B" and "C" in the traces subsequent to the point of change.

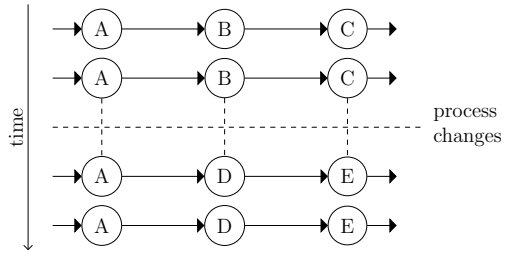

Fig. 3. Evolutionary change: Due to an evolutionary change the events "D" and "E" are recorded as part of the normal process flow instead of the events "B" and "C".

– **Momentary change:** Momentary changes are short-lived and affect only a very few cases. Momentary changes manifest as exceptional executions or outliers in an event log. Fig. 4 exemplifies a momentary change. For a trace, tasks "A" and "C" have been performed and for which events "A" and "C" have been recorded. However, due to an exception, task "B" needed to be performed in between tasks "A" and "C". As a consequence, event "B" has been recorded in between events "A" and "C" which does not correspond to the normal process flow.

Current process mining algorithms assume processes to be in a steady state. In case a log which contains multiple variants of a process is analyzed an overly complex model is obtained. Moreover, for the discovered model there is no information on the process variants that existed during the logged period. Recent efforts in process mining have attempted at addressing this notion of *concept*

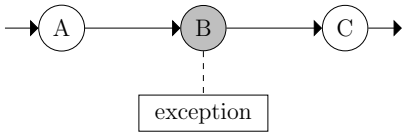

Fig. 4. Momentary change: As part of the normal process flow, the events "A" and "C" are recorded after each other. However, due to an exception, event "B" has occurred in between events "A" and "C".

drifts [23–26]. Several techniques have been proposed to detect and deal with changes (if any) in an event log.

3.2 Quality of the Event Log

With regard to the quality of an event log there can be various issues. we distinguish four broad categories of problems.

- **Missing Data:** This corresponds to the scenario where different kinds of information can be missing in a log although it is mandatory. For example, a certain entity of a log such as an event, event attribute/value, and relation is *missing*. Missing data mostly reflects a problem in the logging framework/process.

- **Incorrect Data:** This corresponds to the scenario where although data may be provided in a log, it may turn out that, based on context information, the data is logged *incorrectly*. For example, an entity, relation, or value provided in a log is *incorrect*.

- **Imprecise Data:** This corresponds to the scenario where the logged entries are too coarse leading to a loss of precision. Such *imprecise* data prohibits in performing certain kinds of analysis where a more precise value is needed and can also lead to unreliable results. For example, the timestamps logged can be too coarse (e.g., in the order of a day) thereby making the order of entries unreliable.

- **Irrelevant Data:** This corresponds to the scenario where the logged entries may be irrelevant *as it is* for analysis but another relevant entity may have to be derived/obtained (e.g., through filtering/aggregation) from the logged entities. However, in many scenarios such filtering/transformation of irrelevant entries is far from trivial and this poses as a challenge for process mining analysis.

In the next section, we elaborate these four categories of event log quality issues in detail.

Table 1. For each entity of an event log depicted in Fig.2 it is indicated whether the "missing", "incorrect", "imprecise", and "irrelevant" problem type applies. For each identified event log quality issue an unique number is given starting with an "I".

	case	**event**	**belongs to**	**c_attribute**	**position**	**activity name**	**timestamp**	**resource**	**e_attribute**
Missing Data	I1	I2	I3	I4	I5	I6	I7	I8	I9
Incorrect Data	I10	I11	I12	I13	I14	I15	I16	I17	I18
Imprecise Data			I19	I20	I21	I22	I23	I24	I25
Irrelevant Data	I26	I27							

4 Event Log Quality Issues

The four categories of problems that are identified in Section 3.2 for an event log, can emanate for different entities within an event log. Table 1 outlines the manifestation of the different classes of problems across the various entities of an event log. In total, 27 different classes of quality issues can be identified. For each identified event log quality issue an unique number is given starting with an "I". Note that not for all combinations of problem classes and log entities an issue has been identified as not all of them are applicable. We now discuss each of these data quality issues in detail. For each of these issues, first a description and an example is given followed by a discussion about the impact of the issue on the process mining application.

4.1 Missing Cases (I1)

This quality issue refers to the scenario where a case has been executed in reality but it has not been recorded in the log. For example, assume that in the period between "01-01-2012" and "30-06-2012" 500 cases have been executed in reality. However, in the event log only 450 cases exist. As another example, we might encounter an event log where we notice case ids missing in consecutive set of numbers, e.g., we might have an event log with case ids ..., case 888, case 889, case 891, case 892, ... It is most likely that case 890 is missing from this event log.

The effect of missing cases may be that the discovered process mining results do not match with the real execution of the cases, e.g., the missing cases might correspond to a critical branch of a process. However, an important aspect of a process mining algorithm is that the discovered result should not restrict behavior to just the examples seen in the log. As an event log only contains example behavior the discovered model should ideally also explain the behavior of another sample log of the same process, i.e. the resulting model should be a generalization of the behavior seen in the event log. In this way, the effect of missing

cases is alleviated. The genetic algorithm described in [27] allows for searching a process model in which generalization is an important quality dimension.

4.2 Missing Events (I2)

This quality issue refers to the scenario where one or more events are missing within the trace although they occurred in reality. For example, Fig. 5 shows a missing event in a trace. In reality, events "A", "B", and "C", have occurred. However, only events "A" and "C" have been logged (event "B" has not been recorded for the trace).

The effect of missing events is that there may be problems with the results that are produced by a process mining algorithm. In particular, relations may be inferred which hardly or even do not exist in reality. To date, we are not aware of any process mining algorithm which is specifically developed for detecting or dealing with missing events. However, algorithms which are most close to dealing with missing events are the ones which can deal with noise (e.g. the fuzzy miner [20] and the heuristics miner [1]).

Fig. 5. Missing events: events "A", "B", and "C" have occurred in reality. Only events "A" and "B" are included in the log whereas event "B" is not included.

Regarding missing events, one specific sub-issue can be identified.

– *Partial / Incomplete Traces*: The prefix and/or suffix events corresponding to a trace are missing although they occurred in reality. This is more prevalent in scenarios where the event data for analysis is considered over a defined time interval (e.g., between Jan'12 and Jun'12). The initial events (prefix) of cases that have started before Jan'12 would have been omitted due to the manner of event data selection. Likewise cases that have been started between Jan'12 and Jun'12 but not yet completed by Jun'12 are incomplete and have their suffixes missing.
  Fig. 6 exemplifies a partial/incomplete trace. In reality, events "A", "B", "C", "D", "E", and "F" have occurred. However, only events "C" and "D" have been logged whereas the prefix with events "A" and "B" and the suffix with events "E" and "F" have not been recorded for the trace.

4.3 Missing Relationships (I3)

This quality issue corresponds to the scenario where the association between events and cases are missing. For example, for the event `register patient` it is not clear for which patient it has been executed. The effect of missing relationships may be similar as for the missing events quality issue.

Fig. 6. Partial / incomplete traces: events "A", "B", "C", "D", "E", and "F" have occurred in reality. Only events "C" and "D" are included in the log whereas the prefix with events "A" and "B" and the suffix with events "E" and "F" are not included.

4.4 Missing Case Attributes (I4)

This quality issue corresponds to the scenario where the values corresponding to case attributes are missing. For example, for patient `John` his weight has not been recorded whereas for all the other patients the weight has been registered. Analysis techniques relying on values of case attributes that are missing need to ignore these affected cases. If many cases need to be ignored this may lead to unreliable results.

4.5 Missing Position (I5)

For this quality issue, we assume that no timestamps have been given for events. As a result, the ordering of events in the log determines the order in which the events occurred in reality. In this scenario, there exist events for which its ordering with respect to other events in the trace is not known. For example, for the `perform CT-scan` event it not clear whether it occurred before or after the `visit outpatient clinic` event. As the ordering of events is not clear within a log, process mining algorithms have problems with discovering the correct control-flow. That is, relations may be inferred which hardly or even do not exist in reality.

4.6 Missing Activity Names (I6)

This quality issue corresponds to the scenario where the activity names of events are missing. For example, for a case there are three events named `check amount` and there are two events with no name. For events for which no activity name has been given it is not clear whether they all refer to the same activity or that they refer to multiple activities. An option may be to remove these activities or to keep them in the log. However, for both situations, the results as produced by a process mining algorithm, relying on activity names, may be questionable.

4.7 Missing Timestamps (I7)

This quality issue corresponds to the scenario where for one or more events no timestamp is given. For example, for the `perform surgery` event it is not clear at which time it has happened. Missing timestamps either hampers the applicability of certain process mining techniques such as performance analysis or might result in incorrect results. For example, relations may be inferred by a control-flow discovery algorithm which hardly or even do not exist in reality.

However, If the position of events (in a case) are guaranteed to be correct (in spite of missing timestamps), one do not have any difficulty in inferring the control-flow. To date, we are not aware of a process mining algorithm in which missing timestamps are inserted based on an event log only. However, in [28], a probabilistic approach is proposed for inferring missing timestamps for events based on a given as-is process.

4.8 Missing Resources (I8)

This quality issue corresponds to the scenario where the resources that executed an activity have not been recorded. For example, for an event corresponding to the activity `perform surgery` it is not given by which person it has been executed although in reality the activity is always performed by a doctor. As an effect, organizational miners such as the Organizational model miner [29] and the Social network miner [29] may discover erroneous information regarding the organizational perspective of a process. Furthermore, process discovery algorithms that also take the resource perspective into account (e.g. the Fuzzy miner [20]) are affected by incorrect resource information.

4.9 Missing Event Attributes (I9)

This quality issue corresponds to the scenario where the values corresponding to event attributes are missing. For example, for the event `register loan`, the amount of the loan has not been recorded. In a similar fashion as for the `missing case attributes` quality issue, unreliable results may be obtained for cases or events that are affected.

4.10 Incorrect Cases (I10)

This quality issue corresponds to the scenario where certain cases in the log belong to a different process. This is manifested mostly due to logging errors from information systems that support more than one process. For example, an instance of process B wrongly gets logged as an instance of process A. Such incorrect cases act as outliers and mislead process mining algorithms e.g., process discovery algorithms would be misled with spurious control-flow relationships.

4.11 Incorrect Events (I11)

This quality issue corresponds to the scenario where certain events in the event log are logged incorrectly. For example, events that were not actually executed for some cases are logged as having been executed.

4.12 Incorrect Relationships (I12)

This quality issue corresponds to the scenario where the association between events and cases are logged incorrectly. For example, an event belonging to case A is logged under case B.

4.13 Incorrect Case Attributes (I3)

This quality issue corresponds to the scenario where the values corresponding to case attributes are logged incorrectly. For example, the amount of claim in a loan/overdraft application process is logged incorrectly for some cases. This hampers the application of analysis techniques that make use of case attributes such as the decision miner [30–32]. One means of mitigating incorrect values is to ignore cases with incorrect entries. Alternatively, one can try to infer the (correct) values based similar instances in the event log. The latter is applicable in scenarios where we have statistically significant number of cases in the event log and assuming that the number of cases with incorrect values are few.

4.14 Incorrect Position (I14)

This quality issue is prevalent in logs that do not have timestamps associated with events. Typically, in logs that do not have timestamps, the order in which the events appear is assumed to be the order in which they have been executed, i.e., the position of events in a case defines its execution order. When the position of events is logged incorrectly, process mining algorithms have problems in discovering the correct control-flow. That is, relations may be inferred which hardly or even do not exist in reality.

4.15 Incorrect Activity Names (I5)

This quality issue corresponds to the scenario where the activity names of events are logged incorrectly.

4.16 Incorrect Timestamps (I16)

This quality issue corresponds to the scenario where the recorded timestamp of (some or all) events in the log do not correspond to the real time at which the events have occurred. For example, Fig. 7(a) illustrates an incorrect recording of the timestamp of an event. Event "A" has been automatically registered at time "03-12-2011 13:45". However, the timestamp of the event recorded in the log is "03-12-2011 13:59". Another example is that of a timestamp recorded as February 29 in a non-leap year. Such discrepancies manifest in systems where there are different clocks that are not synchronized with each other and/or systems where there are delays in logging.

Process mining algorithms discover the control-flow based on behavior observed in the log. The possible effect of incorrect timestamps is that the discovered control-flow relations are unreliable or even incorrect. Moreover, in applications such as the discovery of signature patterns for diagnostic purposes (e.g., fraudulent claims, fault diagnosis, etc.) [33], there is a danger of reversal of *cause* and *effect* phenomenon due to incorrect timestamps. Fig. 7(b) depicts an example of such a scenario. Although the events "A" and "B" occur in that particular order in reality, they are logged as "B" and "A" thereby plausibly leading to incorrect inference that "B" causes "A".

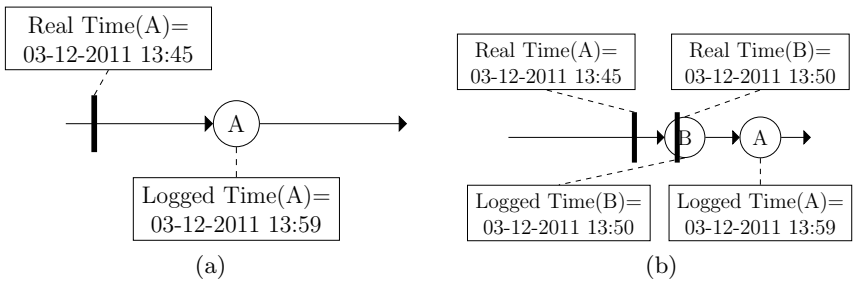

Fig. 7. Incorrect timestamps: (a) event "A" occurred in reality (marked by a dark vertical line) at "03-12-2011 13:45". However, the timestamp of the event recorded in the log is "03-12-2011 13:59" (b) danger of reversal of cause and effect phenomenon due to incorrect timestamps–one can wrongly infer that "B" causes "A" where in reality "A" occurred before "B".

4.17 Incorrect Resources (I17)

This quality issue corresponds to the scenario where the resources that executed an activity are logged incorrectly. For example, an event corresponding to the activity register that has been executed by resource R_1 belonging to department D_1 is logged as being executed by resource R_2 of department D_2. Incorrect resource information in an event log poses problems in the analysis of organizational perspective of processes and in the performance analysis of resources/deparments within an organization. Furthermore, process discovery algorithms such as the Fuzzy miner [20] that take into consideration other attributes of events (such as the resource) besides the activity sequences for inferring control-flow relations are affected by incorrect resource information.

4.18 Incorrect Event Attributes (I18)

This quality issue corresponds to the scenario where the information in various attributes of events are recorded incorrectly. For example, the result of a lab test (such as lipid profiles) of a patient is noted incorrectly in a healthcare process. Incorrect values in event attributes primarily affects the data perspective analysis in process mining such as in the discovery of guards for transitions and conditional branches (choices) in a process [30–32].

4.19 Imprecise Relationships (I19)

This quality issue refers to the scenario in which due to the chosen definition of a case, it is not possible anymore to correlate events in the log to another case type. For example, assume that each case in a log corresponds to an order. For each order, there exist multiple order lines in which a specific product is ordered. Furthermore, for each order line, an action performed for it is saved as part of the corresponding order. Due to the chosen definition of a case, information is lost

about to which order line an event belongs. That is, assume that for each order line the `enter order line` event is always followed by the `secure order line` event. In case two order lines exist, the following four events may be recorded sequentially: `enter order line`, `enter order line`, `secure order line`, and `secure order line`. As there are now multiple order line related events having the same activity name, it is not possible anymore to connotate them with the original order line. As a result, process mining algorithms may discover relations between events which in reality are more fine-grained. For the example this means that it may be inferred that the `enter order line` events occurs in a loop which is not correct as events occur as part of an order line.

4.20 Imprecise Case Attributes (I20)

This quality issue refers to the scenario in which for a case attribute it is not possible to properly use its value as the provided value is too coarse. For example, assume that each case belongs to a patient and that for each patient the age is only given in terms of ten years. As a result, only a very coarse value can be given for the average age of the patients whereas an average age in terms of years is more useful.

4.21 Imprecise Position (I21)

For this specific position quality issue, we again assume that no timestamps have been given for events. So, the ordering of events in the log determines the order in which the events occurred in reality. In this scenario, the ordering of events is imprecise if for some events the ordering is correct but that for some events the ordering is incorrect as they occurred in parallel. Although some events occurred in parallel, they have been recorded sequentially. For example, for an order fulfillment process, the events `prepare route guide` and `estimate trailer usage` have occurred together but in the log it is recorded that event `prepare route guide` occurs after event `estimate trailer usage`. As the ordering of events is not clear within a log, process mining algorithms have problems with discovering the correct control-flow. That is, relations may be inferred which hardly or even do not exist in reality.

4.22 Imprecise Activity Names (I22)

This quality issue corresponds to the scenario in which activity names are too coarse. As a result, within a trace there may be multiple events with the same activity name. These events may have the same connotation as, for example, they occur in a row. Alternatively, the events may have different connotations. That is, the activity corresponding to `Send Acknowledgment` may mean differently depending on the context in which it manifests. In Fig. 8 an example is given. One occurrence of event “A” is immediately followed by another occurrence of event “A”. The two events either belong to the same instance of task “A” or

they belong to separate instances of task "A". The separate instances of task "A" may have the same connotation or a different one.

The effect on the application of process mining is that algorithms have difficulty in identifying the notion of duplicate tasks and thereby produce results that are inaccurate. For example, in process discovery, duplicate tasks are represented with a single node resulting in a large fan-in/fan-out. In case of duplicate events which have the same connotation, a simple filter may suffice in order to aggregate the multiple events into one.

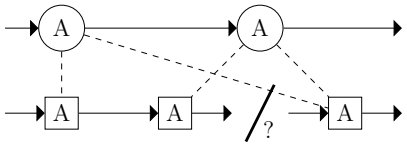

Fig. 8. Imprecise activity names: event "A" occurs two times in succession. As a consequence, both events may belong to the same instance of task "A" or they may each belong to a separate instance of task "A".

4.23 Imprecise Timestamps (I23)

This quality issue corresponds to the scenario where timestamps are imprecise as a too coarse level of abstraction is used for the timestamps of (some of the) events. This implies that the ordering of events within the log may not conform to the actual ordering in which the events occurred in reality, i.e. the ordering of events within a trace is unreliable. In Fig. 9 an example is given of a too coarse granularity of timestamps. For a trace, events "B" and "C" both have "05-12-2011" as the timestamp. It is not clear whether event "B" occurred before "C" or the other way around. For event "A" having timestamp "03-12-2011" it is clear that it occurred before both "B" and "C". Likewise, event "D" occurred after both "B" and "C" as it has the timestamp "06-12-2011".

Process mining algorithms for discovering the control-flow assume that all events within the log are totally ordered. As multiple events may exist within a trace with the same timestamp, process mining algorithms may have problems with identifying the correct control-flow. In particular, discovered control-flow models tend to have a substantial amount of activities which occur in parallel. Furthermore, event logs with coarse granular timestamps are incapacitated for certain types of process mining analysis such as performance analysis.

Another example of imprecise timestamps is concerned with scenarios where the level of granularity of timestamps is not the same across all events in an event log, i.e., there are pairs of events for which the level of granularity of their timestamps is different (e.g., seconds versus days). Fig. 10 exemplifies a mixed granularity of timestamps. For a trace, there is an event "B" with timestamp "05-12-2011" and an event "C" with timestamp "05-12-2012 17:59". It is not

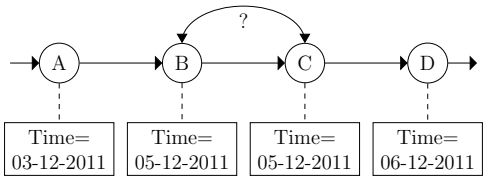

Fig. 9. Coarse granularity of timestamps: for events "B" and "C", it is not clear in which order they occurred as both occurred on the same day.

clear whether event "B" occurred before "C" or the other way around. The effect of event logs with mixed granular timestamps is similar to that of coarse granular event logs.

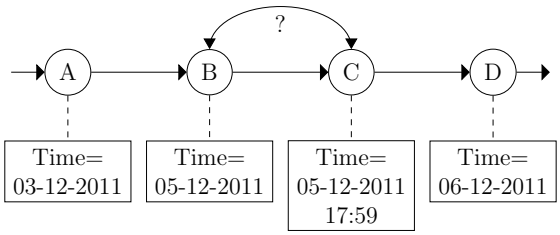

Fig. 10. Mixed granularity of timestamps: for events "B" and "C" it is not clear in which order they occurred as they both occurred on the same day. Furthermore, event "C" has a more fine-grained timestamp than that of event "B".

4.24 Imprecise Resources (I24)

This quality issue refers to the scenario in which for the resource attribute of an event more specific information is known about the resource(s) that performed the activity but that coarser resource information has been recorded. For example, for an event corresponding to the activity `perform X-ray` it is not registered by which radiologist it has been executed. Instead, it is only recorded that the activity has been executed at the radiology department. As a result, process mining algorithms focusing on the discovery of the organizational perspective of a process are limited to inferring relations at the level of detail at which resource information has been recorded.

4.25 Imprecise Event Attributes (I25)

This quality issue refers to the scenario in which for an event attribute it is not possible to properly use its value as the provided value is too coarse. For example, assume that for each case there is an event `pay amount` in which the `amount` attribute records the amount of money that has been paid. However,

the amount of money that is paid is only given in terms of hundreds of euro's. Consequently, for the average amount of euro's that is paid only a very coarse value can be given whereas an average amount in terms of single euro's is more useful.

4.26 Irrelevant Cases (I26)

This quality issue corresponds to the scenario where certain cases in an event log are deemed to be irrelevant for a particular context of analysis. For example, a hospital event log pertaining to treatment of cancer patients in a gynaecology department may contain cases related to different types of cancer. When analyzing the workflow of patients who have been diagnosed with cancer type A, we are only interested in cases pertaining to these patients while cases of all other patients are deemed irrelevant. Considering irrelevant cases will add to the complexity of analysis e.g., irrelevant cases adds to the heterogeneity of event logs that might result in incomprehensible process models using process discovery techniques.

If an event log contains rich information in case attributes indicating the characteristics of cases, we can use that information to filter/remove irrelevant cases, e.g., the hospital event log records the type of cancer in diagnosis code attributes. However, if no such information is available (i.e., missing case attributes), dealing with irrelevant cases becomes a daunting task.

4.27 Irrelevant Events (I27)

In some applications, certain logged events may be irrelevant *as it is* for analysis. As a result, it is needed to filter or aggregate events, in order that only relevant events are derived/obtained. This filtering or aggregation may be a far from trivial task. If the event log contains rich information in the form of event attributes, one may use them to assist in filtering/aggregating irrelevant events.

For example, when analyzing X-ray machine event data, completely different aspects need to be considered when it comes to gaining insights on (i) the real usage of the system and (ii) recurring problems and system diagnosis, the former requiring the analysis of commands/functions invoked on the system while the latter needing error and warning events. As another example, when analyzing the process of an entire hospital, one would be interested in a high-level process flow between the various departments in the hospital. However, when analyzing a particular department workflow, one would be interested in more finer level details on the activities performed within the department.

Failing in filtering and aggregating the correct events creates several challenges for process mining and leads to inaccurate results or capturing results at insufficient level of detail. For example, ignoring critical events required for the context of analysis will lead to not being able to uncover the required results (e.g., unable to discover the causes of failure) while considering more than required will add to the complexity of analysis in addition to generating non-interesting results (e.g., an incomprehensible spaghetti-like model).

5 Evaluation of Event Logs

In this section, we analyze the issues highlighted in Sections 3.1 and 4 against several real-life logs. The objective of this analysis is to provide an insight into the manner and extent to which the identified process related issues and event log issues actually appear in real-life data that is used for process mining. We discuss the results of issues manifested in five real-life logs, viz., Philips Healthcare (P), BPI Challenge 2011 (A) [34], BPI Challenge 2012 (B) [35], Catharina Hospital (C), and CoSeLog (G). Table 2 summarizes the process related issues while Table 3 summarizes the event log issues manifested in these logs. From Table 2

Table 2. Evaluation of process related issues on a selection of real-life logs.

	Philips Healthcare (P)	BPI Challenge 2011 (A)	BPI Challenge 2012 (B)	Catharina Hopsital (C)	CoSeLog (G)
Event granularity	X	X		X	X
Voluminous data	X				
Case heterogeneity	X	X	X	X	
Process flexibility and concept drifts				X	X

Table 3. Evaluation of event log issues on a selection of real-life logs. P: Philips Healthcare, A: BPI Challenge 2011, B: BPI Challenge 2012, C: Catharina Hospital, G: CoSeLog.

	case	**event**	**belongs to**	**c_attribute**	**position**	**activity name**	**timestamp**	**resource**	**e_attribute**
missing		{A,B,C,G}		{C}				{A,B}	{C,G}
incorrect		{B,G}			{P}		{A,P,C}	{B}	
imprecise					{A }	{A,P,C,G}	{A,C}		
irrelevant	{P}	{A,B,P}							

it becomes clear that each process related issue is manifested in at least one log. From Table 3 it can be seen that 12 out of the 27 event log issues are manifested in one or more event logs. Furthermore, as suggested by the results, the process characteristics "fine-granular events" and "case heterogeneity", and the event log data quality issues "missing event" and "imprecise activity name" occur most frequently for event logs. Additionally, it can be seen that timestamp related issues occur within three event logs. Such issues have a severe impact on

the correctness of the obtained process mining results. We discuss each of these event logs and the issues manifested in detail in the following subsections.

5.1 X-ray Machine Event Logs of Philips Healthcare

Philips Healthcare has enabled the monitoring of its medical equipment (X-ray machines, MR machines, CT scanners, etc.) across the globe. Each of these systems records event logs capturing the operational events during its usage. Philips is interested in analyzing these event logs to understand the needs of their customers, identify typical use case scenarios (to test their systems under realistic circumstances), diagnose problems, service systems remotely, detect system deterioration, and learn from recurring problems.

Event logs from Philips healthcare are huge and tend to be fine-granular, heterogeneous, and voluminous. The event logs generated by X-ray machines are a result of write statements inserted in the software supporting the system. A typical event log contains hundreds of event classes (activities). Each Cardio-Vascular (CV) X-ray machine of Philips Healthcare logs around 350 KB of event data in compressed format (5 MB in uncompressed format) every day. Currently Philips has event logs from a few thousand systems installed across the globe. This implies that Philips stores around 875 MB of compressed data with (comprising of millions of events) every day. Furthermore, since X-ray machines are typically designed to be quite *flexible* in their operation. This results in event logs containing a heterogeneous mix of usage scenarios with more diverse and less structured behavior. For example, physicians could perform several different types of cardiac surgery procedures (e.g., bypass surgery, stent procedure, angioplasty, etc.) on a patient using a cardio-vascular X-ray machine. Moreover, different physicians might apply a particular medical procedure in different ways. All these lead to a huge diversity in event logs.

In addition, we observe the following data quality issues in these event logs:

- **Incorrect Timestamps (I16)**: An X-ray machine has dozens of components with each component having a local clock and a local buffer. There could be a mismatch between the times when an event is actually triggered and when an event is recorded in a log (an event is first queued in the internal buffer of a component before it is logged). Furthermore, there are scenarios where the various components in an X-ray machine are not synchronized on clock. These factors lead to incorrect timestamps for logged events.
- **Irrelevant Case (I26)**: X-ray machine event logs also suffer from a lack of proper *scope*. All events that happened on/within the system on a given day is recorded as a log. Different aspects of the log need to be analyzed for different purposes. A proper definition of a case need to be identified based on the contexts and purpose of analysis. For example, the events between startup and shutdown of an X-ray machine define one instance of system operation.
- **Irrelevant Events (I27)**: Not all events logged by the system are necessary for gaining insights on certain aspects of a system. For example, in order to

identify any patterns that manifest during a failure of a particular component/part in a system, we are interested in only error/warning events and not on commands. As another example, in order to analyze how a field service engineer works on a machine during fault diagnosis, we need to consider events logged only by the engineer. Philips records several attributes for each event which makes it possible for defining and selecting an appropriate scope. Use of domain knowledge is essential in this process.

5.2 The 2011 BPI Challenge Event Log–Treat Procedures of Cancer Patients

We now discuss some of the data quality issues identified in another real-life log, provided for the 2011 BPI challenge, from a large Dutch academic hospital [34]. The event log contains 1143 cases and 150,291 events distributed over 624 activities related to the activities pertaining to the treatment procedures that are administered on patients in the hospital. Several data issues highlighted in the paper manifest in the log.

The event log contains a heterogeneous mix of patients diagnosed for cancer (at different stages of malignancy) pertaining to the cervix, vulva, uterus, and ovary. Analyzing the event log in its entirety generates an incomprehensible spaghetti-like model [2]. In addition, the event log exhibits the following data quality issues:

- **Missing Resources (I8)**: The event log contains several events with missing information. For example, 16 events do not contain the laboratory/department information (which constitute the resource perspective of the process) in which a medical test pertaining to the event is performed.
- **Missing Event Attributes (I9)**: The event log contains several events with missing values for event attributes. For example, the event log contains 16 attributes pertaining to diagnosis code for each event (diagnosis code:0, diagnosis code:1, ..., diagnosis code:15). However, the values for all of these attributes are not specified for events.
- **Imprecise Activity Names (I22)**: The activities in the log exhibit mixed granularity. For example, there are coarse grained activities such as the administrative tasks and fine-grained activities such as a particular lab test. Furthermore, there are a few duplicate tasks in the event log, e.g., geb.antistoffen tegen erys - dir.coombs and geb. antistoffen tegen erys - dir. coomb (one is specified as singular and the other is plural), natrium vlamfotometrisch - spoed and natrium - vlamfotometrisch - spoed (the difference between the two is a hyphen), etc.
- **Imprecise Timestamps (I23)**: The event log suffers from several timestamp related problems. Timestamps are recorded at the granularity of a day for each event. This creates a loss of information on the exact timing and ordering of events as executed in the process.
- **Irrelevant Events (I27)**: The event log also suffers from the scoping issue. Each trace in the log contains activities performed in different departments (often concurrently) and at different instances of time over multiple visits to

the hospital. Appropriate scope of the trace/log is to be considered based on the context of analysis, e.g., if a particular department is interested in analyzing its process, then only a subset of events pertaining to that department needs to be considered.

5.3 The 2012 BPI Challenge Event Log–Loan/Overdraft Application Process

Our next log is the one provided for the 2012 BPI Challenge pertaining to the handling of loan/overdraft applications in a Dutch financial institute [35]. The event log contains $13,087$ cases and $262,200$ events distributed over 36 activities having timestamps in the period from 1-Oct-2011 to 14-Mar-2012. The overall loan application process can be summarized as follows: a submitted loan/overdraft application is subjected to some automatic checks. The application can be declined if it does not pass any checks. Often additional information is obtained by contacting the customer by phone. Offers are sent to eligible applicants and their responses are assessed. Applicants are contacted further for incomplete/missing information. The application is subsequently subjected to a final assessment upon which the application is either approved and activated, declined, or cancelled.

The event log contains a heterogeneous mix of cases. One can define several classes of cases in the event log. For examples, cases pertaining to applications that have been approved, declined, and cancelled, cases that have been declined after making an offer, cases that have been suspected for fraud, etc. In addition, the log exhibits the following data quality issues.

- **Missing Events (I2)**: The event log contains certain loan/overdraft applications that have started towards the end of the recorded time period. Such applications are yet to be completed leading to partial/incomplete traces (overall there are 399 cases that are incomplete). Furthermore, the event log contains 1042 traces where an association between the start of an activity and a completion of an activity is missed.
- **Missing Resources (I8)**: The event log contains missing resource information for several events. For example, there are 18009 events across 3528 traces that have missing resource information, i.e., 6.86% of events and 26.96% of the traces have partially missing resource information.
- **Incorrect Events (I11)**: The event log contains several incorrect events, which can be regarded as potential outliers. For example, there are some traces where certain activities are executed even after an application is cancelled or declined.
- **Incorrect Resources (I17)**: The event log contains some events with potentially incorrect resource information. For example, there are three cases where a loan has been approved by an automated resource. It is highly unlikely for the loan to have been approved by automated resources.
- **Irrelevant Events (I27)**: The event log contains events pertaining to three concurrent subprocesses viz., application, offer, and contact customers. If an

analyst is interested in getting insights on a particular subprocess, the events pertaining to other subprocesses are deemed to be irrelevant.

5.4 Catharina Hospital Event Log–Treatment Procedures in Intensive Care Unit

The next event log that will be discussed is describing the various activities that take place in the Intensive Care Unit (ICU) of the Catharina hospital. It contains records mainly related to patients, their complications, diagnosis, investigations, measurements, and characteristics of patients clinical admission. The data belongs to 1308 patients for which 739 complications, 11484 treatments, 3498 examinations, 17775 medication administrations, and 21819 measurements have been recorded. Each action performed has been entered manually into the system. For the log the following data quality problems apply.

The event log contains a heterogeneous mix of cases. This is made clear by the fact that for the 1308 patients in the log there in total 16 different main diagnoses (e.g. aftercare cardiovascular surgery). For these main diagnoses there exist in total 218 different indications for why a patient is admitted to the ICU (e.g. a bypass surgery). Here, a patient may even have multiple indications. Also, the log tends to be fine-granular. That is, for a patient it is recorded which fine-grained lab tests have been performed whereas it is also recorded which less fine-grained examinations have been performed. Finally, for patients admitted at an ICU it can be easily imagined that many different execution behaviors are possible. For example, for the group of 412 patients which received care after the heart surgery, we discovered the care process via the Heuristics miner which can deal with noise. For the discovery, we only focused on the treatments and the *c*omplete event type, but still the obtained model was spaghetti-like as it contained 67 nodes and more than 100 arcs. Clearly, the log is suffering from the *m*omentary change process characteristic problem.

In addition, we observe the following data quality issues in the event log:

- **Missing Events (I2)**: Certain events in the log are recorded manually by the resources performing the action on the patients. As a result, the logging is not done in a structured way e.g., events capturing the state of an activity/action such as scheduled, started, and completed are not logged for certain activities. In particular, for 67% of the actions, no complete event has been registered. As another example, the log contains only data only for the year 2006. Therefore, for some patients the events capturing the actions performed at the start or the end of the stay at the ICU are missing.
- **Missing Case Attributes (I4)**: The event log contains several attributes that give additional information on the context of each case. However, these attributes are not always present in each of the cases. For example, for 1229 patients the referring specialist is not given and for 364 patients the main diagnosis at discharge is missing.
- **Missing Event Attributes (I9)**: The event log contains also several attributes that give additional information on the context of each event. In

particular, for the blood test events, there are 20 attributes pertaining to the outcome of the test. For several attributes often no value has been filled in. For example, for the `partial pressure of venous carbon dioxide (PvCO2)` test, for 1543 of the 9486 events no value has been filled in making it unclear whether the test has been performed or not.

- **Incorrect Timestamps (I16)**: Due to manual recording of actions within the system, typically a user records at the same time that a series of actions have been completed. As a result, these actions have been completed in the same millisecond whereas in reality they have been completed at different times.
- **Imprecise Activity Names (I22)**: Within the event log there are several examples of the duplicate recording of an action. For example, one character of a Foley catheter has been written 906 times with a capital (Catheter a Demeure) and 185 times without a capital (Cathether a demeure).
- **Imprecise Timestamps (I23)**: For some specific class of treatments it is only recorded on which day they have taken place whereas for all the other actions it is recorded in terms of milliseconds when they have taken place.
- **Irrelevant Cases (I26)**: As already indicated earlier for the 1308 patients in the log there in total 16 different main diagnoses For example, there are 434 patients with diagnosis `aftercare cardiac surgery` and 180 patients with diagnosis `aftercare general surgery`. For each main diagnosis a different treatment process is followed and thus needs to be analyzed in isolation. This as consequence that the event log also suffers from a lack of proper scope.

5.5 Event Log from a Large Dutch Municipality–Building Permit Process

Our last real-life log corresponds to one of the processes in a large Dutch municipality. The process pertains to the procedure for granting permission for projects like the construction, alteration or use of a house or building, etc., and involves some submittal requirements, followed by legal remedies procedure and enforcement. The event log contains 434 cases and 14562 events distributed across 206 activities for the period between Aug 26, 2010 and Jun 09, 2012.

The events in this log are too fine-grained. This is evident from a large number (206) of distinct activities. Analysts are mostly interested in high-level abstract view of the process without being bothered about the low-level activities. The process corresponding to this event log has undergone three (evolutionary) changes during the time period of the log and this is manifested as concept drift [23]. In addition, several of the data quality issues can be observed even in this log.

- **Missing Events (I2)**: The event log contains 197 cases that are still running (incomplete). This leads to the situation that there are missing events for some traces in the log.
- **Missing Event Attributes (I9)**: The event log contains several attributes that give additional information on the context of each event. However, these

attributes are not always present in all the events. In other words, there are several events where we see missing information.
- **Incorrect Events (I12)**: The event log also contains several cases with exceptional execution behavior. These are often manifested as an extraneous execution of some activities thereby leading to incorrect events.
- **Imprecise Activity Names (I22)**: The event log also exhibits activity names of mixed granularity. Since the event log emanates from a hierarchical process, we see events with activity names from all levels of hierarchy in the same event log.

6 Related Work

It is increasingly understood that data in data sources is often "dirty" and therefore needs to be "cleansed" [36]. In total, there exist five general taxonomies which focus on classifying data quality problems [37]. Although each of these approaches construct and sub-divide their taxonomies quite differently [36, 38–41], they arrive at very similar findings [36]. Alternatively, as data quality problems for time-oriented data have distinct characteristics, a taxonomy of dirty time-oriented data is provided in [37]. Although some of the problems within the previous taxonomies are very similar to process mining data quality problems, the taxonomies are not specifically geared towards the process mining domain.

So far, process mining has been applied in many organizations. In literature, several scholarly publications can be found in which an application of process mining has been described. Several publications mention the need of event log preprocessing as data quality problems exist. For example, the healthcare domain is a prime example of a domain where event logs suffer from various data quality problems. In [42, 43] the gynecological oncology healthcare process within an university hospital has been analyzed; in [44] several processes within an emergency department have been investigated; in [45] all Computer Tomography (CT), Magnetic Resonance Imaging (MRI), ultrasound, and X-ray appointments within a radiology workflow have been analyzed; in [46] the activities that are performed for patients during hospitalization for breast cancer treatment are investigated; in [47] the journey through multiple wards has been discovered for inpatients; and finally, in [48], the workflow of a laparoscopic surgery has been analyzed. In total, these publications indicate problems such as "missing values", "missing events", "duplicate events", "evolutionary change", "momentary change", "fine-granular events", "case heterogeneity", "noisy data / outliers", and "scoping". In particular, "case heterogeneity" is mentioned as a main problem. This is due to the fact that healthcare processes are typically highly dynamic, highly complex, and ad hoc [44].

Another domain in which many data quality problems for the associated event logs can be found is Enterprise Resource Planning (ERP). Here, scholarly publications about the application of process mining have been published about a procurement and billing process within SAP R/3 [19]; a purchasing

process within SAP R/3; and the process of booking gas capacity in a gas company [49]. The data quality problems arising here concern "fine-granular events", "voluminous data", and "scoping". In particular, the identification of relationships between events together with the large amounts of data that can be found within an ERP system are considered as important problems.

7 Conclusions

Process mining has made significant progress since its inception more than a decade ago. The huge potential and various success stories have fueled the interest in process mining. However, despite an abundance of process mining techniques and tools, it is still difficult to extract the desired knowledge from raw event data. Real-life applications of process mining tend to be demanding due to process related issues manifested in event logs and data quality issues. In this paper we highlighted several classes of data quality issues manifested in event logs. These issues hamper the applicability of several process mining techniques and limit the level/quality of insights gained. There is a pressing need for process mining research to also focus on techniques that address these quality issues.

Acknowledgements

This research is supported by the Dutch Technology Foundation STW, applied science division of NWO and the Technology Program of the Ministry of Economic Affairs.

References

1. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer-Verlag, Berlin (2011)
2. Bose, R.P.J.C., van der Aalst, W.M.P.: Analysis of Patient Treatment Procedures: The BPI Challenge Case Study. Technical Report BPM-11-18, BPMCenter.org (2011)
3. Bose, R.P.J.C., van der Aalst, W.M.P.: Process Mining Applied to the BPI Challenge 2012: Divide and Conquer While Discerning Resources. Technical Report BPM-12-16, BPMCenter.org (2012)
4. Bose, R.P.J.C., van der Aalst, W.M.P.: Analysis of Patient Treatment Procedures. In Daniel, F., Barkaoui, K., Dustdar, S., eds.: Business Process Management Workshops. Volume 99 of Lecture Notes in Business Information Processing. (2012) 165–166
5. Bose, R.P.J.C., van der Aalst, W.M.P.: Process Mining Applied to the BPI Challenge 2012: Divide and Conquer While Discerning Resources. In Rosa, M.L., Soffer, P., eds.: Business Process Management Workshops. Volume 132 of Lecture Notes in Business Information Processing. (2013) 221–222
6. IEEE Task Force on Process Mining: Process Mining Manifesto. In Daniel, F., Dustdar, S., Barkaoui, K., eds.: BPM 2011 Workshops. Volume 99 of Lecture Notes in Business Information Processing., Springer-Verlag, Berlin (2011) 169–194

7. Rogers, S.: Big Data is Scaling BI and Analytics–Data Growth is About to Accelerate Exponentially–Get Ready. Information Management-Brookfield **21**(5) (2011) 14
8. Olofson, C.W.: Managing Data Growth Through Intelligent Partitioning: Focus on Better Database Manageability and Operational Efficiency with Sybase ASE. White Paper, IDC, Sponsored by Sybase, an SAP Company (November, 2010)
9. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H.: Big Data: The Next Frontier for Innovation, Competition, and Productivity. Technical report, McKinsey Global Institute (2011)
10. Song, M., Günther, C.W., van der Aalst, W.M.P.: Trace Clustering in Process Mining. In Ardagna, D., Mecella, M., Yang, J., eds.: Business Process Management Workshops. Volume 17 of Lecture Notes in Business Information Processing., Springer-Verlag, Berlin (2009) 109–120
11. Weerdt, J.D., vanden Broucke, S.K.L.M., Vanthienen, J., Baesens, B.: Leveraging Process Discovery with Trace Clustering and Text Mining for Intelligent Analysis of Incident Management Processes. In: 2012 IEEE Congress on Evolutionary Computation (CEC). (2012) 1–8
12. de Medeiros, A.K.A., Guzzo, A., Greco, G., van der Aalst, W.M.P., Weijters, A.J.M.M., van Dongen, B.F., Sacca, D.: Process Mining Based on Clustering: A Quest for Precision. In ter Hofstede, A.H.M., Benatallah, B., Paik, H., eds.: Business Process Management Workshops. Volume 4928 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2008) 17–29
13. Greco, G., Guzzo, A., Pontieri, L., Sacca, D.: Discovering Expressive Process Models by Clustering Log Traces. IEEE Transactions on Knowledge and Data Engineering **18**(8) (2006) 1010–1027
14. Bose, R.P.J.C., van der Aalst, W.M.P.: Context Aware Trace Clustering: Towards Improving Process Mining Results. In: Proceedings of the SIAM International Conference on Data Mining (SDM). (2009) 401–412
15. Bose, R.P.J.C., van der Aalst, W.M.P.: Trace Clustering Based on Conserved Patterns: Towards Achieving Better Process Models. In: Business Process Management Workshops. Volume 43 of LNBIP., Springer (2010) 170–181
16. Görg, C., Pohl, M., Qeli, E., Xu, K.: Visual Representations. In Kerren, A., Ebert, A., Meye, J., eds.: Human-Centered Visualization Environments. Volume 4417 of Lecture Notes in Computer Science. Springer-Verlag, Berlin (2007) 163–230
17. C. Pedrinaci and J. Domingue: Towards an Ontology for Process Monitoring and Mining. In M.Hepp, Hinkelmann, K., Karagiannis, D., Klein, R., Stojanovic, N., eds.: Semantic Business Process and Product Lifecycle Management. Volume 251., CEUR-WS.org (2007) 76–87
18. de Medeiros, A.K.A., van der Aalst, W.M.P., Carlos, P.: Semantic Process Mining Tools: Core Building Blocks. In Golden, W., Acton, T., Conboy, K., van der Heijden, H., Tuunainen, V.K., eds.: Proceedings of the 16^{th} European Conference on Information Systems (ECIS 2008). (2008) 1953–1964
19. Günther, C.W., Rozinat, A., van der Aalst, W.M.P.: Activity Mining by Global Trace Segmentation. In Rinderle-Ma, S., Sadiq, S., Leymann, F., eds.: Business Process Mangement Workshops. Volume 43 of Lecture Notes in Business Information Processing., Springer-Verlag, Berlin (2010) 128–139
20. Günther, C.W., van der Aalst, W.M.P.: Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics. In: International Conference on Business Process Management (BPM 2007). Volume 4714 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2007) 328–343

21. Bose, R.P.J.C., Verbeek, E.H.M.W., van der Aalst, W.M.P.: Discovering Hierarchical Process Models Using ProM. In Nurcan, S., ed.: CAiSE Forum 2011. Volume 107 of Lecture Notes in Business Information Processing., Springer-Verlag, Berlin (2012) 33–48
22. Bose, R.P.J.C., van der Aalst, W.M.P.: Abstractions in Process Mining: A Taxonomy of Patterns. In Dayal, U., Eder, J., Koehler, J., Reijers, H., eds.: Business Process Management. Volume 5701 of LNCS., Springer-Verlag (2009) 159–175
23. Bose, R.P.J.C., van der Aalst, W.M.P., Žliobaitė, I., Pechenizkiy, M.: Handling Concept Drift in Process Mining. In Mouratidis, H., Rolland, C., eds.: International Conference on Advanced Information Systems Engineering (CAiSE 2011). Volume 6741 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2011) 391–405
24. Carmona, J., Gavaldà, R.: Online Techniques for Dealing with Concept Drift in Process Mining. In Hollmén, J., Klawonn, F., Tucker, A., eds.: International Conference on Intelligent Data Analysis (IDA 2012). Volume 7619 of Lecture Notes in Computer Science. (2012) 90–102
25. D Luengo, M.S.: Applying Clustering in Process Mining to Find Different Versions of a Process that Changes Over Time. In Florian Daniel, K.B., Dustdar, S., eds.: Business Process Management Workshops. Volume 99 of Lecture Notes in Business Information Processing., Springer-Verlag, Berlin (2012) 153–158
26. Stocker, T.: Time-Based Trace Clustering for Evolution-Aware Security Audits. In Florian Daniel, K.B., Dustdar, S., eds.: Business Process Management Workshops. Volume 100 of Lecture Notes in Business Information Processing., Springer-Verlag, Berlin (2012) 471–476
27. Buijs, J., Dongen, B., van der Aalst, W.M.P.: A Genetic Algorithm for Discovering Process Trees. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2012), Piscataway: IEEE (2012) 1–8
28. Rogge-Solti, A., Mans, R., van der Aalst, W.M.P., Weske, M.: Reparing Event Logs Using Timed Process Models (2012)
29. Song, M.S., van der Aalst, W.M.P.: Towards Comprehensive Support for Organizational Mining. Decision Support Systems **46**(1) (2008) 300–317
30. Rozinat, A., van der Aalst, W.M.P.: Decision Mining in ProM. In Dustdar, S., Fiadeiro, J.L., Sheth, A.P., eds.: Business Process Management. (2006) 420–425
31. de Leoni, M., van der Aalst, W.M.P.: Data-Aware Process Mining: Discovering Decisions in Processes Using Alignments. In: Proc. of the 28th ACM symposium on Applied Computing (SAC'13), ACM (2013)
32. de Leoni, M., Dumas, M., Garcia-Bañuelos, L.: Discovering branching conditions from business process execution logs. In: Proc. of 16th International Conference on Fundamental Approaches to Software Engineering (FASE 2013). Volume 7793 of LNCS., Springer (2013)
33. Bose, R.P.J.C.: Process Mining in the Large: Preprocessing, Discovery, and Diagnostics. PhD thesis, Eindhoven University of Technology (2012)
34. 3TU Data Center: BPI Challenge 2011 Event Log (2011) doi:10.4121/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54.
35. 3TU Data Center: BPI Challenge 2012 Event Log (2011) doi:10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f.
36. Kim, W., Choi, B.J., Hong, E.K., Kim, S.K., Lee, D.: A Taxonomy of Dirty Data. Data Mining and Knowledge Discovery **7** (2003) 81–99
37. Gschwandtner, T., Gärtner, J., Aigner, W., Miksch, S.: A Taxonomy of Dirty Time-Oriented Data. In et al., G.Q., ed.: CD-ARES 2012. Volume 7465 of Lecture Notes in Computer Science. (2012) 58–72

38. Rahm, E., Do, H.: Data Cleaning: Problems and Current Approaches. IEEE Bulletin of the Technical Committee on Data Engineering **24**(4) (2000)
39. Müller, H., Freytag, J.C.: Problems, Methods, and Challenges in Comprehensive Data Cleansing. Technical report hub-ib-164, Humboldt University Berlin (2003)
40. Oliveira, P., Rodrigues, F., Henriques, P.: A Formal Definition of Data Quality Problems. In: International Conference on Information Quality (MIT IQ Conference). (2005)
41. Barateiro, J., Galhardas, H.: A Survey of Data Quality Tools. Datenbankspectrum **14** (2005) 15–21
42. Mans, R.S., Schonenberg, M.H., Song, M.S., van der Aalst, W.M.P., Bakker, P.J.M.: Application of Process Mining in Healthcare : a Case Study in a Dutch Hospital. In Fred, A., Filipe, J., Gamboa, H., eds.: Biomedical engineering systems and technologies (International Joint Conference, BIOSTEC 2008, Funchal, Madeira, Portugal, January 28-31, 2008, Revised Selected Papers). Volume 25 of Communications in Computer and Information Science., Springer-Verlag, Berlin (2009) 425–438
43. Mans, R.: Workflow Support for the Healthcare Domain. PhD thesis, Eindhoven University of Technology (June 2011) See http://www.processmining.org/blogs/pub2011/workflow_support_for_the_healthcare_domain.
44. Rebuge, A., Ferreira, D.: Business Process Analysis in Healthcare Environments: A Methodology Based on Process Mining. Information Systems **37**(2) (2012) 99–116
45. Lang, M., Bürkle, T., Laumann, S., Prokosch, H.U.: Process Mining for Clinical Workflows: Challenges and Current Limitations. In: Proceedings of MIE 2008. Volume 136 of Studies in Health Technology and Informatics., IOS Press (2008) 229–234
46. Poelmans, J., Dedene, G., Verheyden, G., van der Mussele, H., Viaene, S., Peters, E.: Combining Business Process and Data Discovery Techniques for Analyzing and Improving Integrated Care Pathways. In: Proceedings of ICDM'10. Volume 6171 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2010) 505–517
47. Perimal-Lewis, L., Qin, S., Thompson, C., Hakendorf, P.: Gaining Insight from Patient Journey Data using a Process-Oriented Analysis Approach. In: HIKM 2012. Volume 129 of Conferences in Research and Practice in Information Technology., Australian Computer Society, Inc. (2012) 59–66
48. Blum, T., Padoy, N., Feuner, H., Navab, N.: Workflow Mining for Visualization and Analysis of Surgeries. International Journal of Computer Assisted Radiology and Surgery **3** (2008) 379–386
49. Maruster, L., Beest, N.: Redesigning Business Processes: a Methodology Based on Simulation and Process Mining Techniques. Knowledge and Information Systems **21**(3) (2009) 267–297