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 growing 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 quality of the events logs used as input is critical for the success of any process mining effort. In this paper, we identify ten categories of data quality issues, e.g., problems related to timestamps in event logs, event granularity, ambiguity of activity names, and mashed logging are described in detail. The systematic identification and analysis of these problems calls for a consolidated effort from the process mining community. Five real-life event logs are analyzed to illustrate the omnipresence data quality issues. We hope that these findings will encourage systematic logging approaches (to prevent data quality problems) and repair techniques (to alleviate data quality problems).
Key words: Process Mining, Data Quality, Event Log, Preprocessing, Data Cleansing, Outliers

1 Introduction

⟨TODO: Abstract has been added. Various parts have been rewritten. In other parts only minor changes.⟩

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 event data recorded by physical devices, web services, ERP 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, 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 *fine-granular*, *heterogeneous*, *voluminous*, *incomplete*, and *noisy*. 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 before applying process mining algorithms. The strongest contributions addressing the 'Business Process Intelligence Challenge' event logs illustrate the significance of log preprocessing [2–4].

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 data quality issues that we encounter in event logs and their impact on process mining. We describe ten categories of data quality problems. For example, there is a category describing three timestamp related issues: (1) course granular timestamps (e.g., logs having just date information such that ordering of events on the same day is unknown), (2) mixed granular timestamps (i.e., event logs that have timestamps with different levels of precision, e.g., milliseconds, seconds, minutes, days), 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 rest of the paper is organized as follows. Section 2 elicits a list of common data quality issues grouped in ten categories. In Section 3, we analyze five real-life logs and evaluate data quality problems using the classification presented in Section 2. Related work is presented in Section 4. Finally, conclusions are presented in Section 5.

⟨TODO: The paper has two weaknesses. First of all, the paper is about event logs, but a rigorous definition is missing. Terminology is changing constantly: trace, case, process instance, etc. Second, the paper only points out problems and not solutions. Only hints are given for solutions. I would also have put more emphasis on prevention: how to get level 5 logs. Given the space problems I would not know how to address these issues.⟩

2 Categories of Process Mining Data Quality Problems

In this section, we present a comprehensive overview of data quality problems related to event logs to be used for process mining. We identify ten broad (not necessarily orthogonal) classes of data quality issues. Each potential quality issue is described in detail using a standard format that includes: a description of the problem and the way it manifests in a log, examples, and its impact on the application of process mining.

⟨TODO: Discussion: why are the categories not orthogonal? Using a formal definition of the log, one could do this I think. See my Visio file to start the discussion⟩

2.1 Event Granularity

Description. Event logs are often fine-grained and too detailed for most stakeholders. The granularity at which events are logged varies widely (across domains/applications) without considering the desired levels of analysis. Events in an event log are often at very different levels of granularity. This problem is due to the lack of good standards and guidelines for logging. 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.

Example. 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.

Effect. 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 [7]. This problem is compounded if the graph is dense (which is often the case in unstructured processes) thereby compromising the comprehensibility of models. Several attempts have been reported in the literature on grouping events to create higher levels of abstraction ranging from the use of semantic ontologies [8, 9] to the grouping

of events based on correlating activities [10, 11] or the use of common patterns of execution manifested in the log [12, 13]. 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).

2.2 Case Heterogeneity

Description. 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. Although it is desirable to also record the scenario chosen for a particular case, it is infeasible to define all possible variants. 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.
Example. 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.
Effect. Process mining techniques have problems when dealing with heterogeneity in event logs. For example, process discovery algorithms produce spaghetti-like incomprehensible process models. 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. Analyzing the whole event log in the presence of heterogeneity fails to achieve this objective. Trace clustering has been shown to be an effective way of dealing with such heterogeneity [14–19]. 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.

2.3 Voluminous Data

Description. 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 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).
Example. 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 [20]. 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 [20], 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 [21]. The term "big data" has emerged to refer to the spectacular growth of digitally recorded data [22].

Effect. 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.

2.4 Timestamp Related Issues

For the application of process mining, correct and precise knowledge about the time at which events occurred is important. However, in practice we often find timestamp-related problems. We distinguish three types of problems.

- **Coarse Granular Timestamps**
  Description. This corresponds to the scenario where a coarse level of granularity is used for the timestamps of 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. For example, the ordering of multiple events on the same day may be lost due to events having only date information.
  Example. Fig. 1 exemplifies 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".
  Effect. 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.
- **Mixed Granular Timestamps**
  Description. The level of granularity of timestamps does not need to be 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

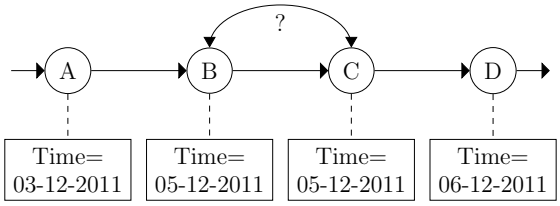

Fig. 1. Coarse granularity of timestamps: for events "B" and "C", it is not clear in which order they occurred as both occurred on the same day.

days).

Example. Fig. 2 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 clear whether event "B" occurred before "C" or the other way around.

Effect. The effect of event logs with mixed granular timestamps is similar to that of coarse granular event logs. For events having a timestamp with a coarse level of granularity, the precise ordering may not be clear in case of events which have a similar but more fine-grained timestamp. As such, process mining algorithms have problems discovering the correct control-flow for these events, e.g., sequential activities are modeled in parallel.

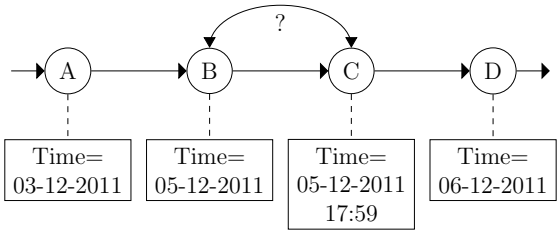

Fig. 2. Mixed granularity of timestamps: for events "B" and "C" it is not clear in which order they occurred as they both occurred on the same day. Furthermore, event "C" has a more fine-grained timestamp than that of event "B".

– **Incorrect Timestamps**

Description. This corresponds to the scenario where the recorded timestamp of (some or all) events in the log does not correspond to the real time at which the event has occurred.

Example. Fig. 3(a) exemplifies 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.

Effect. Process mining algorithms discover the control-flow based on behav-

ior observed in the log. However, due to incorrect timestamps, the discovered control-flow relations may be unreliable or even incorrect. Moreover, in applications such as the discovery of signature patterns for diagnostic purposes (e.g., fraudulent claims, fault diagnosis, etc.) [23], there is a danger of reversal of *cause* and *effect* phenomena due to incorrect timestamps. Fig. 3(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".

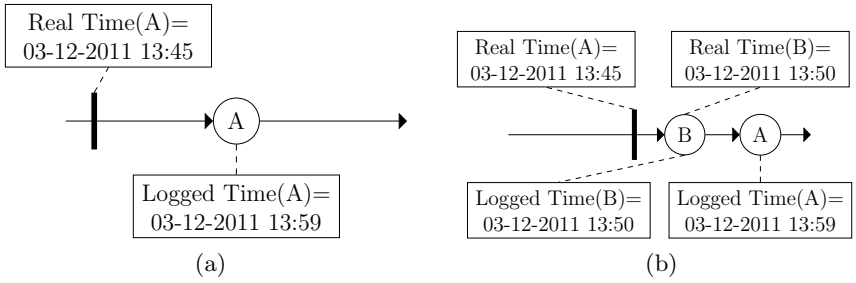

Fig. 3. Incorrect timestamps: (a) event "A" occurred in reality at "03-12-2011 13:45", but the timestamp of the event recorded in the log is "03-12-2011 13:59", (b) as a result one can wrongly infer that "B" causes "A" where in reality "A" occurred before "B".

2.5 Missing Data

Within a log, different kinds of information can be missing. We distinguish three different types of missing information.

– **Missing Attribute Values**
Description. This corresponds to the scenario where in the event log certain attributes are missing or there are attributes that have no value. Such attributes can either belong to a trace (e.g. the identifier of the case or the name of the customer etc.) or an event (e.g. the name of the task to which the event refers to or the timestamp of the event etc.).
Example. Fig. 4 exemplifies missing data for both an event and a trace. For the event, no information has been given about the associated task and the time at which it occurred. For the trace, an unique identifier is missing.
Effect. Event logs with missing attributes/values hinder the application of certain process mining algorithms. For example, control-flow discovery algorithms are impacted by missing timestamps or task information whereas techniques that analyze the organizational-perspective are affected if information is missing about the actor/resource that handled an event. In order to deal with missing values, affected events or traces may be removed from the event log. This may have as counter effect that too little information is remaining in the log in order to obtain reliable results.

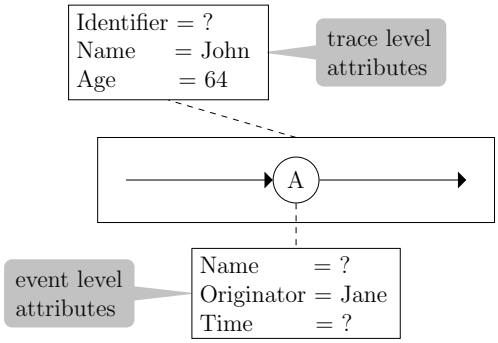

Fig. 4. Missing Values: for event "A" the name of the associated task and the time at which the event occurred are missing. For the trace no unique identifier has been given.

– **Missing Events (Anywhere in a Trace)**
Description. This corresponds to the scenario where some events may be missing anywhere within the trace although they occurred in reality.
Example. Fig. 5 exemplifies 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).
Effect. As events may be missing, 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.

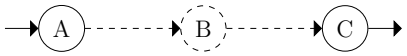

Fig. 5. Missing events (anywhere in a trace): events "A", "B", and "C" have occurred in reality. Only events "A" and "B" are included in the log whereas event "B" is not included.

– **Partial / Incomplete Traces**
Description. This is a special case of missing events where the prefix and/or suffix events corresponding to a trace is 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 started between Jan'12 and Jun'12 but not yet completed by Jun'12 are incomplete and have their suffixes missing.
Example. 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.
Effect. For some algorithms there may be problems with the produced result as different relations may be inferred. In particular, different relations may be inferred for the start or end of a process. However, there are algorithms which can deal with noise (e.g. the fuzzy miner [11] and the heuristics miner [1]). Alternatively, partial / incomplete traces may be filtered from the log. This may have as counter effect that too less information is remaining in the log in order to obtain reliable results.

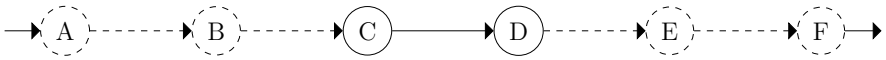

Fig. 6. Partial / incomplete traces: events “A”, “B”, “C”, “D”, “E”, and “F” have occurred in reality. Only events “C” and “D” are included in the log whereas the prefix with events “A” and “B” and the suffix with events “E” and “F” are not included.

2.6 Ambiguity Between Events

– **Duplicate Events**
Description. This corresponds to the scenario where multiple events have 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. For example, the activity corresponding to `Send Acknowledgment` may mean differently depending on the context in which it manifests.
Example. Fig. 7 exemplifies duplicate events within a trace. 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.
Effect. Process mining 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.
– **Overlapping Activity Executions**
Description. This corresponds to the scenario where an instance of an activity is started and before it is completed another instance of the same activity is started.
Example. Fig. 8 exemplifies overlapping instances of execution of an activity. First, an instance of task “A” is started for which event “A (start)” is recorded. Afterwards, another instance of task “A” is started leading to the recording of second event “A (start)”. Next, both instances of task “A” are

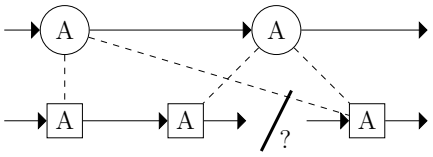

Fig. 7. Duplicate events: event "A" occurs two times in succession. As a consequence, both events may belong to the same instance of task "A" or they may each belong to separate instances of task "A".

completed represented by two recordings of event "A (complete)". As a consequence, there is an ambiguity in associating the complete events with their corresponding start events.

Effect. As there is an ambiguity in associating the complete events with their corresponding start events, process mining algorithms have problems with deciding when a certain instance of an activity is completed. For instance, when calculating the performance of a process, faulty statistics may be obtained for the duration of an activity.

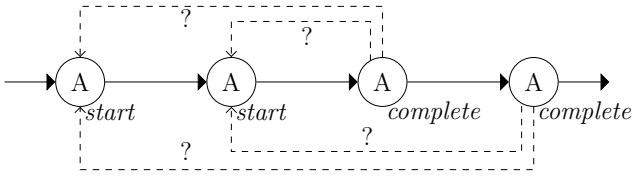

Fig. 8. Overlapping instances of activity executions: two executions of activity "A" are overlapping. As a result, a sequence of events "A (start)", "A (start)", "A (complete)", and "A (complete)" are recorded. For the last two complete events it is not clear to which execution of activity "A (start)" they correspond to.

2.7 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. As a consequence, process flexibility is needed in which a process is able to deal with both foreseen and unforeseen changes. 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 *permanent*. Momentary changes are short-lived and affect only a very few cases while permanent changes are persistent and stay for a while [24].

– **Evolutionary change**
Description. This refers to the scenario where during the period of time in

which the process has been logged, the process has undergone a persistent change. As a consequence, for a 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.
Example. Fig. 9 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.
Effect. 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 drifts* [25–28]. Several techniques have been proposed to detect and deal with changes (if any) in an event log.
Related Data Problems. Case Heterogeneity

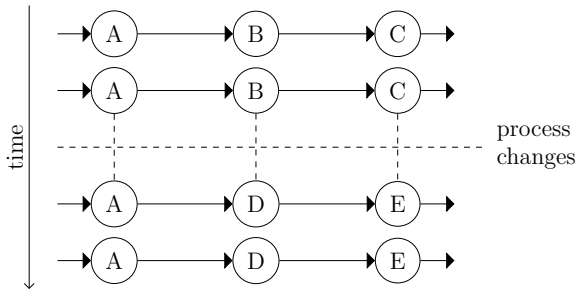

Fig. 9. Evolutionary change: Due to an evolutionary change the events "D" and "E" are recorded as part of the normal process flow instead of the events "B" and "C".

– **Momentary change**
Description. Momentary changes are short-lived and affect only a very few cases. For the process instances that are affected, one can perceive differences in comparison with other traces with regard to the activities performed, the ordering of activities, data involved and/or the people performing the activities.
Example. Fig. 10 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.

Effect. Momentary changes manifest as exceptional executions or outliers in an event log. As such, process mining algorithms need to be able to distinguish common and frequent behavior from any exceptional behavior. Some examples are the heuristics miner [1] and the fuzzy miner [11] which can deal with noise in a log. For any algorithm lacking the aforementioned capabilities, the produced model is typically spaghetti-like and hard to understand as both high and low frequent behavior is visualized. Alternatively, one can also consider techniques for outlier detection as a means of preprocessing the log and filtering/removing them [29, 30].
Related Data Problem. Outliers

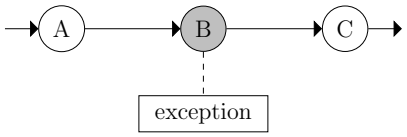

Fig. 10. Momentary change: As part of the normal process flow, the events "A" and "C" are recorded after each other. However, due to an exception, event "B" has occurred in between events "A" and "C".

2.8 Noisy Data/Outliers

Description. Event logs may often contain outliers, i.e., rare, exceptional, or anomalous execution behavior. Outliers are also referred to as noise. There is no clear definition of what constitutes an outlier/noise in a generic sense. It largely varies based on the event log and processes that we are dealing with and the contexts of analysis. There could be several sources of such outlier manifestation in event logs ranging from issues in logging to more serious cases of fraudulent behavior and/or non-compliance. Outliers can be manifested in any of the perspectives, e.g., control-flow resulting in inappropriate sequence of events, resources resulting in non-privileged execution of activities, etc.
Example. It could be the case that in an event log there are certain process instances whose lengths deviate a lot from the average trace length in the log. Such traces constitute some rare executional scenarios. As another example, in the financial institute log provided for the 2012 BPI Challenge [31], three loan applications are approved by an automated resource.
Effect. Most process mining techniques are misled by the presence of outliers, which impacts the goodness of the mined results. For example, outliers can lead to infer incorrect control-flows between activities, which often manifest as spaghetti-like processes. Unfortunately, very limited work [29, 30] has been done in dealing with outliers in process mining. The basic challenge still remains in being able to define what constitutes an outlier and to be able to detect and properly deal with them.

2.9 Mashed Processes

Description. Experiences from applying process discovery techniques on real-life logs revealed that viewing processes as a *flat single monolithic* entity is problematic. In reality, processes are designed and executed in a modular way with a collection of (potentially concurrent) interacting entities. The overall process can be seen as a *mashup* of its constituent sub-processes.
Example. The event log provided for the BPI Challenge 2012 [31] corresponds to a process that is a merger of three intertwined sub-processes in a large Dutch Financial Institute. Such extreme concurrency is also most often seen in healthcare processes. For example, during *resuscitation* of emergency patients in a hospital, several physicians and nurses will be performing activities simultaneously.
Effect. Dealing with concurrency remains one of the core challenges of process mining [1]. For example, process discovery algorithms yield either a flower model or a spaghetti-model on event logs exhibiting extreme concurrency. If the association between activities and the entities (sub-processes) that they correspond to is known, one can think of mining individual sub-processes in isolation. However, such an analysis might not be entirely interesting as an organization would be more interested in finding intricacies in how these entities interact.

2.10 Scoping

Description. In several domains, the definition of what constitutes a case is not explicit. The definition of a case can have different connotations based on the contexts and purpose of analysis. Domain knowledge is often needed in defining such an appropriate scope.
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.
Effect. Not choosing an appropriate scope of the event log 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).

3 Evaluation of Event Logs

In this section, we analyze the issues highlighted in Section 2 against several real-life logs. The objective of this analysis is to provide an insight into the manner and extent to which the identified data quality problems actually appear in real-life data that is used for process mining. Due to space constraints, we depict and discuss the results of only five real-life logs. Table 1 summarizes the classes of data quality issues manifested in these logs.

⟨TODO: Ordering in table does not match text. Also check consistency or names of issues!!⟩

Table 1. Evaluation of process mining data quality problems on a selection of real-life logs.

Data Quality Problem	Philips Healthcare	BPI Challenge 2011 (AMC)	BPI Challenge 2012	Catharina hospital	CoSeLoG
Coarse Granular Timestamps		X			
Mixed Granular Timestamps				X	
Incorrect Timestamps	X			X	
Missing Values			X		X
Missing Events			X	X	
Missing Events (anywhere in a trace)			X	X	
Partial / Incomplete Traces			X	X	X
Duplicate Tasks	X	X		X	X
Overlapping Activity Executions			X	X	
Evolutionary Change					X
Momentary Change				X	
Fine-Granular Events	X	X		X	X
Mashed Processes		X	X	X	
Case Heterogeneity	X	X	X	X	
Voluminous Data	X				
Noisy Data / Outliers	X	X	X	X	X
Scoping	X	X		X	

3.1 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. We observe the following data quality issues in these event logs:

– *Voluminous Data:* 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 2500 systems installed across the globe. This implies that Philips stores around 875 MB of compressed data with (comprising of millions of events) every day.

- *Fine-granular Events:* The event logs generated by X-ray machines are a result of write statements inserted in the software supporting the system. These event logs capture very fine-grained events such as the commands executed on the system and errors/warnings triggered by various components within the system. A typical event log contains hundreds of event classes (activities). An analyst is typically not interested in such low-level details and expects insights at high-levels of abstraction when analyzing these event logs.
- *Case Heterogeneity:* 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.
- *Incorrect Timestamps:* The ordering of events are not always be reliable. 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 (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.
- *Scoping:* 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. For 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.
- *Noisy Data/Outliers:* Several components in an X-ray machine might deteriorate over time exhibiting a compromised functionality. The system is maintained both pro-actively and reactively. The event data during periods of system malfunction manifest as a completely different behavior when compared to a fully functional system. This creates several noisy/outlier data in the event logs.
- *Duplicate Tasks:* When a physician operates a particular command on an X-ray machine, the connotation largely depends on the context of its execution. For example, a Fluoroscopy procedure selection can have different interpretations based on the surgical procedure applied on a patient. Such duplicity of tasks need to be properly disambiguated based on their relevant contexts.

3.2 The 2011 BPI Challenge Event Log

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 [32]. 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.

- *Case Heterogeneity:* 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 [4].
- *Coarse-granular Timestamps:* The event log also 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.
- *Mixed Granular Events:* 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.
- *Missing Values:* The event log contains several events with missing information. For example, 16 events do not contain the laboratory/department information in which a medical test pertaining to the event is performed.
- *Scoping:* 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.
- *Duplicate Tasks:* 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.
- *Noisy Data/Outliers:* The event log also exhibits noisy/outlier behavior such as the skipping of certain events. For more examples on some outlier behavior, the reader is referred to [4].

3.3 The 2012 BPI Challenge Event Log

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 [31]. 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 data quality issues encountered in the log are illustrated in Table 1 and explained below:

- *Mashed Processes:* The global process is defined over three sub-processes, viz., application, offer, and contact customers running concurrently, which manifests as an extreme concurrency problem in the event log. This creates a spaghetti-like process model from most discovery techniques.
- *Partial/Incomplete Traces:* Certain applications that have started towards the end of the recorded time period are yet to be completed leading to partial/incomplete traces (overall there are 399 cases that are incomplete).
- *Missing Events:* The event log contains several missing events. For example, there are 1042 traces where an association between the start of an activity and a completion of an activity is missed.
- *Overlapping Activity Executions:* The event log contains one trace where we see an overlapping execution of start and complete events for two instances of execution of an activity.
- *Missing Values:* The event log contains missing information for some attributes 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.
- *Case Heterogeneity:* 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.
- *Noisy Data/Outliers:* The event log contains several outlier traces. For example, there are some traces where certain activities are executed even after an application is cancelled or declined. There are three cases where a loan has been approved by an automated resource.

3.4 Catharina Hospital Event Log

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.

- *Mixed Granular Timestamps:* 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.
- *Incorrect Timestamps:* 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.
- *Missing Events:* Also, due to manual recording of actions, it is not recorded in a structured way when actions have been scheduled, started, and completed. As a result, for many actions one or more of these events are missing. In particular, for 67% of the actions, no complete event has been registered.
- *Partial / Incomplete Traces:* The log only contains only data for the year 2006. Therefore, for some patients the actions performed at the start or the end of the stay at the ICU are missing.
- *Overlapping Activity Executions:* For 9% of the actions for which both the start and the end have been recorded it is seen that there is an overlap between multiple actions.
- *Duplicate Events:* 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").
- *Momentary Change:* For patients admitted at an ICU it can be 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 "complete" event type, but still the obtained model was spaghetti-like as it contained 67 nodes and more than 100 arcs.
- *Extreme Concurrency:* For around 20% of the patients, one or more complications are registered. When a complication occurs, typically multiple actions are required in order to properly deal with the situation.
- *Case Heterogeneity:* 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 heartsurgery). 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.
- *Fine-Granular Events:* 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.
- *Noisy Data / Outliers:* The event log contains multiple outlier trace. For example, for the 412 patients which received care after the heart surgery the majority of the patients has between 15 and 70 events. Here, there are two patients which have more than 300 events.

– *Scoping:* Closer inspection of the log reveals that actions exist which are performed on a daily basis for patients (e.g. "basic care" or "medium care") whereas also patient-specific actions are performed (e.g. "cardioversion" or "place Swan Ganz catheter"). For each situation, different aspects need to be considered and different insights are obtained.

3.5 Event Log from a Large Dutch Municipality

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. Several of the data quality issues can be observed even in this log.

– *Fine-granular Events:* 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.
– *Mixed-granular Events:* The event log also exhibits events of mixed granularity, i.e., the event log emanates from a hierarchical process and we see events from all levels of hierarchy in the same event log.
– *Evolutionary Change:* 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 [25].
– *Partial/Incomplete Traces:* The event log contains 197 cases that are still running (incomplete).
– *Missing Values:* 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.
– *Noisy Data/Outliers:* The event log also contains several cases with exceptional execution behavior. These are often manifested as a missing (skipped) execution or extraneous execution of some activities.

4 Related Work

It is increasingly understood that data in data sources is often "dirty" and therefore needs to be "cleansed" [33]. In total, there exist five general taxonomies which focus on classifying data quality problems [34]. Although each of these approaches construct and sub-divide their taxonomies quite differently [33, 35–38], they arrive at very similar findings [33]. Alternatively, as data quality problems for time-oriented data have distinct characteristics, a taxonomy of dirty time-oriented data is provided in [34]. 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 [39, 40] the gynecological oncology healthcare process within an university hospital has been analyzed; in [41] several processes within an emergency department have been investigated; in [42] all Computer Tomography (CT), Magnetic Resonance Imaging (MRI), ultrasound, and X-ray appointments within a radiology workflow have been analyzed; in [43] the activities that are performed for patients during hospitalization for breast cancer treatment are investigated; in [44] the journey through multiple wards has been discovered for inpatients; and finally, in [45], 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 [41].

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 [46]. 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.

5 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 data quality issues. We believe that problems related to the quality of event data are the limiting factor. Unfortunately, these problems have received limited attention from process mining researchers. Therefore, we identified ten categories of data-related problems encountered in process mining projects. We hope that our findings will encourage systematic logging approaches (to prevent data quality problems) and repair techniques (to alleviate data quality problems). In the

upcoming big-data era, process mining will not be limited by the availability of data, but by the quality of event data.

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.

⟨TODO: Repair bib file for two styles used, e.g., missing "van der" problems.⟩

References

1. Aalst, W.: 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, 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. Aalst, W., Medeiros, A., Weijters, A.: Genetic Process Mining. In Ciardo, G., Darondeau, P., eds.: Applications and Theory of Petri Nets 2005. Volume 3536 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2005) 48–69
6. on Process Mining, I.T.F.: 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. 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
8. 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
9. 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
10. 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
11. Günther, C., van der Aalst, W.: 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

12. 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
13. 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
14. 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
15. 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
16. 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
17. 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
18. 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
19. 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
20. 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
21. 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)
22. 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)
23. Bose, R.P.J.C.: Process Mining in the Large: Preprocessing, Discovery, and Diagnostics. PhD thesis, Eindhoven University of Technology (2012)
24. Schonenberg, H., Mans, R., Russell, N., Mulyar, N., van der Aalst, W.M.P.: Process Flexibility: A Survey of Contemporary Approaches. In Dietz, J., Albani, A., Barjis, J., eds.: Advances in Enterprise Engineering I. Volume 10 of Lecture Notes in Business Information Processing., Springer-Verlag, Berlin (2008) 16–30
25. 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

26. Carmona, J., Gavaldà, R.: Online Techniques for Dealing with Concept Drift in Process Mining. In: International Conference on Intelligent Data Analysis (IDA 2012). (2012) to appear.
27. 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
28. 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
29. Ghionna, L., Greco, G., Guzzo, A., Pontieri, L.: Outlier Detection Techniques for Process Mining Applications. In An, A., Matwin, S., Ras, Z.W., Slezak, D., eds.: Foundations of Intelligent Systems. Volume 4994 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2008) 150–159
30. Folino, F., Greco, G., Guzzo, A., Pontieri, L.: Mining Usage Scenarios in Business Processes: Outlier-Aware Discovery and Run-Time Prediction. Data & Knowledge Engineering **70**(12) (2011) 1005–1029
31. 3TU Data Center: BPI Challenge 2012 Event Log (2011) doi:10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f.
32. 3TU Data Center: BPI Challenge 2011 Event Log (2011) doi:10.4121/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54.
33. 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
34. 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
35. Rahm, E., Do, H.: Data Cleaning: Problems and Current Approaches. IEEE Bulletin of the Technical Committee on Data Engineering **24**(4) (2000)
36. Müller, H., Freytag, J.C.: Problems, Methods, and Challenges in Comprehensive Data Cleansing. Technical report hub-ib-164, Humboldt University Berlin (2003)
37. Oliveira, P., Rodrigues, F., Henriques, P.: A Formal Definition of Data Quality Problems. In: International Conference on Information Quality (MIT IQ Conference). (2005)
38. Barateiro, J., Galhardas, H.: A Survey of Data Quality Tools. Datenbankspectrum **14** (2005) 15–21
39. Mans, R., Schonenberg, M., Song, M., van der Aalst, W., Bakker, P.: 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
40. 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.
41. Rebuge, A., Ferreira, D.: Business Process Analysis in Healthcare Environments: A Methodology Based on Process Mining. Information Systems **37**(2) (2012)
42. 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

43. 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
44. 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
45. 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
46. 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