Quality-Informed Process Mining: A Case for Standardised Data Quality Annotations

KANIKA GOEL, Queensland University of Technology, Brisbane, Australia
SANDER J.J. LEEMANS, Queensland University of Technology, Brisbane, Australia
NIELS MARTIN, Hasselt University, Hasselt, Belgium and Research Foundation Flanders, Brussels, Belgium
MOE T. WYNN, Queensland University of Technology, Brisbane, Australia

Real-life event logs, reflecting the actual executions of complex business processes, are faced with numerous data quality issues. Extensive data sanity checks and pre-processing are usually needed before historical data can be used as input to obtain reliable data-driven insights. However, most of the existing algorithms in process mining, a field focusing on data-driven process analysis, do not take any data quality issues or the potential effects of data pre-processing into account explicitly. This can result in erroneous process mining results, leading to inaccurate or misleading conclusions about the process under investigation. To address this gap, we propose data quality annotations for event logs, which can be used by process mining algorithms to generate quality-informed insights. Using a design science approach, requirements are formulated, which are leveraged to propose data quality annotations. Moreover, we present the 'Quality-Informed visual Miner' plug-in to demonstrate the potential utility and impact of data quality annotations. Our experimental results, utilising both synthetic and real-life event logs, show how the use of data quality annotations by process mining techniques can assist in increasing the reliability of performance analysis results.

Additional Key Words and Phrases: Process Mining, Data Quality, Annotations, Metadata, Quality-Informed Performance Analysis, Quality-Informed Conformance Checking

ACM Reference Format:
Kanika Goel, Sander J.J. Leemans, Niels Martin, and Moe T. Wynn. 2022. Quality-Informed Process Mining: A Case for Standardised Data Quality Annotations. *ACM Trans. Knowl. Discov. Data.* 1, 1, Article 1 (January 2022), 47 pages. https://doi.org/10.1145/3511707

1 INTRODUCTION

Data is at the heart of modern organisations. Technological advances in the fields of Business Intelligence and Data Science empower organisations to become more data-driven and decipher hidden knowledge from large databases. Within the Business Process Management field, Process Mining is a specialised form of data-driven process analytics. Process mining extracts detailed insights about the real behaviour of business processes from historical process data, known as event logs, collated from different IT systems [47]. An event log is a collection of events associated with the execution of a process. For a case, such as an order or a patient visit, a trace is recorded

Authors' addresses: Kanika Goel, School of Information Systems, Queensland University of Technology, Brisbane, Australia, k.goel@qut.edu.au; Sander J.J. Leemans, School of Information Systems, Queensland University of Technology, Brisbane, Australia, s.leemans@qut.edu.au; Niels Martin, Research Group Business Informatics, Hasselt University, Hasselt, Belgium and Research Foundation Flanders, Brussels, Belgium, niels.martin@uhasselt.be; Moe T. Wynn, School of Information Systems, Queensland University of Technology, Brisbane, Australia, m.wynn@qut.edu.au.

1556-4681/2022/1-ART1 $15.00
https://doi.org/10.1145/3511707

consisting of an ordered set of events. Each event can be annotated with additional attributes such as a timestamp, a transaction type (e.g., start or complete) and the resource [47].

While process mining offers great potential to better understand business processes, the quality of process mining results ultimately depends on the quality of its input, i.e., the event log [5, 48]. It has been shown that real-life event logs suffer from a multitude of data quality issues such as missing and imprecise attribute values [3, 5, 45, 50]. For example, event logs may have timestamps which are incorrect or defined at a varying degree of granularity (e.g., at the date or the hour level instead of at the granularity level of seconds).

Despite the critical importance of data quality, most process mining algorithms do not take data quality or any data pre-processing information into account. This entails the risk of obtaining counter-intuitive or even misleading outcomes. For example, incorrect timestamps can alter the order of events and, moreover, result in the calculation of incorrect activity durations [5]. To avoid such problems, an analyst should identify and remedy data quality issues in an event log, which typically accounts for 80% of analysts' time [55]. However, very limited information about the event log quality tends to be transferred to the analysis phase, which is why the reliability of process mining insights could be questioned. This observation stresses the need for quality-informed process mining techniques, i.e., process mining techniques which explicitly take data quality metadata into account when generating their results.

A key step towards quality-informed process mining is being able to capture data quality metadata [51] in a structured way such that it can be used by algorithms. Current literature does not support the latter. Given this research gap, this paper proposes standardised data quality annotations for event logs that can be used by process mining algorithms for increased reliability of process mining outcomes. To demonstrate the potential utility of these annotations, the *'Quality-Informed visual Miner'* plug-in is developed, which contains initial quality-informed process mining techniques. The performance analysis capabilities of the plug-in are illustrated using both synthetic and real-life event logs in which varying amounts of data quality issues are inserted.

The paper is structured as follows: Section 2 summarises the related work and Section 3 introduces preliminary concepts. Section 4 presents the methodology and elicits requirements for data quality annotations. The data quality annotations are introduced in Section 5. Section 6 demonstrates the potential use of the annotations in a ProM-plugin. Section 7 tests the utility of data quality annotations. The paper ends with a discussion and conclusion in Section 8.

2 RELATED WORK

The importance of data quality is well-recognised in the data mining community, and is considered crucial to manage and interpret real-life data sets. Several data quality metrics have been brought forth [11] and approaches to repair data quality issues have been proposed [22, 52]. Similarly, the significance of data quality has also been recognised in the area of natural language processing, resulting in the introduction of techniques such as word embedding models [27, 32] and meta-heuristic algorithms for clustering [31]. Word embedding provides a more compact and expressive representation model for text documents along with extraction of semantic and syntactic meaning of words [28] resulting in better predictive performance. Word embedding models have been used along with deep learning resulting in better predictive performance for sentiment analysis [29, 30] and sarcasm identification [26]. Overall, prior work conveys that the predictive performance of algorithms can improve by addressing the quality of data through preprocessing. While the aforementioned works illustrate that data quality is a concern in various domains, this paper will focus on data quality within the process mining domain.

The importance of data quality in the process mining domain is well recognised by the community, as exemplified by its inclusion in the Process Mining Manifesto [48]. Despite this awareness, explicit studies on event log quality only started to appear more frequently in recent years. Within the process mining field, three main areas of quality-related research can be distinguished: (i) data quality taxonomies, (ii) log quality assessment, and (iii) log repair.

The first literature area focuses on conceptualising the notion of event log quality and identifying classes of data quality issues. Event log quality taxonomies differ in terms of the level of granularity at which quality issues are defined. While the Process Mining Manifesto [48] defines five high-level maturity levels without specifying specific issues, Bose et al. [5] define 27 types of quality issues classified within the categories missing, incorrect, imprecise, and irrelevant data. Verhulst [51] develops a literature-based framework consisting of 12 quality dimensions, whereas Suriadi et al. [45] propose 11 imperfection patterns based on real-life logs. Recently, Vanbrabant et al. [50] outline 14 fine-grained quality issues categorised as missing, wrong, and not directly usable data.

The second literature area focuses on the identification of event log quality issues. Conceptual process mining frameworks have been proposed in which data quality assessment is explicitly included [3, 23]. From a practical perspective, a more systematic event log quality assessment is supported by the open source R-package DaQAPO [1]. Similarly, Andrews et al. [1] proposes the foundations of the log query language QUELI, Querying Event Log for Imperfections. Additionally, RDB2Log has been proposed to perform assessment while creating an event log from a relational database [2]. Recently, Fischer et al. [9] propose a framework to detect timestamp-related issues and output a set of quality metrics such as granularity and precision.

The third literature area centres around log repair, i.e., enhancing event log quality by addressing data quality issues. Heuristics are proposed to address data quality such as missing events [10, 40, 53], missing case identifiers [4], incorrect timestamps [8], activity labels [41], or deviating sequences [43]. These techniques require domain knowledge in the form of a process model as an input. Conversely, Nguyen et al. [24] recently experimented with the use of autoencoders to handle incorrect/missing attribute values in the absence of domain knowledge, while Conforti et al. [7] present an automated technique that filters out infrequent behaviour in event logs.

From the previous, it follows that increasing research is being carried out on event log quality. However, quality-informed process mining implies that process mining techniques *explicitly* take log quality and repair information into account. This need is reinforced by recent work on data impact analysis [46], which elaborates on how changes to data can impact process analysis. Similary, Pegoraro et al. [34, 35] explore how uncertainty information about attributes in an event log, whereby the actual value is unknown but a set of potential values is known, can be used for conformance checking. These works illustrate the relevance of topics such as quality-informed process mining.

Quality-informed process mining techniques should take data quality issues or data transformations into account. Annotations provide a structured way to make such data quality information accessible to process mining algorithms. Annotations have, for instance, been proposed to record cost [54] and for privacy-preserving transformations [39]. However, annotations are absent for event log quality. Given this research gap, this paper proposes a standardised format for event log quality annotations and demonstrates how annotations can support deriving quality-informed insights from process mining techniques.

3 PRELIMINARIES

In this section, we introduce some preliminary concepts which we will leverage in the paper.

Event Logs. An *event log* is a finite collection of traces. Each *trace* denotes one traversal of the process by a customer/claim/order, and each trace consists of an ordered finite number of events,

[1] https://github.com/nielsmartin/daqapo

which denote the execution of process steps (*activities*). Logs, traces, and events can have attributes. For instance, the `concept:name` attribute of a log or trace denotes the name of the log or trace, and denotes the activity of an event. Other common attributes include timestamps (`time:timestamp`) and the resource that executed an event (`org:resource`). Given an event e, we denote the value of attribute X of e with $\#_X(e)$, and similar for a log L or a trace σ. For instance, the following log fragment consists of one trace, which consists of four events:

Trace	`concept:name`	`time:timestamp`	`org:resource`
1564	register request	10/12/2020 7:35	Femke
1564	process request	10/12/2020 9:54	Emile
1564	notify customer	10/12/2020 10:40	Jesse
1564	archive request	11/12/2020 7:35	Geert
…	…	…	…

Petri Nets. A *labelled Petri net* is a bipartite graph consisting of places (P) and transitions (T) as nodes, and edges ($\subseteq (P \rightarrow T) \cup (T \rightarrow P)$) connecting them. A potentially partial labelling function λ maps transitions to activities. The state of a Petri net, representing the execution of a business process, is a multiset over P, that is, tokens on places. We assume the standard semantics of Petri nets, that is, from an initial marking transitions can *fire* if the places from which they have incoming edges have sufficient tokens. On firing, a transition $t \in T$ consumes tokens from its incoming places, produces tokens on its outgoing places, and, if λ labels t, indicates the execution of its labelled activity $\lambda(t)$. Figure 2b shows an example with *language* $\{\langle a, b\rangle\}$.

Conformance Checking. After discovering a process model, the quality of the model should be evaluated: discovery techniques inherently have to fit the behaviour recorded in the event log into their representational bias, and may choose to leave behaviour of the log out of the model, or include behaviour in the model that was not recorded in the event log. Evaluating the discovered model by studying these differences is essential for any process mining project [47].

The current state-of-art in conformance checking are optimal alignments, which describe a matching between a trace in the log and a path through the model. An alignment contains steps in both trace and paths (*synchronous moves*), steps in only the path (*model move*) and steps in only the trace (*log move*). For a particular trace from the log, an alignment computation aims to choose a path and construct an alignment such that the total cost of moves is minimal [49], where each move inhibits a cost as follows:

Definition 1 (Alignment cost function). *Let e be an event, and let t be a transition. Then,*

$$cost(\genfrac{}{}{0pt}{}{e}{\rightarrow}) = 1$$

$$cost(\genfrac{}{}{0pt}{}{e}{t}) = \begin{cases} 0 & if\, \#_{concept:name}(e) = \lambda(t) \\ \infty & otherwise \end{cases}$$

$$cost(\genfrac{}{}{0pt}{}{\rightarrow}{t}) = \begin{cases} 0 & if\, t \text{ is invisible} \\ 1 & otherwise \end{cases}$$

Figure 2c shows an example of an alignment; the first move $\genfrac{}{}{0pt}{}{b}{\rightarrow}$ is log move, $\genfrac{}{}{0pt}{}{a}{a}$ is a synchronous move and $\genfrac{}{}{0pt}{}{\rightarrow}{b}$ is a model move.

4 METHODOLOGY & REQUIREMENTS

4.1 Methodology

This study follows a design science approach, a paradigm centering around the design, development, and scientific study of an artefact which solves a problem of general interest [15]. The key artefact of this paper are data quality annotations, which enable quality-informed process mining.

To operationalise the design science principles, the six-stage procedure of Peffers et al. [33] has been followed, which is based on a synthesis of prior literature. Firstly, the research problem is specified and motivated: the absence of a standardised way to capture data quality metadata in an event log such that this metadata can be leveraged by process mining algorithms (Section 1 - 2). Secondly, the requirements of the data quality annotations are specified, which are inspired by related work on data quality in process mining (Section 4.2). Thirdly, the annotations are designed and developed (Section 5). Fourthly, the artefact is demonstrated by developing and using the *Quality-Informed visual Miner* (QIvM) ProM plug-in which includes initial process mining techniques using data quality annotations (Section 6-7). Fifthly, the designed, developed and demonstrated artefact is reflected against its objectives (Section 8). Finally, the current paper constitutes the prime communication outlet for the prior research activities.

4.2 Requirements for the Data Quality Annotations

This subsection outlines five requirements of the data quality annotations. They are inspired by the related work on event log quality, taking into account the aim of the annotations: systematically informing process mining algorithms about the quality of a log to generate quality-informed process mining insights.

R1: Data quality annotations should support the extraction of quality-informed process mining insights.

Event log quality issues, e.g., related to timestamps, can generate unreliable results when used without further consideration. For instance, in a hospital setting, physicians have a tendency to see several patients and record their findings in the system afterwards. This causes the timestamps in the log to differ from the time at which an activity has been executed. Inaccurate timestamps can result in an incorrect activity order and duration. Similarly, log changes made during preprocessing can also influence process mining outcomes. E.g., missing timestamps might be inserted based on the values of related timestamps. To make process mining insights quality-informed, they have to take into account event log quality issues and data transformations. Hence, the annotations should feed process mining algorithms with data quality information.

R2: Data quality annotations should be standardised, enabling their systematic use by process mining algorithms.

To enable a widespread use of the data quality annotations in process mining algorithms, it is crucial that the annotations are standardised. One way to achieve this is by proposing a data quality extension to the IEEE *Extensible Event Stream* (XES) format, which is an XML-based standard that ensures interoperability in event logs [56].

R3: Data quality annotations should support the recording of information about the type of data quality issue.

Several types of event log quality issues can be distinguished. The commonly cited framework of Bose et al. [5] distinguishes between the missing, inaccurate, imprecise and irrelevant character of several key event log attributes such as timestamps, and activity labels. For instance, an event might be irrelevant for the analysis (irrelevant), might not carry a timestamp (missing), might have a wrong timestamp (inaccurate), or might be recorded as a date (imprecise). As these issues need to be handled differently by process mining algorithms, data quality annotations should convey information about the type of data quality issue.

R4: Data quality annotations should support the recording of information about data transformations contained in an event log.

For certain data quality issues, an analyst may decide to correct them based on domain knowledge, e.g., using a heuristic [44]. He/she might, e.g., correct timestamps which were recorded after the activity took place. Besides repairing data quality issues, data transformations can also originate from the need to anonymise, generalise, or apply some other privacy-preserving techniques to safeguard privacy (e.g., regarding personal information of patients or staff members) [38, 39]. As data transformations can have an impact on process mining outcomes, data quality annotations should keep track of transformations, which were executed for reasons of data quality or privacy preservation.

R5: Data quality annotations should support the recording of data quality/transformation information at different levels of aggregation.

Data quality/transformation information can be conveyed at different levels of details. At the lowest granularity level, data quality annotations can be recorded for various event attributes. For some process mining algorithms or for an analyst, it may also be of benefit to know how many events in a trace have data quality issues and/or how many traces in an event log have data quality issues. To cater for various information needs, the annotations should support capturing data quality information at different levels in an event log (event, trace, and log).

The aforementioned five requirements form the basis for the annotations presented in Section 5.

5 DATA QUALITY ANNOTATIONS

This section introduces the data quality annotations for event logs. Figure 1 presents a conceptual data model depicting the relations between key entity types (i.e., event, event attribute, trace, and log) and proposed data quality annotations using the Object Role Modelling 2 (ORM2) notation [12]. We acknowledge that there are multiple languages that can be used to build conceptual models (e.g., ER, UML, ORM, and more). We chose ORM2 notation as it is powerful, and its attribute-free nature makes it more stable to changing business environments [13]. Furthermore, the detailed character of ORM2 enables readers not familiar with process mining to understand the structure of an event log and visually position the annotations in the event log [14][2].

Let $QA = \{qa_1, qa_2, .., qa_n\}$ be the set of data quality annotations such that $QA \subset Attributes$. All annotations will share the prefix 'quality' to easily recognise them. We propose to maintain data quality annotations at the event, trace, and log levels, which align with the XES event log structure [56], a standard format for process mining supported by majority of process mining tools. The log is the topmost level, and contains information that relates to a specific process such as the patient journey in a hospital. A log consists of arbitrary number of traces, which describes the execution of a specific process instance (also referred to as a case) of the logged process. For example, the journey of one specific patient in the hospital describes a trace. A trace consists of

[2] For readers who are not familiar with ORM2 notations, we briefly introduce the key concepts of the notation next. Figure 1 captures the relationship between entities, where an entity type is depicted using a round cornered rectangle (e.g., Event and Trace). Each entity type has an identifier label, mentioned between brackets in each round cornered rectangle (e.g., Event ID) and can be associated with one or more label types depicted using dashed round cornered rectangle (e.g., Description). Entity types can share a relationship (known as fact type) with two or more other entity types. For example, 'consists of' is a binary fact type between Trace and Log entity types, expressing that an event log consists of traces. A bar above a fact type represents a uniqueness constraint that applies to that fact type, which can be one to one, one to many, or many to many. A dot at the end of the connector between two entity types denotes a mandatory constraint (e.g., every log must consist of one or many traces). A line with an arrowhead displays a generalisation/specialisation constraint. For example, an activity is a specialisation of the Event attribute entity type.

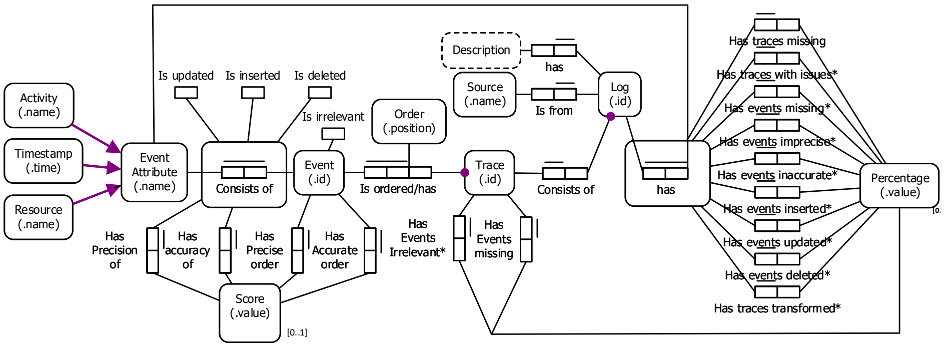


Fig. 1. A conceptual data model of proposed event log quality annotations.

an arbitrary number of events. An event represents the atomic granules of an activity that have been observed during the execution of a process. For instance, the completion of the admission of a specific patient in the hospital constitutes an event. A log, trace, or event does not contain any information by themselves. They define the structure of the log. All information is stored in attributes, which describe their parent element. For example, the attribute 'activity' stores the name of the activity to which an event is associated.

The data quality annotations presented in this section are grouped into three categories: (i) annotations capturing the presence of data quality issues in an event log, (ii) annotations capturing the presence of data transformations performed on an event log, and (iii) annotations capturing general event log characteristics.

5.1 Category I: Event Log Quality

The first category of annotations relates to the presence of data quality issues in an event log (R3). As mentioned earlier, events in an event log have various attributes which provide information about the event. For example: an event corresponds to a particular activity, can have a timestamp, and can be conducted by a particular resource. For every event, event order represents the position of that event in a trace with respect to other events, represented as a ternary fact type in Figure 1. The proposed metadata for this category covers the event order and three key event attributes: activity, timestamp, and resource (depicted as sub-types of the 'Event Attribute' entity in Figure 1). In the remainder of this section, 'attribute X' refers to one of the attributes. The same annotations could be used for any other attributes of interest.

Inspired by the process data quality taxonomy of Bose et al. [5],for every event $e \in \sigma$ and for every $\sigma \in L$ we propose the following annotations related to data quality issues to be recorded at the **event level**:

- Accuracy of Attribute X ($\textsc{acc}_{\rhd X}$) – Denotes the probability that the value is logged correctly. E.g., a nurse may record the patient admission timestamp as 12-01-2021 11:23:45 instead of 12-01-2021 23:11:45. This annotation is recorded as *quality:accuracy:attributeX.*
- Precision of Attribute X ($\textsc{pre}_{\rhd X}$) – Denotes how coarse the logged data is. E.g., a nurse may record the admission time till hours (13-01-2021 09), which is less precise than recording time till seconds (13-01-2021 09:07:44). This annotation is recorded as *quality:precision:attributeX.* For the precision of timestamps, the value can be ms (millisecond), sec (second), min (minute), hr (hour), day, mon (month), and yr (year). ms denotes that the timestamp is precise to

millisecond level (e.g., 13-01-2021 11:23:45.200), whereas hr denotes that the timestamp is precise to hours level (e.g., 13-01-2021 11 a.m.). Example annotations are present in Listing 1.
- Relevance ($\textsc{rel}$) – Denotes whether the data is relevant for the analysis. E.g., a nurse may record the mobile number of a patient's next of kin which may be irrelevant for the analysis. This annotation is recorded as *quality:relevance.*
- Missing Attribute X ($\textsc{mis}_{\triangleright X}$) – Denotes if data is missing for an attribute. E.g., no resource is recorded for an event. This annotation is recorded as *quality:relevance.* The value for missing annotation is 0 or 1, where 0 indicates that the value is missing.
- Order precision ($\textsc{pre}_{\triangleright \text{order}}$) – Denotes how certain the order of events in the logged trace is. Even though a trace consists of a total ordering of events, some events could e.g., have happened at the same time, or it could be unknown in which order they happened (e.g., due to equal or imprecise timestamps). Such events have an imprecise order. The order precision annotation denotes the probability that the order of an event is precise, as will be operationalised in Section 6.1.1.
- Order accuracy ($\textsc{acc}_{\triangleright \text{order}}$) – Denotes the probability that the order of the event has been logged correctly. For instance, a nurse erroneously records a wrong day that a particular treatment step was performed, thereby introducing an order inaccuracy with the other steps performed.

The value of these annotations – unless indicated otherwise – lies between 0 and 1, with 1 being perfectly accurate, precise, or relevant.

Besides annotations at the event level, we propose aggregated measures at the trace and log levels (R4). At the **trace level**, the following two annotations *can* be maintained:

- Percentage of irrelevant events - Denotes the percentage of events irrelevant for analysis in a trace. E.g., a value of 0.2 would indicate that 20% of the events in the trace are irrelevant for analysis. This annotation is recorded as *quality:pctirrelevant.*
- Percentage of events with missing value for attribute X - Denotes the percentage of events in a trace with a missing value for attribute X. E.g., a value of 0.3 for *quality:pctmissing:resource* would mean that 30% of events in the trace have a missing value for resource. This annotation is recorded as *quality:pctmissing:attributeX.*

At the **log level**, the following annotations summarise the presence of quality issues:

- Percentage of traces with any data quality issue - Denotes the percentage of traces in a log that have a data quality issue (precision, accuracy, relevance, or missing). E.g., a value of 0.3 for *quality:pctissuestraces* would indicate that 30% of traces in the log have some data quality issues. This annotation is recorded as *quality:pctissuestraces.*
- Percentage of traces without a data quality issue - Denotes the percentage of traces in a log without a data quality issue (precision, accuracy, or missing) related to attribute X. E.g., a value of 0.6 for *quality:pctprecisetraces:time:timestamp* would mean that 60% of the traces have precise timestamps. This annotation is recorded as *quality:pctprecisetraces:attributeX, quality:pctaccuratetraces:attributeX,* and *quality:pctnotmissingtraces:attributeX.*
- Percentage of traces with relevant events - Denotes the percentage of traces in a log that have events relevant for analysis. E.g., a value of 0.8 for quality:pctrelevanttraces would indicate that 80% traces have relevant events. This annotation is recorded as *quality:pctrelevanttraces.*
- Percentage of events with a data quality issue for attribute X - Denotes the percentage of events in a log with an imprecise, inaccurate, or missing value for attribute X. E.g., a value of 0.5 for *quality:pctinaccurate:resource* would indicate that 50% of the events in the log have inaccurate resources. This annotation is recorded as *quality:pctimprecise:attributeX, quality:pctinaccurate:attributeX,* and *quality:pctmissing:attributeX.*

These aggregate values are derivable (represented with an asterisk in Figure 1) if corresponding fine-grained annotations exist at the event level. Therefore, it is not mandatory to use them. However, it is possible that some annotations at the event level may be omitted due to privacy concerns [36]. In such cases, these aggregated measures can be used to reason about the suitability of an event log for different types of process mining techniques. These annotations can also be used to work on specific traces of interest or as input to quality-informed process mining techniques.

5.2 Category II: Event Log Transformations

The second category of annotations centres around event log transformations (R4). Similar to the previous category, this metadata covers the three key event attributes for process mining: activity, timestamp, and resource. However, these annotations could also be used for other attributes of interest. At the **event level**, the following annotations with boolean values are proposed (three unary fact types in Figure 1):

- Value of attribute X inserted ($\textsc{ins}_{\rhd X}$) – Denotes if a value is inserted for attribute X. E.g., an analyst may insert a missing timestamp for an event. This annotation is recorded as *quality:inserted:attributeX*.
- Value of attribute X updated ($\textsc{upd}_{\rhd X}$) – Denotes if a value is updated for attribute X. E.g., an analyst may update the activity label 'item shipd' to 'item shipped'. This annotation is recorded as *quality:updated:attributeX*.
- Value of attribute X deleted ($\textsc{del}_{\rhd X}$) – Denotes if a value is deleted for attribute X. E.g., an analyst may delete a timestamp or sensitive details of a user from the log for privacy purposes. This annotation is recorded as *quality:deleted:attributeX*.

At the **trace level**, annotations related to the *percentage of events* with a data transformation *can* be maintained. At the **log level**, the following four aggregated annotations can be specified:

- Percentage of traces with a data transformation - Denotes the percentage of traces in a log that underwent a data transformation (insert, update, delete) related to attribute X. E.g., a value of 0.3 for *quality:pcttracesupdated:time:timestamp* would mean that 30% of traces in the log have events with timestamps updated. This annotation is recorded as *quality:pcttracesupdated:attributeX*, *quality:pcttracesinserted:attributeX*, and *quality:pcttracesdeleted:attributeX*.
- Percentage of events with an inserted value for attribute X - Denotes the percentage of events where a value was inserted for attribute X. E.g., a value of 0.7 for *quality:pctinserted:resource* would indicate that 70% of events in the log had values inserted for attribute resource. This annotation is recorded as *quality:pctinserted:attributeX*.
- Percentage of events with an updated value for attribute X - Denotes the percentage of events where a value was updated for attribute X. E.g., a value of 0.4 for *quality:pctupdated:activity* would suggest that 40% of events in the log had values updated for activity label. This annotation is recorded as *quality:pctupdated:attributeX*.
- Percentage of events with a deleted value for attribute X - Denotes the percentage of events where a value was deleted for attribute X. E.g., a value of 1 for *quality:pctdeleted:resource* would indicate that 100% events in the log have the value deleted for the attribute resource. This annotation is recorded as *quality:pctdeleted:attributeX*.

Similar to the previous category, the annotations at the trace and log level can be derived from event level annotations. Nevertheless, annotations are specified at the trace level as trace is a core concept in process mining. An analyst may wish to maintain annotations at the trace level to mark traces where insertions, updates, and deletions are made. Similarly, the log level annotations could be used to gain an overall impression of the transformations made to the event log and can also be used as input for quality-informed process mining techniques.

5.3 Category III: General Event Log characteristics

The third category of annotations provides some general contextual information regarding an event log at the log level to partially support requirements R1, R3, and R4. Two metadata fields are proposed:

- Description ($\textsc{des}_{\triangleright Log}$) – Provides an overview of the event log. The description can also capture any privacy-preserving transformations applied to the log. E.g., 'Emergency department log 2018-2019. All timestamps were shifted for anonymisation purposes'. This annotation is recorded as *quality:description*.
- Source ($\textsc{src}_{\triangleright Log}$) – Highlights the information system(s) from which the event log originates. This annotation allows an understanding of how the log was generated and from which systems it originated. E.g., 'The data was obtained from Healthcare Management System'. The annotation is recorded as *quality:source*.

5.4 Data Quality Annotations in Practice

Data quality annotations should be standardised to ensure its use by process mining algorithms. To facilitate a widespread adoption of data quality annotations, we propose an XES-compliant data quality extension (R1,R2) to the IEEE XES event log standard [56]. The proposed XES extension and an overview of all proposed annotations is present in Appendix A.

Listing 1 shows a snapshot of an annotated event log and demonstrates annotations at log, trace, and event level. At the log level the annotations related to description of the log (*quality:description*), source of the log (*quality:source*), percentage of events with time:timestamp updated (*quality:pctupdated:time:timestamp*), and percentage of events with imprecise resource (*quality:pctimprecise:resource*) are present. The snapshot is for a trace with *concept:name* (or id) P352. The attributes of the trace present in the snapshot are *concept:name*, *quality:pctimprecise:resource* which indicates the percentage of events with imprecise resource, and *quality:pctupdated:activity* that demonstrates the percentage of activity labels updated. At the event level, the attributes are - *concept:name* as 'First physician consult', *org:resource* as 'Physician 34', *lifecycle:transition* as 'complete', and a *time:timestamp* of '2020-02-17 T18:00:00.000". Data quality annotations related to accuracy of timestamp (*quality:accuracy:time:timestamp*) and update of time:timestamp attribute (*quality:updated:time:timestamp*) are present.

```
<log>
    <string key="quality:description" value="Emergency department log 2018-2019. All timestamps were
  shifted for anonymisation purpose"/>
    <string key="quality:source" value="Healthcare management system ..."/>
    <double key="quality:pctupdated:time:timestamp" value=0.20/>
    <double key="quality:pctimprecise:resource" value=0.38/>
    ...
    <trace>
        <string key="concept:name" value="P352"/>
        <double key="quality:pctimprecise:resource" value=0.20/>
        <double key="quality:pctupdated:activity" value=0.10/>
        ...
        <event>
            <string key="concept:name" value="First physician consult"/>
            <string key="org:resource" value="Physician 34"/>
            <string key="lifecycle:transition" value="complete"/>
            <string key="time:timestamp" value="2020-02-17T18:00:00.000"/>
            <double key="quality:accuracy:time:timestamp" value=0/>
            <boolean key="quality:updated:time:timestamp" value=T/>
            ...
        </event>
    </trace>
</log>

```

Listing 1. An example of data quality annotations.

A detailed discussion on how to generate these annotations is beyond the scope of this paper. We point the readers to recent complementary research conducted by the process mining community on the assessment and quantification of data quality issues in event logs. Andrews et al. [2] propose an approach to quantify the extent to which various data attributes in databases are suitable to be used as attribute in an event log (e.g., case, activity, start/complete times, resource). Fisher et al. [9] present an extensive framework to detect and quantify timestamp-related issues in event log. Additionally, several repair techniques to tackle event log quality issues have been proposed (e.g., [6, 8]). From a practical perspective, tools like these can facilitate obtaining the data quality annotations proposed in this paper.

The log, trace and event level quality attributes can be leveraged to the benefit of other perspectives as well. For instance, the *org:resource* and *concept:name* event attributes denote the resource who executed an activity and the name of the activity, respectively. Such categorical event-level attributes yield another level on which data quality issues can materialise and be identified, annotated and studied. For instance, when finishing a task, a computer resource will log an event immediately, thus the *time:timestamp* of that event will be accurate and precise, while a human resource might need time to register a completed task in the system, resulting in an inaccurate and imprecise *time:timestamp* for the corresponding event. Similarly, some activities (*concept:name*) might have an inherent precision of their *time:timestamp*, for instance if the activity is logged by a nurse who records performed procedures at the end of a shift. Such knowledge can be recorded at the log level, thereby providing a fourth virtual level of data quality annotations, that is, in our example, on resources or activities. The techniques described in this paper could be applied to such virtual-level data quality annotations, for instance by propagating their data quality annotations to relevant events.

6 DEMONSTRATION: QUALITY-INFORMED PROCESS MINING

To demonstrate the potential use of data quality annotations within the context of quality-informed process mining, the ProM plug-in *Quality-Informed visual Miner* (QIvM) is proposed. To this end, the influence of data quality issues and annotations on conformance checking (Section 6.1) and performance analysis (Section 6.2) are explored. Afterwards, some pointers regarding the implementation of QIvM are provided (Section 6.3).

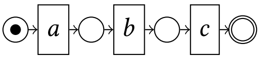
While we focus on conformance checking and performance analysis, we acknowledge that data quality issues may also influence discovery techniques. In particular, noise and incomplete information are data quality issues that have been widely recognised in the field and for which process discovery techniques [18, 19, 42] have been designed to be robust against. Thus, we do not consider process discovery in this paper and leave a detailed study of the influence of data quality annotations (and the data quality issues they indicate) for individual discovery techniques for future work. Instead, we assume that a correct process model has been obtained.

6.1 Conformance Checking

Conformance checking techniques highlight the differences between an event log and a process model. In this section, we study the influence of data quality annotations on alignment computations and provide a technique to reduce this influence. That is, we will provide a data quality-informed cost function to address the identified issues. In the remainder of this section, we discuss how the

$[\langle b^{\text{PRE}_{\triangleright\text{order}}:0.5}, a^{\text{PRE}_{\triangleright\text{order}}:0.25}, c^{\text{PRE}_{\triangleright\text{order}}:0.5}, d^{\text{PRE}_{\triangleright\text{order}}:0.25}\rangle]$

(a) Log L, annotated with order precision.



(b) Model M.

log	b	a	$\rightarrow$	c	d
model	$\rightarrow$	a	b	c	$\rightarrow$

(c) Alignment: cost 3
Quality-informed alignment: cost 1.75.

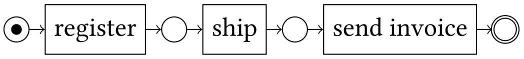
log	$\rightarrow$	b	a	c	d
model	a	b	$\rightarrow$	c	$\rightarrow$

(d) Alignment: cost 3
Quality-informed alignment: cost 1.5.

Fig. 2. An example of alignments with imprecisely ordered events.

$[\langle \text{register}^{\text{PRE}_{\text{concept:name}}:1}, \text{shipped}^{\text{PRE}_{\text{concept:name}}:0.25}, \text{sent invoice}^{\text{PRE}_{\text{concept:name}}:0.5}\rangle]$

(a) Log L, annotated with precision of concept:name.



(b) Model M.

log	register	shipped	$\rightarrow$	sent invoice	$\rightarrow$
model	register	$\rightarrow$	ship	$\rightarrow$	send invoice

(c) Standard alignment: cost 4; Quality-aware alignment: invalid.

log	register	ship~~ped~~	sen~~t~~d invoice
model	register	ship	send invoice

(d) Standard alignment: invalid; Quality-aware alignments: cost 0.

Fig. 3. An example of alignments with imprecise concept:name.

annotations $\text{PRE}_{\triangleright\text{order}}$ and $\text{PRE}_{\triangleright\text{concept:name}}$ can be considered in the cost function. For the other annotations, please refer to Appendix B. We finish with a summary and formalisation of the quality-informed alignments.

6.1.1 Order Precision ($\text{PRE}_{\triangleright\text{order}}$). By definition, a trace consists of a total ordering of events. However, some events could have happened at the same time, or it could be unknown in which order they happened (e.g., due to equal or imprecise timestamps). Such events have an imprecise order, and the $\text{PRE}_{\triangleright\text{order}}$ annotation denotes the probability that the order of an event is precise. That is, the probability that both (1) the set of activities appearing before the event in the trace and (2) the set of activities appearing after the event in the trace are correct. For instance, the $\text{PRE}_{\triangleright\text{order}}$ annotation of 0.25 to event a in Figure 2a indicates that the probability of the set of events before a being $\{b\}$ and the set of events after a being $\{c, d\}$ is 0.25. Consequently, the probability of a difference in either set is 0.75.

Accordingly, our quality-informed alignments make it *cheaper* to skip *imprecise* events to ensure they are less likely to influence the calculations of performance measures. E.g., the log and model shown in Fig. 2 have two optimal alignments (Fig. 2c and 2d) with the same cost when data quality information is disregarded. However, as the $\text{PRE}_{\triangleright\text{order}}$ indicates that b's ordering in the trace is more likely to be precise than a, the alignment of Figure 2d is preferred.

Please note that not every combination of $\text{PRE}_{\triangleright\text{order}}$ annotations in a trace is sensible. For instance, for the trace $\langle b^{\text{PRE}_{\triangleright\text{order}}:0.1}, a^{\text{PRE}_{\triangleright\text{order}}:1}\rangle$, b has likely been moved, but a is in the correct position, which cannot both be true. As annotations encountered in practice may have been estimated, our method is robust against such impossible combinations of these annotations. To fully address the challenge of handling imprecise ordered events, a quality-informed process mining technique could construct a partial order between events, and consequently search an alignment that uses at least one total order of events that fits the identified partial order [21]. However, partial orders are outside the scope of this paper and we leave such a technique for future work.

6.1.2 Precision of concept:name ($\textsc{pre}_{\triangleright concept:name}$). If a `concept:name` value is imprecise, this value might have been recorded with errors, but still resembles the "true" value. Figure 3 shows an example of a model and a trace, where the log's activities have been recorded imprecisely: `ship` has been logged as `shipped` and `send invoice` has been logged as `sent invoice`. Alignments cannot map these logged activities to any model transition (Figure 3c), and, as only synchronous moves contribute to frequency and performance measures, these events will not contribute to these measures.

To address this issue, we remap such events to the closest transition in the model, measured by normalised Levenshtein distance [20], but only if the precision is low enough to warrant this. That is, we remap only if the annotated precision is lower than 1 minus the normalised Levenshtein distance. In our example, a standard alignment cannot map `shipped` and `ship`, thus log and model move costs are incurred (Figure 3c). The quality-informed alignment considers that the normalised Levenshtein distance of `ship` and `shipped` is $\frac{3}{7}$ and, given that $\textsc{pre}_{\triangleright \text{concept:name}}$ of `shipped` is 0.25, will attempt to remap `shipped` to `ship`, which incurs no cost. Similarly, `sent invoice` is remapped to `send invoice` and the quality-informed alignment has a cost of 0 (Figure 3d).

Note that this technique can be leveraged by giving *every* event in the log a $\textsc{pre}_{\triangleright \text{concept:name}}$ of 0, in which case the quality-informed alignment will remap every event to the closest activity in the model.

6.1.3 A Quality-Informed Conformance Checking Technique. In summary, to demonstrate how data quality annotations can be used in conformance checking, we change alignments in two ways: (1) we change the cost function for alignments and (2) we remap imprecise `concept:name` attributes.

In order to provide the formal updated cost function, we first define two helper functions that assist with remapping imprecise `concept:name` values, in which δ is the normalised Levenshtein distance [20]:

$$\text{closest}(c, T) = \{t \mid \delta(c, \lambda(t)) = max_{t' \in T} \delta(c, \lambda(t'))\}$$

$$\text{remap}(e, c, T) = \begin{cases} \lambda(t) & \text{if for } t \in \text{closest}(c, T) \\ & \textsc{pre}_{\triangleright \text{concept:name}}(e) < 1 - \delta(c, \lambda(t)) \\ c & \text{otherwise} \end{cases}$$

Then, we define the new cost function for quality-informed alignments, in which we take the minimum value of all the annotation-based costs introduced in this section. Taking the minimum ensures that each data quality annotation has maximum impact on the total cost function.

Definition 2 (Quality-informed alignment cost function). *Let e be an event, and let $t \in T$ be a transition. Then:*

$$
\begin{aligned}
cost(\overset{e}{\rightarrow}) &= \min(\textit{ACC}_{\triangleright \textit{concept:name}}(e), \textit{PRE}_{\triangleright \textit{order}}, \textit{ACC}_{\triangleright \textit{order}}, \\
&\qquad\qquad \textit{REL}(e)) \\
cost(\tbinom{e}{t}) &= \begin{cases} \max(-2\textit{ACC}_{\triangleright \textit{concept:name}}(e)^2 + 2, 2(1 - \textit{REL}(e))) \\ \qquad \textit{if remap}(e, \#_{\textit{concept:name}}(e), T) = \lambda(t) \\ \infty \quad \textit{otherwise} \end{cases} \\
cost(\underset{t}{\rightarrow}) &= \begin{cases} 0 & \textit{if } t \textit{ is invisible} \\ 1 & \textit{otherwise} \end{cases}
\end{aligned}
$$

6.2 Performance Measures

Once the model is discovered and conformance checking is done, performance measures can be computed. In this section, we study the influence of data quality annotations on alignment-based performance measures and we provide a technique to reduce this influence. First, we mathematically study the influence of data quality issues (as expressed in data quality annotations) on timestamps. Second, as a time measure is the difference between two timestamps – sojourn, waiting, service, queuing time –, we study the influence of data quality issues on these time measures. Third, we lift this to collections of time measures, which are the performance measures sought. Finally, we explicitly revisit the assumptions that were made.

6.2.1 Influence of Data Quality on a Timestamp. As follows from Section 5, three quality annotations are relevant for timestamps: relevance of the event, and accuracy and precision of the timestamp.

Relevance (REL) and Accuracy of time:timestamp ($\textit{ACC}_{\triangleright \textit{time:timestamp}}$). The relevance of an event describes the likelihood that the event is relevant for a particular analysis and the accuracy of a timestamp indicates the probability that the timestamp has been recorded correctly within the bounds specified by its precision. As performance measures are computed after conformance checking, the conformance checking computation (Section 6.1) has already decided on whether the event should be included, thus we can assume the event has been included in the correct position. Thus, we weigh its contribution to performance measures with REL and $\text{ACC}_{\triangleright \text{time:timestamp}}$.

Precision of time:timestamp ($\textit{PRE}_{\triangleright \textit{time:timestamp}}$). Where relevance and timestamp accuracy provide a weight to a timestamp, timestamp precision transforms a timestamp into an interval. Timestamps might be recorded in an imprecise manner, and the annotation PRE of the attribute `time:timestamp` indicates this. This annotation can have the values `ms`, `sec`, `min`, `hr`, `day`, `mon` and `yr`. The timestamp and its precision thus define an interval, given by:

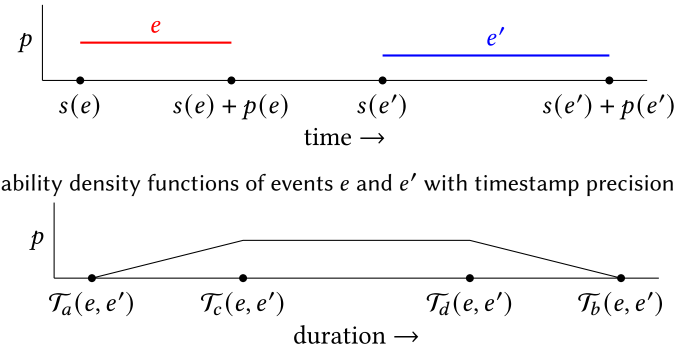
$$s(e) = \text{start of the interval for event } e \tag{1}$$

$$p(e) = \text{length of the interval for event } e \tag{2}$$

For instance, a timestamp of `2020-14-04 09:07:34.936` with a precision of `min` yields t = `2020-14-04 09:07:00.000` and $p = 59999$ms.

We assume that the timestamp is uniformly distributed over this interval, as the uniform distribution offers the maximum entropy and assumes the least presumed knowledge [17, Theorem 1.3] (we will revisit this assumption A1 in Section 6.2.4).

6.2.2 Influence of Data Quality on a Time Measure. In the previous section, we have shown that for each event, a weight and an interval is available. A time measure, such as sojourn time, service



(a) Probability density functions of events e and e' with timestamp precision intervals.

(b) Probability density function of the difference between e and e'.

Fig. 4. Probability density functions of two events e and e' with imprecise timestamps.

time, queuing time or waiting time, is the difference between two events e and e' (in particular, between their timestamps). Next, we study the influence of data quality on such a time measure.

Considering the relevant quality annotations, a time measure between e and e' is weighted according to the minimum of quality annotations regarding relevance and accuracy, to ensure each quality issue has maximum impact:

$$w(e, e') = \min(\textsc{acc}_{\triangleright\text{time:timestamp}}(e), \textsc{rel}(e), \textsc{acc}_{\triangleright\text{time:timestamp}}(e'), \textsc{rel}(e'))$$

Process discovery and conformance checking ensure that the order between these events has been established: without loss of generality, we assume that e happened before e'. Given this constraint, we assume that e and e' are otherwise independent (we will revisit this assumption A2 in Section 6.2.4). Then, we derive that the time measure $e' - e$ follows a trapezoidal distribution with parameters $(\mathcal{T}_a, \mathcal{T}_c, \mathcal{T}_d, \mathcal{T}_b)$ – a detailed derivation from Killmann and von Collani [17] is included in Appendix C.

Figure 4 illustrates two events e and e' having precision intervals (Fig. 4a), and shows the probability density function of the difference between e and e' (Fig. 4b). As the values of $\mathcal{T}_a$, $\mathcal{T}_b$, $\mathcal{T}_c$ and $\mathcal{T}_d$ might coincide, eight different combinations of overlapping parameters can occur [16][3].

6.2.3 *Influence of Data Quality on a Performance Measure.*

From the previous section, we end up with a collection of weighted $w(e, e')$ trapezoid distributions $\mathcal{T}(e, e')$. Next, we compute the weighted mean and standard deviation for all such pairs of events e, e' in an event log L. The weighted mean $\hat{\mu}$ of time measures over the event log L is:

$$\hat{\mu} = \frac{\sum_{(e,e') \in L} \mathcal{T}(e, e') w(e, e')}{\sum_{(e,e') \in L} w(e, e')} \tag{3}$$

Intuitively, the weighted standard deviation considers the average difference between a value and the mean over all values. In our case, the mean over all values is $\hat{\mu}$, while our values are not single values but trapezoidal distributions – one for each pair of timestamps $\mathcal{T}(e, e')$. If we take a small range (dx) centred around x out of $\mathcal{T}(e, e')$, then the contribution of this range to the standard deviation is proportional to the probability of x and is approximated by the probability density function of the trapezoid distribution $pdf(e, e', x)$ dx. Integrating over x yields the total

[3] In addition to the cases identified in [16], we also distinguish the case where all of $\mathcal{T}_a$, $\mathcal{T}_b$, $\mathcal{T}_c$ and $\mathcal{T}_d$ coincide. In that case, the time measure is a constant 0.

contribution to the standard deviation of this single time measure e, e', which leads to:

$$\varphi(e, e') = \int_{\mathcal{T}_a(e,e')}^{\mathcal{T}_b(e,e')} (x - \hat{\mu})^2 pdf(e, e', x)\, \mathrm{d}x \tag{4}$$

$$\hat{\sigma} = \sqrt{\frac{\sum_{(e,e')\in L} \varphi(e, e') w(e, e')}{\sum_{(e,e')\in L} w(e, e')}} \tag{5}$$

Notice that for the base case of one precise timestamp and one imprecise timestamp that are not overlapping, Equation (4) equates the standard deviation of a uniform distribution $\frac{(\mathcal{T}_b - \mathcal{T}_a)^2}{12}$ [16]. A closed expression for φ was derived by considering the three parts separated by $\mathcal{T}_c$ and $\mathcal{T}_d$ of the trapezoidal distribution in isolation, and solving these using Wolfram Alpha. Please note that the eight special cases for coinciding $\mathcal{T}_a$, $\mathcal{T}_b$, $\mathcal{T}_c$ and $\mathcal{T}_d$ are not necessary here, as in these cases the φ_1, φ_2 and φ_3 integrals will consist of empty domains, and hence equal 0 where appropriate (please refer to Appendix Dfor more details).

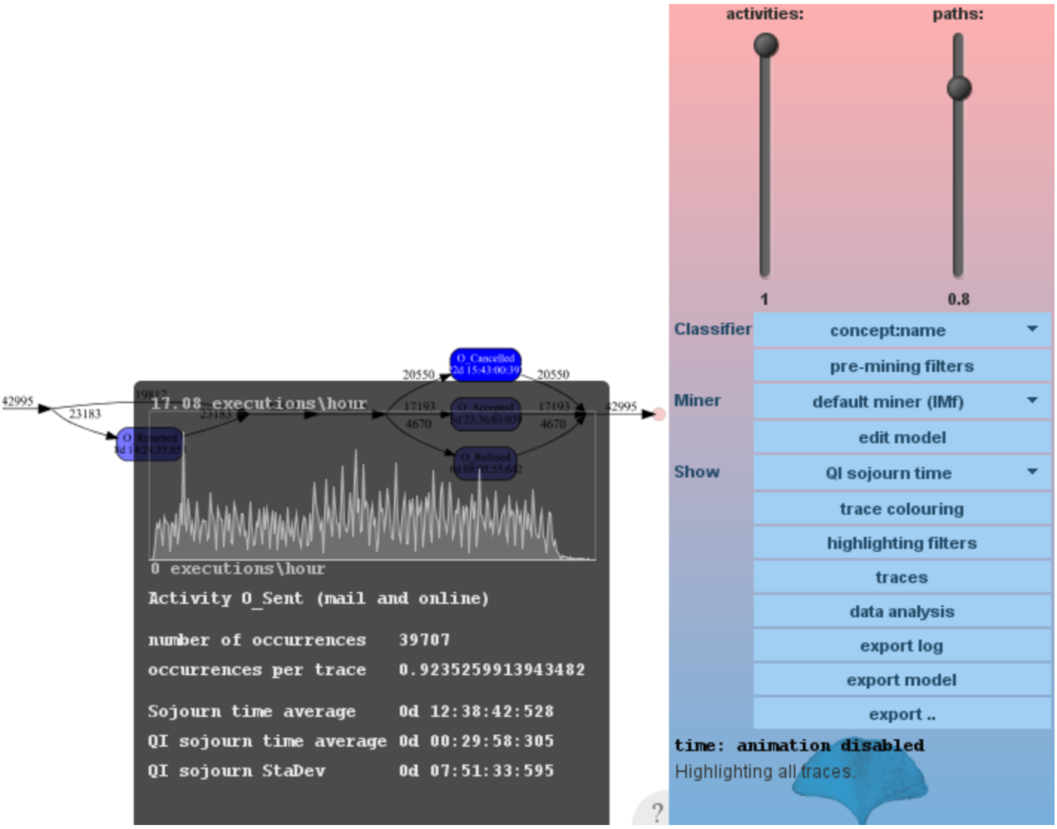
In our implementation, we report on $\hat{\mu}$ and $\hat{\sigma}$ to indicate the weighted time measure and to give an idea of its reliability.

6.2.4 Discussion: Assumptions. In this section, we revisit the assumptions and explore when they could hold, and we argue why stronger assumptions would not be realistic.

A1 Uniformly distributed timestamps. Given two events e and e' that are being considered for a time measure, process discovery and conformance checking have already established their order. According to the semantics of process models, e is finished first before e' can happen. If we denote the actual, unknown, timestamps of e and e' with $\widetilde{e}$ and $\widetilde{e'}$, then we know that $\widetilde{e} \leq \widetilde{e'}$. Thus, we assume that e is uniformly distributed over $[s(e), \min(s(e) + p(e), s(e') + p(e'))]$, and that e' is uniformly distributed over interval of $[\max(\widetilde{e}, s(e')), s(e') + p(e')]$.
We argue that these assumptions are reasonable in most cases: if a human task takes a couple of minutes and the precision is seconds, then there is no reason why a millisecond value of 2 would be more likely than a millisecond value of 369. Similarly, if a nurse registers in the evening that a procedure was performed on a patient earlier that day, then, without further information, we argue that a uniform distribution is reasonable. However, in some cases this assumption might not hold, for instance in manufacturing settings where a machine might have a normally or otherwise distributed timing profile. Our derivation is flexible towards these cases, as pdf in Equation (4) could be replaced with a matching probability density function.

A2 Independent timestamps. It is assumed that the timestamps of e and e' are independent, apart from assuming that e' happens at the same time or after e, which is guaranteed by the absence of concurrency whenever two timestamps are compared. In standard BPM contexts, e' occurs whenever e has completed and the necessary resources are available. After occurrence of e, no more resources are held by it, thus, apart from their order, e and e' can be assumed to be independent.

A3 Dependent time measures. Over a log L, the time measures are typically not independent if they include waiting time (waiting time and sojourn time): in typical BPM models, it is assumed that activities are only executed when all resources are available. Thus, when observing a high time measure, which indicates that there was a long waiting time for resources, it is likely that the next time measure is also high. Consequently, the Central Limit Theorem does not apply and normality cannot be assumed.



```
17.08 executions\hour
0 executions\hour
Activity O_Sent (mail and online)

number of occurrences    39707
occurrences per trace    0.9235259913943482

Sojourn time average     0d 12:38:42:528
QI sojourn time average  0d 00:29:58:305
QI sojourn StaDev        0d 07:51:33:595
```

activities: 1

paths: 0.8

Classifier: concept:name

Miner: default miner (IMf)

Show: QI sojourn time

time: animation disabled

Highlighting all traces.

Fig. 5. Quality-Informed visual Miner (QIvM) in ProM.

6.3 Implementation

We implemented the techniques introduced in Section 6 as a plug-in of the ProM framework[4]. Our implementation, *Quality-Informed visual Miner* (QIvM) extends the Visual Miner [18, 19]. The Visual Miner enables analysts to study an event log by automatically discovering a process model, applying conformance checking and measuring performance, and offers quick filters such that analysts can drill down into areas of interest. QIvM extends this with the techniques introduced in Section 6 by replacing the conformance checking computations (invisible for the end user) and performance measures with quality-informed ones, while indicating the reliability of these measures using the quality-informed standard deviation (Equation (5)). Figure 5 shows a screenshot, where the quality-informed sojourn time is projected on the activities in the model. Furthermore, QIvM allows analysts to inspect data quality annotations aggregated over traces and events.

7 DEMONSTRATION: THE IMPACT OF DATA QUALITY ANNOTATIONS ON PROCESS MINING INSIGHTS

In this section, we test the utility of data quality annotations by reporting on three experiments. These experiments aim to study the complex influence of data quality issues on (i) conformance checking and (ii) the reliability of performance measures.

[4] http://www.promtools.org/doku.php?id=nightly

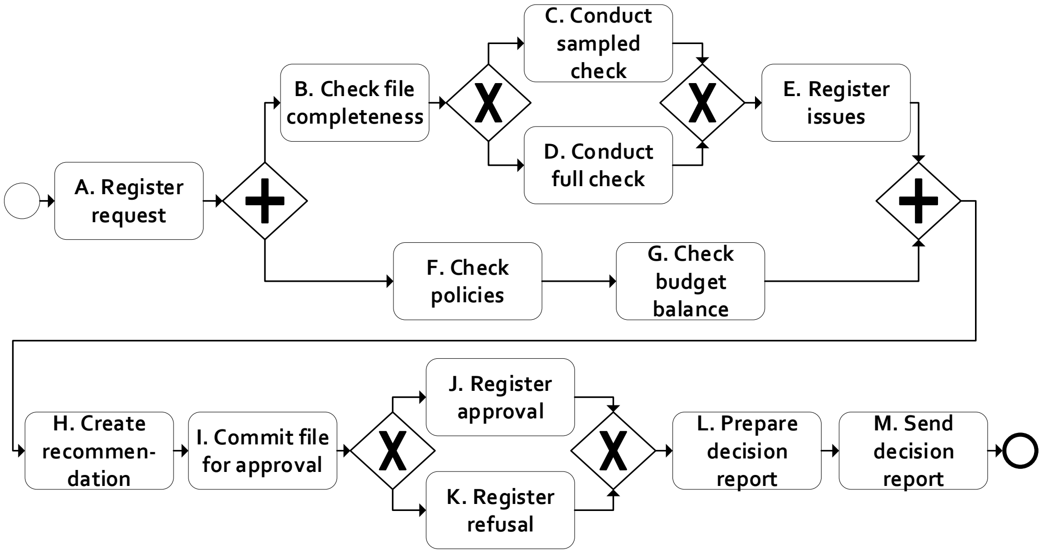


Fig. 6. Process Model for Synthetic Log Generation.

7.1 Data

Both synthetic and real-world event logs are used in the experiments.

Synthetic log. To demonstrate the approach in a setting with a ground truth, we created an event log resulting from the execution of the process model shown in Figure 6, consisting of 13 activities. The process model is fully parameterised to make it executable. Parameters include the activity durations, routing probabilities, and resource availability calendars. Specifying these parameters enables the creation of a synthetic event log[5]. The synthetic log was generated using `SimPy`[6], an open-source framework for discrete event simulation in `Python`. The synthetic log is the basis to generate a series of logs with varying degrees of imputed data quality issues. These data quality issues were imputed (discussed in Section 7.2) using programming language, `R`[7].

Real-world logs. To demonstrate the practical applicability of our approach we used three publicly available event logs. The BPI Challenge 2017-Offer log[8] for its complexity, the BPI Challenge 2012-W activities log[9] as it has both start and complete timestamps, and the BPIC 2018-control summary log[10], which has a high degree of parallelism.

7.2 Experimental Design

Three experimental settings are considered: (i) Experiment 1 with event logs containing inaccurate timestamps, (ii) Experiment 2 with event logs containing imprecise timestamps, and (iii) Experiment 3 with event logs containing both inaccurate and imprecise timestamps.

[5] The synthetic logs with the full parameter specification are available online at https://tinyurl.com/ceh8z8f4

[6] https://simpy.readthedocs.io.

[7] https://www.r-project.org

[8] https://doi.org/10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b

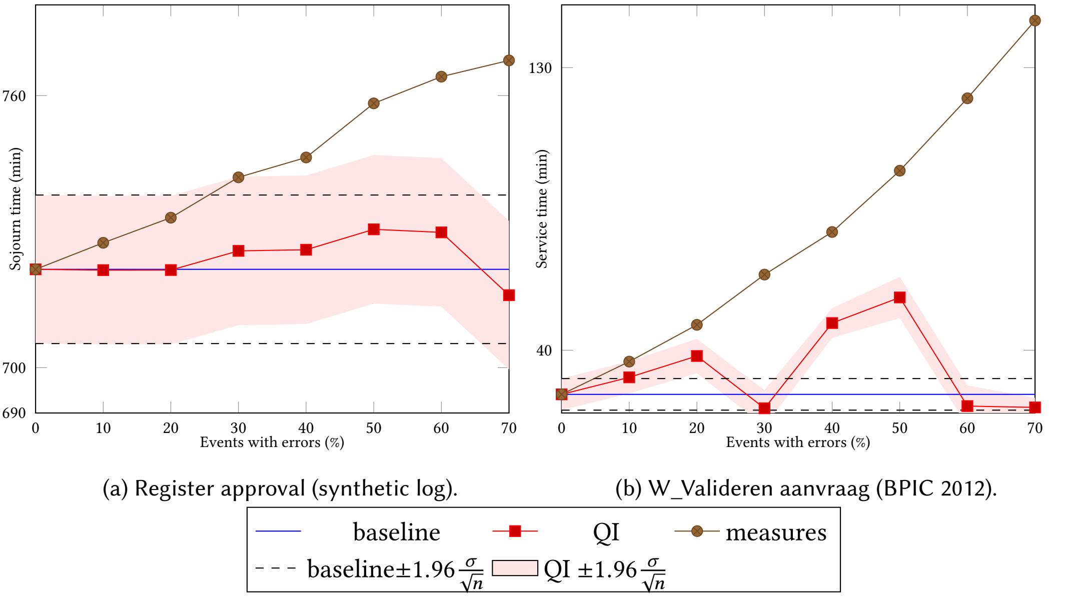
[9] https://doi.org/10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b

[10] https://doi.org/10.4121/uuid:3301445f-95e8-4ff0-98a4-901f1f204972

From each log (synthetic and real-world), eight variants are created where each variant has an increasing proportion of the log being influenced by the respective data quality issue(s). In each variant for experiment 1 and 2, a data quality issue is purposefully introduced for $X\%$ of the events (with X ranging from 0 to 70 with steps of 10) for arbitrarily chosen activities that are not next to each other. For experiment 3, each variant consists of data quality issues introduced in experiment 1 and 2, i.e., each variant has two data quality issues. The use of eight variants enables us to study the ability of data quality annotations in handling increasing levels of noise. Details related to the number of activities impacted with the data quality issue are explained below for each experiment:

- **Experiment 1 – Inaccurate timestamps:** The timestamps of up to three activities in an event log were shifted by a certain amount of time. We chose up to three activities such that they do not overlap and we can illustrate the influence of inaccurate timestamps on performance measures in isolation, as well as address the data quality issue by our technique, in this case inaccurate timestamps. The modified events were given an annotation of *quality:accuracy:time:timestamp* being 0, and if events need to be re-ordered, *quality:order:accuracy* was set to 0. In the synthetic log, we randomly shifted the timestamp between 600-720 seconds for `Register issues` and 300-500 seconds for `Register approval` and `Register refusal`. In BPIC 2017, the timestamp of `O_sent (mail and online)` and `O_cancelled` was shifted by 24 hours. In BPIC 2012, the complete timestamp of `W_Completeren aanvraag` and `W_Valideren aanvraag` was shifted by 15 minutes. In BPIC 2018, the timestamp of `save` and `finish editing` was shifted by 24 hours. The timeshifts were chosen such that they result in re-ordering of events and influence the performance measures.
- **Experiment 2 – Imprecise timestamps:** The milliseconds and seconds of timestamps were set to 0 for certain activities as discussed next. Similar to Experiment 1, choosing certain activities enabled us to focus on the influence of data quality annotations on imprecise timestamps. The modified events were given an annotation of *quality:precision:time:timestamp* being 'min' to express that the precision level is minutes. Moreover, if events need to be re-ordered, *quality:order:precision* was set to 0. For the synthetic log, the precision of timestamps of `Register issues`, `Register approval`, and `Register refusal` was set to minutes. In BPIC 2017, timestamp of activities of five resources (`User_1`, `User_3`, `User_49`, `User_10`, and `User_28`), was set to minutes. We chose activities of five resources to account for situations when particular resources populate imprecise timestamps. Finally, the timestamp of `W_Nabellen offertes` and `W_Completeren aanvraag` in BPIC 2012, and the timestamp of `save` in BPIC 2018 was set to minutes.
- **Experiment 3 – Inaccurate and Imprecise timestamps:** In this experiment, both, inaccurate (experiment 1) and imprecise (experiment 2) data quality issues were inserted in logs using the same parameters. This experiment was performed to study the influence of two data quality issues (inaccurate and imprecise timestamps) on performance measures and understand how data quality annotations can assist in obtaining quality-informed measures.

For each log without data quality issues, we obtain a process model: for the synthetic log, we use the ground-truth model (Figure 6), while for the real-life logs, process models were discovered using Inductive Miner - infrequent (IM) [18]. As process discovery is not part of this evaluation, these models are assumed to be the ground truth. Next, for each combination of a model and one of the data quality imputed logs, we apply alignments and measure alignment-based performance [49] (standard measures), as well as the quality-informed measures (QImeasures) introduced in Section 6. We believe that by taking data quality annotations into account, QImeasures should be closer to the baseline measures or ground truth. In addition, we calculate the quality-informed standard deviation. Following Pika et al. [37] and Zhang and Serban [57], and after empirical verification



(a) Register approval (synthetic log).

(b) W_Valideren aanvraag (BPIC 2012).

Fig. 7. Experiment 1 results.

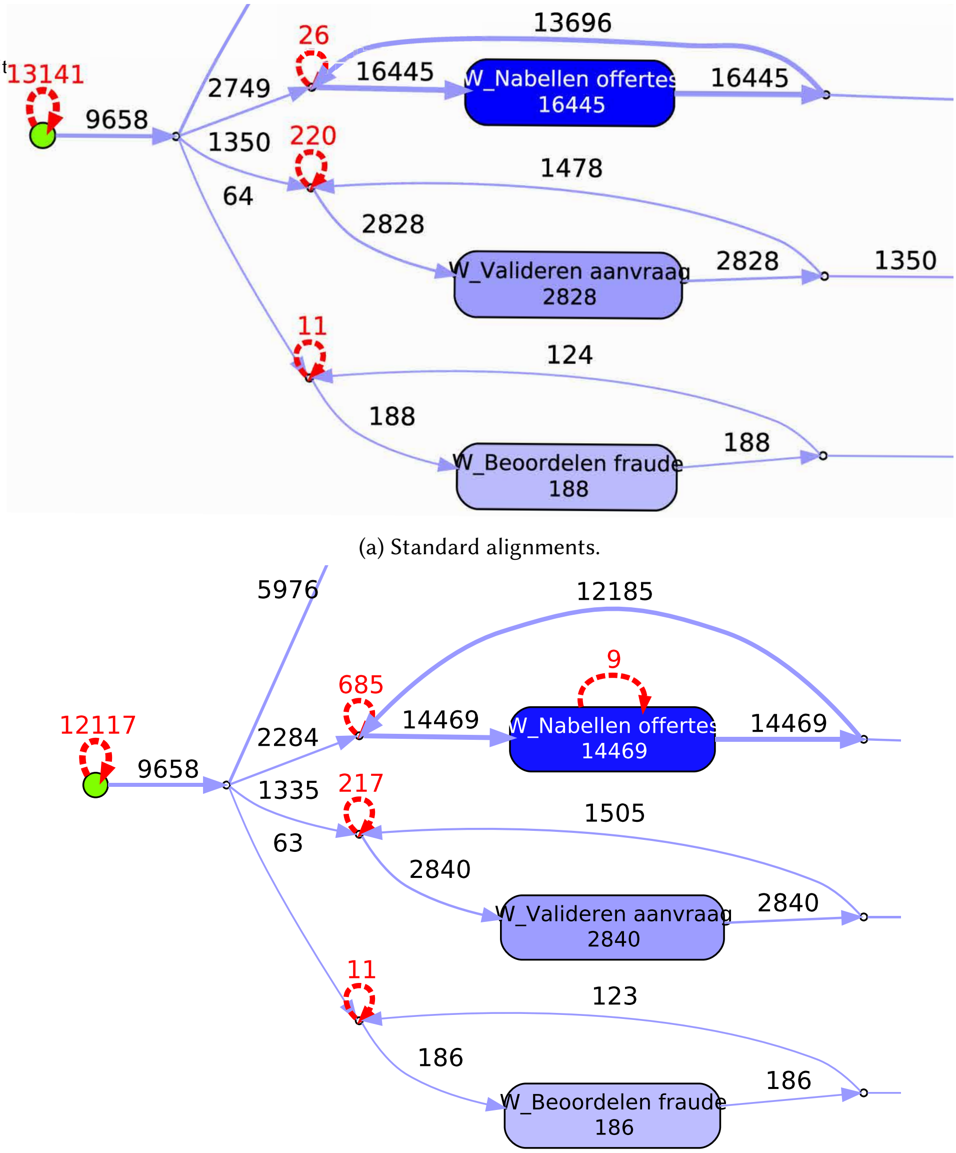
using statistical plots of activity durations, we assume that activity durations follow log-normal distribution, therefore logarithms of activity duration follow a normal distribution. In accordance with Olsson [25] we use the formula $QImeasures \pm z\frac{\sigma}{\sqrt{n}}$ to calculate the 95% confidence interval of the average of QI performance measure. The same formula is used to calculate the confidence interval of the baseline measures.

7.3 Results

The results convey that taking data quality annotations into account generates performance measures which are closer to the baseline and, hence, more accurate and reliable. The use of annotations also enable quality-informed alignments and the calculation of quality-informed standard deviation, which gives an indication of the range in which the accurate value lies. While the complete results for all experiments are present in Appendix E, this section shows the results of a few activities for illustrative purposes.

7.3.1 Experiment 1. The results of Experiment 1 show that QIvM stays closer to the baseline measure than the standard measures obtained using Inductive miner - infrequent (IM) as events with inaccurate timestamps were disregarded in the calculations of performance measures. Figure 7 shows the results for two activities `Register approval` (synthetic log) and `W_Valideren aanvraag` (BPIC 2012). Moreover, our quality-informed alignments would have corrected for the events that were re-ordered due to shift in timestamp. Furthermore, we observe that QI measures vary from baseline with increasing noise (greater than 30%). This is because a greater number of events are disregarded when calculating the performance measures, impacting the overall value. Experiment 1 results clearly demonstrate that by taking data quality annotations into account, the performance measures are more accurate and hence reliable.

7.3.2 Experiment 2. The results of Experiment 2 convey that the performance measures of QIvM are similar to IvM, both not being close to the baseline measures. While an accurate value is not

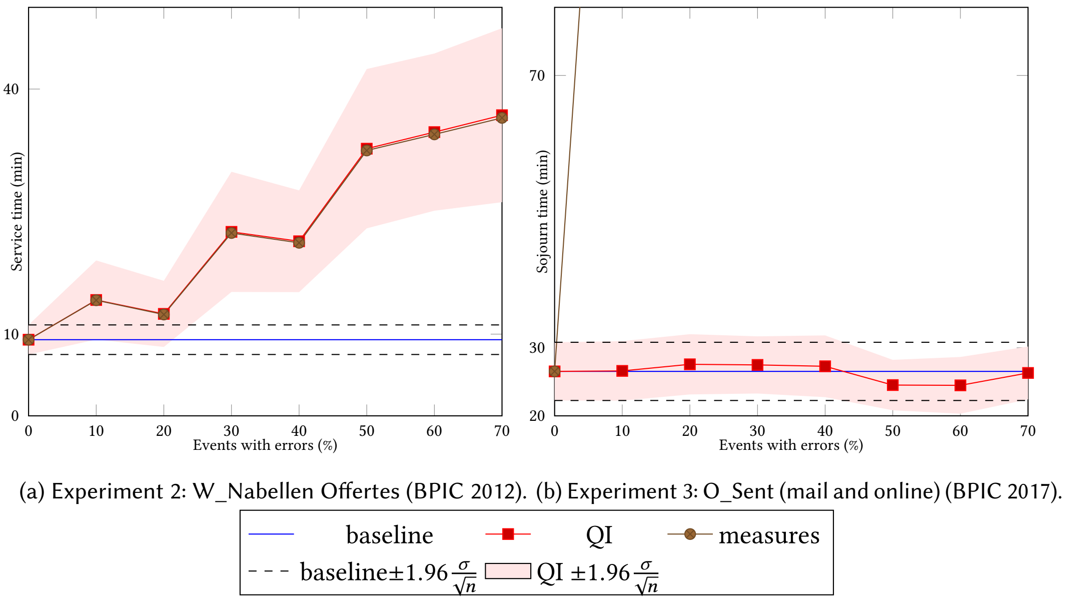


(a) Standard alignments.

(b) Quality-informed alignments.

Fig. 8. Alignments and quality-informed alignments.

attained, we believe that it is better to have an indication of the range in which the accurate value lies. This indication is provided by the QI standard deviation, which is unique to QIvM. The results of `W_Nabellen Offertes` (BPIC 2012) are present in Figure 9a. As can be seen from Figure 9a, the QI confidence interval of service time overlaps the baseline CI for low noise levels, which can therefore be used to predict the actual service time and the range in which the services time lies. The interval deviates when the noise is greater than 40%. It is interesting to see how the imprecision of minute amplifies to a difference of more than 30 minutes on average in sojourn time. These observations illustrate how data quality issues can have a significant impact on process performance insights.



(a) Experiment 2: W_Nabellen Offertes (BPIC 2012). (b) Experiment 3: O_Sent (mail and online) (BPIC 2017).

Fig. 9. Experiment 2 and 3 results.

Furthermore, Figure 8 demonstrates the influence of data quality annotations on conformance checking. The number of deviations are considerably different for quality-informed alignments, as described in Section 6.1. Quality-informed alignments use *quality:order:precision*, which increases the number of model moves for W_Nabellen Offertes (BPIC 2012) as compared to standard alignments. Additionally, the QI-alignments project nine log moves during the execution of W_Nabellen Offertes, which indeed was made imprecise in our experiment. The example shows how event re-ordering has been considered and acted upon by the quality-informed alignments.

7.3.3 Experiment 3. The results of Experiment 3 demonstrates that by using annotations related to accuracy and precision, QIvM provides measures closer to the baseline than IvM. The results of Experiment 3 are illustrated in Figure 9b. IvM performs as expected because 24 hours were added to the timestamps of O_Sent (mail and online) (BPIC 2017). However, by using data quality annotation related to timestamp accuracy, the QI performance measures and their confidence intervals stay closer to the baseline. For instance, in Figure 9b, the QI standard deviation closely overlaps the baseline confidence interval. The results reinforce the observation that taking data quality annotations into account can instil increased confidence in process mining results.

8 DISCUSSION AND CONCLUSION

Real-life event logs are often accompanied with data quality issues, which can lead to counterintuitive or even misleading process mining outcomes if they are not taken into account. As a first step to enable quality-informed process mining insights, we proposed a set of standardised data quality annotations. The annotations were defined based on requirements inspired by data quality literature in the process mining field. Being aware of event logs' shortcomings by means of data quality annotations can significantly contribute to the reliability and trustworthiness of results, empowering the field of process mining. The data quality annotations we propose are standardised

(R2), support recording of information about the type of data quality issue (R3), allow recording of information about data transformations (R4), and can enable maintaining information at different levels of aggregation (R5). To demonstrate the potential of the proposed data quality annotations, we developed the *Quality-Informed visual Miner* plug-in, and tested the utility of the annotations in obtaining quality-informed process mining insights. We showed that the data quality annotations can be used by process mining algorithms (R2) and their use can result in more reliable process mining insights (R1).

The contributions of this paper must be reflected against some potential limitations. First, we do not claim that the proposed data quality annotations are complete. They are based on data quality literature and fulfil the formulated requirements inspired by literature. We intend to further validate these requirements and propose new annotations by seeking input from the process mining community. Second, we acknowledge that techniques to insert the annotations proposed in this paper into an event log are required. Having an annotated log is a premise to conduct quality-informed process mining. This is beyond the scope of this paper that focuses on introducing structured data quality annotations for event logs, as well as demonstrating their potential for quality-informed process mining. Section 5 also pointed to some valuable contributions in literature, which can form a basis to develop an efficient and user-friendly event log annotation technique that builds upon the conceptual foundations in this paper. Third, our experiments demonstrate the applicability of a subset of the proposed annotations within the context of novel quality-informed process mining algorithms, which can also be configured according to organisational context. Nevertheless, we also highlight how the remaining annotations could be used in Section 6 and Appendix B. Moreover, the applicability is also discussed in Section 5.4 where we propose an XES extension that utilises the annotations present in this paper. Details of the proposed extension are present in Appendix A. We also explain the applicability of annotations using a snapshot of an annotated event log.

This work open various avenues for future work. First, algorithms that detect data quality issues and populate logs with data quality annotations are required. Second, the plug-in presented in this paper can be improved to include transformations applied to the log. For example, for each activity instance, information regarding whether it was created or updated can be added. This also suggests the need for algorithms that repair data quality issues in the log and populate it with annotations. Third, the data quality annotations can be used in many other interesting ways such as the use of partial orders to address imprecise ordering of events. Fourth, data quality annotations can be extended to other levels such as activity to account for different contexts in which data quality errors can be introduced. Finally, to further evaluate our technique, we aim to evaluate and optimise it using specific contexts: for instance, in an emergency healthcare context, a precision of seconds in timestamps may be critical, while in other contexts this precision might not be as crucial; to which our technique could be adapted by assigning weights appropriately.

REFERENCES

[1] Robert Andrews, Suriadi Suriadi, Chun Ouyang, and Erik Poppe. 2018. Towards event log querying for data quality. In *On the Move to Meaningful Internet Systems*. Springer, 116–134. https://doi.org/10.1007/978-3-030-02610-3_7

[2] Robert Andrews, Christopher GJ van Dun, Moe Thandar Wynn, Wolfgang Kratsch, MKE Röglinger, and Arthur HM ter Hofstede. 2020. Quality-informed semi-automated event log generation for process mining. *Decision Support Systems* 132 (2020), 113265. https://doi.org/10.1016/j.dss.2020.113265

[3] Robert Andrews, Moe T Wynn, Kirsten Vallmuur, Arthur HM ter Hofstede, Emma Bosley, Mark Elcock, and Stephen Rashford. 2019. Leveraging data quality to better prepare for process mining: an approach illustrated through analysing road trauma pre-hospital retrieval and transport processes in Queensland. *Int. J Environ Res Public Health* 16, 7 (2019), 1138. https://doi.org/10.3390/ijerph16071138

[4] Dina Bayomie, Ahmed Awad, and Ehab Ezat. 2016. Correlating Unlabeled Events from Cyclic Business Processes Execution. In *Int. Conf. on Advanced Information Systems Engineering (LNCS, Vol. 9694)*. Springer, 274–289. https://doi.org/10.1007/978-3-319-39696-5_17

[5] Jagadeesh Chandra Bose, R. S. Mans, and Wil M. P. van der Aalst. 2013. Wanna improve process mining results?. In *IEEE Symposium on Computational Intelligence and Data Mining*. IEEE, 127–134. https://doi.org/10.1109/CIDM.2013.6597227

[6] Raffaele Conforti, Marcello La Rosa, and A ter Hofstede. 2018. Timestamp Repair for Business Process Event Logs. (2018). https://minerva-access.unimelb.edu.au/handle/11343/209011

[7] R. Conforti, M. L. Rosa, and A. H. M. ter Hofstede. 2017. Filtering Out Infrequent Behavior from Business Process Event Logs. *IEEE Transactions on Knowledge and Data Engineering* 29, 2 (2017), 300–314. https://doi.org/10.1109/TKDE.2016.2614680

[8] Prabhakar M. Dixit, Suriadi Suriadi, Robert Andrews, Moe Thandar Wynn, Arthur H. M. ter Hofstede, Joos C. A. M. Buijs, and Wil M. P. van der Aalst. 2018. Detection and Interactive Repair of Event Ordering Imperfection in Process Logs. In *Int. Conf. on Advanced Information Systems Engineering (LNCS, Vol. 10816)*. Springer, 274–290. https://doi.org/10.1007/978-3-319-91563-0_17

[9] Dominik Andreas Fischer, Kanika Goel, Robert Andrews, Christopher G. J. van Dun, Moe Thandar Wynn, and Maximilian Röglinger. 2020. Enhancing Event Log Quality: Detecting and Quantifying Timestamp Imperfections. In *Int. Conf. on Business Process Management (LNCS, Vol. 12168)*. Springer, 309–326. https://doi.org/10.1007/978-3-030-58666-9_18

[10] Chiara Di Francescomarino, Chiara Ghidini, Sergio Tessaris, and Itzel Vázquez Sandoval. 2015. Completing Workflow Traces Using Action Languages. In *Int. Conf. on Advanced Information Systems Engineering (LNCS, Vol. 9097)*. Springer, 314–330. https://doi.org/10.1007/978-3-319-19069-3_20

[11] Aindrila Ghosh, Mona Nashaat, James Miller, and Shaikh Quader. 2021. Context-Based Evaluation of Dimensionality Reduction Algorithms—Experiments and Statistical Significance Analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 2 (2021), 1–40. https://doi.org/10.1145/3428077

[12] Terry Halpin. 2005. ORM 2 graphical notation. *Technical Report ORM2–02* (2005).

[13] Terry Halpin. 2010. Object-role modeling: Principles and benefits. *International Journal of Information System Modeling and Design (IJISMD)* 1, 1 (2010), 33–57. https://doi.org/10.4018/jismd.2010092302

[14] Mustafa Jarrar. 2007. Mapping ORM into the SHOIN/OWL description logic. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 729–741. https://doi.org/10.1007/978-3-540-76888-3_95

[15] Paul Johannesson and Erik Perjons. 2014. *An Introduction to Design Science*. Springer. https://doi.org/10.1007/978-3-319-10632-8

[16] Raghu N Kacker and James F Lawrence. 2007. Trapezoidal and triangular distributions for Type B evaluation of standard uncertainty. *Metrologia* 44, 2 (2007), 117. https://doi.org/10.1088/0026-1394/44/2/003

[17] Frank Killmann and Elart von Collani. 2001. A note on the convolution of the uniform and related distributions and their use in quality control. *Stochastics and Quality Control* 16, 1 (2001), 17–41. https://doi.org/10.1515/EQC.2001.17.

[18] Sander J. J. Leemans, Dirk Fahland, and Wil M. P. van der Aalst. 2013. Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour. In *Business Process Management Workshops (LNBIP, Vol. 171)*. Springer, 66–78. https://doi.org/10.1007/978-3-319-06257-0_6

[19] Sander J. J. Leemans, Erik Poppe, and Moe Thandar Wynn. [n.d.]. Directly Follows-Based Process Mining: Exploration & a Case Study. In *Int. Conf. on Process Mining*. IEEE, 25–32. https://doi.org/10.1109/ICPM.2019.00015

[20] V. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics-Doklady* 10, 8 (1966), 707–710.

[21] Xixi Lu, Dirk Fahland, and Wil M. P. van der Aalst. 2014. Conformance Checking Based on Partially Ordered Event Data. In *Business Process Management Workshops (LNBIP, Vol. 202)*. Springer, 75–88. https://doi.org/10.1007/978-3-319-15895-2_7

[22] Qian Ma, Yu Gu, Wang-Chien Lee, Ge Yu, Hongbo Liu, and Xindong Wu. 2020. REMIAN: Real-Time and Error-Tolerant Missing Value Imputation. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14, 6 (2020), 1–38. https://doi.org/10.1145/3412364

[23] Niels Martin, Antonio Martinez-Millana, Bernardo Valdivieso, and Carlos Fernández-Llatas. 2019. Interactive Data Cleaning for Process Mining: A Case Study of an Outpatient Clinic's Appointment System. In *Business Process Management Workshops (LNBIP, Vol. 362)*. Springer, 532–544. https://doi.org/10.1007/978-3-030-37453-2_43

[24] Hoang Thi Cam Nguyen, Suhwan Lee, Jongchan Kim, Jonghyeon Ko, and Marco Comuzzi. 2019. Autoencoders for improving quality of process event logs. *Expert Systems with Applications* 131 (2019), 132–147. https://doi.org/10.1016/j.eswa.2019.04.052

[25] Ulf Olsson. 2005. Confidence intervals for the mean of a log-normal distribution. *Journal of Statistics Education* 13, 1 (2005).

[26] Aytuğ Onan. 2019. Topic-enriched word embeddings for sarcasm identification. In *Computer Science On-line Conference*. Springer, 293–304. https://doi.org/10.1007/978-3-030-19807-7_29

[27] Aytuğ Onan. 2019. Two-stage topic extraction model for bibliometric data analysis based on word embeddings and clustering. *IEEE Access* 7 (2019), 145614–145633. https://doi.org/10.1109/ACCESS.2019.2945911

[28] Aytuğ Onan. 2020. Mining opinions from instructor evaluation reviews: A deep learning approach. *Computer Applications in Engineering Education* 28, 1 (2020), 117–138. https://doi.org/10.1002/cae.22179

[29] Aytuğ Onan. 2020. Sentiment Analysis in Turkish Based on Weighted Word Embeddings. In *2020 28th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 1–4. https://doi.org/10.1109/SIU49456.2020.9302182

[30] Aytuğ Onan. 2020. Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. *Concurrency and Computation: Practice and Experience* (2020), e5909. https://doi.org/10.1002/cpe.5909

[31] Aytug Onan, Hasan Bulut, and Serdar Korukoglu. 2017. An improved ant algorithm with LDA-based representation for text document clustering. *Journal of Information Science* 43, 2 (2017), 275–292. https://doi.org/10.1177/0165551516638784

[32] Aytuğ Onan and Mansur Alp Toçoğlu. 2020. Weighted word embeddings and clustering-based identification of question topics in MOOC discussion forum posts. *Computer Applications in Engineering Education* (2020). https://doi.org/10.1002/cae.22252

[33] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. 2008. A Design Science Research Methodology for Information Systems Research. *J. Management Information Systems* 24, 3 (2008), 45–77. https://doi.org/10.2753/MIS0742-1222240302

[34] Marco Pegoraro, Merih Seran Uysal, and Wil M. P. van der Aalst. 2020. Conformance Checking over Uncertain Event Data. *CoRR* abs/2009.14452 (2020).

[35] Marco Pegoraro and Wil M. P. van der Aalst. 2019. Mining Uncertain Event Data in Process Mining. In *Int. Conf. on Process Mining*. IEEE, 89–96. https://doi.org/10.1109/ICPM.2019.00023

[36] Anastasiia Pika et al. 2020. Privacy-Preserving Process Mining in Healthcare. *Int J Environ Res Public Health* 17, 5 (2020), 1612. https://doi.org/10.3390/ijerph17051612.

[37] Anastasiia Pika, Wil M. P. van der Aalst, Colin J. Fidge, Arthur H. M. ter Hofstede, and Moe Thandar Wynn. 2012. Predicting Deadline Transgressions Using Event Logs. In *Business Process Management Workshops (LNBIP, Vol. 132)*. Springer, 211–216. https://doi.org/10.1007/978-3-642-36285-9_22

[38] Anastasiia Pika, Moe Thandar Wynn, Stephanus Budiono, Arthur H. M. ter Hofstede, Wil M. P. van der Aalst, and Hajo A. Reijers. 2019. Towards Privacy-Preserving Process Mining in Healthcare. In *Business Process Management Workshops (LNBIP, Vol. 362)*. Springer, 483–495. https://doi.org/10.1007/978-3-030-37453-2_39

[39] Majid Rafiei and Wil M. P. van der Aalst. 2020. Privacy-Preserving Data Publishing in Process Mining. 392 (2020), 122–138. https://doi.org/10.1007/978-3-030-58638-6_8

[40] Andreas Rogge-Solti, Ronny Mans, Wil M. P. van der Aalst, and Mathias Weske. 2013. Repairing Event Logs Using Timed Process Models. In *On the Move to Meaningful Internet Systems (LNCS, Vol. 8186)*. Springer, 705–708. https://doi.org/10.1007/978-3-642-41033-8_89

[41] Sareh Sadeghianasl, Arthur H. M. ter Hofstede, Suriadi Suriadi, and Selen Turkay. 2020. Collaborative and Interactive Detection and Repair of Activity Labels in Process Event Logs. In *Int. Conf. on Process Mining*. IEEE, 41–48. https://doi.org/10.1109/ICPM49681.2020.00017

[42] Wei Song, Hans-Arno Jacobsen, Chunyang Ye, and Xiaoxing Ma. 2016. Process Discovery from Dependence-Complete Event Logs. *IEEE Trans. Serv. Comput.* 9, 5 (2016), 714–727. https://doi.org/10.1109/TSC.2015.2426181

[43] Wei Song, Hans-Arno Jacobsen, and Pengcheng Zhang. 2019. Self-Healing Event Logs. *IEEE Transactions on Knowledge and Data Engineering* (2019), 1–1. https://doi.org/10.1109/TKDE.2019.2956520

[44] Wei Song, Xiaoxu Xia, Hans-Arno Jacobsen, Pengcheng Zhang, and Hao Hu. 2015. Heuristic Recovery of Missing Events in Process Logs. In *IEEE Int. Conf. on Web Services*. IEEE Computer Society, 105–112. https://doi.org/10.1109/ICWS.2015.24

[45] Suriadi Suriadi, Robert Andrews, Arthur H. M. ter Hofstede, and Moe Thandar Wynn. 2017. Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. *Information Systems* 64 (2017), 132–150. https://doi.org/10.1016/j.is.2016.07.011

[46] Arava Tsoury, Pnina Soffer, and Iris Reinhartz-Berger. 2020. Data Impact Analysis in Business Processes. *Bus. Inf. Syst. Eng.* 62, 1 (2020), 41–60. https://doi.org/10.1007/s12599-019-00611-5

[47] Wil M. P. van der Aalst. 2016. *Process Mining - Data Science in Action, Second Edition*. Springer. https://doi.org/10.1007/978-3-662-49851-4

[48] Wil M. P. van der Aalst et al. 2011. Process Mining Manifesto. 99 (2011), 169–194. https://doi.org/10.1007/978-3-642-28108-2_19

[49] Wil M. P. van der Aalst, Arya Adriansyah, and Boudewijn F. van Dongen. 2012. Replaying history on process models for conformance checking and performance analysis. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 2, 2 (2012), 182–192. https://doi.org/10.1002/widm.1045

Table 1. Data quality annotations

Level	Key	Type	Description
Log	quality:description	string	Records a brief description of the event log and details on any transformations applied
Log	quality:dource	string	Records the source of the event log
Log	quality:pcissuestraces	continuous	Records percentage of traces with a data quality issue
Log	quality:pctprecisetraces:attributeX	continuous	Records percentage of traces without an imprecise value for attribute X
Log	quality:pctaccuratetraces:attributeX	continuous	Records percentage of traces without an inaccurate value for attribute X
Log	quality:pctnotmissingtraces:attributeX	continuous	Records percentage of traces without a missing value for attribute X
Log	quality:pctrelevanttraces	continuous	Records percentage of traces with relevant events
Log	quality:pctimprecise:attributeX	continuous	Records percentage of events with an imprecise value for attribute X
Log	quality:pctinaccurate:attributeX	continuous	Records percentage of events with an inaccurate value for attribute X
Log	quality:pctmissing:attributeX	continuous	Records percentage of events with a missing value for attribute X
Log	quality:pcttracesupdated:attributeX	continuous	Records percentage of traces with updated value of attribute X
Log	quality:pcttracesinserted:attributeX	continuous	Records percentage of traces with inserted value for attribute X
Log	quality:pcttracesdeleted:attributeX	continuous	Records percentage of traces with deleted value for attribute X
Log	quality:pctinserted:attributeX	continuous	Records percentage of events with values inserted for attribute X
Log	quality:pctupdated:attributeX	continuous	Records percentage of events with values updated for attribute X
Log	quality:pctdeleted:attributeX	continuous	Records percentage of events with values deleted for attribute X
Trace	quality:pctmissing:attributeX	continuous	Records percentage of events with missing values related to an attribute X
Trace	quality:pctirrelevant	continuous	Records percentage of irrelevant events in a Trace
Event	quality:accuracy:attributeX	continuous	Records the level of accuracy of the attribute X
Event	quality:precision:attributeX	string or double	Records the level of precision of the attribute X
Event	quality:relevance	continuous	Records the relevance of an event for analysis
Event	quality:missing:attributeX	continuous	Records is data is missing for an attribute in an event
Event	quality:inserted:attributeX	boolean	Records if the value related to attribute X has been inserted
Event	quality:deleted:attributeX	boolean	Records if the value related to attribute X has been deleted
Event	quality:updated:attributeX	boolean	Records if the value related to attribute X has been updated

*The attribute X can refer to the event order, timestamp, activity label, case identifier, resource identifier, or any other attribute in the log.

[50] Lien Vanbrabant, Niels Martin, Katrien Ramaekers, and Kris Braekers. 2019. Quality of input data in emergency department simulations: Framework and assessment techniques. *Simul. Model. Pract. Theory* 91 (2019), 83–101. https://doi.org/10.1016/j.simpat.2018.12.002

[51] Rick Verhulst. 2016. *Evaluating quality of event data within event logs: an extensible framework.* Master's thesis. Eindhoven University of Technology.

[52] Vassilios S Verykios, Ahmed K Elmagarmid, and Elias N Houstis. 2000. Automating the approximate record-matching process. *Information sciences* 126, 1-4 (2000), 83–98. https://doi.org/10.1016/S0020-0255(00)00013-X

[53] Jianmin Wang, Shaoxu Song, Xiaochen Zhu, and Xuemin Lin. 2016. Efficient Recovery of Missing Events. *IEEE Transactions on Knowledge and Data Engineering* 28, 11 (2016), 2943–2957. https://doi.org/10.1109/TKDE.2016.2594785

[54] Moe Thandar Wynn, Wei Zhe Low, Arthur H. M. ter Hofstede, and Wiebe Nauta. 2014. A Framework for Cost-Aware Process Management: Cost Reporting and Cost Prediction. *Journal of Universal Computer Science* 20, 3 (2014), 406–430. https://doi.org/10.3217/jucs-020-03-0406

[55] Moe Thandar Wynn and Shazia W. Sadiq. 2019. Responsible Process Mining - A Data Quality Perspective. In *Int. Conf. on Business Process Management (LNCS, Vol. 11675)*. Springer, 10–15. https://doi.org/10.1007/978-3-030-26619-6_2

[56] XES Working Group, others. 2016. IEEE standard for eXtensible Event Stream (XES) for achieving interoperability in event logs and event streams. (2016).

[57] Ping Zhang and Nicoleta Serban. 2007. Discovery, visualization and performance analysis of enterprise workflow. *Computational statistics & data analysis* 51, 5 (2007), 2670–2687. https://doi.org/10.1016/j.csda.2006.01.008

A DATA QUALITY ANNOTATIONS: ADDITIONAL DETAILS AND PROPOSED XES EXTENSION

This Appendix provides additional details related to the data quality annotations proposed in this paper.

A.1 Description of Data Quality Annotations

Table 1 provides details related to the annotations proposed in the paper as presented in Figure 1. Here X refers to any attribute present in the event log.

A.2 Proposed XES Extension

This section presents the XES Extension we propose that incorporates data quality annotations.

```
        <xesextension name="Quality" prefix="quality">
        <log>
        <string key="source">
            <alias mapping="EN" name="Source of the log"/>
        </string>
        <string key="description">
            <alias mapping="EN" name="Description of the log"/>
        </string>
        <double key="pctissuestraces">
            <alias mapping="EN" name="Percentage of traces with data quality issues" />
        </double>
        <double key="pctprecisetraces:time:timestamp">
            <alias mapping="EN" name="Percentage of traces with precise value of timestamp" />
        </double>
        <double key="pctprecisetraces:activity">
            <alias mapping="EN" name="Percentage of traces with precise value of activity" />
        </double>
        <double key="pctprecisetraces:resource">
            <alias mapping="EN" name="Percentage of traces with precise value of resource" />
        </double>
         <double key="pctprecisetraces:attributeX">
            <alias mapping="EN" name="Percentage of traces with precise value of attributeX" />
        </double>
        <double key="pctaccuratetraces:time:timestamp">
            <alias mapping="EN" name="Percentage of traces with accurate value of timestamp" />
        </double>
        <double key="pctaccuratetraces:activity">
            <alias mapping="EN" name="Percentage of traces with accurate value of activity" />
        </double>
        <double key="pctaccuratetraces:resource">
            <alias mapping="EN" name="Percentage of traces with accurate value of resource" />
        </double>
         <double key="pctaccuratetraces:attributeX">
            <alias mapping="EN" name="Percentage of traces with accurate value of attributeX" />
        </double>
        <double key="pctnotmissingtraces:time:timestamp">
            <alias mapping="EN" name="Percentage of traces with a value for timestamp" />
        </double>
        <double key="pctnotmissingtraces:activity">
            <alias mapping="EN" name="Percentage of traces with a value for activity" />
        </double>
        <double key="pctnotmissingtraces:resource">
            <alias mapping="EN" name="Percentage of traces with a value for resource" />
        </double>
         <double key="pctnotmissingtraces:attributeX">
            <alias mapping="EN" name="Percentage of traces with a value for attributeX" />
        </double>
        <double key="pctrelevanttraces">
            <alias mapping="EN" name="Percentage of traces with relevant events" />
        </double>
        <double key="pctmissing:time:timestamp">
            <alias mapping="EN" name="Percentage of events with missing values for timestamp"/>
        </double>
        <double key="pctmissing:activity">
            <alias mapping="EN" name="Percentage of events with missing values for activity"/>
        </double>
        <double key="pctmissing:resource">
            <alias mapping="EN" name="Percentage of events with missing values for resource"/>
        </double>
        <double key="pctmissing:attributeX">
            <alias mapping="EN" name="Percentage of traces with missing values related to attributeX"/>
        </double>
        <double key="pctimprecise:time:timestamp">
            <alias mapping="EN" name="Percentage of events with imprecise values for timestamp"/>
        </double>
        <double key="pctimprecise:activity">
            <alias mapping="EN" name="Percentage of events with imprecise values for activity"/>
        </double>
        <double key="pctimprecise:resource">
            <alias mapping="EN" name="Percentage of events with imprecise values for resource"/>
        </double>
        <double key="pctimprecise:attributeX">
            <alias mapping="EN" name="Percentage of events with imprecise values related to attribute X"/>
        </double>
        <double key="pctinaccurate:time:timestamp">
            <alias mapping="EN" name="Percentage of events with inaccurate values for timestamp"/>
        </double>
        <double key="pctinaccurate:activity">
            <alias mapping="EN" name="Percentage of events with inaccurate values for activity"/>
        </double>
        <double key="pctinaccurate:resource">
            <alias mapping="EN" name="Percentage of events with inaccurate values for resource"/>
        </double>
       <double key="pctinaccurate:attributeX">
            <alias mapping="EN" name="Percentage of events with inaccurate values related to attribute X"/>
       <double key="pcttracesupdated:time:timestamp">
            <alias mapping="EN" name="Percentage of traces with updated value for timestamp"/>
        </double>
        <double key="pcttracesupdated:activity">
            <alias mapping="EN" name="Percentage of traces with updated value for activity"/>
        </double>
        <double key="pcttracesupdated:resource">
            <alias mapping="EN" name="Percentage of traces with updated value for resource"/>
        </double>
        <double key="pcttracesupdated:attributeX">
            <alias mapping="EN" name="Percentage of traces with updated value for attribute X"/>
        </double>
        <double key="pcttracesinserted:time:timestamp">
            <alias mapping="EN" name="Percentage of traces with inserted value for timestamp"/>
        </double>
        <double key="pcttracesinserted:activity">
            <alias mapping="EN" name="Percentage of traces with inserted value for activity"/>
        </double>
        <double key="pcttracesinserted:resource">
            <alias mapping="EN" name="Percentage of traces with inserted value for resource"/>
        </double>
        <double key="pcttracesinserted:attributeX">
            <alias mapping="EN" name="Percentage of traces with inserted value for attribute X"/>
        </double>
        <double key="pcttracesdeleted:time:timestamp">
            <alias mapping="EN" name="Percentage of traces with deleted value for timestamp"/>
        </double>
        <double key="pcttracesdeleted:activity">
            <alias mapping="EN" name="Percentage of traces with deleted value for activity"/>
        </double>
        <double key="pcttracesdeleted:resource">
            <alias mapping="EN" name="Percentage of traces with deleted value for resource"/>
        </double>
        <double key="pcttracesdeleted:attributeX">
            <alias mapping="EN" name="Percentage of traces with deleted value for attribute X"/>
        </double>
        <double key="pctinserted:time:timestamp">
            <alias mapping="EN" name="Percentage of events where timestamp was inserted"/>
        </double>
        <double key="pctinserted:activity">
            <alias mapping="EN" name="Percentage of events where activity was inserted"/>
        </double>
        <double key="pctinserted:activity">
            <alias mapping="EN" name="Percentage of events where resource was inserted"/>
        </double>
        <double key="pctinserted:attributeX">
            <alias mapping="EN" name="Percentage of events inserted related to attribute X"/>
        </double>
        <double key="pctdeleted:time:timestamp">
            <alias mapping="EN" name="Percentage of events where timestamp was deleted"/>
        </double
        <double key="pctdeleted:time:activity">
            <alias mapping="EN" name="Percentage of events where activity was deleted"/>
        </double
        </double<double key="pctdeleted:resource">
            <alias mapping="EN" name="Percentage of events where resource was deleted"/>
        </double>
        </double<double key="pctdeleted:attributeX">
            <alias mapping="EN" name="Percentage of events deleted related to attribute X"/>
        </double>
        <double key="pctupdated:time:timestamp">
            <alias mapping="EN" name="Percentage of events where timestamp was updated"/>
        </double>
        <double key="pctupdated:activity">
            <alias mapping="EN" name="Percentage of events where activity was updated"/>
        </double>
        <double key="pctupdated:resource">
            <alias mapping="EN" name="Percentage of events where resource was updated"/>
        </double>
        <double key="pctupdated:attributeX">
            <alias mapping="EN" name="Percentage of events updated related to attribute X"/>
        </double>
    </log>
    <trace>
        <double key="pctmissing:time:timestamp">
            <alias mapping="EN" name="Percentage of missing values related to timestamp"/>
        </double>
        <double key="pctmissing:activity">
            <alias mapping="EN" name="Percentage of missing values related to activity"/>
        </double>
        <double key="pctmissing:resource">
            <alias mapping="EN" name="Percentage of missing values related to resource"/>
        </double>
        <double key="pctmissing:attributeX">
            <alias mapping="EN" name="Percentage of missing values related to attribute X"/>
        </double>
        <double key="pctirrelevant">
            <alias mapping="EN" name="Percentage of irrelevant events in the trace"/>
        </double>
    </trace>
    <event>
        <boolean key="relevance">
            <alias mapping="EN" name="Event is relevant"/>
        </boolean>
        <string key="precision:time:timestamp ">
            <alias mapping="EN" name="Units to which timestamp is precise"/>
        </string>
        <double key="precision:activity">
            <alias mapping="EN" name="Precision of the activity"/>
        </double>
        <double key="precision:resource">
            <alias mapping="EN" name="Precision of the resource"/>
        </double>
        <double key="precision:attribute X">
            <alias mapping="EN" name="Precision of attribute X"/>
        </double>
        <double key="accuracy:time:timestamp">
            <alias mapping="EN" name=" Accuracy of timestamp"/>
        </double>
         <double key="accuracy:activity">
            <alias mapping="EN" name=" Accuracy of activity"/>
        </double>
         <double key="accuracy:resource">
            <alias mapping="EN" name=" Accuracy of resource"/>
        </double>
         <double key="accuracy:attributeX">
            <alias mapping="EN" name=" Accuracy of attribute X"/>
        </double>
         <double key="missing:time:timestamp">
            <alias mapping="EN" name=" Value of timestamp is missing"/>
        </double>
         <double key="missing:activity">
            <alias mapping="EN" name=" Value of activity is missing"/>
        </double>
         <double key="missing:resource">
            <alias mapping="EN" name=" Value of resource is missing"/>
        </double>
         <double key="missing:attributeX">
```

```
            <alias mapping="EN" name=" Value of attribute X is missing"/>
        </double>
        <boolean key="inserted:time:timestamp">
            <alias mapping="EN" name=" Value of Timestamp is inserted"/>
        </boolean>
        <boolean key="inserted:activity">
            <alias mapping="EN" name=" Value of Activity is inserted"/>
        </boolean>
        <boolean key="inserted:resource">
            <alias mapping="EN" name=" Value of Resource is inserted"/>
        </boolean>
        <boolean key="inserted:attributeX">
            <alias mapping="EN" name=" Value of Attribute X is inserted"/>
        </boolean>
        <boolean key="deleted:time:timestamp">
            <alias mapping="EN" name=" Value of Timestamp is deleted"/>
        </boolean>
        <boolean key="deleted:activity">
            <alias mapping="EN" name=" Value of Activity is deleted"/>
        </boolean>
        <boolean key="deleted:resource">
            <alias mapping="EN" name=" Value of Resource is deleted"/>
        </boolean>
        <boolean key="deleted:attributeX">
            <alias mapping="EN" name=" Value of Attribute X is deleted"/>
        </boolean>
    </event>
</xesextension>


```

Listing 2. Proposed Quality Extension in XML Format.

B QUALITY-INFORMED CONFORMANCE CHECKING

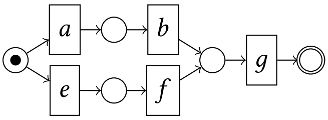
This Appendix provides details of how other annotations can be used for quality-informed conformance checking techniques.

B.1 Relevance (rel)

The relevance data quality annotation may indicate that the information in an event is not relevant to the analysis and should therefore be ignored as much as possible. Accordingly, data-aware conformance checking techniques should prefer relevant events over irrelevant events.

For instance, consider Figure 10 that contains a model M and a log L consisting of a single trace, where events a and b are not fully relevant, indicated by their data quality annotation. Figure 10c shows the standard optimal alignment, which has a cost of 1. This alignment does not do justice to the known data quality issue: the events a and b are not relevant but they have been mapped as synchronous moves, while e is relevant and is mapped as a log move. Consequently, the relevant e will not be included in frequency and performance measures, while the irrelevant a and b will be included.

To address this issue, we change the cost function of alignments, by making irrelevant events *cheaper* to classify as log moves and *more expensive* to classify as synchronous moves. That is, events are assigned a cost of log move equal to their relevance, and a synchronous cost of $2 * (1 - \textsc{rel})$. In our example, a is given a log-move cost of 0.25, b of 0.5 and f is given a log-move cost of 1. Then, the alignment shown in Figure 10d has a total cost of 1.75. As alignment computations select the alignment with the lowest cost, consequently, the relevant e will be included in frequency and performance measures, while a and b will not.



(a) Model M.

$[\langle e^{\text{REL}:1}, a^{\text{REL}:0.25}, b^{\text{REL}:0.5}, g^{\text{REL}:1}\rangle]$

(b) Log L, annotated with relevance.

log	e	a	b	g
model	$\rightarrow$	a	b	g

(c) Standard alignment: cost 1.
Quality-informed alignment: cost 3.5.

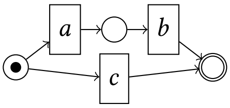
log	e	a	b	$\rightarrow$	g
model	e	$\rightarrow$	$\rightarrow$	f	g

(d) Standard alignment: cost 3.
Quality-informed alignments: cost 1.75.

Fig. 10. An example of alignments with irrelevant events.

$[\langle c^{\text{ACC}_{\text{concept:name}}:1}, a^{\text{ACC}_{\text{concept:name}}:0.25}, b^{\text{ACC}_{\text{concept:name}}:0.5}\rangle]$

(a) Log L, annotated with accuracy.



(b) Model M.

log	c	a	b
model	$\rightarrow$	a	b

(c) Alignment: cost 1. Quality-informed alignment: cost 4.375.

log	c	a	b
model	c	$\rightarrow$	$\rightarrow$

(d) Alignment: cost 2. Quality-informed alignments: cost 0.75.

Fig. 11. An example of alignments with inaccurate concept:name values.

B.2 Order Accuracy (ACC$_{\triangleright\text{order}}$)

The order accuracy (ACC$_{\triangleright\text{order}}$) annotation indicates the probability that the position of an event in a trace is incorrect. As such, we *decrease* its log-move cost accordingly in order to *decrease* the likelihood of the event being considered a synchronous move.

B.3 Accuracy of concept:name (ACC$_{\triangleright\text{concept:name}}$)

If a concept:name value is inaccurate, this value has been recorded with errors, and does not resemble the “true" value. In this case, it would be challenging to remap the inaccurate concept:name to the “true" one. In this paper, we assume that the process model is correct.

If the inaccurate concept:name does not appear in the model, then alignments will classify the corresponding event as a log move, and consequently the event will be ignored in frequency and performance computations. However, if the inaccurate concept:name appears in the model, then it might be erroneously classified as a synchronous move. Figure 11 shows an example that contains a model M and a log L consisting of a single trace, where events a and b have an inaccurate concept:name, indicated by their data quality annotation. Figure 11c shows the standard alignment, which has a cost of 1. This alignment does not do justice to the known data quality issue: the events a and b are most likely inaccurate but have been mapped as synchronous moves, while c is accurate but is mapped as a log move. Consequently, the accurate c will not be included in frequency and performance measures, while the inaccurate a and b will be included in the measures *for the wrong activities.*

To address this issue, we increase the cost of a synchronous move on an event with an inaccurate concept:name value (as this value is wrong, the event should not be mapped), and we decrease the cost of it being a log move. An event e with inaccurate concept:name can either be mapped on a transition t (with the same erroneous label as e) as a synchronous move, or can be mapped as a log move while t is mapped as a model move. That is, either $\begin{array}{|c c|}\hline {\scriptstyle \text{log}} & e \\ {\scriptstyle \text{model}} & t \\ \hline\end{array}$ or $\begin{array}{|c c c|}\hline {\scriptstyle \text{log}} & e & \longrightarrow \\ {\scriptstyle \text{model}} & \longrightarrow & t \\ \hline\end{array}$. We identified three requirements for these new costs of events with inaccurate `concept:name`:

(1) Given that the data quality annotation $\textsc{acc}_{\rhd\text{concept:name}}$ indicates a *probability* that the `concept:name` value is inaccurate, we require that for $\textsc{acc}_{\rhd\text{concept:name}}(e) = \frac{1}{2}$, these two alignments should have the same cost.

(2) As a base case, we require that if $\textsc{acc}_{\rhd\text{concept:name}}(e) = 1$, synchronous move and log move costs should be equal to Definition 1.

(3) For the case that we are sure that `concept:name` is inaccurate, the synchronous move cost should not be higher than 2, as to avoid having a non-local influence on alignments.

To summarise these requirements:

$\textsc{acc}_{\rhd\text{concept:name}}(e)$	synchronous move $\frac{e}{T}$ cost	log move $\overset{e}{\rightarrow}$ cost
0	2	0
0.5	1.5	0.5
1	0	1

Please note that model move costs cannot be influenced by $\textsc{acc}_{\rhd\text{concept:name}}$, as no log event is involved in a model move and thus no data quality annotation is available. In accordance with these requirements, we define a synchronous move cost of $-2\textsc{acc}_{\rhd\text{concept:name}}(e)^2 + 2$ and a log move cost of $\textsc{acc}_{\rhd\text{concept:name}}(e)$, which are the simplest functions satisfying these requirements.

In our example, the first alignment (Figure 11c) gets a cost of 1 (log move c) + 1.875 (synchronous move a) + 1.5 (synchronous move b). A more desirable alignment (Figure 11d) gets a cost of 0.75. Thus, where a and b were inaccurately represented and thus would contribute erroneously to frequency and performance measures, while c would not, the issue is addressed.

C DERIVATION OF TRAPEZOID PARAMETERS

This Appendix provides details regarding derivation of trapezoid parameters. From [17], where U is the uniform distribution and $\mathcal{T}$ is the trapezoid distribution:

$$X = X_1 + X_2 \text{ with} \tag{6}$$

$$X_1 \sim U[a_1, b_1] \tag{7}$$

$$X_2 \sim U[a_2, b_2] \text{ then} \tag{8}$$

$$X \sim \mathcal{T}[\mathcal{T}_a, \mathcal{T}_c, \mathcal{T}_d, \mathcal{T}_b] \text{ with} \tag{9}$$

$$\mathcal{T}_a = a_1 + a_2 \tag{10}$$

$$\mathcal{T}_c = \begin{cases} a_1 + b_2 & \text{if } a_1 + b_2 < a_2 + b_1 \\ a_2 + b_1 & \text{if } a_1 + b_2 > a_2 + b_1 \\ \frac{(a_1+a_2)+(b_1+b_2)}{2} & \text{if } a_1 + b_2 = a_2 + b_1 \end{cases} \tag{11}$$

$$\mathcal{T}_d = \begin{cases} a_2 + b_1 & \text{if } a_1 + b_2 < a_2 + b_1 \\ a_1 + b_2 & \text{if } a_1 + b_2 > a_2 + b_1 \\ \frac{(a_1+a_2)+(b_1+b_2)}{2} & \text{if } a_1 + b_2 = a_2 + b_1 \end{cases} \tag{12}$$

$$\mathcal{T}_b = b_1 + b_2 \tag{13}$$

For our case, we take $-e = X_1$ and $e' = X_2$. Assume for the moment that $s(e) + p(e) \leq s(e')$, and thus that e and e' are independent:

$$T' = e' + (-e) \text{ with} \tag{14}$$

$$-e \sim U[-(s(e) + p(e)), -s(e)] \tag{15}$$

$$e' \sim U[s(e'), s(e') + p(e')]\text{then} \tag{16}$$

$$T' \sim \mathcal{T}[\mathcal{T}_a', \mathcal{T}_c', \mathcal{T}_d', \mathcal{T}_b']\text{with} \tag{17}$$

$$\mathcal{T}_a' = -(s(e) + p(e)) + s(e') \tag{18}$$

$$\mathcal{T}_c' = \begin{cases} -(s(e) + p(e)) + s(e') + p(e') \\ \qquad\qquad \text{if } -(s(e) + p(e)) + s(e') + p(e') < s(e') + -s(e \\ s(e') - s(e) \\ \qquad\qquad \text{if } -(s(e) + p(e)) + s(e') + p(e') > s(e') + -s(e \\ \frac{(-(s(e)+p(e))+s(e'))+(-s(e)+s(e')+p(e'))}{2} \\ \qquad\qquad \text{if } -(s(e) + p(e)) + s(e') + p(e') = s(e') + -s(e \end{cases} \tag{19}$$

$$\mathcal{T}_d' = \begin{cases} s(e') - s(e) \\ \qquad\qquad \text{if } -(s(e) + p(e)) + s(e') + p(e') < s(e') + -s(e \\ -(s(e) + p(e)) + s(e') + p(e') \\ \qquad\qquad \text{if } -(s(e) + p(e)) + s(e') + p(e') > s(e') + -s(e \\ \frac{(-(s(e)+p(e))+s(e'))+(-s(e)+s(e')+p(e'))}{2} \\ \qquad\qquad \text{if } -(s(e) + p(e)) + s(e') + p(e') = s(e') + -s(e \end{cases} \tag{20}$$

$$\mathcal{T}_b' = -s(e) + s(e') + p(e') \tag{21}$$

Next, we weaken the assumption that $e < e'$, and that e and e' are independent (A2). Instead, we use the fact that $T \geq 0$, which is guaranteed by the conformance checking technique, and apart from this, e and e' are independent. Accordingly, we ensure that none of the parameters can be < 0:

$$T = e' + (-e) \text{ with} \tag{22}$$

$$-e \sim U[-(s(e) + p(e)), -s(e)] \tag{23}$$

$$e' \sim U[s(e'), s(e') + p(e')] \text{ then} \tag{24}$$

$$T \sim \mathcal{T}[\mathcal{T}_a, \mathcal{T}_c, \mathcal{T}_d, \mathcal{T}_b] \text{ with} \tag{25}$$

$$\mathcal{T}_a = \max(0, \mathcal{T}_a') \tag{26}$$

$$\mathcal{T}_c = \max(0, \mathcal{T}_c') \tag{27}$$

$$\mathcal{T}_d = \max(0, \mathcal{T}_d') \tag{28}$$

$$\mathcal{T}_b = \max(0, \mathcal{T}_b') \tag{29}$$

D PERFORMANCE INFLUENCE

This Appendix presents the formulae used to calculate the influence of data quality on performance measures as discussed in Section 6.2.

$$\varphi(e,e') = \begin{cases} (a-\hat{\mu})^2 & \text{if } a=b=c=d \\ \varphi_1(e,e')+\varphi_2(e,e')+\varphi_3(e,e') & \text{otherwise} \end{cases} \tag{30}$$

$$\varphi_1(e,e') = \int_{\mathcal{T}_a}^{\mathcal{T}_c} (x-\hat{\mu})^2 \frac{x-\mathcal{T}_a}{\mathcal{T}_c-\mathcal{T}_a} h \,\mathrm{d}x \tag{31}$$

$$= -\tfrac{1}{12} h(\mathcal{T}_a-\mathcal{T}_c)(\mathcal{T}_a{}^2 + 2\mathcal{T}_a(\mathcal{T}_c-2\hat{\mu}) + 3\mathcal{T}_c{}^2 - 8\mathcal{T}_c\hat{\mu} + 6\hat{\mu}^2) \tag{32}$$

$$\varphi_2(e,e') = \int_{\mathcal{T}_c}^{\mathcal{T}_d} (x-\hat{\mu})^2 h \,\mathrm{d}x \tag{33}$$

$$= -\tfrac{1}{3} h(\mathcal{T}_c-\mathcal{T}_d)(\mathcal{T}_c{}^2 + \mathcal{T}_c(\mathcal{T}_d-3\hat{\mu}) + \mathcal{T}_d{}^2 - 3\mathcal{T}_d\hat{\mu} + 3\hat{\mu}^2) \tag{34}$$

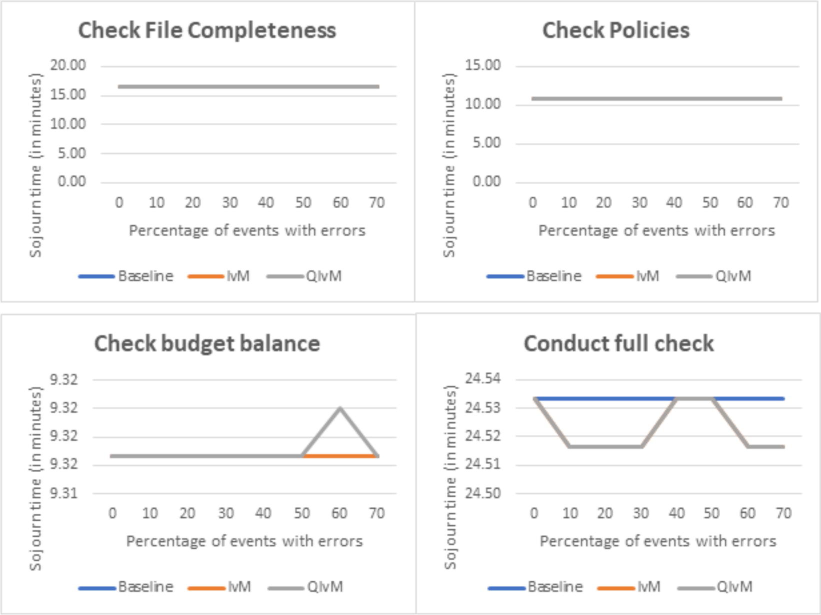
$$\varphi_3(e,e') = \int_{\mathcal{T}_d}^{\mathcal{T}_b} (x-\hat{\mu})^2 \frac{\mathcal{T}_b-x}{\mathcal{T}_b-\mathcal{T}_d} h \,\mathrm{d}x \tag{35}$$

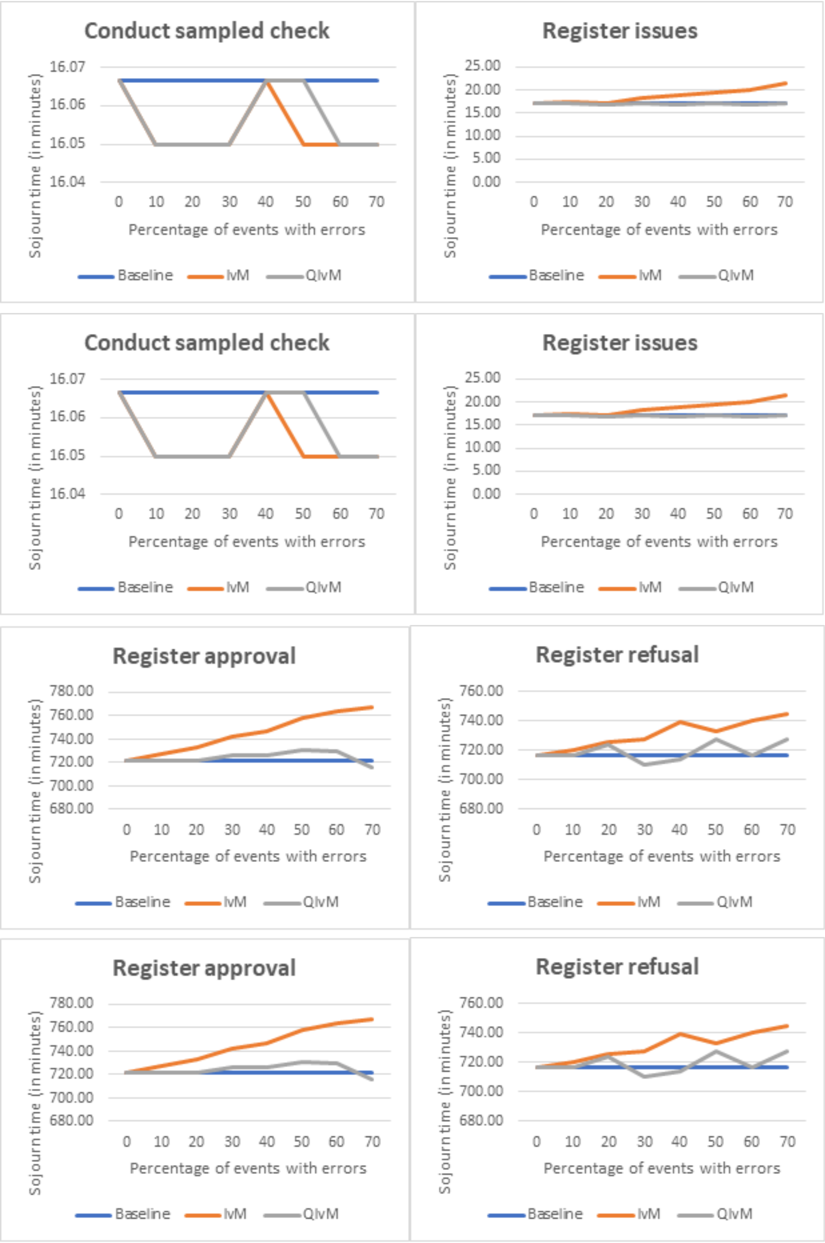
$$= \tfrac{1}{12} h(\mathcal{T}_b-\mathcal{T}_d)(\mathcal{T}_b{}^2 + 2\mathcal{T}_b(\mathcal{T}_d-2\hat{\mu}) + 3\mathcal{T}_d{}^2 - 8\mathcal{T}_d\hat{\mu} + 6\hat{\mu}^2) \tag{36}$$

E DETAILED EXPERIMENT RESULTS

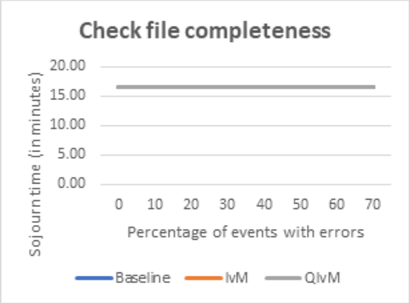
This Appendix provides results for all the experiments conducted in this study.

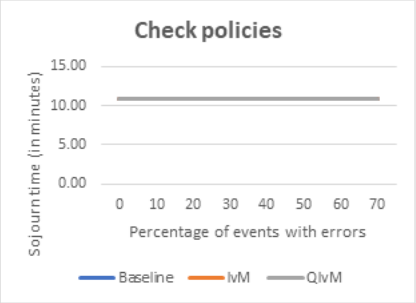
E.1 Synthetic Log - Experiment 1: Inaccurate Timestamps

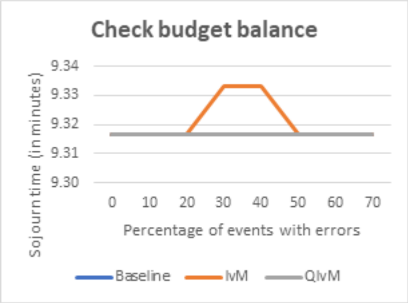


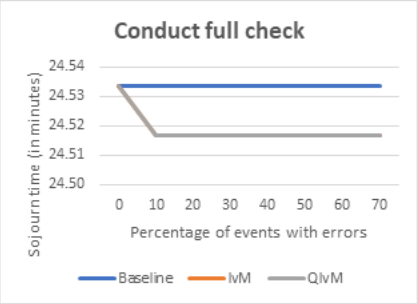

Conduct sampled check
Sojourn time (in minutes)
16.07
16.06
16.05
16.04
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Register issues
Sojourn time (in minutes)
25.00
20.00
15.00
10.00
5.00
0.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Conduct sampled check
Sojourn time (in minutes)
16.07
16.06
16.05
16.04
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Register issues
Sojourn time (in minutes)
25.00
20.00
15.00
10.00
5.00
0.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Register approval
Sojourn time (in minutes)
780.00
760.00
740.00
720.00
700.00
680.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Register refusal
Sojourn time (in minutes)
760.00
740.00
720.00
700.00
680.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Register approval
Sojourn time (in minutes)
780.00
760.00
740.00
720.00
700.00
680.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Register refusal
Sojourn time (in minutes)
760.00
740.00
720.00
700.00
680.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM

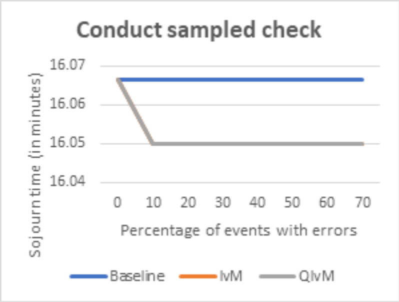
E.2 Synthetic Log - Experiment 2: Imprecise Timestamps

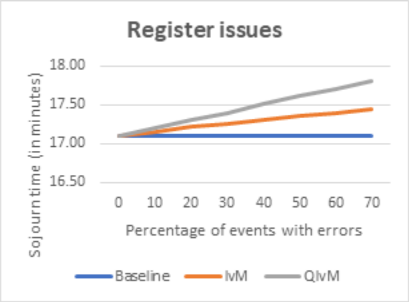


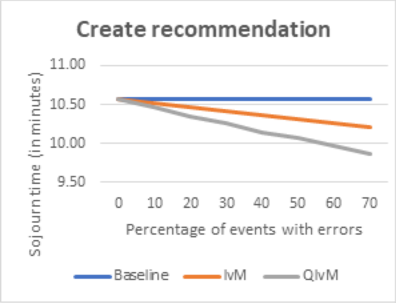


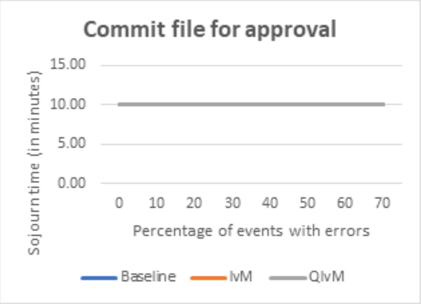


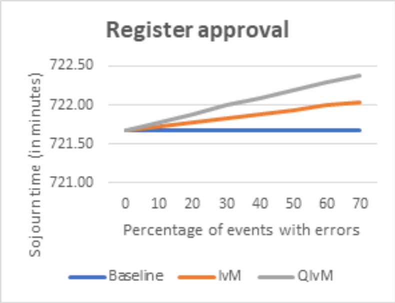


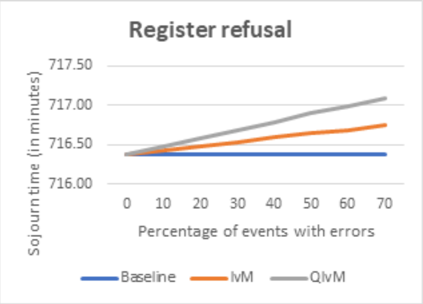


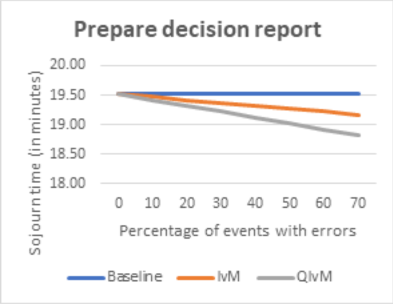


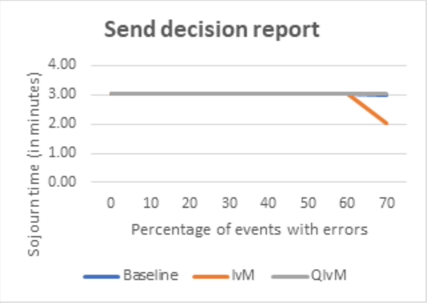




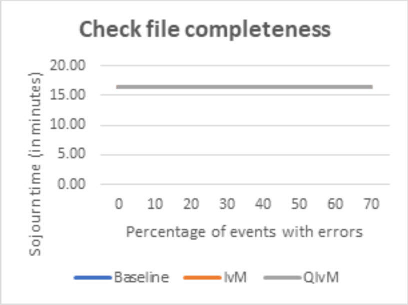


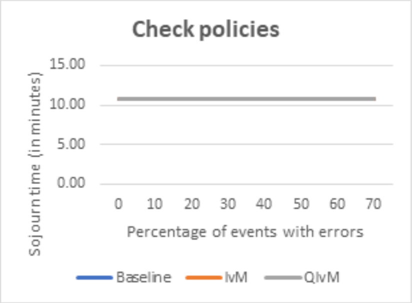


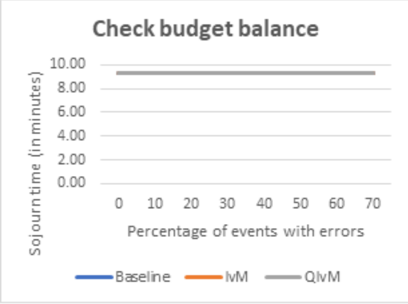


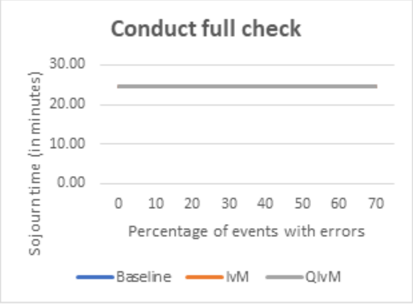


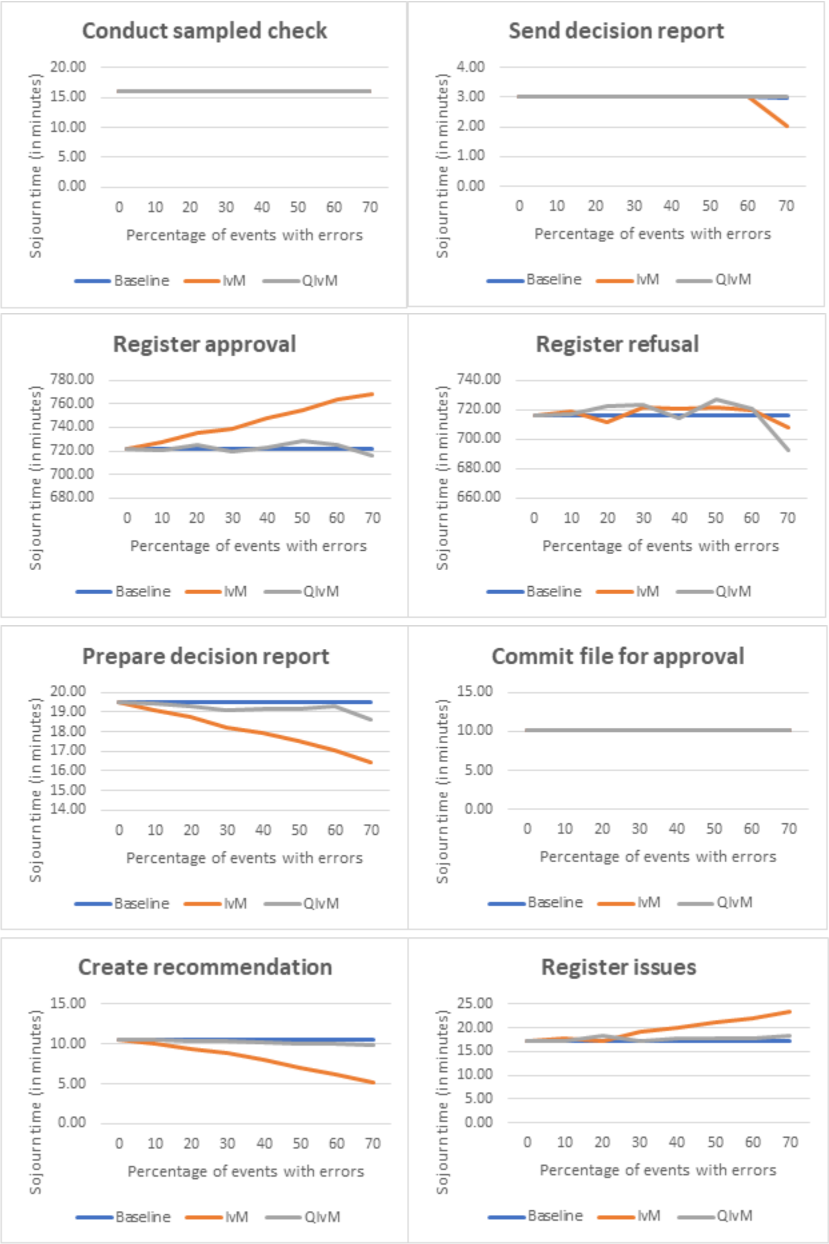
E.3 Synthetic Log - Experiment 3: Inaccurate and Imprecise Timestamps



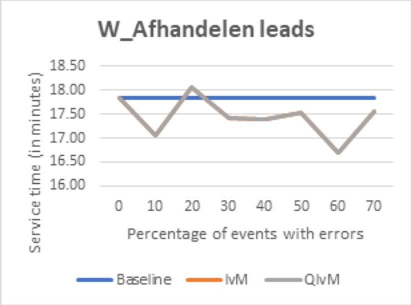


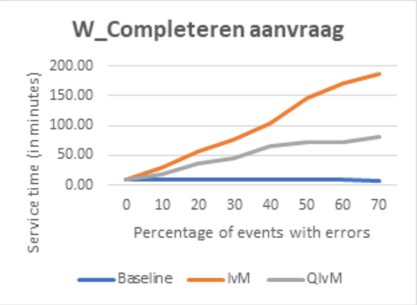


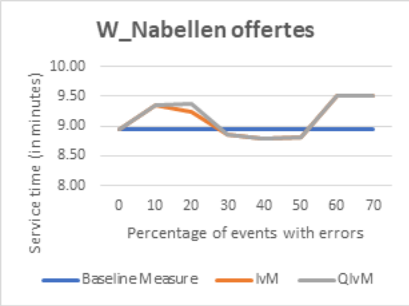


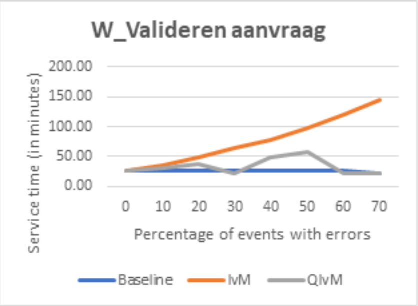

Conduct sampled check
Sojourn time (in minutes)
20.00
15.00
10.00
5.00
0.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Send decision report
Sojourn time (in minutes)
4.00
3.00
2.00
1.00
0.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Register approval
Sojourn time (in minutes)
780.00
760.00
740.00
720.00
700.00
680.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Register refusal
Sojourn time (in minutes)
740.00
720.00
700.00
680.00
660.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Prepare decision report
Sojourn time (in minutes)
20.00
19.00
18.00
17.00
16.00
15.00
14.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Commit file for approval
Sojourn time (in minutes)
15.00
10.00
5.00
0.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Create recommendation
Sojourn time (in minutes)
15.00
10.00
5.00
0.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM
Register issues
Sojourn time (in minutes)
25.00
20.00
15.00
10.00
5.00
0.00
0 10 20 30 40 50 60 70
Percentage of events with errors
Baseline IvM QIvM

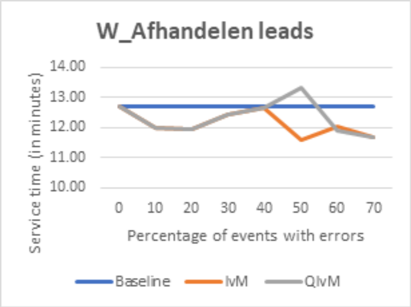
E.4 BPIC 2012 (W Activities) - Experiment 1: Inaccurate Timestamps



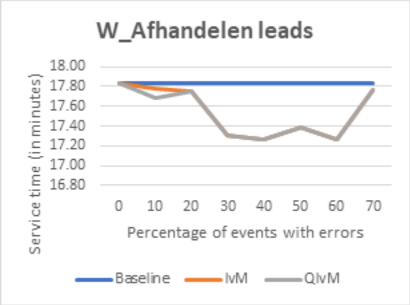


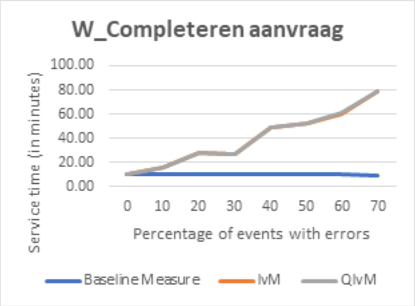


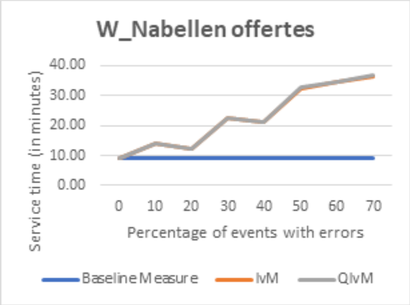


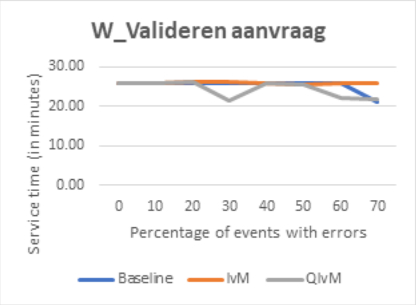


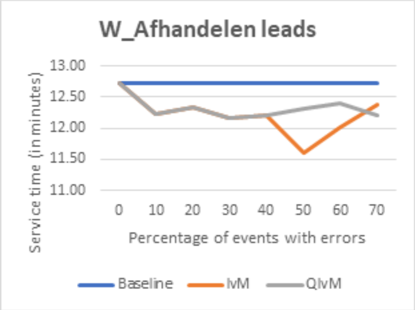
E.5 BPIC 2012 (W Activities) - Experiment 2: Imprecise Timestamps



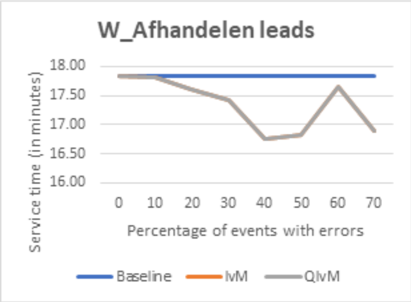


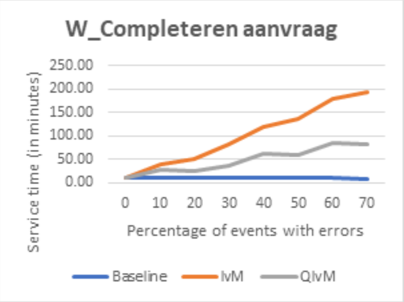


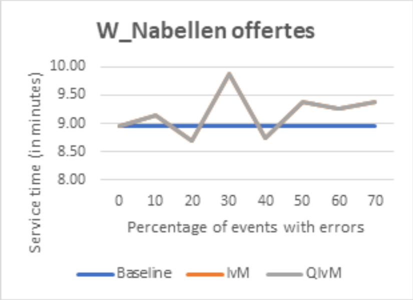


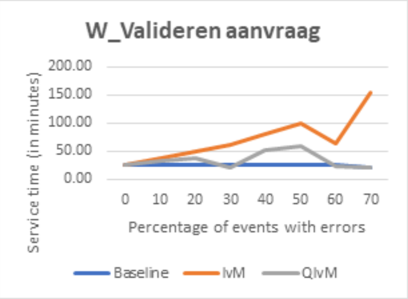


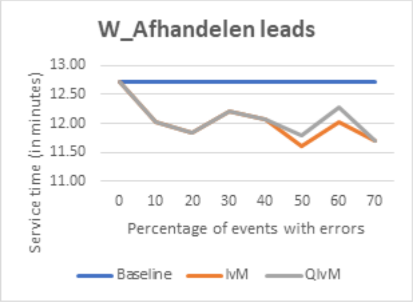
E.6 BPIC 2012 (W Activities) - Experiment 3: Inaccurate and Imprecise Timestamps



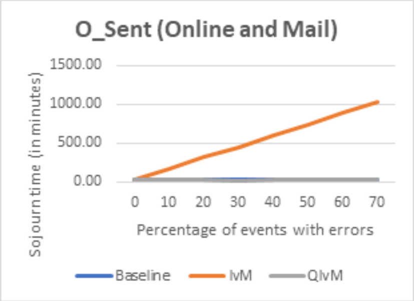


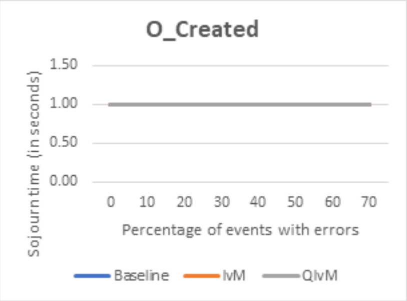


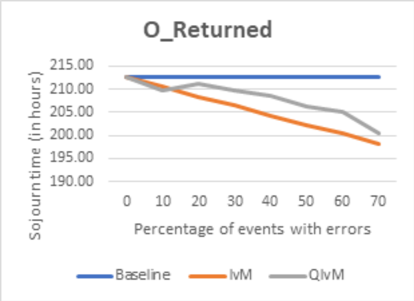


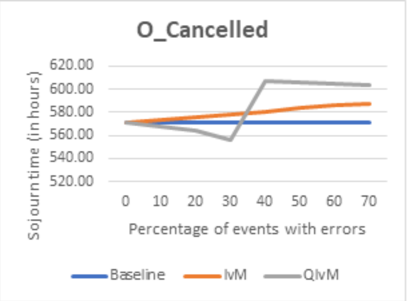


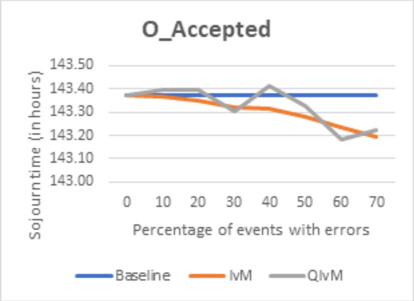
E.7 BPIC 2017 (Offer Log) - Experiment 1: Inaccurate Timestamps

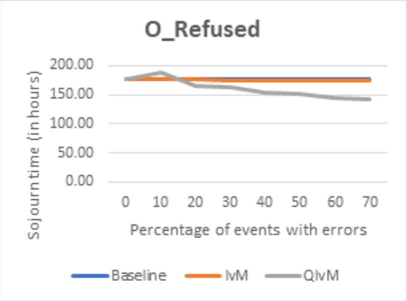




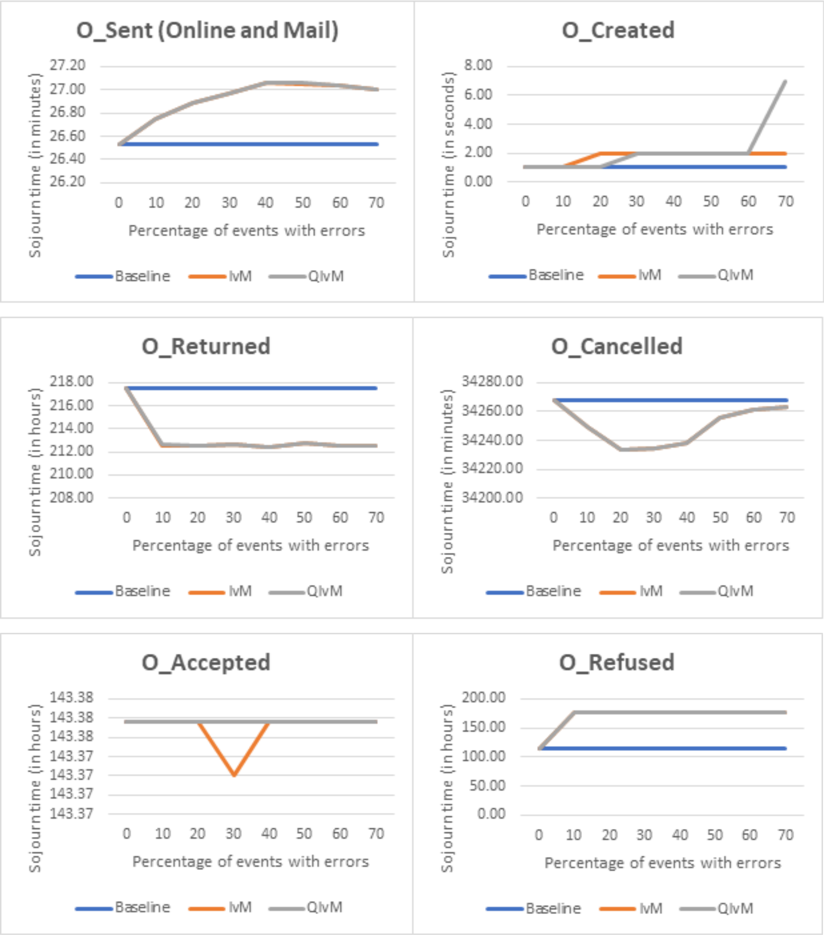




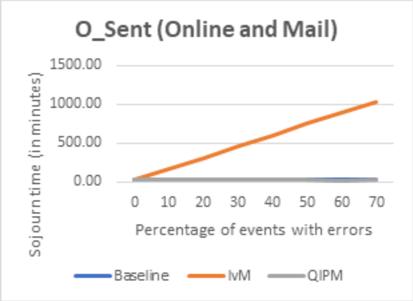


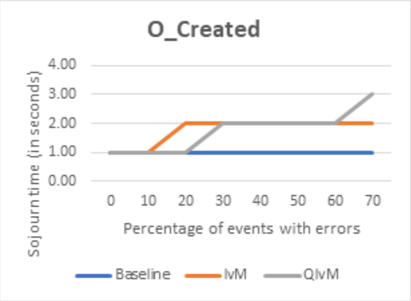


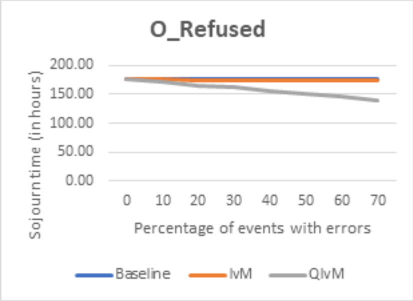
E.8 BPIC 2017 (Offer Log) - Experiment 2: Imprecise Timestamps

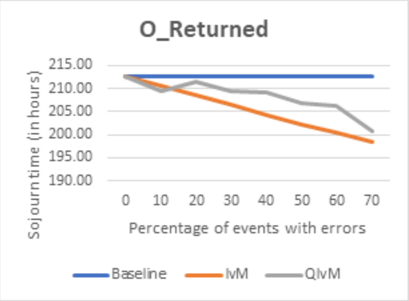


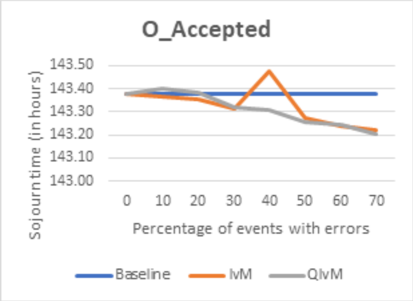
E.9 BPIC 2017 (Offer Log) - Experiment 3: Inaccurate and Imprecise Timestamps

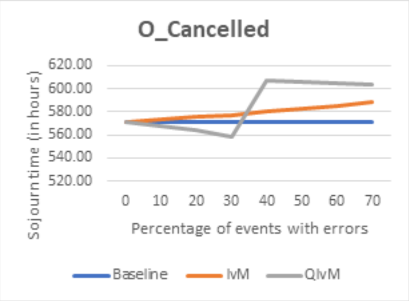




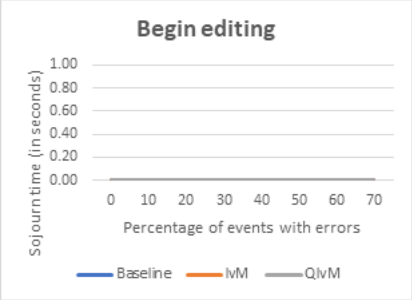


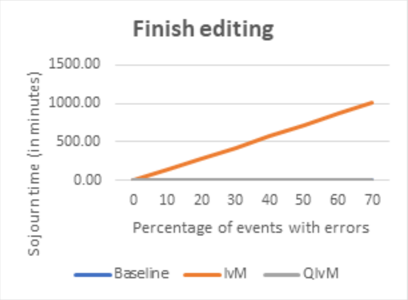


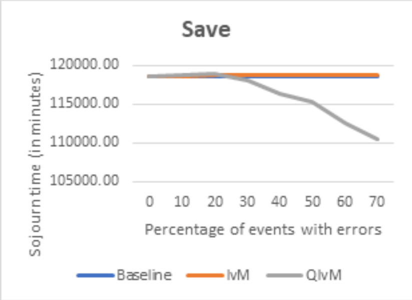


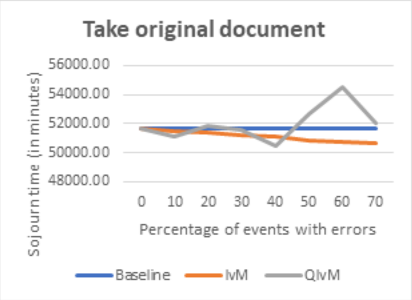


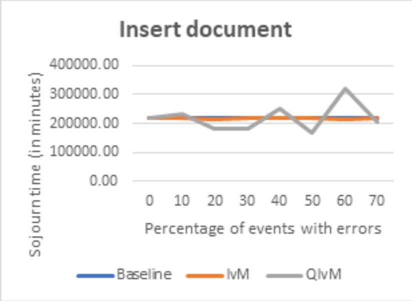
E.10 BPIC 2018 (Control Summary) - Experiment 1: Inaccurate Timestamps

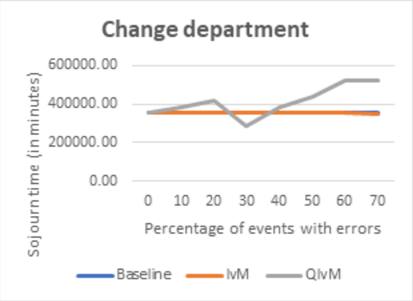

Begin editing


Finish editing

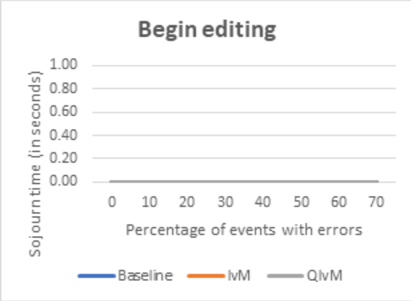

Save

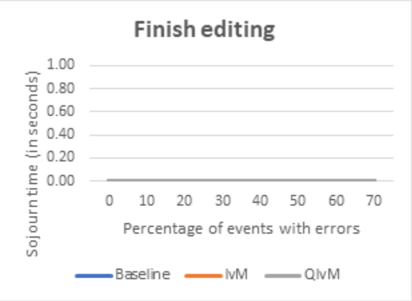

Take original document

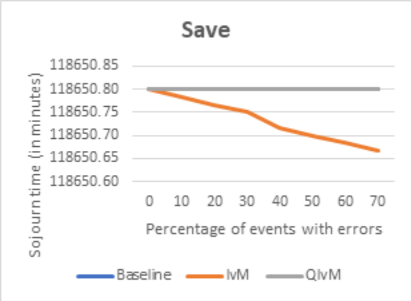

Insert document

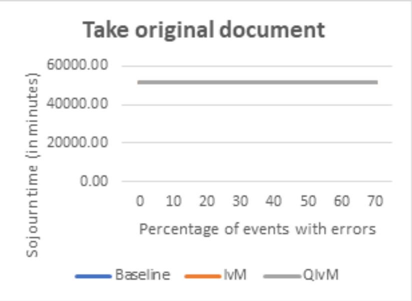

Change department

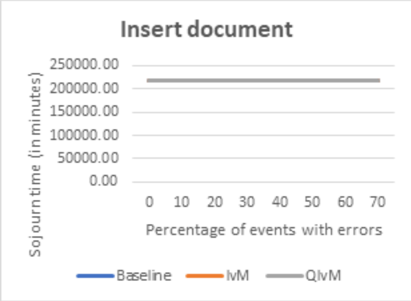
E.11 BPIC 2018 (Control Summary) - Experiment 2: Imprecise Timestamps

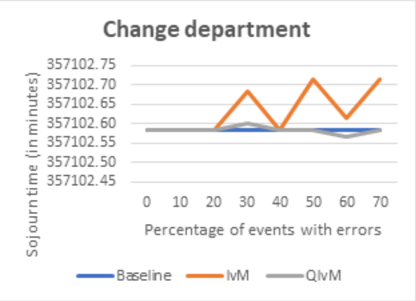




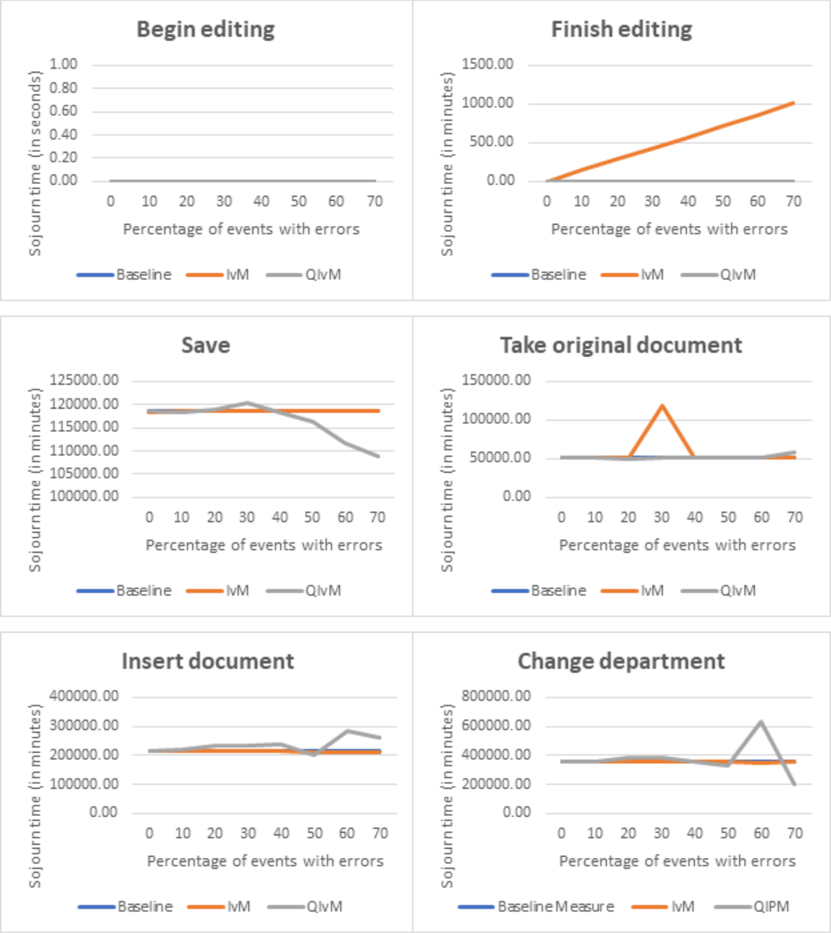








E.12 BPIC 2018 (Control Summary) - Experiment 3: Inaccurate and Imprecise Timestamps


Begin editing
Sojourn time (in seconds)
Percentage of events with errors
Baseline
IvM
QIvM
Finish editing
Sojourn time (in minutes)
Percentage of events with errors
Baseline
IvM
QIvM
Save
Sojourn time (in minutes)
Percentage of events with errors
Baseline
IvM
QIvM
Take original document
Sojourn time (in minutes)
Percentage of events with errors
Baseline
IvM
QIvM
Insert document
Sojourn time (in minutes)
Percentage of events with errors
Baseline
IvM
QIvM
Change department
Sojourn time (in minutes)
Percentage of events with errors
Baseline Measure
IvM
QIPM