universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Data Quality Challenges of Event Logs for Process Mining

verfasst von | submitted by

Lukas Sorer BSc (WU)

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 926
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Wirtschaftsinformatik
Betreut von | Supervisor:	Mag. Dipl.-Ing. Dr.techn. Maria Leitner Bakk.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Leitner, for her unwavering patience and support throughout the creation of this thesis. Her insightful feedback continuously challenged me to reflect on the problems at hand, pushing me to find solutions that not only addressed the technical aspects but were also well-grounded in academic thoroughness.

I would also like to extend my sincere thanks to the Austrian IT Service Provider for providing the real-world data that played a crucial role in verifying the practical application of the Event Data Quality Dashboard. Their willingness to share their data greatly enriched this work and ensured that the dashboard could be tested in a realistic, business-relevant context.

Abstract

This thesis explores the critical role of data quality in process mining, focusing on the identification and assessment of quality issues within event logs. As process mining gains traction in business process analysis, the quality of event logs becomes essential to ensure reliable and actionable insights. Through a comprehensive review of the literature, this study synthesizes existing knowledge on Data Quality (DQ) dimensions both in general and within the specific context of process mining. Drawing on foundational frameworks and definitions from data quality literature, we identify key dimensions relevant to event log data — such as completeness, accuracy, validity, interpretability, and relevancy. These dimensions serve as basis for a novel Data Quality Assessment Framework (DQAF) designed to detect and assess specific issues in event logs. The DQAF includes systematic indicators and assessment measures for each dimension, allowing a structured evaluation of data quality in a process mining context.

To implement this framework practically, the Event Data Quality Dashboard (EDQD) was developed as a diagnostic tool for data quality analysis. The EDQD visualizes data quality assessments and provides analysts with insights into potential deficiencies within event logs that could skew process mining results. Evaluating the EDQD involved three types of event logs: synthetic datasets tailored to isolate particular data quality issues, publicly available event logs from the Business Process Intelligence Challenge (BPIC), and industry-provided real-world datasets supplied by an Austrian IT service provider. These evaluations demonstrated the EDQD's ability to identify a wide range of data quality issues across various log structures and domains, validating the robustness of the framework and the practical application of the tool in real world settings.

The results confirm that the EDQD, grounded in a structured assessment framework, can effectively support process analysts in diagnosing and addressing data quality issues within event logs. By applying this tool across diverse data types, this thesis underscores the importance of rigorous data quality assessment to achieve accurate and reliable results in process mining.

Kurzfassung

Diese Masterarbeit untersucht die zentrale Rolle der Datenqualität im Process Mining und fokussiert sich auf die Identifizierung und Bewertung von Qualitätsproblemen in Ereignisprotokollen. Da Process Mining zunehmend an Bedeutung für die Analyse von Geschäftsprozessen gewinnt, ist die Qualität der zugrunde liegenden Ereignisprotokolle entscheidend, um verlässliche und umsetzbare Erkenntnisse zu gewährleisten. Durch eine umfassende Literaturrecherche wird bestehendes Wissen über allgemeine und speziell für das Process Mining relevante Datenqualitätsdimensionen zusammengefasst. Aufbauend auf grundlegenden Modellen und Definitionen der Datenqualitätsforschung werden zentrale Dimensionen für Ereignisdaten identifiziert, darunter Vollständigkeit, Genauigkeit, Validität, Interpretierbarkeit und Relevanz. Diese Dimensionen bilden die Grundlage für ein neuartiges Data Quality Assessment Framework (DQAF), das entwickelt wurde, um spezifische Qualitätsprobleme in Ereignisprotokollen zu erkennen und zu bewerten. Das DQAF umfasst systematische Indikatoren und Bewertungsmaße für jede Dimension und ermöglicht so eine strukturierte Evaluierung der Datenqualität im Kontext des Process Mining.

Zur praktischen Implementierung dieses Frameworks wurde das Event Data Quality Dashboard (EDQD) als diagnostisches Tool für die Datenqualitätsanalyse entwickelt. Das EDQD visualisiert Datenqualitätsbewertungen und liefert Prozessanalysten Einblicke in potenzielle Mängel in Ereignisprotokollen, die die Ergebnisse des Process Mining verfälschen könnten. Die Evaluierung des EDQD umfasste drei Arten von Ereignisprotokollen: synthetische Datensätze, die speziell zur Isolierung bestimmter Datenqualitätsprobleme entwickelt wurden, öffentlich verfügbare Ereignisprotokolle der Business Process Intelligence Challenge sowie reale Datensätze eines österreichischen IT-Dienstleisters. Diese Evaluierungen zeigten die Fähigkeit des EDQD, eine Vielzahl von Datenqualitätsproblemen über verschiedene Protokollstrukturen und Domänen hinweg zu identifizieren, was die Robustheit des Frameworks und die praktische Anwendbarkeit des Tools in realen Umgebungen bestätigte.

Die Ergebnisse zeigen, dass das EDQD, das auf einem strukturierten Bewertungsframework basiert, Prozessanalysten effektiv dabei unterstützen kann, Datenqualitätsprobleme in Ereignisprotokollen zu diagnostizieren und zu erkennen. Durch die Anwendung des Tools auf verschiedene Datentypen verdeutlicht diese Arbeit die Bedeutung einer rigorosen Datenqualitätsbewertung für verlässliche und präzise Ergebnisse im Process Mining.

Contents

List of Tables

List of Figures

List of Algorithms

Listings

1. Introduction

Process mining is according to Van der Aalst the missing link between data mining and traditional model-driven Business Process Management (BPM) [16]. It is an analysis technique designed to unveil, inspect and enhance the process reality by processing event data of an information system. This reality is in most cases tremendously different to the human designed business process models within the organization [17].

Although the degree of knowledge is at a level never seen before there are still some areas in this domain that pose childhood disease that have been addressed by researchers around the globe. One of these issues, i.e. problems with the quality of data, is the very fabric of this master's thesis. As data, in form of an event log, is the basis of any process mining project issues can harm the results in manifold fashion.

Event data is the starting point of every process mining endeavor; it highly influences the quality and validity of the mining results. This means that unrecognized issues regarding the quality of event data might lead to faulty or even misleading results. We can subdivide the problems and challenges that arise when analyzing event logs into two high level categories: *process characteristics* and *quality of the event log* [7]. While the class of process characteristics is concerned with challenges such as fine-granular activities or process heterogeneity, problems in the quality of the event log class stem from issues related to the quality of logging.

The resulting master's thesis aims at synthesizing state of the art knowledge in regard to data quality in process mining. Furthermore, it is planned to assess a real-world information system in the Information Technology Service Management (ITSM) domain, gather its event data and check for data quality issues that might occur. This will be done by proposing a dashboard that analyses the event log and reveals quality issues to the process analyst.

1.1. Goals and Research Questions

The goal of the master's thesis is to investigate and evaluate data quality issues of event logs. These issues are manifold; there might be data missing, the data provided might be imprecise, wrong or irrelevant. In order to expose potential data quality issues it is planned to introduce a data quality dashboard, that gives an broad overview of how well-suited a given event log is in terms of processing that data by means of process mining.

The following research questions will be answered:

1. What are possible data quality issues of event logs?
2. How can a concept for the identification of data quality issues be specified?
3. How can a dashboard, that showcases identified data quality issues, be implemented?
4. How to evaluate the dashboard with a use case study?

1.2. Approach

First, the current state of knowledge will be synthesized, giving the reader an informed outlook on the topic itself. Second, during conceptualization, we will elaborate on common event log quality issues, what are ways to spot them, and how severe the issues found in the data really are. As an example we can imagine an event log that consists of the columns: case, event, timestamp and resource. As it would not be that problematic if we found several rows (process instances) where the resource information is missing this would be fatal in case that e.g. the timestamp was missing. Without proper timestamps it is not possible to perform process mining algorithms at all. This is just one example of many problems that arise due to data quality issues. Third, the implementation phase will take place. It is planned to create a data quality dashboard that takes an user provided event log and analyses it in regard to quality issues. After the analysis is finished the dashboard will display the findings to the user to help that person understand potential sources of error when performing process mining techniques with the log. Ultimately and forth, we will evaluate the implementation in a threefold manner: first by synthetic event logs where we are able to incorporate one specific quality issue at a time in a controlled manner. Secondly, by publicly available sample logs e.g. event logs of the Business Process Intelligence Challenge (BPIC)[1]. Here we are excited to see if these logs pose any quality issues at all. Third, the implementation will be tested with real-world data originating from the IT service sector. Hereby, the empirical utility of the dashboard will be stress-tested, leading to the realization whether its application is of value to the community.

1.3. Results

At the end of the master's thesis there should be following artefacts present in order to consider the thesis successful:

- Summarized state of knowledge in regards to event log quality issues.
- A concept for a data quality dashboard, consisting of a systematic list of potential quality problems, how they become apparent, what characteristics they exhibit, how we can search for them and what limitations there are.

[1] `https://data.4tu.nl/search?search=bpic`

- The implementation of a data quality dashboard that assesses the quality of event logs and reveals the presence of issues to the user.
- Evaluation results of the dashboard when testing the implementation with synthetic, publicly available and real-world data.

1.4. Outline

This thesis is structured into several key chapters, each building upon the previous one. After giving the reader a general overview of the scope of this thesis, Chapter 2 provides the necessary background information, introducing the reader to core concepts such as process mining and the use of event logs, which form the foundation for the subsequent discussions. In Chapter 3, the thesis delves into the literature on data quality, identifying the most significant dimensions of event data quality and reviewing previous research in this area. Chapter 4 introduces the proposed Data Quality Assessment Framework (DQAF), outlining the various checks and assessments that can be used to evaluate the quality of event data across multiple dimensions. Building on this framework, Chapter 5 presents the development of the Event Data Quality Dashboard (EDQD), a tool designed to operationalize these assessments in practice. The results of applying the dashboard to both synthetic and real-world event logs are discussed in Chapter 6, showcasing its practical application. Finally, Chapter 7 concludes the thesis by summarizing key findings, discussing limitations, and suggesting avenues for future research and development.

2. Background

2.1. Business Process Management

A business process consists of a set of activities that take input factors and transform these inputs into output factors. These activities are performed in a coordinated manner, with respect to the organizational and technical environment of the enterprise [18]. The main objective in this transformation procedure is to achieve predefined business goals, such as generating profit or increasing customer satisfaction. More often than not business processes interact with each other, hence increasing the complexity of the transformation. In order to guarantee consistent output factors business process management offers concepts, methods and techniques to design, steer, adapt and analyse these business processes. Dumas et al. [14] defined the so-called BPM life-cycle, showcasing the different stages of business processes along their lifeline.

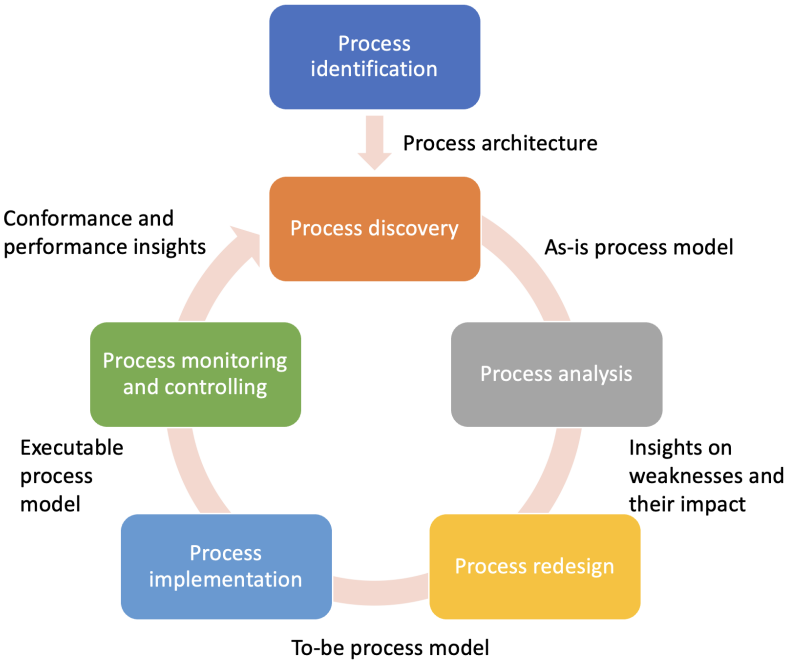

Figure 2.1.: The BPM life-cycle according to Dumas et al. [14]

As the discipline of business process management is an intersection of the domains of business administration and computer science it is natural that these stakeholder

view the subject from different angles. While the business administration community is interested in business orientated aspects, such as improving the operations, reducing cost or increasing production flexibility members of the computer science community are more interested about the technical circumstances of business processes. As business processes are realized in a diverse technology landscape the computer science community is furthermore interested in the dependencies and prerequisites to integrate processes into new information systems.

Coming back to the BPM life-cycle we can see that the very first step is *process identification*. This step is considered as starting point, as the realisation is crucial that even if there is no documentation about processes, no write-ups, no drawings - these processes exists. They exist on an implicit layer, where the operator of the processes knows what steps to do first, and which ones last. As an example we can imagine the everyday visit of a restaurant. We will elaborate on this example throughout the background section in order to give the reader a sound understanding of the concepts and techniques that are the basis of the later chapters of this paper. As the name business process management suggests, it is the business that is the origin of considerations when we inspect and analyse the processes. Just to give the reader the full scale overview the *process identification* stage would also involve determining all core, management and support processes that are established in order to enable value creation. We will leave this aspect aside as it is not necessary for the basic understanding of the BPM life-cycle. During the *process discovery* phase the processes are investigated in a one-by-one manner. We will focus on the customer serving process, without going into details about procurement, meal preparation, personal management, cleaning services and so forth.

On a trivial level, we can summarize the serving process to the following steps:

1. (Greet customer and) provide the menu
2. Take beverage and meal orders
3. Serve beverage
4. Serve meal
5. Bill (and farewell) customer

As this is an almost rudimentary summary of the process, it neglects the fact that there is a loop between taking the orders and serving them, as the customer can reorder items of the menu, order dessert after the main course, and many other aspects. The important thing is that we are now in possession of an ordered list of activities that have to be fulfilled in order to complete the customer serving process. We can obtain this list by the usage of different methods such as observation, interviews with staff or if available documentation. As we now have a clear understanding of the process we can model it using one of several modeling languages, Business Process Model and Notation (BPMN)[1]

[1] `https://www.bpmn.org`

being one of the widely used ones. The discovered and human crafted process model would then look something like this:

Figure 2.2.: Rudimentary serving process model in BPMN

The process model at hand is the so-called as-is process model. It is the initial point for the next stage of the BPM life-cycle the *process analysis* step. Here we can differentiate between analysis methods that are of qualitative or quantitative nature. Analysis methods of the *qualitative process analysis* sphere are:

- Value-Added Analysis
- Root Cause Analysis
- Issue Documentation
- Impact Assessment

When we look back at our restaurant example and the serving process that we have derived we can perform, depending on the information that are accessible to the process analyst[2], several quantitative process analysis. We could perform e.g. the value-added analysis to find out which steps of the process actually generate value to the business and which ones are supporting to enable the value creation. The main goal of the value-added analysis is to identify process steps that bind resources but do not generate any value at all. If activities that meet the conditions are found it is desirable to eliminate these activities, thus remove waste from the process. Waste is in this context considered as work that has to be done, but is not beneficial for neither the customer nor the enterprise. As the designed process model is already highly compact all of the steps are necessary, we can not eliminate a process step as it would make the process incomplete. Nevertheless we can argue that the steps *serve beverage* and *serve meal* generate the most value for the customer, whereas the step *bill and farewell customer* generate the most value for the enterprise. This leaves the steps *greet customer and provide the menu* and *take beverage and meal orders* as not directly value-adding steps, that are still important in order to guarantee the overall completion of the process.

On the other side, there are analysis techniques used in the field of *quantitative process analysis*, including:

[2]This is the person that is responsible for the creation, adaption and enrichment of the Business Process Management System (BPMS)

- Performance Measures
- Flow Analysis
- Queues
- Simulation

Again, putting these concepts into practise the process analyst can for example perform a flow analysis of the serving process. This quantitative process analysis technique focuses on the calculation of cycle time. Cycle time is considered the time it takes a team or person to produce or fulfil a service until it is delivered. In a one-by-one manner the process analyst would measure the time it takes to fulfil every task along the process model. We can subdivide the time an activity takes to be considered completed into several categories, the most prominent are working time (value-added time) and waiting time (non-value-added time) [19]. There are hence a multitude of possibilities to reduce, eliminate or shift working as well as waiting time, with the grand goal of reducing the overall cycle time. The gathering of the cycle time can be done in a manual manner, by the usage of a stop watch or by means of data that is collected by a logging system of the enterprise. Both approaches have their advantages and disadvantages, manually recorded cycle time is prone to the Hawthorne effect [20] meaning that persons that are observed tend to act in a different way than they would normally do. When we think about our restaurant example, this means that the waiter that is observed by the process analyst while performing the serving process would work much faster than in a setting were he is not monitored. Thus leading to skewed time measures that are way shorter than the actual time it takes to perform a task. Automatically recorded time measures incline to factor out the Hawthorne effect but tend to be incomplete, hence measure only partitions of the processes and not each and every step.

To make this point clear lets inspect the touch-points with information systems that the waiter has when performing the serving process. When providing the customer with a menu and greeting that person there are no information system in play, it is just manual work that does not depend on any kind of technology. So we do not obtain the actual staring point of the customer journey when we solely depend on IT systems that document what happened. Only at the second step of our identified serving process, when the waiter takes the beverage and meal orders and forwards them to the kitchen[3] the first touch-point with IT systems has taken place. The next steps of the process, serving the beverage and serving the meal also do not depend in most cases on any information system, it is just manual work. Only once we get to the last activity of the process, when the waiter bills the customers orders we then have our second and last touch-point with an information system. If we look at the example we can see that only two of five tasks, that is 40% of the overall

[3]Some restaurants still work with hand-written notes as communication system between service staff and kitchen, but their number is decreasing in a fast way. Most modern restaurants use smartphones or tablets and an IT supported ordering system that logs the time when a order was taken, and of course, what kind of order it was.

process, is documented by means of information systems. This is in fact a huge problem when performing process analysis that exclusively depend on automatically generated data as basis for analysis. As less than half of the real-world information is covered in the data collected the results of the analysis should be treated carefully by the process analysts.

During the process analysis phase the process analyst notes down the results of the performed process analysis. The process analyst then takes the newly generated knowledge into account when documenting the insights of weaknesses and their impact. With these insights at hand the process analyst proceeds to the next step of the BPM life-cycle, i.e., the *process redesign* phase. Due to the fact that the first steps of the BPM life-cycle are of high importance to the overall understanding of the following chapters and the later steps are not, we will finish up this subsection without elaborating further. If the reader wishes to continue to learn more about the BPM life-cycle and get a deeper understanding of it, we highly recommend the book *Fundamentals of Business Process Management* by Dumas et al. [14].

2.2. Data Mining

Data mining is the process of identifying patterns and extracting information from big data sets using techniques that combine machine learning, statistics, and database systems. With the overarching objective of extracting information from a data collection and structuring the information into a comprehensible form for subsequent use, data mining is an interdisciplinary branch of computer science and statistics [21]. The analytical stage of the Knowledge Discovery in Databases (KDD) process is known as data mining [22]. Along with the raw analysis stage, other components of the process include database and data management, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of found structures, visualization, and online updating.

Data mining can be subdivided into six common classes of tasks [23] summarized in Table 2.1 and 2.2. We will introduce these tasks and give an example to each one following up on our illustration of the restaurant example.

While data mining helps enterprises to understand the underlying structures of its data it lacks in terms of understanding the enterprises processes and procedures. Data mining can be of great help when planning marketing campaigns, adapting the pricing of products and services and many other aspects. Nevertheless, behavioral aspects e.g., how certain tasks are performed, where bottlenecks lie and so forth are not uncovered by classical data mining techniques. In order to close this gap, the process analysis technique of *process mining* was developed.

Class of task	Description	Example
Anomaly detection	Identification of unexpected data records that may be intriguing or data problems that need to be investigated further.	While analysing the billing data of the restaurant we find an invoice that was € 4682.68 which was an absolute outlier as the median invoice amount is € 54.21. We now have to investigate if this invoice was wrongly documented (data quality issue) or if a company celebration or some kind of large scale event has taken place that would legitimise the height of the invoice.
Association rule learning	Is a machine learning strategy that uses rules to uncover novel relationships between variables in large data sets. Its goal is to detect strong rules identified in databases using various interestingness criteria.	The restaurants invoice database can be searched for products or services that are often bought together. Furthermore, this knowledge about products that are frequently bought together might be used for marketing purposes; e.g., for bundling or cross-selling. In literature this is sometimes referred to as market basket analysis [24].
Clustering	Is the task of arranging a set of objects in such a way that items in the same group (cluster) are more similar to each other than to items of another group.	When analysing the order data we will uncover that there are two or many types of events that take place at the restaurant. The first and probably most common event will be a group (this can be a family or a group of friends) dinner where there will be more meals consumed than drinks. In contrast to the first group we will find a set of visits where more drinks are consumed than dishes, leading to the realisation that there is a cluster of social gathering that takes place at the restaurant.
Classification	Generalization of known structure that is applied to new data	Following up on the clustering example, when we know that there are more meals consumed at a group dinner, we can classify these occurrences in run-time as they evolve.

Table 2.1.: Classes of Data Mining tasks

Class of task	Description	Example
Regression	Attempts to discover a function that, for estimating the relationships between data or data sets, models the data with the least amount of error.	The management of the restaurant could approximate the number of waiters that is needed in order to optimise customer satisfaction and cost.
Summarization	Is a task of visualizing the data in a comprehensive and compact representation. Dashboards and reports are the format of choice.	The owner of the restaurant wants an weekly report of how many guests where served, what the average consumption was, et cetera.

Table 2.2.: Classes of Data Mining tasks

2.3. Process Mining

Process mining can be positioned as an intersection between the two previously mentioned disciplines of business process management and data mining. It combines, by the usage of algorithms, the procedural character of business processes models and generates insights in an automated manner from data [16]. Process models that are designed by humans often differ from models that are discovered from the data of the enterprise. This is due to the difference of where the information, that is essential for the generation of a process model, came from.

The common business process discovery technique involves process analysts and domain experts and business experts. The latter provide information of the process in an interview or workshop with the process analyst. With no doubt this method is biased by subjective viewpoints as well as department interests of the involved parties. By taking out the subjective factor we primarily focus on a data-driven and objective picture of the process reality. A great number of research already point out the advantages of process mining in many sectors, such as healthcare [25, 26], the pubic sector [27], and finance [28]. There are many more, this is just a short list of all the publications that were published in the last years.

Generally speaking, process mining consists of three types, i.e.; *discovery*, *conformance* and *enhancement* [10]. The aim of discovery is to point out the process reality which is stored within the event data of the information system of an organisation. The variety of already explored discovery-algorithms makes it easy for the analyst to choose the notation and representation that fits the scope of the analysis the most. There are many commercial as well as open source software tools that allow to achieve proper results fast

and simple [29].

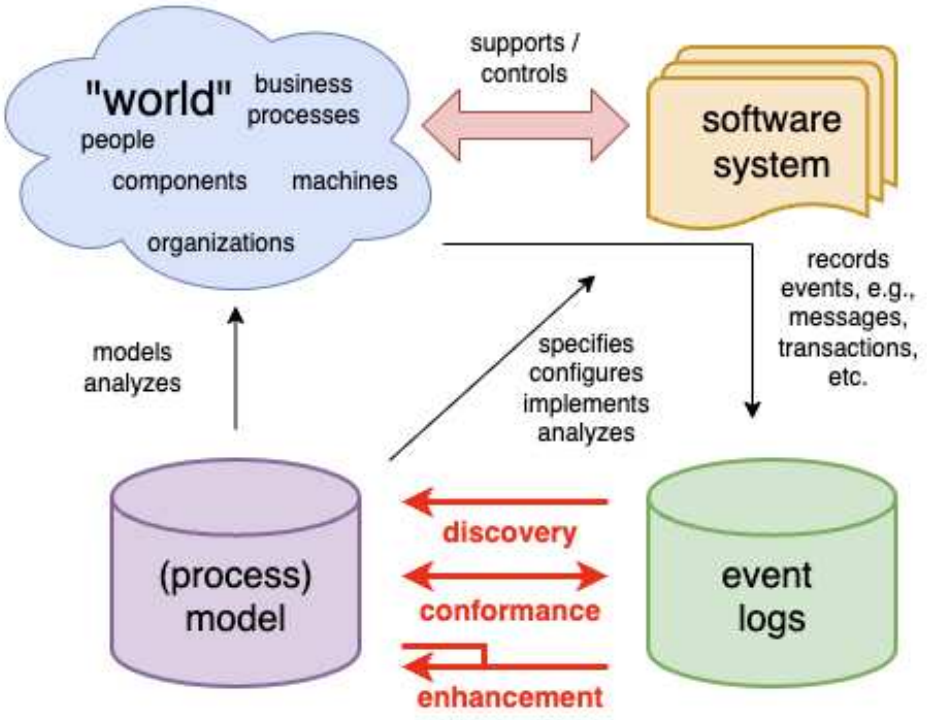

Figure 2.3.: Positioning of the three main types of process mining: *discovery*, *conformance*, and *enhancement* according to Van der Aalst [10]

In contrast to discovery conformance, or sometimes also known as conformance checking, is concerned with the analysis of the similarities as well as differences between the designed process model and the de facto process reality, which is harvested by the process mining algorithm. More often than not the process reality, i.e. the process model gathered by analysing the event log of the infomation system, is significantly different than the designed process model which was sketched by an process analyst. Conformance checking can be of great benefit, as it unveils unknown, misleading or falsely assumed characteristics of the inspected process and hence be of help in adapting the process model in a more accurately manner. The third and last type of process mining is enhancement. Here we take the process model as basis and analyse during run-time if the system is running according to plan or not. This aspect is often described as operational support, it is concerned with detecting and predicting process instances that are running out of hand. Figure 2.3, gratefully taken from [10], p. 9, showcases the different types of process mining in a compact and comprehensible way. We see how aspects of interest, be it the real world, the software system, a process model or event logs are connected to each other and how they might influence one another. The red arrows represent the previously stated types of process mining and illustrate the important relationship between process model and event logs.

Case	Activity sequence
C_1	$< a, b, c, d, e >$
C_2	$< a, b, b, c, d, e >$
C_3	$< a, b, c, d, e >$
C_4	$< a, b, b, b, c, d, e >$
C_5	$< a, c, b, d, e >$

Table 2.3.: Simplified event log [10]

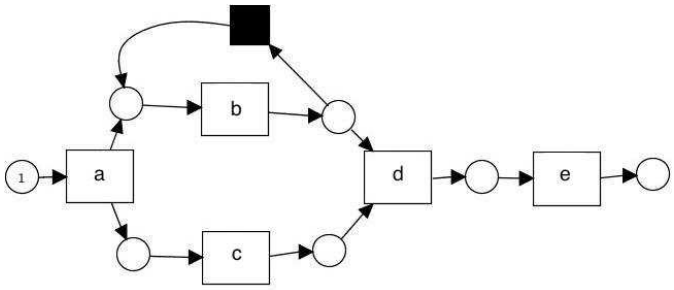

Figure 2.4.: Petri net of the example log [10]

Nevertheless, process mining is not limited to pointing out process models, which is considered as *control-flow perspective*. There are four perspectives that can be distinguished that are subject to process mining. We are also able to access information on how often certain resources work together and how well this cooperation is situated. This is considered as *organisational perspective*. The *case perspective* focuses on homogeneous characteristics of cases, whereas the *time perspective* emphasises the time-frame and frequency of events [10].

Starting point for any process mining analysis is an *event log*, which represents a selected set of the overall event data. In this event log process instances, i.e. *cases* sometimes *traces*, and the corresponding process steps, i.e. *events*, are recorded sequentially in such way that every event refers to an *activity* and a case. Activities are well-defined steps in a process. To be able to perform a process mining analysis three data elements are mandatory; the *case identifier*, activities and the related *timestamp*, which logs when a particular activity was performed. Additionally, more detailed information about events can be stored. There are two categories of additional information, *resources* and *data elements*. Resources might be persons, groups, roles, machines, or digital entities such as agents or bots, which perform a certain activity. Data elements can be considered as additional attributes enriching events with more in-depth information about the execution parameters, such as description, documentation, size of order, temperature, et cetera.

Table 2.3 and Figure 2.4 visualise process mining in a simplified manner. In Table 2.3 we see the traces (C_1 to C_5) in the left column and the corresponding executed activities (a to e), i.e. events, in the right column. Figure 2.4 is the result of performing the *Alpha #* algorithm in ProM[4] version 6.12 on the information in Table 2.3. The visualising notation is *PetriNets*. It represents the event log as a Petri net that allows all sequence flows that were recorded in the log. The black box is a so called τ-activity that is used as a non-recorded activity enabling loop behaviour.

For better understanding we can imagine our serving process, in which activity a

[4] `https://promtools.org/`

represents the process step *greet customer and provide menu*, activity b: *take beverage and meal orders*, activity c: *serve beverage*, activity d: *serve meal* and activity e: *bill and farewell customer*. If we take a closer look into Table 2.3, we see that not all cases follow the same sequence. Let's assume that the activity sequence of C_1 and C_3 is the desirable process sequence of the organisation in regard to completeness and cycle-time. We see that three cases do not follow this desired process sequence and are able to derive differences between the executions. When we reminisce about our self-designed serving process model, i.e. Figure 2.2, we can see how tremendously different the process looks like when it is reconstructed based on recorded data. It is of high importance that the process analyst has a sound understanding of the process analysed, as it enables that person to tell with certainty if there are issues with the a priori process model or if there are issues concerning the recorded data. Domain experts are the go-to place to grasp this knowledge, they should be involved frequently and not only to verify already found results.

One example of deviation when we compare the discovered process model (Fig. 2.4) and our a priori process model (Fig. 2.2) is that activity b(take beverage and meal orders) can, but must not, be executed multiple times within one case. In practice the process analyst would discuss this finding with domain experts in order to find out if there is a problem with the designed process model, or if the recorded data is faulty. In this case the a priori process model is lacking the possibility to loop, i.e., allow the customer to order several times during that persons restaurant visit. We found a deviation between the two process models and should now alter our self-designed process model, as it is the one that lacks important information, and adapt it in a way, that allows for a reoccurring happening of activity b.

Another example can be found in C_5, to be precise in the succession of activity b and activity c that are interchanged. The serving of the beverage was performed first and afterwards the beverage and meal order has been taken. When we think about this case, it comes natural as some guests of the restaurant already know its beverage menu, and are thirsty, hence want to get something to drink as soon as possible. We see that our self designed process model lacks this common practice and should be remodeled in a way to allow such behaviour. In this case the simple introduction of a parallel gateway would solve the issue. Nevertheless, this alternation only makes sense when this behaviour is often recorded. If the number of cases in which the ordering of beverage and meal is done before the serving of the beverage is significantly higher then the other way around it would be advisable to keep the order of the sequence flow as is. The danger that is anchored in this consideration is called *over-fitting*. This happens when we try to allow every sequence pattern that we found in the data, leading to spaghetti-like process models that are hard, maybe even unreadable, for human[10].

In conclusion, we have performed all three types of process mining on a very fundamental level. By transforming the event log into a process model, we accomplished discovery (Fig. 2.4). By defining a reference model (Fig. 2.2) and comparing the activity sequences of all

cases with this model we completed conformance and found several deviations between a priori model and the discovered process model (multiple executions of activity b, the interchange of activity b with activity c in C_5). Finally, by investigating the root-cause for the non-compliant behaviour we proposed improvements to the process (changes of the a priori model) and hereby facilitated a generic understanding of enhancement.

2.4. Event Logs

It is impossible to perform a process mining analysis without event logs. Additionally, not all event logs entail the information needed in order to perform process mining of a certain perspective. While information of resources, i.e., the entity that performs specific activities of the process, are mandatory if the process analyst wants to mine for the organisational perspective, the resource is not essential when mining for the control-flow perspective. Generally speaking, depending on the process mining technique performed, the information needed variegates [10].

We can find event logs in a variety of sources, e.g. document management systems, Enterprise Resource Planning (ERP) systems, databases, message logs, flat files and transaction logs. Furthermore, we are not obliged to use information from a single source. It is possible, and sometimes even necessary, to combine information from two or more data sources in order to generate the data that holds the information that one requires. Questions should be the guiding principle when extracting and merging event data, as the abundance of data in today's world is exhausting.

Table 2.4 illustrates the typical information present in an event log. For the sake of consistency we stay with our example of the restaurant and the serving process. We have already learned in Section 2.3 that every row of the table is called an event. Every event corresponds to one case, represented by the *Case id*. Properties, that are located in the columns *Timestamp*, *Activity*, *Resource* and *Table*, are considered as so-called *attributes*. We can distinguish between two kinds of attributes; ones that are a property of an event and others that are a characteristic of a whole case. At its core, we can spot the difference when we take a closer look at the modification of these attributes. Basically, if the attribute is changing along the progress of a case, we consider it an *event attribute*. And if it does not change and stays throughout the case the same we consider it a *case attribute*. When we look more closely at Table 2.4 we can see that the attribute of the resource column is changing along the progress of every case. We would consider the attribute of resource as an event attribute. On the other hand, the attribute of the table column would be regarded as case attribute as its value stays the same throughout the whole case.

Event logs contain a lot of information, some are accessible by the bare eye, others have to be analysed in depth to generate new insights. Without jumping to conclusions we can see that certain activities are performed by different actors, and that some activities are

Case id	Timestamp	Activity	Resource	Table
Guest9723	2022-10-25 17:33:23	Check-in table	Tom	5
Guest9723	2022-10-25 17:38:39	Take orders	Tom	5
Guest9723	2022-10-25 17:46:19	Drinks ready	Sue	5
Guest9723	2022-10-25 17:56:11	Meals ready	Tony	5
Guest9723	2022-10-25 18:34:51	Bill customer	Tom	5
Guest9724	2022-10-25 17:44:41	Check-in table	Alice	11
Guest9724	2022-10-25 17:45:39	Take orders	Alice	11
Guest9724	2022-10-25 17:59:46	Drinks ready	Sue	11
Guest9724	2022-10-25 18:11:09	Meals ready	Tony	11
Guest9724	2022-10-25 18:41:37	Take orders	Alice	11
Guest9724	2022-10-25 18:59:05	Meals ready	Tony	11
Guest9724	2022-10-25 19:56:45	Bill customer	Alice	11
Guest9725	2022-10-25 17:56:17	Check-in table	Tom	3
Guest9725	2022-10-25 18:12:01	Take orders	Tom	3
Guest9725	2022-10-25 18:26:48	Drinks ready	Sue	3
Guest9725	2022-10-25 18:42:53	Meals ready	Tony	3
Guest9725	2022-10-25 19:28:44	Bill customer	Tom	3

Table 2.4.: Example event log

bound to these actors while others are not. Without the concept of roles being present in the event log, we can approximate which person is responsible for what tasks, hence determine their roles. For example, Sue is always responsible for getting the drinks ready, while Tony is always in charge of the meals. Given this observation, we can assume that Sue is the barkeeper of the restaurant, and Tony is a chef. Alice and Tom are responsible for the same tasks, suggesting that they possess the same role, most likely as waitress and waiter. Again, we have to stress the fact that newly generated insights have to be reviewed by domain experts as these people can verify the validity of the insight or dismiss them. The main takeaway of this example is that we can find new insights in the data that was not present in the first place. We are able to then take the new information into account when performing additional analysis.

Key Terminology Overview:

- **Event log:** A tabular or list-like representation of all logged events.
  - Composed of multiple traces.
- **Trace:** A sequence of events with defined start and end points, representing stages (activities) in a process.
  - Made up of a series of events.
  - Can contain trace attributes.

- **Event:** A single occurrence of an activity within a trace.
  - Contains multiple attributes, including a mandatory timestamp.
- **Attribute:** A defining characteristic of a log, trace, or event.
  - Composed of a type, value, and key.
  - Common attributes include activity name, timestamp, and resource.
  - Attributes may include nested sub-attributes.

More often then not event logs are documented in plain text or in databases. Most practitioners of process mining obtain their event logs in a *.txt*, i.e., plain text file format or in a *.csv*, i.e. comma-separated values format. Although these formats are perfectly fine for process mining techniques, they might lack proper semantics and ontology. Depending on which process mining tool the process analyst uses, a transformation to the de facto standard for storing and exchanging event logs, i.e. the eXtensible Event Stream (XES) format, has to be performed.

2.5. XES

What better way to introduce a standard file format as quoting the purpose of it in the very words of its creators:

> The purpose of this standard is to provide a generally acknowledged XML format for the interchange of event data between information systems in many applications domains on the one hand and analysis tools for such data on the other hand. As such, this standard aims to fix the syntax and the semantics of the event data which, for example, is being transferred from the site generating this data to the site analyzing this data. As a result of this standard, if the event data is transferred using the syntax as described by this standard, its semantics will be well understood and clear at both sites.[5]

In essence, it is an XML format that helps information systems understand the semantics of the event log. We as human understand the meaning of activity name, resource and timestamp by name, machines do not. Machines understand due to the syntax (YYYY-MM-DD hh:mm:ss or some deviation of it) of a timestamp that it is one. This is not the case when we think about the case id, the attributes of activity name, resource and so forth. For information systems these are all just strings, the machine can not differentiate between those strings. In order to resolve this issue, the XES format defines a standardised way of how certain attributes are saved and interpreted later on. On top of general aspects of events and their attributes, XES allows the user to design ones very own extension, incorporating domain, or even organisation, specific information into the event log. Figure 2.5 illustrates the meta-model of the XES standard. The

[5]Purpose section of https://xes-standard.org

structure of XES is similar to the structure we defined in Section 2.4 as we summarized the terminology. On the top level there is one *XLog* object (tag name <log>). This object contains all event information captured by one process. The XLog object contains multiple traces, i.e. *XTrace* objects (tag name <trace>). Every trace consists of multiple events, i.e. *XEvent* objects (tag name <event>). XLog, XTrace and XEvent objects hold no information themselves, they only serve as the data structure. The real information is stored in attributes.

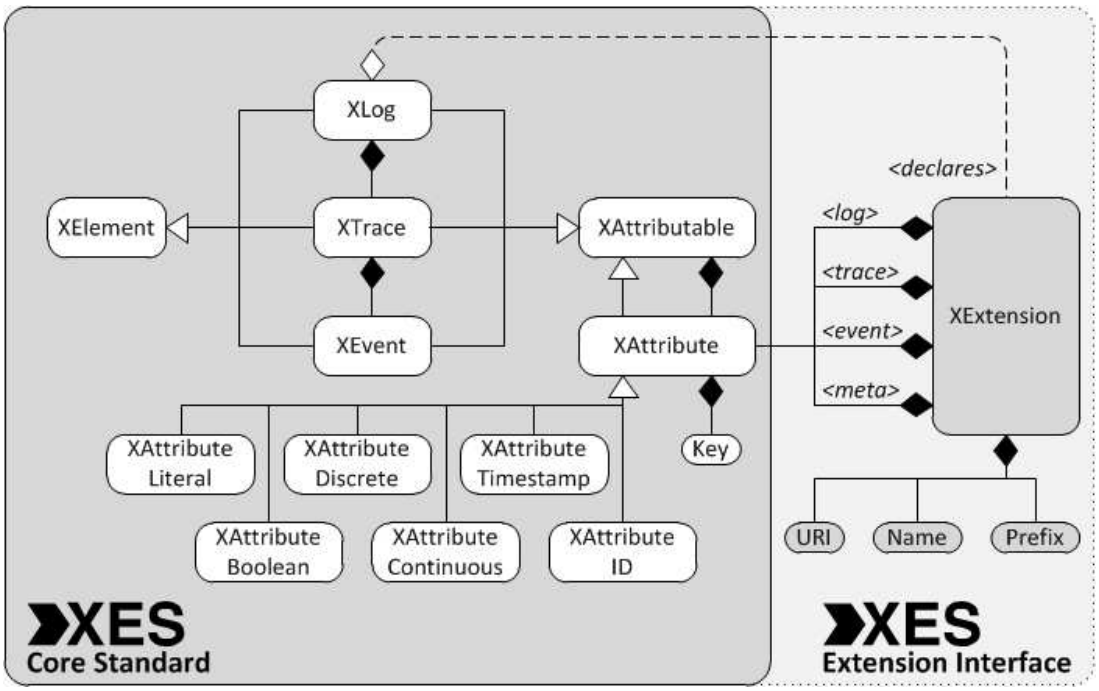

Figure 2.5.: Meta-model of the XES standard [15]

To conclude this chapter, Listing 2.1 showcases the very same event log that we already know from Table 2.4 in the XES format. Although Table 2.4 is easier to read and understand for human its semantics are not clear to the process mining tool. With the help of XES the data now has clear, machine readable semantics that are needed in order to perform process mining techniques.

At last we want to emphasise the fact that all of these examples suggest the presence of well defined IT systems that work in a coordinated manner with a shared name-space of data entities. This example is, when you will, the best case scenario. In practise we can not expect these perfect conditions. Software systems will be manifold, they will document the same piece of information in different databases, with different names meaning exactly the same object. The format of timestamps will differ depending on the information system that logs a certain event. The identifier of each case will be named

different, depending on the purpose of the software. These and many more issues will arise when we take a deep dive into the world of process mining. In the next chapter we will elaborate on different data quality issues that arise when working with event logs.

```
<?xml version="1.0" encoding="UTF-8" ?>
<log xes.version="1.0" xes.features="nested-attributes" openxes.version="1.0RC7">
  <extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xes)ext"/>
  <extension name="Lifecycle" prefix="lifecycle" uri="http://www.xes)-standard.org/lifecycle.xesext"/>
  <extension name="Concept" prefix="concept" uri="http://www.xes-standard.org/concept.xesext"/>
  <classifier name="Event Name" keys="concept:name"/>
  <string key="concept:name" value="xes Event Log"/>
  <trace>
    <string key="concept:name" value="Guest9723"/>
    <event>
      <string key="org:resource" value="Tom"/>
      <int key="Table" value="5"/>
      <string key="concept:name" value="Check-in table"/>
      <string key="lifecycle:transition" value="complete"/>
      <date key="time:timestamp" value="0025-10-22T17:33:00.000+01:00"/>
    </event>
    <event>
      <string key="org:resource" value="Tom"/>
      <int key="Table" value="5"/>
      <string key="concept:name" value="Take orders"/>
      <string key="lifecycle:transition" value="complete"/>
      <date key="time:timestamp" value="0025-10-22T17:38:00.000+01:00"/>
    </event>
    <event>
      <string key="org:resource" value="Sue"/>
      <int key="Table" value="5"/>
      <string key="concept:name" value="Drinks ready"/>
      <string key="lifecycle:transition" value="complete"/>
      <date key="time:timestamp" value="0025-10-22T17:46:00.000+01:00"/>
    </event>
    <event>
      <string key="org:resource" value="Tony"/>
      <int key="Table" value="5"/>
      <string key="concept:name" value="Meals ready"/>
      <string key="lifecycle:transition" value="complete"/>
      <date key="time:timestamp" value="0025-10-22T17:56:00.000+01:00"/>
    </event>
    <event>
      <string key="org:resource" value="Tom"/>
      <int key="Table" value="5"/>
      <string key="concept:name" value="Bill customer"/>
      <string key="lifecycle:transition" value="complete"/>
      <date key="time:timestamp" value="0025-10-22T18:34:00.000+01:00"/>
```

```
    </event>
  </trace>
...
</log>
```

Listing 2.1: XES format of case id Guest9723 of the example event log

3. Literature Review

This thesis covers a number of event Data Quality (DQ) metrics, all of which are based on literature-based research. Studies about data quality that serve as foundation for this thesis are mentioned in this chapter. The content focuses on the many definitions, frameworks, aspects and dimensions of DQ. Additionally, the second section puts special emphasis on event data and the needs of process mining. Several academic papers about DQ dimensions that describe the goodness of generic data are presented in Section 3.2. This entails not paying close attention to event data. Event data quality dimensions and particularities when dealing with event data are the main subject of Section 3.3. In Section 3.4 we will point out differences between general data quality dimensions and those dimensions specific to process mining. We will conclude this section with an overview table of all mentioned DQ dimensions and where they appear in literature.

3.1. Literature Review Approach

This chapter is designed as a semi-systematic literature review [30]. The reasons for choosing this type of literature review type was the fact that the available publications differ in their approaches and have been conceptualized differently making it difficult, almost impossible to conduct the paper as fully systematic literature review. Furthermore, it was the aim of synthesising the state of knowledge and exploring areas of interest that had not been researched before that led to the writing of this paper. This reinforces the need to choose this type of literature review.

To conduct our literature research we focused on the terms *data quality* in combination with *dimension* or *assessment* or*event log* or *process mining* which we searched in the online databases u:search[1], IEEE Xplore[2] as well as google scholar[3].

As it surprised us that there were only few publications in this field of interest we took an additional approach in searching for academic papers that match the scope of this thesis. After skimming through the abstracts of the publications that were found during regular search and confirming that their content matches the scope of this survey we inspected other publications that reference their work based upon the papers found.

[1] `https://usearch.univie.ac.at`
[2] `https://ieeexplore.ieee.org`
[3] `https://scholar.google.com`

3.2. Data Quality

In this section, state of the art on data quality is presented. The subsections are named after the authors of the papers we found during the literature review phase of this thesis. We will pay particular attention to the dimensions of DQ and how they are differently named and described throughout these publications.

3.2.1. Pipino et al. [1]

Pipino et al. [1] place data quality as a multi-dimensional concept. That is, the assessment of DQ must be divided into two aspects: the subjective perception of the individuals that work with the data in question and an objective measurement of the data.

1. **Subjective data quality assessments:** Summarize the needs and experiences of different stakeholders that use the data. The authors differentiate between collectors, custodians, managers and consumers of data products. One way, that has proven useful, to gather subjective perception of a given data set is the usage of a questionnaire. Stakeholders are invited to rate the data quality dimensions as seen in table 3.1 on a scale from 0 to 9.

2. **Objective data quality assessments:** Can be further divided into task-independent and task-dependent assessments. Task-independent metrics can be applied to any data set and do not take contextual or domain knowledge of the data into account. On the other hand, task-dependent metrics incorporate contextual information such as constraints of the database system, business rules and government regulations when dealing with data.

The authors also describe three functional forms for assessing specific quality dimensions. According to [1], simple ratio, min or max operation and weighted average are the main building blocks for most objective quality assessments. Finally, we would like to emphasise that in this paper we will mainly focus on the objective assessment of DQ.

Dimensions	Definitions
Accessibility	the extent to which data is available, or easily and quickly retrievable
Appropriate Amount of Data	the extent to which the volume of data is appropriate for the task at hand
Believability	the extent to which data is regarded as true and credible
Completeness	the extent to which data is not missing and is of sufficient breadth and depth for the task at hand
Concise Representation	the extent to which data is compactly represented
Consistent Representation	the extent to which data is presented in the same format
Ease of Manipulation	the extent to which data is easy to manipulate and apply to different tasks
Free-of-Error	the extent to which data is correct and reliable
Interpretability	the extent to which data is in appropriate languages, symbols, and units, and the definitions are clear
Objectivity	the extent to which data is unbiased, unprejudiced, and impartial
Relevancy	the extent to which data is applicable and helpful for the task at hand
Reputation	the extent to which data is highly regarded in terms of its source or content
Security	the extent to which access to data is restricted appropriately to maintain its security
Timeliness	the extent to which the data is sufficiently up-to-date for the task at hand
Understandability	the extent to which data is easily comprehended
Value-Added	the extent to which data is beneficial and provides advantages from its use

Table 3.1.: Data quality dimensions according to Pipino et al. [1]

3.2.2. DAMA UK Working Group [2]

The Data Management Association (DAMA) is a global community of data management professionals organised around local chapters. DAMA UK is a local chapter formed in 2002. In October 2013 they released a paper detailing their view of the six key dimensions of data quality assessment. In contrast to Pipino et al. [1], the authors focus on a compact list of only six DQ dimensions but go into more detail in explaining them. The paper is structured in such a manner that after giving the reader some preliminary information on the context of the paper they proceed to outline the six core data quality

dimensions according to them. After establishing an generic understanding of the matter they continue by systematically describing each and every quality dimension on its own. Listed in tabular form the authors elaborate on the aspects: title, definition, reference, measure, scope, unit of measure, related dimension, optionality, examples and pseudo code of every dimension.
In the following we will condense these tables, mainly focusing on the quality dimension and its definition, and illustrate it in table 3.2.

Dimension	**Definition**
Completeness	The proportion of stored data against the potential of "100% complete"
Uniqueness	No thing will be recorded more than once based upon how that thing is identified.
Timeliness	The degree to which data represent reality from the required point in time.
Validity	Data are valid if it conforms to the syntax (format, type, range) of its definition.
Accuracy	The degree to which data correctly describes the "real world" object or event being described.
Consistency	The absence of difference, when comparing two or more representations of a thing against a definition.

Table 3.2.: Data quality dimensions according to DAMA UK [2]

We will take into account the other attributes mentioned in the paper (reference, measure, etc.) when we later on synthesise our own descriptive representation of the relevant data quality dimensions for assessment.

3.2.3. Caballero et al. [3]

The study by Caballero et al. [3] points out and tries to solve one problem of the literature that is concerned with data quality measurement, i.e. the lack of unification of the nomenclature of DQ issues. That is, different authors call the same concepts in different ways, leading to confusion when a novice DQ management practitioner starts learning about data quality management and measurements thereof in particular. We can see this realisation when looking at our first two sources of data quality studies mentioned earlier. While [1] name the concept of data that conforms to a predefined syntax *Consistent Representation* we realise that [2] call this concept *Validity*. This is just one example of the differences in naming-schemes throughout the DQ literature, we are going to experience this a lot. In order to resolve this issue the authors propose a *Data Quality Measurement Information Model* which provides a standardization of the referred terms taking ISO/IEC 15939 as basis. In a similar vain as in [1], Caballero et al. subdivide the measures into the two aspects we already know from Pipino et al., i.e., subjective and objective measurement concepts. Table 3.3 shows their synthesised measurement

concepts divided into subjective and objective measures and enriched by methods of how to generate metadata for these measures.

Metadata Source	Measurement Concept	Methods of generation metadata
Subject (user)	Believability	User Experience
	Concise Representation	User Sampling
	Interpretability	User Sampling
	Relevancy	Continuous user assessment
	Reputation	User Experience
	Understandability	User Sampling
	Value-Added	Continuous user assessment
Object (data store)	Completeness	Parsing, Sampling
	Customer Support	Parsing, Contract
	Documentation	Parsing
	Objectivity	Expert input
	Price	Contract
	Reliability	Continuous assessment
	Security	Parsing
	Timeliness	Parsing
	Verifiability	Expert input

Table 3.3.: Data quality dimensions according to Caballero et al. [3]

Instead of following a ISO/IEC standard we will take a different approach in consolidating the varying terms of data quality measurement concepts that we find in literature. We will focus on concepts that are mentioned most often and point out similar data quality dimensions in an tabular overview at the end of this chapter.

3.2.4. UNECE Big Data Quality Task Team [4]

Dufty et al. [4] elaborate on a comprehensive framework tailored for assessing the quality of Big Data, addressing the unique challenges it presents beyond traditional data sources. It introduces three so-called hyperdimensions, i.e., *Source*, *Metadata*, and *Data* as pivotal in evaluating the quality of data. These hyperdimensions encapsulate various quality dimensions such as Institutional Environment, Privacy, Security, Complexity, Completeness, Usability, Timeliness, Accuracy, Coherence, and Validity. The framework emphasizes the significance of considering the data origin, the metadata that describes it, and the data itself through these dimensions, ensuring a holistic assessment approach.

Hyperdimension	Quality Dimension	Factors to consider
Source	Institutional/Business Environment	Sustainability of the entity-data provider Reliability status Transparency and interpretability
	Privacy and Security	Legislation Data Keeper vs. Data provider Restrictions Perception
Metadata	Complexity	Technical constraints Whether structured or unstructured Readability Presence of hierarchies and nesting
	Completeness	Whether the metadata is available, interpretable and complete
	Usability	Resources required to import and analyse Risk analysis
	Time-related factors	Timeliness Periodicity Changes through time
	Linkability	Presence and quality of linking variables Linking level
	Coherence - consistency	Standardisation Metadata available for key variables (classification variables, construct being measured)
	Validity	Transparency of methods and processes Soundness of methods and processes
Data	Accuracy and selectivity	Total survey error approach Reference datasets Selectivity
	Linkability	Quality of linking variables
	Coherence - consistency	Coherence between metadata description and observed data values
	Validity	Coherence between processes and methods and observed data values

Table 3.4.: Data quality dimensions according to UNECE [4]

Table 3.4 gives an structural overview of the hyperdimentions, their quality dimensions and factors that the authors emphasise to consider when dealing with Big Data.

By integrating these hyperdimensions, the framework aims to guide practitioners through the structured evaluation of Big Data across its lifecycle, from collection (input) through

processing (throughput) to its final form (output). This methodology acknowledges the distinct attributes of Big Data, including its volume, velocity, variety [31], and the complexities associated with its processing and analysis. The dimensions underscore the necessity for robust governance, ethical considerations, and the adaptability of DQ measures to maintain the integrity and usefulness of Big Data within statistical and analytical practices.

This Big Data Quality Framework represents a paradigm shift in data quality assessment, recognizing the evolving nature of data ecosystems and the technological advancements that drive them. It seeks to provide a standardized approach to assessing Big Data quality, facilitating better decision-making and enhancing trust in data-driven insights. Through the detailed examination of each hyperdimension and its associated quality dimensions, the framework offers a path forward for organizations navigating the complexities of Big Data, ensuring that such data is leveraged responsibly and effectively for the benefit of all stakeholders involved.

3.2.5. Wang et al. [5]

In the article, the dimensions of data quality are presented as critical components for assessing and ensuring the effectiveness and reliability of data within organizational databases. These dimensions include **accuracy**, which measures the correctness of the data in representing the real-world construct it aims to depict; **completeness**, which evaluates whether all necessary data is present; **consistency**, ensuring that the data does not present contradictions and is coherent across different datasets; and **timeliness**, which assesses the data relevance and currency relative to its intended use.

Other important dimensions mentioned are **reliability and relevance**, which focus on the trustworthiness of the data source and the alignment of the data with user needs and contexts, respectively. The framework also considers data **accessibility and clarity**, emphasizing the importance of data being readily available and understandable to users. Each dimension plays a vital role in the comprehensive assessment of DQ, highlighting the multifaceted nature of data quality issues and the need for a holistic approach to address them.

The article underscores the significance of these dimensions in the evaluation and improvement of data quality. It presents a structured way to categorize existing research within these dimensions and suggests that addressing DQ issues requires attention to detail across these varied dimensions. This approach enables organizations to systematically identify and rectify data quality issues, thereby enhancing the overall integrity and value of their data assets.

3.2.6. Sidi et al. [6]

The article *Data Quality: A Survey of Data Quality Dimensions* [6] is a comprehensive literature review that stands out for its in-depth examination and synthesis of DQ dimensions. It summarizes the most important quality dimensions found in the literature

from 1985 to 2009, combining them into one holistic table. This survey identifies forty different data quality dimensions, categorizing them into intrinsic, accessibility, contextual, representational, and additional dimensions. It serves as a pivotal resource for researchers and practitioners, providing a unified framework that encapsulates the complexity and interrelatedness of data quality dimensions. This work is notable for its systematic approach to consolidating a wide range of DQ dimensions into a single, extensive table, offering a thorough overview of the field and its key components.

In order to avoid repetitive mentioning of already described data quality dimensions we have altered the original table from Sidi et al. [6] and modified it so that only novel dimensions are listed. Additionally, we removed multiple definitions of one and the same data quality dimension for readability purposes.

Dimension	**Definition**
Currency	Currency is the degree to which a datum is up-to-date. A datum value is up-to-date if it is correct is spite of possible discrepancies caused by time-related changes to the correct value [32].
Duplication	A measure of unwanted duplication existing within or across systems for a particular field, record, or data set [33].
Data specification	A measure of the existence, completeness, quality and documentation of data standards, data models, business rules .meta data and reference data [33].
Presentation Quality	A measure of how information is presented to and collected from does how utilize it. Format and appearance support appropriate use of information [33].
Safety	It is the capability of the function to achieve acceptable levels of risk of harm to people, process, property or the environment [34].
Effectiveness	It is the capability of the function to enable to users to achieve specified goals with accuracy and completeness in a specified context of use [35].
Ease of Use and maintainability	A measure of the degree to which data can be accessed and used and the degree to which data can be updated, maintained, and managed [33].
Usability	The extent to which information is clear and easily used [36].
Amount of data	The extent to which the quantity or volume of available data is appropriate [37].
Freshness	Freshness represents a family of quality factors which each one representing some freshness aspect and having on its metrics [38].
Learn ability	It means the capability of the function to enable to user to learn it [34].
Data Decay	A measure of the rate of negative change to data [33].

Consistency and Synchronization	A measure of the equivalence of information used in various data stores, applications, and systems, and the processes for making data equivalent [33].
Data integrity fundamentals	A measure of the existence, validity, structure, content, and other basic characteristics of the data [33].
Navigation	Extent to which data are easily found and linked to [36].
Useful	Extent to which information is applicable and helpful for the task at hand [37].
Efficiency	Extent to which data are able to quickly meet the information needs for the task at hand [37].
Availability	Extent to which information is physically accessible [36].
Data Coverage	A measure of the availability and comprehensiveness of data compared to the total data universe or population of interest [33].
Transactability	A measure of the degree to which data will produce the desired business transaction or outcome [33].
Timeliness and Availability	A measure of the degree to which data are current and available for use as specified and in the time frame in which they are expected [33].

Table 3.6.: Data quality dimensions according to Sidi et al. [6]

3.2.7. Conclusion of Data Quality Literature

This section focuses on the most commonly mentioned data quality dimensions identified in various studies and frameworks discussed in the literature review. These dimensions are foundational to understanding and improving data quality across different contexts and applications. Here, we delve deeper into each of these dimensions, providing definitions, significance, and the implications of their effective management. It is notable that, in one form or another, each of the following four core dimensions is present in every study that deals with data quality.

Completeness

- **Definition:** Completeness refers to the extent to which all required data is present in a dataset. It evaluates whether any data is missing and whether the data present covers all necessary aspects for the task at hand.

- **Significance:** The completeness of data is crucial for ensuring that decisions and analyses are based on the full picture. Incomplete data can lead to erroneous conclusions and actions, which may have significant adverse effects on project outcomes and business decisions.

- **Implications:** To manage completeness, organizations must implement checks and balances that regularly assess the presence of necessary data elements. Techniques such as data profiling and auditing are useful to detect missing data [39].

Accuracy

- **Definition:** Accuracy measures the extent to which data correctly describes the real-world objects or events it is supposed to represent.
- **Significance:** High accuracy ensures that data reflects reality as closely as possible, which is critical for operational integrity and reliability. Accuracy impacts the trust users place in the data and, by extension, the systems that process and present this data.
- **Implications:** Enhancing accuracy involves the validation of data input processes, continual data cleaning, and reconciliation against known reliable sources to correct inaccuracies as they are identified.

Consistency

- **Definition:** Consistency refers to the uniformity and harmony in data across different datasets. It ensures that data does not present contradictions and remains coherent across multiple data stores.
- **Significance:** Consistent data helps maintain the reliability and credibility of data systems, especially in environments where data integration from various sources is common. It is vital for analytical processes where data from different sources are combined to produce insights.
- **Implications:** Maintaining consistency requires robust data governance policies, standardization of data formats, and regular audits to check for discrepancies. Conflict resolution mechanisms are essential when inconsistencies arise.

Timeliness

- **Definition:** Timeliness refers to the relevance and currency of data at the time of its use. It measures whether the data is up-to-date and available when needed.
- **Significance:** Timely data is critical for decision-making processes that rely on the latest information. The value of data often diminishes over time, so timeliness ensures that decisions are made based on the most current and relevant data available.
- **Implications:** To ensure timeliness, organizations need to implement processes that guarantee quick data updates and efficient data delivery mechanisms. This often involves automated systems for data capture and integration.

The dimensions of *Completeness*, *Accuracy*, *Consistency*, and *Timeliness* form the core pillars of data quality management. Each dimension is integral to ensuring that data serves its intended purpose effectively and reliably. By focusing on these key areas, organizations can significantly enhance the quality of their data, thereby improving their operational effectiveness and decision-making capabilities. There are several additional data quality dimensions worth mentioning that are not consistently represented throughout publications covered in the literature review.

- **Validity**: The data is valid if it conforms to its defined syntax specification such as format, type and range.
- **Believability** and **Reputation**: Reflect the perceived truthfulness and credibility of the data and its sources.
- **Accessibility** and **Security**: Focus on the ease of data retrieval and ensuring appropriate restrictions on data access.
- **Interpretability** and **Understandability**: Deal with the clarity of data presentation and ease of comprehension by users.
- **Relevancy**: Pertains to the usefulness of data for the task at hand.

Through rigorous management and continuous improvement of these DQ dimensions, organizations can better leverage their data assets to achieve strategic goals and maintain a competitive edge.

3.3. Data Quality specific to Process Mining

3.3.1. Bose et al. [7]

The article discusses the importance of addressing data quality issues in process mining to improve results. It identifies four categories of process characteristics issues and 27 classes of event log quality issues, emphasizing the need for systematic logging, repair techniques, and analysis techniques to deal with these issues. Bose et al. [7] explored the field of event DQ issues in a multi-layered fashion. The first realisation that they made is that there are two high level categories of the subject, i.e.:

1. **Process characteristics:** Here challenges arise from the very nature of processes as there might be fine-granular activities, process variability, process changes (concept drift) and high volume processes.
2. **Quality of the event log:** This category is concerned with problems that emerge from the quality of logging.

We will start by elaborating on challenges that arise from process characteristics.

- *Event Granularity:* Processes that contain a huge number of different activities produce event logs that are fine-granular. If the information system allows any arbitrary sequence of activities the resulting process model gained by discovery will be spaghetti-like and hard to understand.

- *Case Heterogeneity:* Modern-day processes management tools come in many flavours, as such flexible frameworks have arisen and allow process models to change according to company needs. Consequently the resulting event logs cover a high variety of one and the same process, but performed in many different ways. The large number distinct traces once again lead to spaghetti-like process models that are difficult to grasp.

- *Concept Drifts:* Changes to the process model manifest with latency in event logs. The authors differentiate, based upon the duration for which the change is active, between *momentary* and *evolutionary* changes.

- *High Volume Processes:* In times of Big Data every little aspect of our life is being recorded, leading to massive amounts of data. Unfortunately, present-day process mining tools are unable to cope with this amount of data.

On the other side there are quality issues that originate from the way of how the logging was performed.

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

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

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

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

	case	**event**	**belongs to**	**c attribute**	**position**	**activity name**	**timestamp**	**resource**	**e attribute**
missing data	In reality a case has been executed but it has not been recorded in the log	Events are missing within the trace although they occurred in reality.	Association between events and cases is lost (correlation problem)	Case attribute was not recorded.	Ordering of events in the trace is lost.	Activity names of events are missing.	Timestamps of events are missing.	Resources that executed an activity have not been recorded.	Event attribute was not recorded.
incorrect data	Some cases in the log belong to a different process.	Events that were not actually executed for some cases are logged	Association between events and cases are logged incorrectly.	Values corresponding to case attributes are logged incorrectly.	Order is mixed up.	Wrong activity names are recorded.	Incorrect timestamps.	Incorrect resource assigned to event.	Attributes of events are recorded incorrectly.
imprecise data			Difficult to correlate events to specific cases (too coarse).	Provided value is too coarse, e.g., city but no address.	For example concurrent events may have become been totally ordered.	Activity names are too coarse.	Days rather than minutes or seconds.Hence, precise order cannot be derived.	Just role or department is recorded.	Provided value is too coarse.
irrelevant data	Irrelevant cases are included and cannot be removed easily.	Events may be irrelevant and difficult to remove							

Table 3.7.: Manifestation of the different classes of problems across the various entities of an event log according to [7]

Table 3.7 outlines the manifestation of different classes of problems across various entities of an event log, identifying specific issues like missing, incorrect, imprecise, and irrelevant data for each entity. The table serves as a comprehensive guide to understanding and addressing the complex data quality challenges in process mining.

3.3.2. Process Mining Manifesto [8]

The issue of data quality is as old as process mining itself, we can find this facet in the guiding principles of the process mining manifesto [8]. The very first principle of the manifesto *Event Data Should Be Treated as First-Class Citizens*, highlights the inherent importance of event log quality. Van der Aalst et al. defined five event log maturity levels, ranging from excellent quality ($\star\star\star\star\star$) to poor quality ($\star$). They argue that process mining techniques can be applied to logs of level $\star\star\star\star\star$, $\star\star\star\star$ and $\star\star\star$. Process mining results of event logs with this log maturity level will be trustworthy, reliable and safe. On the other side, event logs of level $\star\star$ or $\star$ can be analysed by means of process mining, but the results are most likely not trustworthy and problematic to interpret.

Level	Characterization
⋆⋆⋆⋆⋆	Highest level: the event log is of excellent quality (i.e., trustworthy and complete) and events are well-defined. Events are recorded in an automatic, systematic, reliable, and safe manner. Privacy and security considerations are addressed adequately. Moreover, the events recorded (and all of their attributes) have clear semantics. This implies the existence of one or more ontologies. Events and their attributes point to this ontology. *Example*: semantically annotated logs of BPM systems.
⋆⋆⋆⋆	Events are recorded automatically and in a systematic and reliable manner, i.e., logs are trustworthy and complete. Unlike the systems operating at level ⋆⋆⋆, notions such as process instance (case) and activity are supported in an explicit manner. *Example*: the events logs of traditional BPM/workflow systems.
⋆⋆⋆	Events are recorded automatically, but no systematic approach is followed to record events. However, unlike logs at level ⋆⋆, there is some level of guarantee that the events recorded match reality (i.e., the event log is trustworthy but not necessarily complete). Consider, for example, the events recorded by an ERP system. Although events need to be extracted from a variety of tables, the information can be assumed to be correct (e.g., it is safe to assume that a payment recorded by the ERP actually exists and vice versa). *Examples*: tables in ERP systems, events logs of CRM systems, transaction logs of messaging systems, event logs of high-tech systems, etc.
⋆⋆	Events are recorded automatically, i.e., as a by-product of some information system. Coverage varies, i.e., no systematic approach is followed to decide which events are recorded. Moreover, it is possible to bypass the information system. Hence, events may be missing or not recorded properly. *Examples*: event logs of document and product management systems, error logs of embedded systems, worksheets of service engineers, etc.
⋆	Lowest level: event logs are of poor quality. Recorded events may not correspond to reality and events may be missing. Event logs for which events are recorded by hand typically have such characteristics. *Examples*: trails left in paper documents routed through the organization ("yellow notes"), paper-based medical records, etc.

Table 3.8.: Maturity levels for event logs according to Van Der Aalst [8]

Table 3.8 gives an overview of the different event log maturity levels and briefly summarizes what aspects the event log has to fulfil in order to be considered of high or low quality according to [8].

Additionally we can find in **Guiding Principle 1 (GP1)** of the manifesto one text passage that exclusively focuses on quality criteria of event data, that is of special interest to us:

> There are several criteria to judge the quality of event data. Events should be *trustworthy*, i.e., it should be safe to assume that the recorded events actually happened and that the attributes of events are correct. Event logs should be *complete*, i.e., given a particular scope, no events may be missing. Any recorded event should have well-defined *semantics*. Moreover, the event data should be *safe* in the sense that privacy and security concerns are addressed when recording the events. For example, actors should be aware of the kind of events being recorded and the way they are used.[4]

3.3.3. Van Der Aalst 2015 [9]

The guidelines for logging, as proposed in "Extracting event data from databases to unleash process mining" by Van Der Aalst [9] aim at enhancing the quality of event data for process mining. The article focuses on the input side of process mining, i.e., the event log. It encompasses twelve guidelines that are stated as recommendations for the analyst to keep in mind, when working with event data. The guidelines are formulated as follows:

GL1 *Reference and attribute names should have clear semantics, i.e., they should have the same meaning for all people involved in creating and analyzing event data.* Different stakeholders should interpret event data in the same way.

GL2 *There should be a structured and managed collection of reference and attribute names.* A new reference or attribute name can only be added after there is consensus on its value and meaning. Also consider adding domain or organization specific extensions.

GL3 *References should be stable (e.g., identifiers should not be reused or rely on the context).* For example, references should not be time, region, or language dependent. Some systems create different logs depending on the language settings. This is unnecessarily complicating analysis.

GL4 *Attribute values should be as precise as possible. If the value does not have the desired precision, this should be indicated explicitly (e.g., through a qualifier).* For example, if for some events only the date is known but not the exact timestamp, then this should be stated explicitly.

GL5 *Uncertainty with respect to the occurrence of the event or its references or attributes should be captured through appropriate qualifiers.* For example, due to communication errors, some values may be less reliable than usual. Note that uncertainty is different from imprecision.

[4] [8]: page 179

GL6 *Events should be at least partially ordered. The ordering of events may be stored explicitly (e.g., using a list) or implicitly through an attribute denoting the event's timestamp.* If the recording of times- tamps is unreliable or imprecise, there may still be ways to order events based on observed causalities (e.g., usage of data).

GL7 *If possible, also store transactional information about the event (start, complete, abort, schedule, assign, suspend, resume, withdraw, etc.).* Having start and complete events allows for the computation of activity durations. It is recommended to store activity references to be able to relate events belonging to the same activity instance. Without activity references it may not always be clear which events belong together, which start event corresponds to which complete event.

GL8 *Perform regularly automated consistency and correctness checks to ensure the syntactical correctness of the event log.* Check for missing references or attributes, and reference/attribute names not agreed upon. Event quality assurance is a continuous process (to avoid degradation of log quality over time).

GL9 *Ensure comparability of event logs over time and different groups of cases or process variants.* The logging itself should not change over time (without being reported). For comparative process mining, it is vital that the same logging principles are used. If for some groups of cases, some events are not recorded even though they occur, then this may suggest differences that do not actually exist.

GL10 *Do not aggregate events in the event log used as input for the analysis process.* Aggregation should be done during analysis and not before (since it cannot be undone). Event data should be as "raw" as possible.

GL11 *Do not remove events and ensure provenance. Reproducibility is key for process mining.* For example, do not remove a student from the database after he dropped out since this may lead to misleading analysis results. Mark objects as not relevant (a so-called "soft delete") rather than deleting them: concerts are not deleted - they are canceled, employees are not deleted - they are fired, etc.

GL12 *Ensure privacy without losing meaningful correlations.* Sensitive or private data should be removed as early as possible (i.e., before analysis). However, if possible, one should avoid removing correlations. For example, it is often not useful to know the name of a student, but it may be important to still be able to use his high school marks and know what other courses he failed. Hashing can be a powerful tool in the trade-off between privacy and analysis.

In contrast to the previously described papers, which aim at conceptualizing (event) data quality aspects and issues, [9] focuses on the *how* of improving event data quality while, or respectively after, the creation of the data. When we take a closer look at both approaches, we can find a lot of overlap. For example lets consider *"**GL1** Reference and attribute names should have clear semantics, i.e., they should have the same meaning for all people involved in creating and analyzing event data."* this guideline corresponds with the quality dimension *Understandability* that we know from [37, 1].

3.4. Differences and Overlaps in Data Quality Literature

In the following, we will make a comparison between general data quality as discussed in Section 3.2 and data quality specific to process mining outlined in Section 3.3 of this study. This comparison aims to identify both overlaps as well as gaps between these two domains, facilitating a better understanding of how general DQ principles apply to the specialized field of process mining. By examining the congruence and discrepancies between these sections, we can highlight the unique challenges of ensuring data quality in event logs and derive insights that may be less apparent or emphasized in the broader context of data quality. This comparative analysis will also help in refining our chosen DQ dimensions for assessment by aligning them more closely with the specific needs of process mining, ultimately leading to more accurate and effective process analysis outcomes.

To visually illustrate the comparison between general data quality dimensions and those specific to process mining, we will introduce Table 3.9. This comprehensive table will be organized such that the y-axis lists all the DQ dimensions identified in our review, while the x-axis will indicate which papers or frameworks mention each of these dimensions. This format will allow readers to quickly ascertain the frequency and context in which each data quality dimension is discussed across different sources. It will also highlight dimensions that are universally recognized versus those that are more specific to certain contexts or methodologies.

Key differences between the two domains:

- **Contextual Relevance**: While general DQ dimensions address broader data integrity issues, the dimensions in process mining are specifically tailored to deal with the nuances and specific requirements of process event data.
- **Detail Orientation**: Process mining dimensions tend to be more granular, dealing with the intricacies of event relationships and log specifics, which are critical for the accurate representation of process flows.
- **Practical Implications**: In process mining, data quality directly impacts the ability to derive meaningful insights from process analytics, making the specific dimensions more operationally critical in that context.

By comparing these dimensions, it becomes evident that while there is overlap in the fundamental aims of ensuring DQ (such as accuracy and completeness), the specific concerns and focus points vary significantly. Process mining requires a more detailed approach to data quality that considers the dynamic and interconnected nature of process data, which differs from the more static and general data quality considerations in other domains.

	General DQ Literature						PM specific DQ Literature		
DQ Dimension	**[1]**	**[2]**	**[3]**	**[4]**	**[5]**	**[6]**	**[7]**	**[8]**	**[9]**
Completeness	[x]	[x]	[x]	[x]	[x]	[x]	Missing Data	GP1 (complete)	GL8, GL11
Accuracy (Correctness)	Free-of-Error	[x]	Believability	[x]	[x]	[x]	Incorrect Data	GP1 (trustworthy)	GL4
Consistency	Consistent Representation	[x]	-	[x] (Coherence)	[x]	[x]	-	-	GL8
Timeliness	[x]	[x]	[x]	Time Factors	[x]	[x]	Imprecise Data	-	GL8
Validity	Interpretability	[x]	Interpretability	[x]	data validation	Data integrity fundamentals	-	-	-
Accessibility	[x]	Confidence	-	[x] (and Clarity)	-	[x]	-	GP1 (safe)	-
Security	[x]	Confidence	[x]	(Privacy and) Security	-	[x]	-	GP1 (safe)	-
Interpretability (Understandability)	[x]	Usability	[x]	Clarity	-	[x]	Imprecise Data	GP1 (semantics)	GL1, GL4
Relevancy	[x]	Usability	[x]	Relevance	[x]	[x]	Irrelevant Data	GP2	-

[x] DQ Dimension is present in the study

- DQ Dimension is not considered in paper

Table 3.9.: Overview of Data Quality Dimensions and their Occurrences in Literature

The table presents a summary of how various DQ dimensions are treated across a selection of general data quality literature compared to process mining specific literature.

There is a high degree of consensus within the general data quality community regarding the importance of various DQ dimensions; most are acknowledged in some capacity across the literature. Dimensions such as Completeness, Accuracy, Timeliness, and Validity are marked by almost all the general DQ literature sources, signifying a broad agreement on their significance.

However, there are notable gaps when comparing these dimensions to the process mining specific literature. Here are four examples that illustrate these disparities:

1. **Completeness**: This dimension is universally mentioned across all the general DQ literature sources, signifying its foundational importance. It is also recognized in the process mining specific literature, particularly with regard to missing data, highlighting its criticality in both domains.

2. **Validity**: This dimension, while frequently discussed in the general DQ literature, does not appear in the process mining specific literature. It suggests that while the concept of data conforming to predefined formats and ranges is relevant in a broader sense, it may not be as explicitly addressed within the process mining domain.

3. **Consistency**: The dimension of Consistent Representation appears in the general literature and correlates to *syntactical correctness* in the process mining literature. However, it's not explicitly labeled as 'consistency' in the process mining context, which may indicate differences in terminologies or focal points between the two fields.

4. **Accuracy**: Recognized across both general data quality and process mining literature, accuracy is sometimes referred to as *Free-of-Error* or *Believability* or even *Correctness* in broader contexts. In process mining, this dimension is addressed through terms like *Incorrect Data*, emphasizing the importance of error-free event logs, and *Trustworthiness*, highlighting the reliability of data in reflecting actual real-life occurrences. Regardless of the terminology, the imperative is clear: data must accurately represent the observed reality to be considered of high quality.

The comparison suggests that while there is a shared understanding of key DQ dimensions between general and process mining specific literature, there are gaps that may arise from the specialized nature of process mining. Some dimensions such as Completeness are universally recognized due to their fundamental impact on data quality, whereas others like Validity might be implicit or underrepresented in process mining discussions, potentially calling for a deeper exploration in future process mining specific DQ research.

4. Data Quality Assessment Framework

4.1. Data Quality Dimensions applicable for Assessment

In this section, we aim to provide a practical guide on which data quality dimensions discussed in the earlier sections are directly applicable for assessment in the context of process mining. Using the detailed analysis of data quality issues outlined by Bose et al. [7] in Subsection 3.3.1, we will draw parallels to the more general data quality dimensions specified in Section 3.2. This approach allows us to bridge the gap between theoretical data quality dimensions and their practical application in evaluating the quality of event logs in process mining scenarios.

4.1.1. Mapping Event Data Quality Issues to General Quality Dimensions

This mapping not only clarifies which general quality dimensions are applicable for assessing event log quality but also assists process mining practitioners in systematically identifying and rectifying specific data quality issues within their process mining endeavors. For example, by understanding how "missing data" affects "completeness," analysts can implement targeted methods to detect and mitigate data gaps, thereby improving the integrity and reliability of their process models.

1. **Missing Data**
   - *Related Data Quality Dimension:* **Completeness**
   - *Rationale:* Missing data directly affects the completeness of the dataset. In process mining, missing events can lead to incomplete traces, which hampers the ability to accurately reconstruct and analyze processes. This aligns with the general dimension of completeness, which assesses whether all necessary data is present and sufficiently detailed to support the intended tasks.
2. **Incorrect Data**
   - *Related Data Quality Dimension:* **Accuracy**
   - *Rationale:* Incorrect data pertains to errors in the data that do not truthfully represent the underlying processes. This issue correlates with the accuracy dimension from general data quality management, where the focus is on ensuring that data correctly reflects the real-world situations it is supposed to represent.
3. **Imprecise Data**

- *Related Data Quality Dimensions:* **Interpretability (Precision)**
- *Rationale:* Imprecise data in event logs, such as coarse timestamps or generalized event descriptions, affects the data's utility for detailed and sequential process analysis. While precision itself is not detailed as a separate dimension in our general data quality literature, it is a component of accuracy and interpretability, impacting the exactitude and usefulness of the data in a given temporal context.

4. **Irrelevant Data**
   - *Related Data Quality Dimension:* **Relevancy**
   - *Rationale:* Irrelevant data includes any information that does not contribute meaningfully to process understanding or analysis. The relevancy dimension assesses whether the data at hand is relevant and beneficial for the tasks being executed, ensuring focus on data that facilitates insights and decisions.

Recognizing the alignment of specific event data quality issues with broader quality dimensions enables organizations to apply a structured approach to data quality improvement across various data-driven applications. This structured alignment fosters a unified strategy in data quality management, bridging theoretical principles with practical application to derive more reliable and actionable insights from data analyses.

In our comprehensive study of data quality within the context of process mining, we recognize the paramount importance of closely examining the event log issues discussed by Bose et al.[7]. However, our analysis has led to the identification of an additional, critical data quality issue that appears to have been overlooked in their framework. We introduce *Valid Data* as a new, fifth category of data quality issues specific to the process mining domain. This category specifically addresses the necessity for data to adhere to established rules and constraints, ensuring that all data elements are both appropriate and logical within their specific context. The introduction of *Valid Data* aims to encapsulate the concerns related to the validity of data, emphasizing its critical role in maintaining the integrity and utility of process analyses.

5. **Valid Data**
   - *Related Data Quality Dimension:* **Validity**
   - *Rationale:* This newly identified issue pertains to the presence of data that strictly conforms to predefined formats, types, and other criteria specified within a dataset's schema or the business rules governing the data. The *Valid Data* issue is closely aligned with the quality dimension of validity, which assesses whether data elements are valid based on whether they adhere to

specific rules and standards, such as correct formats, allowable values, and logical constraints.

By formally recognizing *Valid Data* as a distinct quality issue and aligning it with the dimension of validity, our study enhances the framework provided by Bose et al.[7], offering a more robust tool-set for assessing and improving data quality in process mining environments. This addition ensures that the data not only exists and is accurate but also is correct in form and context, making it truly reliable for making informed decisions and generating insightful analyses.

4.2. Data Quality Dimension - Descriptive Framework

Now that we have an comprehensive understanding of data quality aspects in general and event log quality problems in particular, we will draw our attention towards a descriptive framework for data quality dimensions and how to assess them. In Section 3.2 *Data Quality* we have already presented several tables that encompass data quality dimensions with additional parameters. It was our intention to introduce data quality dimensions by name and give an brief explanation about each dimension. While some references (i.e., [1, 6, 5]) only entail the information about the name of the dimension and its definition, others include further aspects such as metadata source and methods of generation metadata (i.e., [3]) or hyperdimension and factors to consider (i.e., [4]) that we have pointed out in Section 3.2. Each paper approaches the granularity of data quality dimensions differently, offering unique perspectives and methodologies. We place special emphasis on the paper by the Data Management Association (DAMA) UK Working Group [2], which rigorously details six primary data quality dimensions in structured tables, providing a comprehensive breakdown of each dimension's attributes.

4.2.1. Extracting Table Essence for Data Quality Assessment Framework

The DAMA UK Working Group's approach [2] to detailing data quality dimensions serves as a benchmark for creating a robust framework for our own data quality assessments. Each dimension is encapsulated in a table that typically includes several key attributes as seen in Table 4.1.

Title	The name of the data quality dimension
Definition	A concise description of what the dimension entails
Reference	Supplementary details such as business rules, reference datasets, or benchmarks provide essential criteria against which the given data can be evaluated
Measure	How the quality dimension can be quantitatively or qualitatively assessed
Scope	The applicability of the dimension across different data types or contexts
Unit of Measure	The specific units or scales used for measurement
Related Dimensions	Other quality dimensions that are closely related or dependent
Optionality	Indicates whether the dimension is mandatory or optional depending on the context
Example(s)	Practical examples to illustrate the dimension in action
Pseudo Code	A brief code snippet that demonstrates how to calculate or assess the dimension

Table 4.1.: Essence of the Descriptive Data Quality Tables from [2]

4.2.2. Data Quality Assessment Framework

Before we delve into examining each applicable data quality dimension as structured by the DAMA UK framework, it is crucial to first direct our focus towards the hierarchical structure of the measurements themselves. This approach underscores that a data quality dimension is evaluated based on one or several elements, each of which is further assessed through one or multiple indicators. For instance, consider the dimension of completeness: this can be evaluated by an element termed missing values. The element of missing values, in turn, is assessed using various indicators, such as missing timestamps, missing attributes, or missing resources. This multilevel structure enables a more granular and comprehensive analysis, ensuring that each aspect of data quality is thoroughly addressed and quantified, enhancing the precision and utility of the overall assessment process.

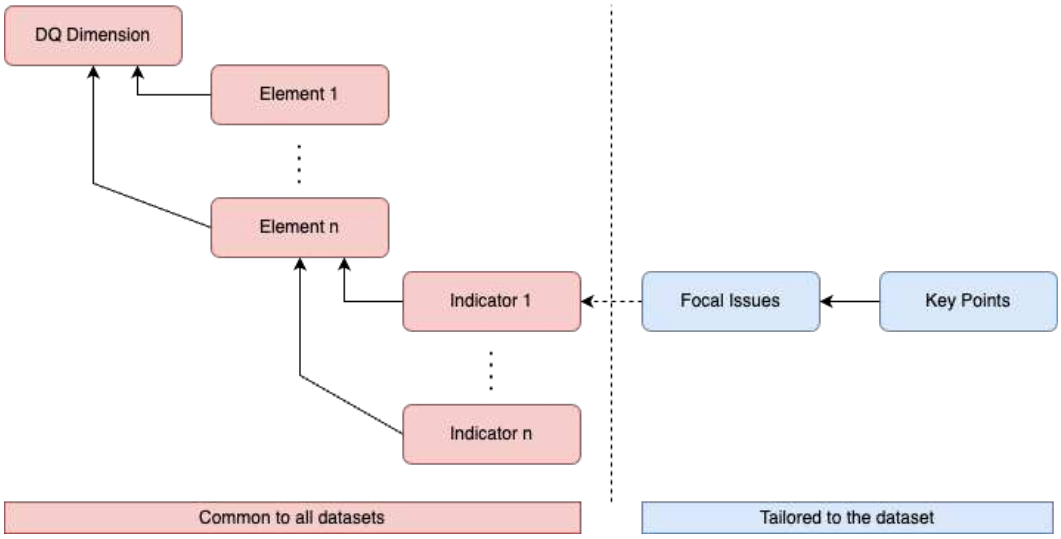

Figure 4.1.: Adapted Data Quality Assessment Framework

The taxonomy of the multilevel approach to data quality assessment — where dimensions are evaluated based on elements, which are in turn assessed through various indicators, was adopted from the Data Quality Assessment Framework (DQAF)[1] developed by the International Monetary Fund (IMF). This framework provides a structured and systematic methodology for assessing the quality of data, enhancing the rigor and comprehensiveness of evaluations. In this thesis, our focus is specifically limited to the left part of the figure presented in the DQAF, as presented in Fig. 4.1. This portion of the framework concentrates on assessments that are generic, meaning they are applicable across all datasets (i.e. event logs) regardless of their specific characteristics or contexts. By incorporating the DQAF taxonomy into our analysis, we align our assessment structure with internationally recognized standards, ensuring that our data quality evaluations are both thorough and globally relevant.

4.3. Integrating Quality Dimensions into the Framework

At this juncture in our discussion, we will recapitulate and consolidate the insights gathered thus far. Our exploration has spanned an extensive range of quality dimensions highlighted in both general and process mining-specific literature. Having identified these dimensions as critical in both contexts, our focus now shifts towards those that are not only prevalent in academic discussions but also practically applicable for assessment in process mining environments.

We will utilize the structured approach of the DAMA UK Working Group (Tab. 4.1), which effectively categorizes data quality dimensions into detailed tables. Each dimension

[1] `https://unstats.un.org/unsd/dnss/docs-nqaf/IMF-dqrs_factsheet.pdf`

we have found applicable for assessment, as mapped against the data quality issues outlined by Bose et al. [7] (Sect. 4.1.1), will be methodically integrated into this table structure. This exercise will not only place these dimensions within a conceptual framework but also initiate the development of corresponding elements and indicators for each dimension's assessment. Further guidance are drawn from the DQAF as depicted in Fig. 4.1, which serves as a foundational reference in our methodology.

4.3.1. Completeness

Title	Completeness
Definition	All values for a certain variable are recorded. This includes traces, events and attributes of an event log. If values of these categories are missing, the event log is considered not 100% complete.
Reference	A meta data file A high level description of the event log
Measure	1. Percentage of not recorded traces versus all recorded traces 2. Percentage of missing events versus all documented events 3. Percentage of not associated events and traces versus all event - traces pairs that are associated 4. Percentage of missing trace attributes versus all recorded trace attributes 5. Percentage of not ordered events in a trace versus all ordered events in all traces 6. Percentage of missing activity names of events versus all recorded activity names of events 7. Percentage of missing timestamps versus all documented timestamps 8. Percentage of missing resources to activities versus all recorded resources 9. Percentage of not recorded event attributes versus all recorded event attributes 10. Verify that each trace with a start event also possesses a corresponding end event
Scope	Event logs for process mining

Unit of Measure	Percentage
Related Dimensions	-
Optionality	Mandatory quality dimension for assessment
Example(s)	Data fields including activity name, timestamp or resource that are defined and expected to be documented, are recorded as empty in the event log.
Pseudo Code	Algorithm 1 outlines the pseudo code for the completeness assessment

```
<?xml version="1.0" encoding="UTF-8" ?>
  <trace>
    <string key="concept:name" value="Guest9723"/>
    <event>
      <string key="org:resource" value=""/>
      <int key="Table" value="5"/>
      <string key="concept:name" value="null"/>
      <string key="lifecycle:transition" value="
    complete"/>
      <date key="time:timestamp" value="na"/>
    </event>
  </trace>
```

Listing 4.1: XES example of completeness issues

Table 4.2.: Assessment Framework for Completeness

Data: an event log
Result: percentage of completeness
initialize emptyCount to 0;
initialize totalCount to 0;
for *each event in dataset* **do**
  increment totalCount;
  if *event.activityName is in ["", "null", "empty"] or*
   event.timestamp is in ["", "null", "empty"] or
   event.resource is in ["", "null", "empty"] **then**
    increment emptyCount
  end
end
completeness = ((totalCount - emptyCount) / totalCount) * 100

Algorithm 1: Pseudo Code for the Assessment of Completeness

4.3.2. Accuracy

Title	Accuracy
Definition	All recorded values are correct and represent real-world occurrences. Traces belong to the analysed process at hand, containing no fictitious events that did not occur and the timestamps reflect the actual times at which the events took place. If discrepancies in the correctness of the event log arise, it is considered not 100% accurate.
Reference	A reference event log Contextual information about the process
Measure	1. Number of traces that belong to a different process 2. Number of fictitious events that were never actually executed 3. Number of wrong associations between traces and events 4. Number of false trace attributes 5. Amount of events in a mixed up order 6. Number of incorrect activity names 7. Number of erroneous timestamps 8. Number of invalid resource to event association 9. Number of falsely recorded event attributes
Scope	Event logs for process mining
Unit of Measure	Sum of occurrence
Related Dimensions	-
Optionality	Mandatory quality dimension for assessment

Example(s)	Resources that are incorrect or nonexistent are documented. Timestamps that are dated in the future are recorded. Event attributes that should be trace-constant change and paint an incorrect picture.
Pseudo Code	Algorithm 1 outlines the pseudo code for the completeness assessment

```
<?xml version="1.0" encoding="UTF-8" ?>
  <trace>
    <string key="concept:name" value="Guest9723"/>
    <event>
      <string key="org:resource" value="Waiter05"/>
      <int key="Table" value="5"/>
      <string key="concept:name" value="Take order"/>
      <date key="time:timestamp" value ="0025-10-28
    T17:33:00.000+01:00"/>
    </event>
      <event>
      <string key="org:resource" value="Waiter05"/>
      <int key="Table" value="071239123"/>
      <string key="concept:name" value="Take order"/>
      <date key="time:timestamp" value="0025-10-28
    T17:33:00.000+01:00"/>
    </event>
  </trace>
```

Listing 4.2: XES example of accuracy issues

Table 4.3.: Assessment Framework for Accuracy

Data: an event log
Result: number of false timestamps
initialize futureTimestampCount to 0;
getCurrentDate();
for *each event in dataset* **do**
  if *event.timestamp* > *getCurrentDate()* **then**
    increment futureTimestampCount
  end
end
return futureTimestampCount

Algorithm 2: Pseudo Code for the Assessment of Accuracy

4.3.3. Validity

The data quality dimensions of *Accuracy* and *Validity* are closely related and often intertwine, making it sometimes challenging to distinguish between the two. Both dimensions focus on the correctness and appropriateness of data within a dataset. Accuracy pertains to the extent to which data correctly represents the true values or states of the real-world entities it is supposed to depict. This involves ensuring that the data is free from errors and precisely reflects what it is intended to capture. On the other hand, validity refers to the degree to which the data conforms to the specified rules, formats, or constraints set out by the data model or business requirements. It assesses whether the data is logically correct and appropriate within its context.

In practical scenarios, these two dimensions can blur. For example, if a data entry of a date is supposed to represent a past event but is erroneously recorded as a future date, it would be considered both inaccurate (because it is wrong) and invalid (because it does not meet the logical expectation of a past date). Such overlap can make it difficult to determine whether a particular issue should be classified under inaccuracies or invalidities, as both dimensions address elements of data correctness and adherence to expected norms. This close relationship necessitates a careful and nuanced approach to data quality assessment, where the specific criteria for accuracy and validity may need to be tailored to the unique characteristics and requirements of the dataset being analyzed.

Title	Validity
Definition	Data are valid if it conforms to the syntax (format, type, range) of its definition[2].
Reference	A meta data file The documentation rules for logging A list of allowed attribute manifestations
Measure	1. Check all values of the event log against the standards specified by XES 2. Check if predefined data types actually hold true 3. Check if the syntax of data elements does not change over time
Scope	Event logs for process mining
Unit of Measure	Score
Related Dimensions	Accuracy
Optionality	Mandatory quality dimension for assessment

Example(s)	An event attribute e.g. *table*[2] that should be an integer data type, contains a string. The date data format is recorded with the syntax "0025-10-22T17:33:00.000+01:00" for the first 20 occurrences and then changes to the format "25/10/2022 17:33:00".
Pseudo Code	Algorithm 3 outlines the pseudo code for the validity assessment

```
<?xml version="1.0" encoding="UTF-8" ?>
  <trace>
    <string key="concept:name" value="Guest9723"/>
    <event>
      <string key="org:resource" value="Tom"/>
      <int key="Table" value="table5"/>
      <string key="concept:name" value="Take order"/>
      <date key="time:timestamp" value ="0025-10-28
    T17:33:00.000+01:00"/>
    </event>
      <event>
      <string key="org:resource" value="Tom"/>
      <int key="Table" value="table5"/>
      <string key="concept:name" value="Take order"/>
      <date key="time:timestamp" value="25/10/2022 17
    :33:00"/>
    </event>
  </trace>
```

Listing 4.3: XES example of validity issues

Table 4.4.: Assessment Framework for Validity

[2]From our restaurant example that we started elaborating in Section 2.1

```
Data: an event log
Result: number of data type mismatches
initialize typeMismatchCount to 0;
define expectedDataTypes as
 "activityName": "string",
 "timestamp": "datetime",
 "resource": "string",
 "duration": "integer",
 "cost": "float"

 for each event in dataset do
 |  for each field, value in event do
 |  |  expectedType = expectedDataTypes[field]
 |  |  if not checkDataType(value, expectedType then
 |  |  |  increment typeMismatchCount
 |  |  end
 |  end
end
return typeMismatchCount
```

Algorithm 3: Pseudo Code for the Assessment of Validity

4.3.4. Interpretability (Precision)

Title	Interpretability (Precision)
Definition	The extent to which data is in appropriate language, symbols, and units, and the definitions are clear [1].
Reference	Contextual information about the process
Measure	1. Number of events that can not be correlated to specific traces as they are documented to coarsely 2. Attributes (trace and event) that are too coarse 3. Concurrent events that have been totally ordered 4. Activity names that have weak semantics (named too coarsely) 5. Timestamps that only capture the day an event occurred, without information of the hour or minute it happened 6. Resources that just represent roles or departments instead of concrete persons or machines
Scope	Event logs for process mining
Unit of Measure	Score
Related Dimensions	-
Optionality	Optional quality dimension for assessment

Example(s)	Timestamps are recorded on a too coarse level, making it hard to put events in a meaningful chronological order. Activity names are gritty, obscuring the clarity and detailed understanding of the process.
Pseudo Code	Algorithm 4 outlines the pseudo code for the interpretability assessment

```
<?xml version="1.0" encoding="UTF-8" ?>
  <trace>
    <string key="concept:name" value="Guest9723"/>
    <event>
      <string key="org:resource" value="Sue"/>
      <int key="Table" value="5"/>
      <string key="concept:name" value="Service
    customer"/>
      <date key="time:timestamp" value ="0025-10-28
    T00:00:00.000+01:00"/>
    </event>
      <event>
      <string key="org:resource" value="Sue"/>
      <int key="Table" value="071239123"/>
      <string key="concept:name" value="Service
    customer"/>
      <date key="time:timestamp" value="0025-10-28
    T00:00:00.000+01:00"/>
    </event>
  </trace>
```

Listing 4.4: XES example of interpretability issues

Table 4.5.: Assessment Framework for Interpretability

Data: an event log
Result: number of imprecise timestamps
initialize insufficientPrecisionCount to 0;
getTimePrecision();
for *each event in dataset* **do**
  if *getTimePrecision(event.timestamp) == "day"* **then**
    increment insufficientPrecisionCount
  end
end
return insufficientPrecisionCount

Algorithm 4: Pseudo Code for the Assessment of Interpretability

4.3.5. Relevancy

Title	Relevancy
Definition	The extent to which data is applicable and helpful for the task at hand [1].
Reference	Contextual information about the process
Measure	1. Number of irrelevant traces within the log 2. Number of irrelevant events, as they are duplicates or just system status information, within the log
Scope	Event logs for process mining
Unit of Measure	Score
Related Dimensions	Accuracy
Optionality	Mandatory quality dimension for assessment
Example(s)	Traces that do not belong to the analysed process are captured in the event log, making it over-fraught with unnecessary data. Information system status information, such as system logs, are recorded in the log causing it to be bloated with useless data.
Pseudo Code	Algorithm 5 outlines the pseudo code for the relevancy assessment

```
<?xml version="1.0" encoding="UTF-8" ?>
  <trace>
    <string key="concept:name" value="INFO 57968 --- [
    restartedMain] o.s.s.restaurant.
    RestaurantApplication"/>
  </trace>
  <trace>
    <string key="concept:name" value="INFO 57968 --- [
    restartedMain] .e.
    DevToolsPropertyDefaultsPostProcessor"/>
  </trace>
        <trace>
    <string key="concept:name" value="INFO 57968 --- [
    restartedMain] .s.d.r.c.
    RepositoryConfigurationDelegate"/>
  </trace>
         <trace>
    <string key="concept:name" value="INFO 57968 --- [
    restartedMain] o.s.b.w.embedded.tomcat.
    TomcatWebServer"/>
  </trace>
```

Listing 4.5: XES example of relevancy issues

Table 4.6.: Assessment Framework for Relevancy

```
Data: an event log
Result: number of irrelevant activity names and resources
initialize irrelevantCount to 0;
define relevantActivityName as ["Take orders", "Bill customer", ...];
define relevantResource as ["Tom", "Sue", "Tony", ...];
for each event in dataset do
    if not event.activityName in validActivityName then
        increment irrelevantCount
    end
    if if not event.resource in validResource then
        increment irrelevantdCount
    end
end
return irrelevantCount
```

Algorithm 5: Pseudo Code for the Assessment of Relevancy

5. Event Data Quality Dashboard

5.1. Overview of the Event Data Quality Dashboard (EDQD)

The Event Data Quality Dashboard (EDQD) represents the culmination of the thesis work, designed to enhance the capability of process mining practitioners by providing immediate insights into the quality of the data they are handling. This software tool is instrumental in process mining endeavours, ensuring that data integrity and quality are upheld before in-depth analysis proceeds.

Key Features of the EDQD:

1. **User-Friendly Interface**: The dashboard is engineered with simplicity in mind. Users can quickly upload an event log file in the XES format, which is a standard format widely used in process mining for logging event data.
2. **Automated Quality Assessment:** Upon uploading their data, the dashboard autonomously executes a series of algorithms — similar to the pseudo code examples previously described. These algorithms assess the data across several crucial dimensions of quality which were found during literature research and synthesized in the conceptual chapter 4 of this thesis: Completeness, Accuracy, Interpretability, Relevancy, and Validity.
3. **Visual Quality Indicators:** The results of the quality assessments are visually represented in the dashboard through an intuitive traffic light color-coding system:
    - **Green:** This color signals that the data within a specific dimension meets all the necessary criteria for quality and integrity, indicating that the data is well-suited for further process mining tasks.
    - **Yellow:** A yellow light denotes that there are some minor issues detected within the data. While these issues are not severe, they may require attention before proceeding to ensure optimal results from process mining analyses.
    - **Red:** A red indicator is a warning that there are significant problems within the data concerning the particular quality dimension. These issues are likely to obstruct meaningful analysis and should be addressed immediately.

Benefits of the EDQD:

- **Immediate Feedback:** The dashboard provides instant feedback on the quality of the data, which is vital for process mining where the accuracy and detail of the input data can significantly influence the outcome of the mining results.

- **Guidance for Data Cleaning:** By identifying specific areas where data quality issues exist, the EDQD helps practitioners prioritize data cleaning efforts, focusing on the most critical issues first.
- **Enhanced Data Trustworthiness:** With an easy-to-understand overview of data quality, practitioners can feel confident in the reliability of their data, leading to more trustworthy and accurate process mining results.

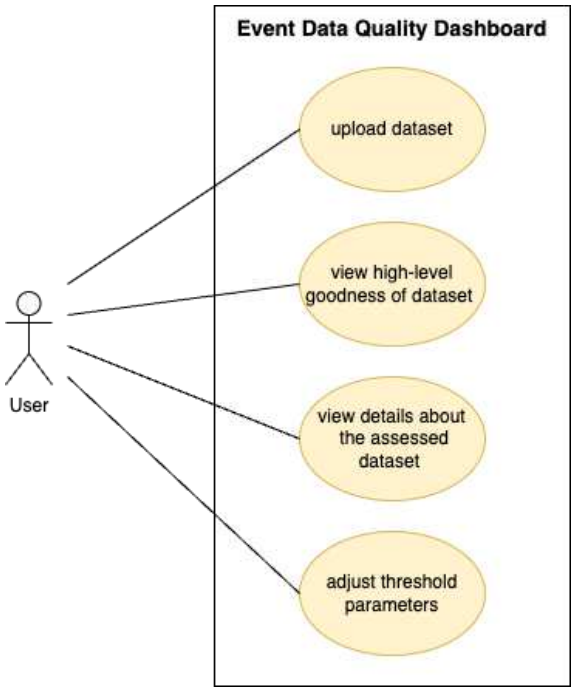

Figure 5.1.: Use-case Diagram of the EDQD

Figure 5.1, the use-case diagram for the Event Data Quality Dashboard (EDQD) illustrates the straightforward workflow and functionality available to the user. The diagram outlines four primary actions:

1. **Upload Dataset:** This is the initial step where the user uploads their event log file in XES format. This function is essential as it serves as the entry point for the data to be assessed by the dashboard.
2. **View high-level Goodness of the Dataset:** After the dataset is uploaded, the user can view a high-level summary of the data quality. This overview provides a quick snapshot of the overall health of the dataset across various quality dimensions using the traffic light color system to indicate the status of the data quality.
3. **View Details about the assessed Dataset:** For users needing more in-depth information, this function allows them to dive into detailed analyses of the data

quality. It breaks down the specifics of the issues found in each quality dimension, offering insights into where problems are located in the log.

4. **Adjust Threshold Parameters:** Allows users to configure various threshold values in the Event Data Quality Dashboard, customizing color-coding limits for score categories and adjusting specific function thresholds for analysis sensitivity.

Together, these use-cases define the core capabilities of the EDQD, designed to assist process mining practitioners in ensuring the quality of their data before proceeding with more complex analyses. The dashboard simplifies the process of data quality assessment, making it accessible even for users with limited technical expertise in data quality metrics.

5.2. Design of the EDQD

Before diving into the implementation of the Event Data Quality Dashboard (EDQD), several key considerations shaped the design and functionality of the tool. This section outlines the foundational thoughts that influenced the development process, providing insights into the overall design strategy, the chosen technical framework, and the user interface.

5.2.1. Design Decisions

At the outset, we had to establish clear design principles to guide the development of the dashboard. A primary consideration was ensuring that the tool was intuitive and user-friendly, catering to both technical and non-technical users. This meant prioritizing simplicity in the user interface, while still providing comprehensive data quality assessments.

One of the key design decisions made during the development of the Event Data Quality Dashboard (EDQD) was the preference for genericity over specialization. This principle guided the creation of the dashboard to ensure that it could be applied across a wide range of use-cases, rather than being tailored to a specific type of event log or industry. By opting for a more generic design, the dashboard is capable of handling various scenarios, making it a versatile tool for different users and organizations.

For example, instead of designing the dashboard to only analyze event logs from a specific process or domain, we ensured that it could assess the quality of any event log that adheres to the XES standard. This decision meant that the dashboard could be used across industries, whether in manufacturing, healthcare, or finance, to assess the quality of their event logs without needing significant modifications. This approach allows for broader applicability and ensures that the tool remains useful in diverse contexts, ultimately providing greater value to a wider audience.

A significant design decision was to implement a color-coded system to convey the results of the data quality assessments. This approach aimed to make the results immediately understandable, allowing users to quickly identify areas that needed attention.

However, this decision also led to the exclusion of certain complex visualizations that, while potentially informative, might have overwhelmed the user or detracted from the dashboard's clarity and accessibility.

Our goal was to minimize the burden on the analyst and streamline the quality assessment process. Therefore, we require only the event log itself, as it is the essential file that the analyst must possess to perform process mining. By not demanding additional files or information, we make our framework more accessible and easier to use, ensuring that analysts can quickly and efficiently evaluate the quality of their event logs without needing to gather and manage additional documentation. This decision reflects our commitment to simplicity and user-friendliness in our approach.

Another critical design choice was the decision to compartmentalize the dashboard into distinct quality dimensions. This modular approach allows users to focus on specific aspects of data quality, making analysis more manageable and targeted. An additional benefit of this design decision is the possibility to extend the dashboard with more quality dimensions, as there is no interdependence between the modules.

5.2.2. Software Architecture

The software architecture of the EDQD is designed to balance simplicity, scalability, and modularity, enabling it to provide comprehensive data quality assessments while being adaptable to different user needs and environments. The architecture is structured around several key components that interact efficiently to deliver a seamless user experience: the front-end interface, back-end processing, data parsing and assessment modules. Below is a breakdown of the core architectural components and their roles in the overall system.

Front-end Architecture
The front-end of the dashboard is built as a web application, designed to be user-friendly and intuitive. It leverages HTML, CSS, and JavaScript to create an interactive user interface, allowing users to upload event logs and view results in a clean, visually appealing manner. A key architectural decision for the front-end is the use of color-coded boxes representing the five main quality dimensions. Each box contains sub-components that represent more granular assessments of data quality within that dimension.

The front-end interacts with the back-end through API calls, passing the uploaded event logs for assessment and receiving the results in return. The dashboard is designed to handle large files by providing visual feedback during the upload and processing phases, illustrated by a loading bar animation, ensuring that users are aware of the progress being made.

Back-end Architecture
The back-end is developed using Flask, a lightweight Python-based web framework. Flask is well-suited for this application due to its simplicity, scalability, and ability to handle

the event-driven requests made by the front-end. The back-end is responsible for several key functions:

- Accepting event log uploads from the front-end
- Parsing the event logs and processing the data
- Running data quality assessments across the various dimensions
- Returning results to the front-end for display

The back-end is designed to be stateless, meaning each request (i.e., event log upload) is processed independently, making it easier to scale the application horizontally if needed. This design choice ensures that the dashboard can handle multiple users simultaneously without significant performance degradation.

Data Parsing and Assessment Modules

At the core of the back-end is the data parsing component, which reads and processes the uploaded event logs. These logs, typically in XES format, are parsed using XML parsing libraries. The parser is responsible for extracting traces, events, and attributes, ensuring that all data is correctly structured for analysis.

Once the event logs are parsed, they are passed to the assessment modules, which are organized around the five quality dimensions. Each dimension has a dedicated module that performs a set of checks and assessments, depending on the nature of the dimension. The modules are:

- The **Accuracy module** detects issues such as duplicate events, unrealistic timestamps, and traces that may belong to a different process.
- The **Completeness module** checks for missing attributes, incomplete traces, and unrecorded events.
- The **Interpretability module** assesses the coarseness of several key event attributes such as timestamps, activity names and resource information.
- The **Relevancy module** checks for noise event that might skew process mining results.
- The **Validity module** ensures that the data adheres to the XES standard, verifying attribute keys, values, and data types.

These modules are highly modular, meaning new quality dimensions or indicators can be added to the system without disrupting the overall architecture. This modularity ensures that the dashboard remains adaptable to future requirements and evolving standards.

Dockerization and Deployment

To ensure consistency and ease of deployment across different environments, the entire

application is containerized using Docker. Docker allows the application to be packaged with all its dependencies, ensuring that it runs the same way regardless of the environment in which it is deployed. The use of Docker is particularly advantageous for the Event Data Quality Dashboard (EDQD), as it simplifies the process of deploying the application in cloud environments or on-premise servers, allowing for quick scalability and reducing the risk of configuration issues.

In addition to the core application, the Docker setup includes configurations for mounting the upload folder and exposing the application through a web server, making it accessible to users via a simple web browser interface. This approach ensures that the dashboard is easy to install, configure, and use for both developers and end-users.

Error Handling and Resilience
A critical aspect of the architecture is its resilience and error-handling capabilities. Given the variability in event logs, the system is designed to handle incomplete or malformed logs gracefully. Each assessment module includes error detection mechanisms that, if triggered, notify the user of specific issues via the front-end. Rather than breaking the application, these errors are communicated clearly in the form of error status messages, and the affected assessment results are excluded from the overall quality score. This ensures that users are informed of any issues without the system becoming unusable or providing unreliable results.

Accessibility
The EDQD is an open-source project available on GitHub. Readers are invited to visit my GitHub account[1] to explore, download, and run the code. The dashboard is designed with flexibility and openness in mind, encouraging developers and data scientists to contribute by forking the project and extending its functionality. This open approach ensures that the tool can continue to evolve and adapt to the changing needs of process mining and data quality assessment communities.

5.2.3. User Interface

In this section, we will provide a detailed illustration of the Event Data Quality Dashboard (EDQD) user interface, structured around the three primary use cases outlined in the use case diagram from Figure 5.1. Each use case represents a key interaction between the user and the dashboard, showcasing how the system responds to various tasks. Through these examples, we aim to demonstrate the dashboard's capabilities in a practical context, highlighting the interface's functionality and how it supports users in evaluating event log quality.

1. **Upload Dataset:** In the first use case, *Upload Dataset*, the initial interface of the EDQD is displayed to the user upon opening the application, as seen in Figure 5.2. The interface is clean and straightforward, offering the user two primary options:

[1] `https://github.com/LSOrer/EDQD`

Choose File and *Upload*. The user begins by selecting a file using the *Choose File* button, which is limited to the XES file type. Once the file is chosen, its name will replace the *No file chosen* text shown in the middle of the interface. After selecting the correct file, the user can press the green *Upload* button to proceed to the next menu. This interaction starts the process of uploading the event log for quality assessment. For larger event logs with extended processing times, a loading bar is displayed to inform the user that the data analysis is still in progress.

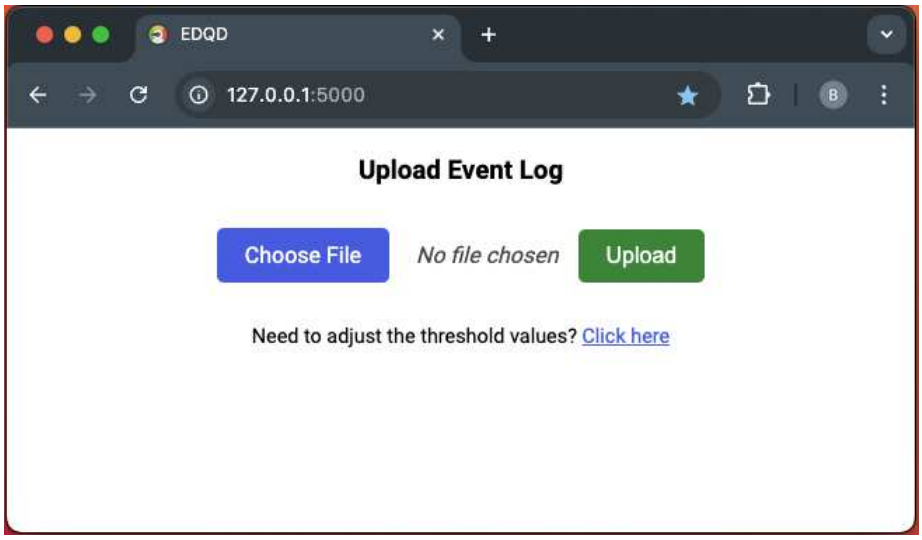

Figure 5.2.: Upload Menu of EDQD

2. **View high-level Goodness of the Dataset:** The second use case, *View high-level goodness of dataset*, is represented in Figure 5.3. After successfully uploading the dataset, the user interface remains easy to navigate with the upload section still accessible at the top, allowing users to upload a new dataset if needed without navigating away from the page.

   Below the upload section, each key data quality dimension is prominently displayed in its own rectangular box. At the top of each box, the overall score for that dimension is presented, quickly highlighting any areas where issues may exist, anything below 100% suggests room for concern. These large boxes contain smaller *indicator boxes* within them, each representing a specific sub-assessment relevant to the overarching quality dimension.

   The color-coded traffic light system is used to visually convey the severity of each issue: green for high quality, yellow for moderate, and red for low quality. Tool-tips appear when the user hovers over the smaller indicator boxes, providing short descriptions of each indicator and what it assesses. Additionally, a *Details* button is

available at the bottom of each dimension, allowing users to explore specific issues in-depth. Finally, at the bottom of the results view, a legend is displayed, guiding users on how to interpret the color codes for each dimension and indicator.

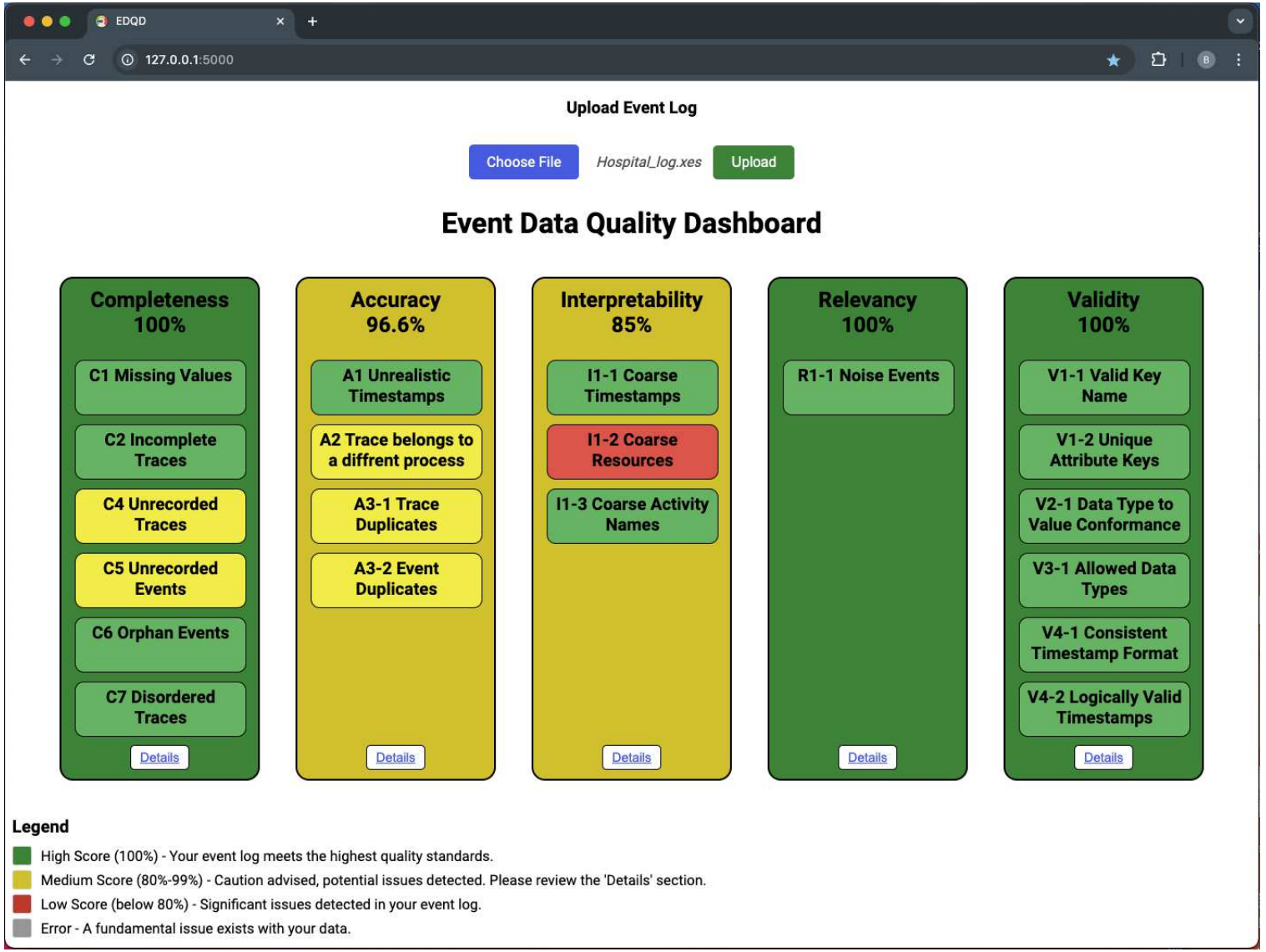

Figure 5.3.: Quality Assessment Results of the EDQD

3. **View Details about the assessed Dataset:** Third, the details view use-case is shown in Figure 5.4. As described in the previous use case, each quality dimension box has a *Details* button at the bottom. When the user clicks on this button, a modal window appears, displaying the detailed results of all assessments. This window allows the user to examine the specific scores that have been calculated and understand how the overall dimension score was derived.

   The detailed results are organized by their corresponding indicators, keeping the view structured and easy to navigate. The user can click on any indicator of interest to expand and view the underlying data. For large logs, where detail results may consist of hundreds or thousands of elements, the system includes a *show more* button, which ensures the interface remains user-friendly while still providing access to the complete set of results if needed.

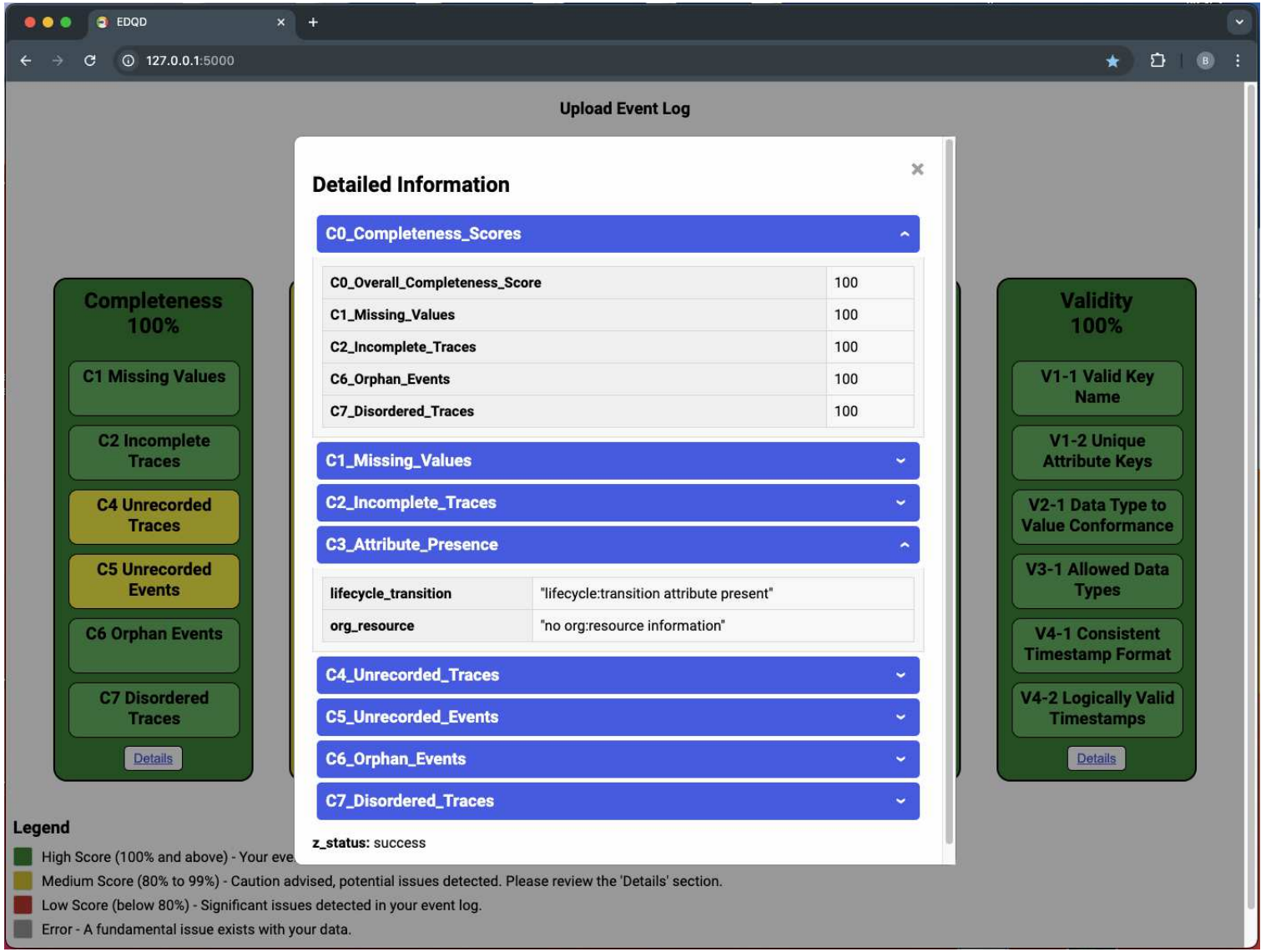

Figure 5.4.: Details View of the EDQD

4. **Adjust Threshold Parameters:** Lastly, the use-case for adjusting the threshold parameters is shown in Figure 5.5. This menu provides users with a configurable interface in the dashboard to fine-tune parameters related to both color-coded scoring thresholds and function-specific thresholds. This use case is divided into two primary sections:

   - Color-Code Thresholds: Users can adjust the boundaries that determine when dimension scores appear as green, yellow, or red in the dashboard. By setting specific percentages, users control the exact values at which scores will trigger each color. This customization helps users visually interpret results based on their specific quality standards.
   - Function Thresholds: Users can also adjust thresholds related to key functions within the assessment, such as parameters for detecting unrecorded events or traces. For instance, users may modify sensitivity levels for identifying patterns in trace lengths or gaps in event timestamps. Lowering or raising these values allows for greater flexibility, accommodating variations in event log characteristics and user preferences.

   The *Adjust Threshold Parameters* screen provides input fields for each threshold, allowing precise numerical entry. Once saved, these settings directly influence subsequent data quality assessments, ensuring that the results align with the user's desired standards. This functionality makes the dashboard highly adaptable for different datasets and analysis requirements, enhancing both usability and interpretability for analysts.

5.3. Composition of Quality Dimensions, Elements and Indicators

We described our conceptual consideration about the assessment framework, how we want to cluster quality dimensions, their elements and indicators that assess them in section 4.2.2. In the following section we will go into the practical application of the conceptual framework and showcase how each and every quality dimension is composed of elements that are in turn composed of indicators. We would like to highlight two visual components of the following diagrams, as they are of high importance for the understanding how they interact with the actual implementation of the dashboard;

1. **Color-code**: Quality dimensions, elements and indicators of the following figures will ether have a yellow or green box color. This color indicates whether the element or indicator has a direct influence on the calculated quality dimension score. We will go into detail why some assessments do not influence the dimension score at the concerned diagrams.
   - **Green**: This element or indicator has a direct impact on the calculation of the quality dimension score.

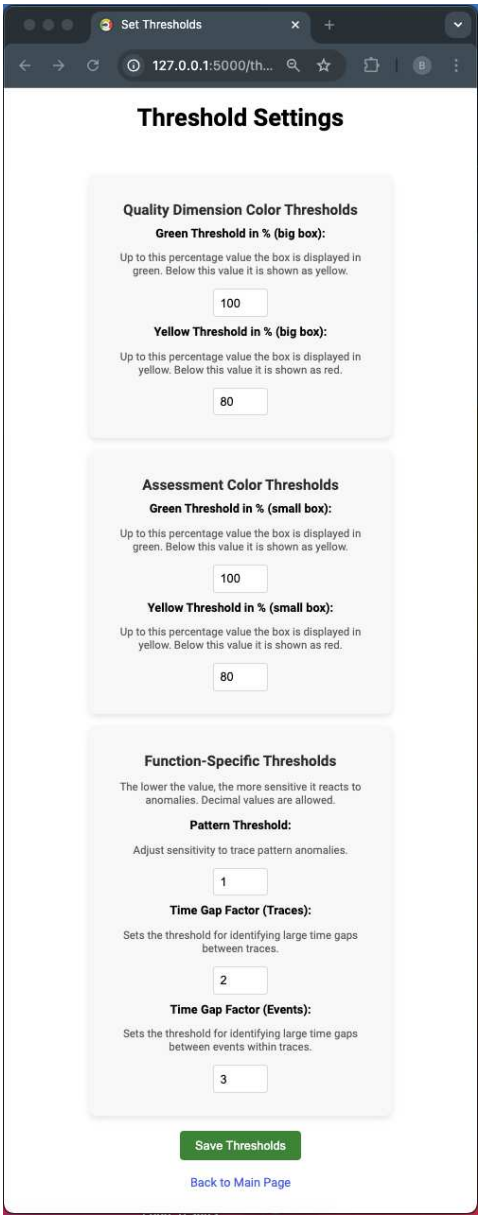

Figure 5.5.: Threshold Configuration Menu of EDQD

- **Yellow**: This element or indicator does not have a direct influence when calculating the dimension score. It is rather a hint for the process analyst that there might be something wrong within the concerned dataset.

2. **Border-thickness**: The front-end of the dashboard is composed of two graphical components: big boxes, that display the quality dimensions and small boxes placed inside the big dimension boxes. These small boxes represent the assessments that derive how flawless a given dataset is in respect to each quality dimension. With simplicity and user-friendliness in mind these small boxes are either elements or indicators according to the assessment framework logic (see section 4.2.2). Sometimes a grouping of indicators (i.e. an element) makes things easier to represent, sometimes it is better to visualise the indicator itself as it bears a lot of information. The border thickness of the following figures indicates which elements or indicators are shown in the front-end. Thick borders of an element or indicators indicate the presents in the front-end, whereas a thin border indicates the absence in the front-end.

5.3.1. Composition of the Accuracy Assessment

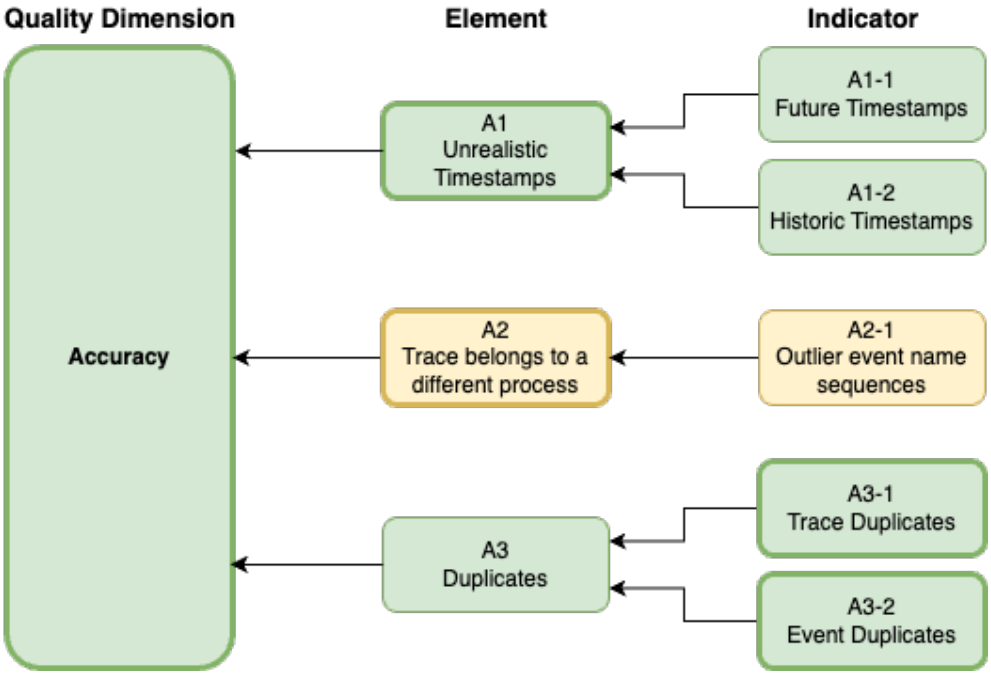

Figure 5.6.: Components of the Accuracy Assessment

Fig. 5.6 illustrates the components of the accuracy assessment. The assessment is composed of three elements, i.e. A1 Unrealistic Timestamps, A2 Trace belongs to a different process, and A3 Duplicates, and five indicators, i.e. A1-1 Future Timestamps,

A1-2 Historic Timestamps, A2-1 Outlier event name sequences, A3-1 Trace Duplicates and A3-2 Event Duplicates. When we look at the border thickness of the boxes in the diagram, we can see that four assessment results, two elements (A1 and A2) and two indicators (A3-1 and A3-2), are shown to the user in the actual implementation of the dashboard. The reasoning behind choosing on the one hand side a grouped representation of two indicators (A1) and on the other side a more granular exposition of duplicates (A3-1 and A3-2) is straightforward the representation format. While we can represent unrealistic timestamps in a uniform format, i.e. by printing the trace-id in which future or historic timestamps are recorded, we cannot do this for trace and event duplicates. Trace duplicates are shown in an easy-to-understand way by a list of trace-ids. Event duplicates, on the other hand, are not only grouped by the trace-id in which they occur, but also hold the activity-name that suggests an event duplicate. This quite substantial difference in representation was the reason why we chose to group unrealistic timestamps in one assessment result box and split the representation of trace and event duplicates into two separate boxes in the implemented dashboard.

When it comes to the assessment results that directly influence the overall quality dimension score of the accuracy dimension, we have to look at the green squares of Fig. 5.6. We can see that A1 and A3 (and also the indicators that are grouped by those elements) have a influence on the calculation of the overall accuracy score. On the other hand we see that A2, as it is displayed in a yellow square, does not influence the overall accuracy score. Our deliberations, in this particular instance and also for all other assessment results that do not influence the dimension score, are in respect to the definitive nature of the outcomes of these assessments. While we can confidently identify a problem when a timestamp is in the future, we cannot, with the same level of certainty, determine if a trace with an event name sequence that differs from the standard is truly an issue or simply a coincidence. This underlying level of confidence in the true-fullness of the results is the reason why we decided in this and also other cases, that we will see in the description of the components of the completeness assessment, to not include the results when computing the overall dimension score.

Description of the Accuracy Sub-scores

1. **A1 Unrealistic Timestamps**: Assesses whether there are any timestamps that are in the future or unrealistically far in the past, which would suggest inaccuracies in logging.

2. **A2 Trace belongs to a different process**: Flags traces that may belong to a different process based on deviations in the event name sequence.

3. **A3-1 Trace Duplicates**: Identifies traces that have been duplicated in the log, leading to inflated or incorrect analysis.

4. **A3-2 Event Duplicates**: Detects duplicate events within the same trace, which could result from logging errors or system misbehavior.

Calculation of the Accuracy Scores

1. **A0 Overall Accuracy Score**: Calculated by applying weighted averages to the three sub-scores A1, A3-1 and A3-2. The final accuracy percentage is then determined by summing up each sub-score multiplied by its weight.

2. **A1 Unrealistic Timestamps**: Weighted at 20%. If there are zero inaccurate timestamps (no timestamps in the future or past), this sub-score is set to 100%. If there are any inaccurate timestamps, the sub-score is set to 0%.

3. **A2 Trace belongs to a different process**: This sub-score has no influence on the overall accuracy score.

4. **A3-1 Trace Duplicates**: Weighted at 40%. Calculated by normalizing the number of duplicate events. The formula decreases the score based on the count of duplicates relative to all traces of the log and adjusts it to a scale between 0 and 100%.

5. **A3-2 Event Duplicates**: Weighted at 40%. Similar to the duplicate events score, it is normalized by the number of duplicate traces relative to a maximum count.

Lastly, we would like to address a critical point that someone with a deep understanding of data quality dimensions might raise: Why is duplication/uniqueness not a own quality dimension but embedded into accuracy. While there is some, yet little, scientific evidence in the sphere of data quality that researchers see uniqueness as its own, standalone quality dimension [33], we would beg to differ, especially in case of event data. Duplicates, whether they are entire traces or individual events, typically result from a malfunction in the logging engine, causing traces or events to be recorded incorrectly. This inaccurate reflection of real-world events, in our view, is more an issue of correctness, and therefore falls under the dimension of accuracy, rather than warranting its own separate quality dimension.

5.3.2. Composition of the Completeness Assessment

Fig. 5.7 outlines the components of the completeness assessment. This dimension is composed of seven primary elements: C1 Missing Values, C2 Incomplete Traces, C3 Attribute Presence, C4 Unrecorded Traces, C5 Unrecorded Events, C6 Orphan Events, and C7 Disordered Traces. These elements are further broken down into specific indicators, such as C1-1 Missing Trace Attributes, C1-2 Missing Event Attributes, C4-1 Trace Pattern Anomalies, and C5-1 Inter-Event Time Gap Anomalies, resulting in a detailed assessment of completeness across multiple facets.

The diagram again differentiates between the components shown to the user and those that are evaluated but do not influence the overall completeness score. As seen in the diagram, elements like C1, C2, C4, C5, C6 an C7 are represented with thicker borders, indicating that their results are displayed to the user. For instance, C1-1 and C1-2 highlight missing trace and event attributes, which are implementation-wise assessed

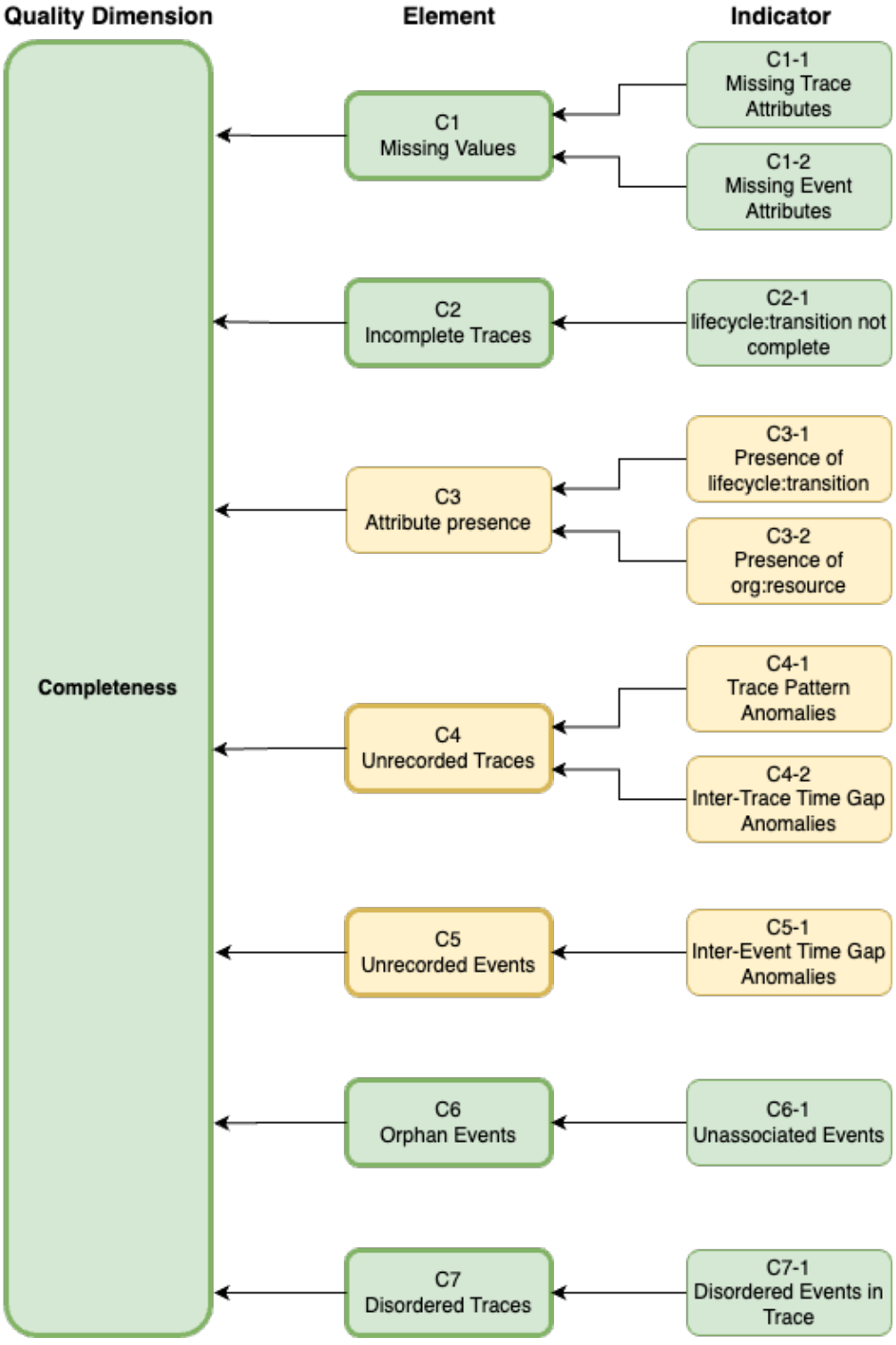

Figure 5.7.: Components of the Completeness Assessment

separately, but displayed to the user in a combined manner.

On the other hand, C3 Attribute Presence, C4 Unrecorded Traces and C5 Unrecorded Events, shown in yellow, are important for identifying gaps in trace or event recording but do not directly influence the overall score calculation. These anomalies are evaluated through indicators like C4-1 Trace Pattern Anomalies and C5-1 Inter-Event Time Gap Anomalies. While these results are available to the user for review, their degree of uncertainty means they do not factor into the overall completeness score. This approach mirrors the rationale seen in the accuracy dimension, where certain results, like A2 Trace Belongs to a Different Process, were excluded from the score due to uncertainty about their impact.

An important design decision for completeness is how missing values are represented uniformly in the assessment results. Both trace and event attributes are checked for missing values, and the results are displayed to the user in a consolidated manner (C1). Equally, when dealing with issues such as unrecorded traces, the system needs to track both trace pattern anomalies and inter-trace time gap anomalies, which require separate considerations but uniform representation. Therefore, while these are grouped and displayed under the same general concept of unrecorded traces, they are displayed as distinct assessment results to the user in the details section of the completeness box for a clear understanding of the issue.

The overall completeness score is directly influenced by the elements represented in green, namely C1, C2, C6 and C7, as these aspects are the most concrete indicators of the completeness of the event log. These elements include the detection of missing values, incomplete traces, missing linkage between events and traces and faulty ordering of events within a trace. Elements like C4 Unrecorded Traces and C5 Unrecorded Events, displayed in yellow, do not impact the score directly because of the uncertainty associated with identifying unrecorded data. While such issues are important, their statistical nature makes it harder to assess them with absolute certainty, hence the decision to exclude them from the score calculation.

Description of the Completeness Sub-scores

1. **C1 Missing Values**: This indicator checks if important attributes are missing in traces or events. Missing values may lead to incomplete or unreliable data.

2. **C2 Incomplete Traces**: Evaluates whether any traces in the event log are incomplete, meaning that crucial transitions or events may be missing.

3. **C3 Attribute Presence**: Checks wether commonly used attributes such as lifecycle:transition or org:resource are present.

4. **C4 Unrecorded Traces**: Identifies traces that are potentially unrecorded due to anomalies in patterns or unusual time gaps between traces.

5. **C5 Unrecorded Events**: Detects missing events within traces, typically identified by unusually large time gaps between consecutive events.

6. **C6 Orphan Events**: Identifies events that are not properly linked to any trace, leading to potential data inconsistencies.

7. **C7 Disordered Traces**: This metric checks whether events within traces are ordered correctly based on timestamps.

Calculation of the Completeness Scores

1. **C0 Overall Completeness Score**: Calculated as the weighted sum of four sub-scores, while each sub-score is weighted equally at 25%. The final completeness percentage is calculated by multiplying the weighted sum by 100.

2. **C1 Missing Values**: Calculates the number of missing attribute values for both trace and event attributes. The score is normalized based on the total missing values against all trace and event attributes. A score of 100% means that there are no missing values, while more missing values reduce the score accordingly.

3. **C2 Incomplete Traces**: The score is normalized based on the number of incomplete traces relative to all traces. Zero incomplete traces give a score of 100%, with the score decreasing as the number of incomplete traces increases.

4. **C3 Attribute Presence**: No sub-score is calculated and therefore it has no influence on the overall completeness score.

5. **C4 Unrecorded Traces**: This sub-score has no influence on the overall completeness score. It is a hint for the process analyst to inspect anomalies.

6. **C5 Unrecorded Events**: Same as C4 Unrecorded Traces.

7. **C6 Orphan Events**: This score is normalized against all events of the event log. A score of 100% means no orphan events, with the score decreasing as the number of orphan events increases.

8. **C7 Disordered Traces**: The score is normalized by subtracting a factor based on the number of disordered traces. If no disordered traces are found, the score is 100%; more disordered traces reduce the score proportionally.

Finally, we want to emphasize a critical aspect of the completeness dimension. The aim is to ensure that every necessary piece of data is recorded and present in the event log. While some gaps, like unrecorded traces, may arise from the way logs are extracted, others, like missing values, represent clear-cut problems that need to be addressed.

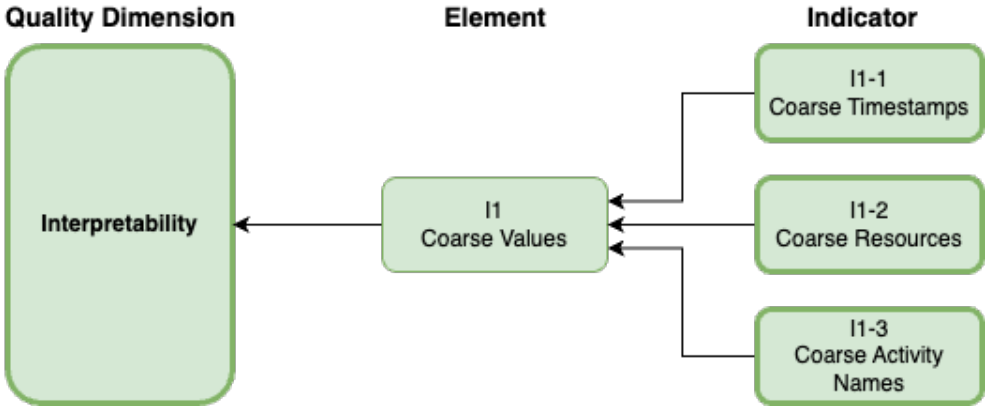

Figure 5.8.: Components of the Interpretability Assessment

5.3.3. Composition of the Interpretability Assessment

Fig. 5.8 displays the components of the interpretability assessment in the dashboard. This dimension is more streamlined compared to the completeness and accuracy dimensions, as it primarily focuses on the coarseness of values within the event log. The assessment consists of one core element, I1 Coarse Values, which is further broken down into three specific indicators: I1-1 Coarse Timestamps, I1-2 Coarse Resources, and I1-3 Coarse Activity Names.

The interpretability dimension is critical for ensuring that the event log data is not only present but also meaningful and detailed enough for analysis. Coarse data refers to values that lack the necessary granularity to provide useful insights. For instance, timestamps that only capture the day and not the specific time of events, or activity names that are overly generic, could make interpreting the process flow challenging. Each of the indicators addresses a different type of coarseness that could impact the interpretability of the event log.

Description of the Interpretability Sub-scores

1. **I1-1 Coarse Timestamps**: This metric checks if the timestamps are too coarse, meaning they might only record days without accounting for hours, minutes, or seconds.

2. **I1-2 Coarse Resources**: Verifies if the resource-related attributes (such as the responsible entity for an event) are recorded in a detailed or overly general way.

3. **I1-3 Coarse Activity Names**: Detects whether activity names are too vague or not descriptive enough to allow for meaningful process analysis.

Calculation of the Interpretability Scores

1. **I0 Overall Interpretability Score**: Calculated as the weighted average of three sub-scores with weights defined in the weights dictionary. The final interpretability score is multiplied by 100 to convert it into a percentage.

2. **I1-1 Coarse Timestamps**: Weighted at 30%. Evaluates whether timestamps are granular enough for meaningful analysis. If timestamps are sufficiently granular, the score is 100%; otherwise, it is 0%.

3. **I1-2 Coarse Resources**: Weighted at 30%. If resource information is fully granular, the score is 100%. Partial granularity (only org:group or org:role recorded) results in a score of 50%, if both are present but no org:resource it gives a score of 80%. If resources are not recorded at all, the score is 0%.

4. **I1-3 Coarse Activity Names**: Weighted at 40%. Evaluates the granularity and descriptiveness of activity names in the log, calculating the average word count and length of activity names. Activity names are considered coarse if they have an average word count of fewer than 2 words or if the average length of names is less than 6 characters. If these conditions are met, it suggests that activity names lack sufficient detail, making analysis less meaningful for process evaluation. If the activity names meet the granularity criteria (average word count larger or equal 2 and average length larger or equal 6), the sub-score is 100%. If either criterion is not met, the sub-score is set to 0%.

In terms of the results shown to the user, the interpretability dimension is entirely represented by its associated indicators. Each of the indicators is almost equally important (40% I1-3, 30% I1-1 and 30% I1-2) in contributing to the interpretability score, and the results are shown to the user in their entirety. As indicated by the green borders in the diagram, all three indicators directly contribute to the overall score for the interpretability dimension.

Unlike other dimensions, such as accuracy, where certain components may be excluded from the overall score due to uncertainty, the interpretability dimension includes all its indicators in the score calculation. This is because coarse data can be definitively identified, and its impact on the interpretability of the event log is clear.

5.3.4. Composition of the Relevancy Assessment

Fig. 5.9 illustrates the components of the relevancy assessment, which is notably simpler than other dimensions, with only one core element, R1 Irrelevant Events, and one associated indicator, R1-1 Noise Events.

The relevancy dimension focuses on identifying events within the log that do not contribute meaningfully to the analysis of the process. These irrelevant events, often referred to as noise events, include system-generated notifications (e.g., "system start" or

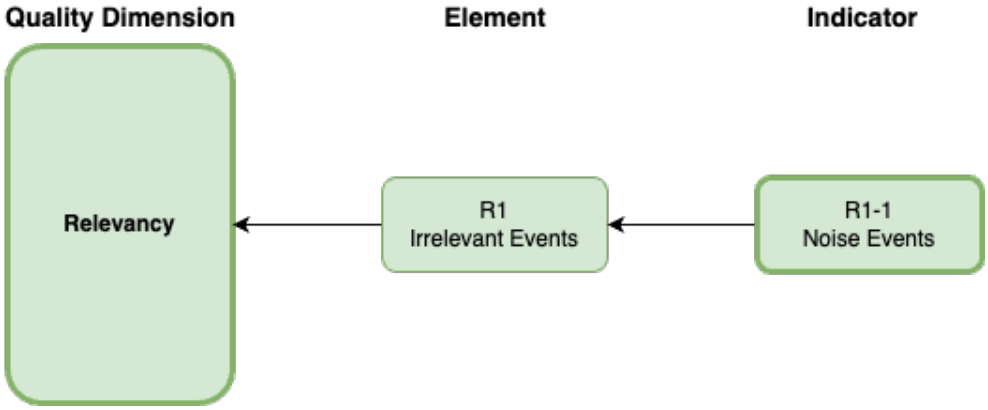

Figure 5.9.: Components of the Relevancy Assessment

"maintenance") that, while informative about the system's status, do not provide insights into the actual process being analyzed. Such events can skew the analysis, making it more complex and less accurate by introducing data that is not tied to the core process under investigation.

In the R1-1 Noise Events indicator, the dashboard detects and flags these irrelevant or extraneous events. This check is crucial because noise events, though valid in the context of the system, do not contribute to process understanding and can even distort the analysis results.

Calculation of the Relevancy Score

1. **R0 Overall Relevancy Score**: This score reflects the proportion of relevant events in the event log, taking into account any noise events identified in the data. The score is calculated as a percentage, where a high score indicates a low presence of noise events.

2. **R1-1 Noise Events**: The function first extracts noise events from the results dictionary by summing the length of events classified as noise within each type of noise event. If noise events are found to be significant, the relevancy score is reduced. If there are no noise events, the score remains at 100%, reflecting high relevancy.

Given the simplicity of this dimension, all results are shown to the user, and the overall relevancy score is based on whether noise events are detected. If the system identifies noise events, this lowers the score to reflect the presence of irrelevant data in the event log. The color-coded display directly represents the result of the R1-1 Noise Events check, with green indicating no noise events and red signaling that such events were found.

5.3.5. Composition of the Validity Assessment

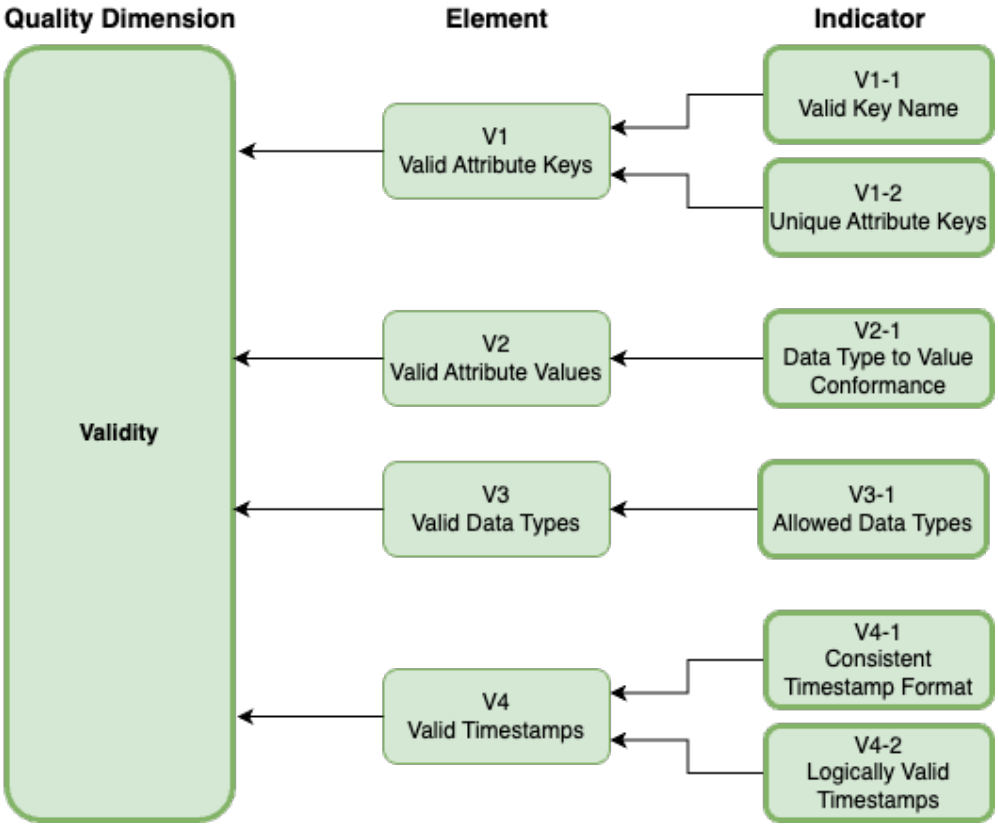

Figure 5.10.: Components of the Validity Assessment

Fig. 5.10 presents the components of the validity assessment. This dimension is comprised of four primary elements: V1 Valid Attribute Keys, V2 Valid Attribute Values, V3 Valid Data Types, and V4 Valid Timestamps. These elements collectively ensure that the data in the event log conforms to the expected standards and formats.

Description of the Completeness Sub-scores

1. **V1-1 Valid Key Name**: Checks if attribute keys conform to the XES standard, ensuring they do not contain illegal characters like line breaks, tabs, or carriage returns.
2. **V1-2 Unique Attribute Keys**: Verifies that attribute keys are unique within their respective containers, avoiding potential conflicts or confusion in analysis.
3. **V2-1 Data Type to Value Conformance**: Ensures that the values of attributes match their expected data types, e.g., integers should contain numeric values, and booleans should be either true or false.

4. **V3-1 Allowed Data Types**: Validates that only allowed data types (such as string, date, integer, float, etc.) are present in the event log.
5. **V4-1 Consistent Timestamp Format**: Assesses whether timestamps are formatted consistently across the entire log, avoiding issues with differing time formats.
6. **V4-2 Logically Valid Timestamps**: Checks if timestamps follow logical constraints, such as valid days, hours, and months (e.g., no 25th hour or 45th month).

Calculation of the Validity Scores

1. **V0 Overall Validity Score**: This score reflects the overall data validity based on specific aspects like attribute keys, data type conformance, and timestamp accuracy. It is calculated as the average of six sub-scores, each covering a critical validity dimension, and represented as a percentage.
2. **V1-1 Valid Key Name**: Checks if all attribute keys conform to validity standards (e.g., no illegal characters). If there are no invalid keys, the score is 100; otherwise, it's 0.
3. **V1-2 Unique Attribute Keys**: Evaluates whether all attribute keys are unique within their respective traces or events. A score of 100 is assigned if there are no duplicate keys, otherwise 0.
4. **V2-1 Data Type to Value Conformance**: Checks if attribute values conform to their expected data types (e.g., integers, strings). If no invalid values are found, the score is 100; if there are any, the score is 0.
5. **V3-1 Allowed Data Types**: Ensures only allowed data types are used as per the XES standard. A score of 100 is given if all data types are compliant; otherwise, 0.
6. **V4-1 Consistent Timestamp Format**: Verifies timestamp consistency across the dataset. If all timestamps follow a consistent format, the score is 100; any inconsistency results in a score of 0.
7. **V4-2 Logically Valid Timestamps**: Checks that timestamps are logically correct (e.g., no. 25th hour). The score is 100 if all timestamps are valid; otherwise, it's 0.

The diagram shows that all indicators are displayed directly to the user in the dashboard, indicated by the thick borders. Each of these components contributes to the overall Validity score, as they all play an important role in determining whether the event log is structured and formatted in a way that allows for accurate process analysis.

The decision to include all results in the validity score reflects the certainty and definitive nature of these checks. Unlike other dimensions where uncertainty might lead to excluding certain indicators from the score calculation (as seen with completeness and accuracy), the results of the validity assessment are clear-cut: the data either conforms to the standards or it does not. As such, any issues detected directly impact the overall score for this dimension.

5.4. Comparison between envisioned Measures and actually implemented Assessments

In this section, we will explore the gap between the envisioned measures conceptualized in Chapter 4 and the actual implemented assessments presented in Section 5.3. During the conceptual phase, a comprehensive framework for assessing data quality dimensions was laid out, outlining a set of measures aimed at ensuring the accuracy, completeness, interpretability, relevancy, and validity of event log data. These envisioned measures were intended to form the foundation of the Event Data Quality Dashboard.

However, as is often the case in the process of transforming theoretical concepts into practical implementations, challenges arose during development that required adjustments to the original framework. Certain envisioned measures, while theoretically sound, proved difficult to implement due to technical limitations or the complexities inherent in working with diverse event log formats. Furthermore, as the development process progressed, new insights emerged that led to the introduction of additional assessments not originally planned in the conceptual phase.

Table 5.1 compares the initial set of envisioned measures with the final set of implemented assessments.

For a more in-depth comparison of the envisioned measures and the actual indicators implemented, detailed tables have been created to highlight the differences and similarities. These tables provide a comprehensive overview and can be found in the Appendix A of this thesis.

Dimension	Measure No.	Implemented	Indicator
Completeness	Measure 1	[x]	C4
4.2	Measure 2	[x]	C5
	Measure 3	[x]	C6
	Measure 4	[x]	C1
	Measure 5	[x]	C7
	Measure 6	[x]	C1
	Measure 7	[x]	C1
	Measure 8	[x]	C1
	Measure 9	[x]	C1
	Measure 10	[x]	C2
Accuracy	Measure 1	[x]	A2
4.3	Measure 2	[x]	A3-2
	Measure 3	[-]	-
	Measure 4	[-]	-
	Measure 5	[x]	C7
	Measure 6	[-]	-
	Measure 7	[x]	A1
	Measure 8	[-]	-
	Measure 9	[-]	-
	New Measure	[x]	A3-1
Interpretability	Measure 1	[-]	-
4.5	Measure 2	[-]	-
	Measure 3	[x]	A3-2 and C7
	Measure 4	[x]	I1-3
	Measure 5	[x]	I1-1
	Measure 6	[x]	I1-2
Relevancy	Measure 1	[-]	-
4.6	Measure 2	[x]	R1-1 and A3-2
Validity	Measure 1	[x]	V1-1 and V1-2 and V3-1
4.4	Measure 2	[x]	V2-1
	Measure 3	[x]	V4-1
	New Measure	[x]	V4-2

[x] Measure implemented

[-] Measure not implemented

Table 5.1.: Overview of implemented Measures

6. Results

The results chapter presents a comprehensive evaluation of the Event Data Quality Dashboard (EDQD), demonstrating its effectiveness across a range of scenarios. In Section 6.1, Evaluation with Synthetic Data Sets, we assess the dashboard's functionality by applying it to self-generated datasets designed to test each feature and function in isolation. This ensures that the underlying mechanisms work as expected across different quality dimensions. Section 6.2, Evaluation with Public Data Sets, extends the testing to datasets from well-known sources, particularly focusing on event logs from the Business Process Intelligence Challenge (BPIC). This allows us to examine the dashboard's performance on realistic and widely recognized data, providing an external validation. Finally, Section 6.3, Evaluation with Real-World Data Sets, takes the testing further by using data from an IT service management enterprise to demonstrate the dashboard's utility in a practical business setting, ensuring its relevance for real-world application.

6.1. Evaluation with Synthetic Data Sets

In this section, we dive into the comprehensive testing process that was undertaken to rigorously validate the Event Data Quality Dashboard. To ensure that every feature and function of the dashboard performed as expected, we created a total of 47 synthetic event logs. These logs were designed to test each function individually, allowing us to guarantee the accuracy and robustness of the dashboard's results.

The foundation for all synthetic event logs is the 0_base_log.xes, which serves as a standard baseline for testing. This base log consists of ten process instances, each containing five events. These traces are modeled based on the process introduced in Chapter 2, Section 2.4 where a guest's interaction with a restaurant — from checking in at a table to paying the bill — is captured. This consistent structure provided us with a controlled environment in which specific data quality issues could be inserted and evaluated.

For each function within the dashboard, we typically developed at least two synthetic event logs. In the first log, only one trace would demonstrate the issue, allowing us to assess the function's ability to detect isolated instances of the problem. The second log would include the issue across all traces, helping us to ensure that the function could handle more pervasive occurrences. For more complex functions, such as *A2 Trace belongs to a different process*, we went even further, creating three logs to show how the function behaves under various conditions and how it handles nuanced cases in real-world scenarios.

At the beginning of each test log, we included detailed comments explaining the nature of the data quality issue present in that particular log. These comments not only describe what the issue is but also indicate where it can be found and, in some cases, specify which hyperparameters need to be set to trigger its detection. This level of transparency ensures that the synthetic logs are easy to follow and allows users to understand exactly how the dashboard identifies and flags data quality problems.

The structure of each synthetic event log follows a consistent format:

1. **Log Naming**: The name of the log provides a clear indication of which quality dimension, element, or indicator is being tested. For instance, C1_missing_values_1.xes tests the handling of missing values in the dashboard's completeness dimension.
2. **Comment Section**: A comment at the top of the log details the problem, where it is located, and any necessary settings for the dashboard to detect the issue. This ensures users have all the information they need to replicate or understand the testing process.
3. **Event Log Content**: The event log itself contains the quality issue embedded within a realistic process flow, allowing for direct observation of how the dashboard interacts with the data.

In some cases, such as with missing values or disordered traces, the issue is straightforward and can be easily observed within the log. However, for more subtle problems, such as potential traces of a different process, the comments and carefully structured logs provide necessary context and guidance for understanding the expected behavior of the dashboard.

Beyond individual function tests, we also created a comprehensive synthetic log known as *Z_murphys_event_log.xes*. This log is a deliberate attempt to showcase all potential data quality issues that the dashboard is capable of detecting, consolidated into one extreme test case. It contains examples of missing values, duplicates, disordered traces, unrealistic timestamps, and more, demonstrating the full range of the dashboard's capabilities. By testing the dashboard against this log, we can evaluate how well it handles complex and overlapping issues in a single dataset.

This extensive evaluation with synthetic data provides a crucial foundation for the subsequent testing with publicly available and real-world data, ensuring that every feature of the Event Data Quality Dashboard has been thoroughly vetted in a controlled environment before moving to more unpredictable data sources. The following sections will build upon this work, exploring how the dashboard performs when applied to actual event logs from different domains.

```
<?xml version="1.0" encoding="UTF-8" ?>
<log xes.version="1.0" xes.features="nested-attributes" openxes.version="
    1.0RC7">
```

```
<extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xesext"/>
<extension name="Lifecycle" prefix="lifecycle" uri="http://www.xes-standard.org/lifecycle.xesext"/>
<extension name="Concept" prefix="concept" uri="http://www.xes-standard.org/concept.xesext"/>
<classifier name="Event Name" keys="concept:name"/>
<classifier name="(Event Name AND Lifecycle transition)" keys="concept:name lifecycle:transition"/>
<string key="concept:name" value="XES Event Log"/>
<!-- This event log incorporates missing values, which can manifest as empty strings, multiple null values, or a series of spaces. -->
<!-- Several types of null-values or empty strings, or a combination of both, in trace and event attributes of "Guest9720". -->
    <trace>
    <string key="concept:name" value="Guest9720"/>
    <fload key="check sum" value="nan"/>
        <event>
            <string key="org:resource" value="null"/>
            <int key="Table" value="NULL"/>
            <string key="concept:name" value="N/A"/>
            <string key="lifecycle:transition" value="      "/>
            <date key="time:timestamp" value="2023-10-22T10:33:00.000"/>
        </event>
        <event>
            <string key="org:resource" value="n/a"/>
            <int key="Table" value="NaN"/>
            <string key="concept:name" value="    n/a     "/>
            <string key="lifecycle:transition" value="None"/>
            <date key="time:timestamp" value="2023-10-22T10:38:00.000"/>
        </event>
        <event>
            <string key="org:resource" value="n/a"/>
            <int key="Table" value=""/>
            <string key="concept:name" value="None"/>
            <string key="lifecycle:transition" value="complete"/>
            <date key="time:timestamp" value="2023-10-22T10:46:00.000"/>
        </event>
        <event>
            <string key="org:resource" value="  "/>
            <int key="Table" value="none"/>
            <string key="concept:name" value="Meals ready"/>
            <string key="lifecycle:transition" value="complete"/>
            <date key="time:timestamp" value="2023-10-22T10:56:00.000"/>
        </event>
        <event>
            <string key="org:resource" value="Tom"/>
            <int key="Table" value="5"/>
            <string key="concept:name" value="Bill customer"/>
            <string key="lifecycle:transition" value="complete"/>
            <date key="time:timestamp" value="2023-10-22T11:34:00.000"/>
        </event>
    </trace>
```

```
50     ...
51 </log>
```

Listing 6.1: Test event log C1_missing_values_1.xes

Listing 6.1 demonstrates how we approach the testing process for each data quality issue within the dashboard. The structure is straightforward, with the log name indicating the issue being tested (in this case, *C1* refers to missing values in the completeness dimension), while the comment section at the beginning of the log explicitly describes where and how the issue is manifested.

In this case, the issue revolves around missing values, which can take various forms, such as empty strings, null values, or a series of spaces. The comment provides guidance on what to look for and hints at which trace (e.g., "Guest9720") holds these issues. As we move into the log itself, we can see examples of the problem:

- **Null values** such as "NULL" and "n/a"
- **Empty strings** or spaces in attributes like "org:resource" and "lifecycle:transition"
- **Variations** of how missing or undefined data might appear, such as "NaN" or "None"

This log is specifically designed to isolate and expose the issue of missing values so that the dashboard can effectively identify and flag it. Similar test logs were created for each function and quality dimension in the system, ensuring comprehensive testing coverage across all potential data quality issues. The clarity provided by the name, commentary, and log structure makes this a useful tool for both verifying the function's performance and demonstrating its behavior to users.

6.2. Evaluation with Business Process Intelligence Challenge Data Sets

In this section, we will evaluate the dashboard using publicly available academic data sets that represent real-life event logs from various domains. All data sets used are sourced from 4TU Research Data[1], a well-known repository for scientific data. Our focus will be on event logs that have been utilized in the Business Process Intelligence Challenge (BPIC), as these logs align closely with the dashboard's objectives.

These event logs are particularly relevant because they often reflect real-world business processes and come from domains like healthcare, finance, and customer service. Moreover, they have been the subject of extensive research, allowing us to compare the findings of our dashboard with those of other scholars. By examining whether our quality assessments match the insights derived from process mining studies, we can cross-validate the accuracy

[1] https://data.4tu.nl/

and relevance of our findings.

However, it is important to note that most of the literature on these data sets focuses on using process mining techniques to extract business insights rather than conducting a comprehensive analysis of data quality. Discussions of data quality are frequently limited to side notes or preliminary observations. Nevertheless, this provides a valuable opportunity to benchmark our dashboard against real-life scenarios and ensure its effectiveness in identifying data quality issues that might otherwise go unnoticed.

6.2.1. Hospital Log BPIC 2011

The first public event log we will analyze using the dashboard is the Hospital Log [11], which was part of the BPIC 2011. This log originates from the healthcare domain, specifically focusing on cancer treatment and diagnosis processes. The event log (doi:10.4121/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54) captures the various steps patients go through, from initial diagnosis to different treatment stages, including consultations, chemotherapy, radiotherapy, and follow-up appointments. The event log is analyzed in [40], where the authors demonstrate the potential of process mining and discuss the unique characteristics of the event log. These specialties are of particular interest to us, as they align with or provide explanations for issues identified by the dashboard.

The assessment of the Hospital Log from the BPIC 2011 reveals that while most data quality dimensions score perfectly, there are notable concerns in the accuracy and interpretability dimensions. Specifically:

1. **Accuracy Issues:**
    - A3-2 Event Duplicates (95%): The dashboard identifies a significant number of event duplicates, with long traces containing dozens of such duplicates. In some cases, there are even three consecutive duplicate events within a trace. This may explain why the authors of [40] observed bursts of activity, as the event duplicates, which share the same timestamp, likely distort the calculation of inter-event time gaps and heavily influence this finding.
    - A3-1 Trace Duplicates (96.5%): Not explicitly mentioned in the research, we identified four traces duplicate pairs where event sequences are identical, suggesting potential logging system malfunctions or re-recordings.
2. **Interpretability Issues**:
    - I1-2 Coarse Resources (50%): This issue highlights the lack of granularity in the recorded resource data, with only the org:group attribute being logged. While this may align with the hospital's documentation standards, the literature suggests that this level of resource information is too imprecise for in-depth

resource-based process analysis. Not recording specific individuals limits the analysis of resource involvement in processes, which is crucial for a detailed understanding of the organisational perspective of process mining.

These findings underscore key areas where the quality of the event log can affect subsequent process mining or analysis.

Assessment	**Score**
Overall Completeness Score	**100%**
C1 Missing Values	100%
C2 Incomplete Traces	100%
C6 Orphan Events	100%
C7 Disordered Traces	100%
Overall Accuracy Score	**96.6%**
A1 Unrealistic Timestamps	100%
A3-1 Trace Duplicates	96.5%
A3-2 Event Duplicates	95%
Overall Interpretability Score	**85%**
I1-1 Coarse Timestamps	100%
I1-2 Coarse Resources	50%
I1-3 Coarse Activity Names	100%
Overall Relevancy Score	**100%**
R1-1 Noise Events	100%
Overall Validity Score	**100%**
V1-1 Valid Key Name	100%
V1-2 Unique Attribute Keys	100%
V2-1 Data Type to Value Conformance	100%
V3-1 Allowed Data Types	100%
V4-1 Consistent Timestamp Format	100%
V4-2 Logically Valid Timestamps	100%

Table 6.1.: Assessment Results of the BPI Challenge 2011 Log [11]

In addition to the clear data quality issues reflected in the assessment scores, the dashboard reveals some potential data quality concerns within the hospital log. These concerns primarily relate to missing or unrecorded events, trace anomalies, and irregular patterns:

1. **C5 Unrecorded Events:**
   - Significant time gaps between the execution of consecutive events suggest that certain activities might have been skipped or were not logged. These gaps, sometimes extending to as long as 19 days between bursts of simultaneous activity, indicate potential unrecorded events, flagged by the dashboard based on deviations in the expected timeframe between events.

2. **C4 Unrecorded Traces:**
    - Some traces contain an unusually high number of events, with counts reaching up to 1814 events in a single trace, which may indicate potential issues, such as traces containing multiple distinct processes or missing events that were not recorded. These anomalies were flagged by the dashboard for further analysis, as such patterns deviate from typical traces in the log.
3. **A2 Trace belongs to a different process:**
    - **Urgent and Non-urgent Cases:** The dashboard flags 24 traces as anomalies based on their activity names. Upon further investigation, these traces reveal a distinctive pattern: at the end of certain activity names, the adjective "spoed" (Dutch for "speed" or "emergency") is appended with a dash. This directly aligns with the findings in Section 2.5 of [40], where the authors describe a distinction between urgent and non-urgent cases. The rarity and distinct nature of these activities led to the dashboard flagging them as potential traces from another process, aligning with findings in the literature where such urgent cases were observed.

These findings, though subtle, highlight areas where the hospital log may benefit from further scrutiny and suggest that there may be additional, hidden data quality issues beyond what is immediately visible in the assessment scores.

6.2.2. Loan Application BPIC 2012

The next event log that we will analyze was provided for the Business Process Intelligence Challenge 2012 [12]. It originates from the financial industry and details the loan application process of a Dutch financial institution. The process includes both automated steps and manual interventions, which will be relevant when reviewing dashboard results. This event log (doi:10.4121/UUID:3926DB30-F712-4394-AEBC-75976070E91F) has also been examined by researchers in [41], which offers us yet another opportunity to compare our findings with those of other scholars.

The evaluation of the BPIC 2012 log shows that while most data quality dimensions score the highest quality standards, there are significant concerns within the interpretability dimension. Specifically:

1. **Interpretability Issues**:
    - I1-3 Coarse Activity Names (0%): Assessment I1-3 Coarse Activity Names evaluates the semantic depth of activity names. In this log, the dashboard flags the activity naming as lacking sufficient descriptive detail, with an average of fewer than two words per activity name. This is evident in the naming of automatic tasks, such as "A_SUBMITTED," "A_ACTIVATED," "A_APPROVED," and "A_REGISTERED." While these names may be adequate for a process

Assessment	Score
Overall Completeness Score	**100%**
C1 Missing Values	100%
C2 Incomplete Traces	100%
C6 Orphan Events	100%
C7 Disordered Traces	100%
Overall Accuracy Score	**100%**
A1 Unrealistic Timestamps	100%
A3-1 Trace Duplicates	100%
A3-2 Event Duplicates	100%
Overall Interpretability Score	**60%**
I1-1 Coarse Timestamps	100%
I1-2 Coarse Resources	100%
I1-3 Coarse Activity Names	0%
Overall Relevancy Score	**100%**
R1-1 Noise Events	100%
Overall Validity Score	**100%**
V1-1 Valid Key Name	100%
V1-2 Unique Attribute Keys	100%
V2-1 Data Type to Value Conformance	100%
V3-1 Allowed Data Types	100%
V4-1 Consistent Timestamp Format	100%
V4-2 Logically Valid Timestamps	100%

Table 6.2.: Assessment Results of the BPI Challenge 2012 Log [12]

analyst within the Dutch financial institution that generated the log, they present challenges for external users trying to understand the actual operations behind these activities. In contrast, manual tasks are named with greater semantic detail, often using at least two descriptive words, following the best practice of combining a verb and a noun (e.g., "W_Completeren aanvraag," meaning "Complete application"). However, due to the overall low average of 1.7 descriptive words per activity name, the dashboard flags this as a data quality issue.

In addition to the issues identified in the interpretability dimension, the dashboard detected potential data quality concerns that indicate there may be underlying problems within the log.

1. **C4 Unrecorded Traces**:
   - We identified several trace pattern anomalies, with the number of events per trace ranging from 41 to as many as 167 (such as in trace 183175). This could be entirely reasonable for complex traces that involve multiple department

handovers or activity loops. When examining the inter-trace time gaps, no irregularities were observed, as the gaps typically range from three to seven hours, which is quite expected.

2. **C5 Unrecorded Events**:
   - The dashboard highlights a significant number of inter-event time gaps, which is entirely expected and aligns with the findings of paper [41]. A closer inspection reveals that these extended time gaps consistently occur between activities beginning with the letter "W," indicating manual tasks. In contrast, activities marked with "A" and "O" represent automated tasks. Paper [41], particularly in Section 2.1, explains this distinction, noting that automated tasks are executed within milliseconds, which skews the average time between events downward. Manual tasks, however, often involve decision-making or delays due to weekends, frequently taking days to complete, thus appearing as outliers.

While the dashboard appropriately flags these anomalies as potential quality issues, they are, in fact, normal for real-world processes and should not be considered problematic in this context.

6.2.3. Incident Management at Volvo IT BPIC 2013

The next data set (doi:10.4121/uuid:500573e6-accc-4b0c-9576-aa5468b10cee) we will analyse is from the BPIC Challenge 2013, which includes three event logs. Our focus will be on the incident management event log [13], which details the incident management process of Volvo IT. Once again, academic papers provide valuable insights into this log, allowing us to verify the dashboard's results. Specifically, we will compare our findings with the analysis presented in [42], ensuring that our evaluation aligns with the broader research community's conclusions.

At first glance, significant issues emerge in the completeness and interpretability dimensions of the log. These findings suggest that the event log may suffer from missing data and semantic clarity, both of which can impact the overall quality and usability of the data for process analysis.

1. **Completeness Issues**:
   - C2 Incomplete Traces (0%): The dashboard identifies a significant number of incomplete traces, as they lack an end event with a lifecycle:transition value indicating closure or completion. Since over one quarter of all traces are incomplete, the dashboard assigns a score of 0% for this assessment. This finding aligns with the research referenced earlier [42], where the authors also highlight in Section 2.2 "Understanding of the Data" that 1,980 of the 7,554 incidents in the log remain uncompleted.

2. **Interpretability Issues**:

Assessment	Score
Overall Completeness Score	**75%**
C1 Missing Values	100%
C2 Incomplete Traces	0%
C6 Orphan Events	100%
C7 Disordered Traces	100%
Overall Accuracy Score	**100%**
A1 Unrealistic Timestamps	100%
A3-1 Trace Duplicates	100%
A3-2 Event Duplicates	100%
Overall Interpretability Score	**60%**
I1-1 Coarse Timestamps	100%
I1-2 Coarse Resources	100%
I1-3 Coarse Activity Names	0%
Overall Relevancy Score	**100%**
R1-1 Noise Events	100%
Overall Validity Score	**100%**
V1-1 Valid Key Name	100%
V1-2 Unique Attribute Keys	100%
V2-1 Data Type to Value Conformance	100%
V3-1 Allowed Data Types	100%
V4-1 Consistent Timestamp Format	100%
V4-2 Logically Valid Timestamps	100%

Table 6.3.: Assessment Results of the BPI Challenge 2013 Log [13]

- I1-3 Coarse Activity Names (0%): The analysis of activity name coarseness once again suggests that the event log lacks proper semantic depth. On average, activity names consist of just one descriptive word, making them overly generic. For example, the activity "Accepted" appears 40,117 times in the log, while others like "Queued" or "Completed" are also simple, one-word verbs or adjectives. These terms are so broad that they could apply to almost any scenario, making it difficult for someone external to the process to fully understand what is happening based solely on these activity names.

Potential anomalies assessed in C4 Unrecorded Traces indicate a not surprising level of process variance, typical for a dynamic system. The top three traces, with event counts of 123, 116, and 90 (in traces 1-687082195, 1-660915327 and 1-627973205), are not surprising given the nature of the process. However, C5 Unrecorded Events, which focuses on analyzing inter-event time gaps, reveals some noteworthy insights.

1. **C5 Unrecorded Events**:
   - Several traces exhibit an unusually long time gap of 8-9 months between events

(e.g., trace 1-506071646). Upon further investigation, no clear reason for this anomaly could be identified. The only common factor among these traces is that they were handled by the org:group "V32 2nd" or "V37 2nd," indicating second-level support involvement. It's possible that these gaps could be due to the resolution of a major production bug or the initiation of efforts to close long-running incidents. However, these are speculative explanations, and it is recommended that these traces be further analyzed within the organization for more concrete answers.

6.3. Evaluation with Industry-Provided Real-World Data Sets

Lastly, we analyze real-world data provided by an Austrian IT Service Provider and data center facility. This event log captures the enterprise's incident management process throughout 2015 and includes all reported incidents within that period. The log comprises 15,034 cases and a total of 218,792 events, detailing the steps taken in response to incidents. The process itself involves 37 different activities, encompassing both automatic tasks, such as bridging to other Information Technology Service Management (ITSM) systems, and manual tasks like incident analysis and resolution. The log also records the involvement of 482 resources responsible for handling various tasks within the incident management lifecycle. Key attributes in the log include the traceID, timestamp, resource, activity, and a detailed event description field. However, we encountered some challenges with the description field during the data conversion, which we will explore in greater detail later in this section.

We received the event log in CSV format, which necessitated converting the data into XES format to facilitate the quality assessment. This conversion process required the development of a custom script to address certain issues, particularly the non-standard timestamp format and the handling of commas within the description field. To ensure compatibility with XES, the timestamps had to be converted to the ISO 8601 standard, making them interpretable by the quality analysis tools in our dashboard.

The description field posed additional challenges, as it contained commas that were not properly enclosed in quotation marks. Since commas are used as delimiters in CSV files, these instances disrupted the file's structure by splitting the description into multiple fields. To resolve this, we opted to remove any text after the first comma in the description field to prevent further misinterpretation, as the additional information was causing misalignment with the expected data structure of five fields.

Despite these necessary adjustments, the core attributes—traceID, timestamp, resource, and activity—remained intact. This allowed us to successfully convert the log and proceed with a thorough quality assessment using our dashboard, which will be discussed in the following.

Assessment	Score
Overall Completeness Score	**67.18%**
C1 Missing Values	68.71%
C2 Incomplete Traces	100%
C6 Orphan Events	100%
C7 Disordered Traces	0%
Overall Accuracy Score	**99.6%**
A1 Unrealistic Timestamps	100%
A3-1 Trace Duplicates	99.93%
A3-2 Event Duplicates	99.08%
Overall Interpretability Score	**60%**
I1-1 Coarse Timestamps	100%
I1-2 Coarse Resources	100%
I1-3 Coarse Activity Names	0%
Overall Relevancy Score	**100%**
R1-1 Noise Events	100%
Overall Validity Score	**100%**
V1-1 Valid Key Name	100%
V1-2 Unique Attribute Keys	100%
V2-1 Data Type to Value Conformance	100%
V3-1 Allowed Data Types	100%
V4-1 Consistent Timestamp Format	100%
V4-2 Logically Valid Timestamps	100%

Table 6.4.: Assessment Results of the real-world Incident Management Event Log

The dashboard uncovered significant issues in both the completeness and interpretability dimensions of the log, indicating notable data quality concerns. There was also a slight deviation observed in the accuracy dimension, though it was less pronounced. No problems were found in the relevancy and validity dimensions, suggesting that the log maintains a high level of conformance in these areas.

1. **Completeness Issues**:
   - C1 Missing Values (68.71%): The dashboard identifies an alarming number of missing values, particularly in the "description" attribute. Upon closer examination, it was found that this frequently occurs when an incident ticket reaches its final state of being resolved and subsequently transitioned to the "closed" state. As no new information is typically added during this transition, the occurrence of "null" in the description field can be considered non-critical and does not indicate a harmful data quality issue.
   - C7 Disordered Traces (0%): All events in the provided event log are recorded in reverse chronological order within the traces—meaning that the first event

based on time appears last in terms of position, while newly logged events are recorded at the top of a case instead of the bottom. This is a crucial realization because it necessitates sorting the events within each trace by timestamp before conducting any process mining, particularly when evaluating the time or control-flow perspectives. Without this correction, analyses could lead to inaccurate insights.

2. **Interpretability Issues**:
   - I1-3 Coarse Activity Names (0%): Once again, we observe issues related to coarse activity names. The most frequent activity, "Phase Change," occurs 62,990 times, and with an average of 1.6 descriptive words per activity, the dashboard flags it as too coarse. Activities such as "Open," "Resolved," "Assignment," or "Update," while likely comprehensible to domain experts within the field, lack the semantic depth necessary for external analysts to fully understand the process. Simply adding a noun like "Incident" would significantly increase the clarity and semantic depth of these activity names, improving interpretability.
3. **Accuracy Issues**:
   - A3-1 Trace Duplicates (99.93%): The dashboard identified two traces with identical event sequences and event attributes. Given the size of the log, this finding is neither shocking nor concerning.
   - A3-2 Event Duplicates (99.08%): The dashboard identifies several event duplicates in the event log. Upon closer inspection, we find that these duplicates primarily occur within the activities "Communication with Vendor" and "Attachment Added," with the majority concentrated in the "Communication with Vendor" activity. This pattern suggests a potential misconfiguration of the ticket bridge used to forward tickets to external vendors for resolution. The repeated logging of these events might reflect a technical issue in how these interactions are recorded.

The successful assessment of real-world event logs demonstrates that the dashboard can effectively identify data quality issues, providing valuable insights to process analysts conducting process mining. By detecting anomalies and potential data inconsistencies, the dashboard ensures that the data used for analysis is reliable, ultimately enhancing the accuracy and depth of the process analysis. This proves that the dashboard is a beneficial tool for analysts aiming to ensure high-quality event logs and improve process mining outcomes.

7. Conclusion

7.1. Summary of results

In this section, we revisit the core research questions outlined in the introduction, structuring the summary of results around each question to provide a comprehensive overview of the insights gained through this research. By addressing each question in turn, this summary connects the theoretical and practical findings with the initial goals of the thesis, guiding the reader through the main contributions of the study.

1. **What are possible data quality issues of event logs?** Data quality issues in event logs primarily stem from dimensions such as completeness, accuracy, interpretability, relevancy, and validity. These issues can manifest as missing data, inconsistent event sequences, incorrect timestamps, or unclear labels, all of which impact process mining outcomes by potentially introducing misleading insights or failing to represent the actual process. Completeness, for example, is critical in ensuring all necessary events are captured, while accuracy and validity check that events are correctly sequenced and conform to expected formats.
   References: For an in-depth discussion, see Chapter 3 (Literature Review), specifically Section 3.3 which covers data quality specific to process mining, and Chapter 4 where the Data Quality Assessment Framework (DQAF) identifies and categorizes these issues.

2. **How can a concept for the identification of data quality issues be specified?** The Data Quality Assessment Framework (DQAF) in this thesis provides a structured approach for identifying data quality issues, mapping general data quality dimensions to those required in process mining. The DQAF specifies indicators and assessment measures across dimensions like completeness, accuracy, validity, interpretability, and relevancy, which are necessary to detect and categorize issues in event logs. This framework enables the structured identification of data quality issues by defining criteria and measurable indicators that can highlight potential data deficiencies.
   References: Refer to Chapter 4 (Data Quality Assessment Framework), especially Sections 4.1 to 4.3, where data quality dimensions and indicators are mapped to specific assessment criteria.

3. **How can a dashboard that showcases identified data quality issues be implemented?** The Event Data Quality Dashboard (EDQD) was implemented to operationalize the DQAF, providing a user-friendly interface for visualizing data

quality issues within event logs. The dashboard's software architecture incorporates design decisions aimed at representing data quality dimensions in an accessible manner, enabling analysts to quickly assess log quality. It allows users to view quality assessment results, configure thresholds, and examine detailed information on specific issues.
References: See Chapter 5 (Event Data Quality Dashboard), particularly Sections 5.2 and 5.3, which cover the design, architecture, and user interface, detailing how the dashboard incorporates and displays the assessments based on data quality dimensions.

4. **How to evaluate the dashboard with a use case study?** The EDQD was evaluated through a three-stage approach: using synthetic data, publicly available event logs, and industry-provided real-world datasets from an Austrian IT service provider. Synthetic data allowed for isolated testing of individual quality issues, while public and real-world data provided insights into how the dashboard performs in varied, complex scenarios. This comprehensive evaluation demonstrates the dashboard's capability to detect issues across different contexts and confirms its practical applicability for process analysts.
References: Chapter 6 (Results) provides a detailed account of the evaluation process, with Section 6.1 focusing on synthetic data sets, Section 6.2 on public datasets, and Section 6.3 on real-world data, illustrating the dashboard's diagnostic utility across different log structures and domains.

By answering each question, this thesis contributes valuable insights into data quality assessment in process mining and offers practical tools to support analysts in ensuring event log quality. Through the systematic development, implementation, and testing of the Event Data Quality Dashboard, this work underscores the importance of addressing data quality in process analysis and provides a robust foundation for future research and applications.

7.2. Discussion

In the discussion section, we would like to draw attention to several important topics that reflect the strategic decisions made during the development of the Event Data Quality Dashboard and its assessments.

Firstly, there is the consideration of how specific indicators were grouped into broader quality dimensions. For example, the assessment of *A3-2 Duplicate Events* could arguably fit better in a dimension dedicated to uniqueness rather than accuracy. Similarly, *V4-1 Consistent Timestamps* might belong in a separate consistency dimension. We recognize and acknowledge these perspectives, and to a degree, we agree that such categorization adjustments could be valid. However, our primary focus was to design a dashboard that is clean, easy to understand, and focused on delivering actionable insights. This approach of grouping assessments into broader, well-established dimensions aligns with the existing

literature but also ensures logical cohesion and simplicity in interpreting the results. We believe this decision, although debatable in certain cases, offers a user-friendly experience without diluting the significance of each quality check.

Secondly, the design of the synthetic data sets presented its own set of challenges. Striking the right balance between creating logs that were almost pristine, yet injected with specific data quality issues, was essential. On the one hand, having a clean event log with minimal variance ensured that when we introduced an issue, it would stand out clearly—both to the dashboard and to the reader analyzing the test data. While this level of homogeneity is often unrealistic for real-world data, it was a necessary step to verify that the individual indicators functioned correctly. To mitigate this concern, we created two to three different test logs for each indicator to add complexity and demonstrate that the dashboard could identify issues even when they were less apparent or scattered within more complex logs. This method provided a balance between ensuring clarity for testing and maintaining relevance to real-world data.

Lastly, the granularity of the assessment results was a significant point of deliberation. Deciding how much detail to present to the analyst for each issue discovered was not straightforward. For instance, in the case of duplicate events within a trace, our initial approach was to flag the trace containing the duplicates. However, for long traces, this required the analyst to manually search through potentially hundreds of events to locate the problem. In response, we enhanced the feedback by including not just the trace name but also the event name where the issue occurred. This change greatly improved usability, enabling analysts to quickly locate and address the problem. However, providing too much detail can also be overwhelming, particularly in large logs where excessive information may become burdensome. The key challenge was thus finding the right balance between offering precise, actionable details and maintaining a high-level overview that remains manageable and interpretable for the user.

7.3. Limitations

The main limitation of the dashboard is its focus on the XES file format. This restricts its use to event logs already structured in XES, making it less applicable to logs stored in other common formats like CSV or JSON. Expanding support to these formats would increase the tool's versatility across different systems and industries.

Additionally, while the dashboard has been tested on large event logs, scalability might become an issue when handling extremely large logs containing millions of events. Future work should focus on optimizing performance for such large datasets.

Finally, the dashboard's generic assessments may not fully address data quality issues that are unique to certain industries. Customizing assessments for specific domains would improve its relevance and effectiveness in those contexts.

7.4. Future work

In the future, we envision expanding the Event Data Quality Dashboard in several key areas. First, incorporating more indicators would improve the comprehensiveness of the assessments. This includes adding checks that address more nuanced data quality issues, such as detecting patterns of missing data in complex traces or identifying subtle anomalies that may not be captured by existing indicators.

Another major improvement would be the ability to suggest data cleaning steps based on the issues found during the assessment. The dashboard could provide actionable recommendations, such as removing duplicates, correcting timestamps, or flagging inconsistent data for review. This would not only highlight issues but also assist users in improving the overall quality of their event logs more efficiently.

Lastly, a domain-specific design could further enhance the tool's relevance. Customizing the assessments to meet the unique needs of different industries would ensure that the dashboard remains adaptable and useful in a variety of contexts. By tailoring indicators and recommendations to specific sectors like healthcare, finance, or manufacturing, we can ensure more meaningful and practical data quality evaluations.

Bibliography

[1] Leo L Pipino, Yang W Lee, and Richard Y Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218, 2002.

[2] Nicola Askham, Denise Cook, Martin Doyle, Helen Fereday, Mike Gibson, Ulrich Landbeck, Rob Lee, Chris Maynard, Gary Palmer, and Julian Schwarzenbach. The six primary dimensions for data quality assessment. *DAMA UK working group*, 2013. `https://www.sbctc.edu/resources/documents/colleges-staff/commissions-councils/dgc/data-quality-deminsions.pdf/`.

[3] Ismael Caballero, Eugenio Verbo, Coral Calero, and Mario Piattini. A data quality measurement information model based on iso/iec 15939. In *ICIQ*, pages 393–408. Cambridge, MA, 2007.

[4] David Dufty, Helene Bérard, Sylvie Lefranc, and Marina Signore. A suggested framework for the quality of big data. *UNECE big data quality task team*, 2014.

[5] Richard Y Wang, Veda C Storey, and Christopher P Firth. A framework for analysis of data quality research. *IEEE transactions on knowledge and data engineering*, 7(4):623–640, 1995.

[6] Fatimah Sidi, Payam Hassany Shariat Panahy, Lilly Suriani Affendey, Marzanah A Jabar, Hamidah Ibrahim, and Aida Mustapha. Data quality: A survey of data quality dimensions. In *2012 International Conference on Information Retrieval & Knowledge Management*, pages 300–304. IEEE, 2012.

[7] RP Jagadeesh Chandra Bose, Ronny S Mans, and Wil MP van der Aalst. Wanna improve process mining results? In *2013 IEEE symposium on computational intelligence and data mining (CIDM)*, pages 127–134. IEEE, 2013.

[8] Wil van der Aalst, Arya Adriansyah, Ana Karla Alves de Medeiros, Franco Arcieri, Thomas Baier, Tobias Blickle, Jagadeesh Chandra Bose, Peter van den Brand, Ronald Brandtjen, Joos Buijs, et al. Process mining manifesto. In *International conference on business process management*, pages 169–194. Springer, 2011.

[9] Wil MP Van der Aalst. Extracting event data from databases to unleash process mining. In *BPM-Driving innovation in a digital world*, pages 105–128. Springer, 2015.

[10] W.M.P. Van der Aalst. *Process mining : discovery, conformance and enhancement of business processes*. Springer, Germany, 2011.

[11] Boudewijn van Dongen. Real-life event logs - hospital log (version 1) [data set], 2011. Eindhoven University of Technology. doi:10.4121/UUID:D9769F3D-0AB0-4FB8-803B-0D1120FFCF54.

[12] Boudewijn van Dongen. Bpi challenge 2012 (version 1) [data set], 2012. Eindhoven University of Technology. doi:10.4121/UUID:3926DB30-F712-4394-AEBC-75976070E91F.

[13] Ward Steeman. Bpi challenge 2013, incidents (version 1) [data set], 2013. Eindhoven University of Technology. 10.4121/uuid:500573e6-accc-4b0c-9576-aa5468b10cee.

[14] Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A Reijers, et al. *Fundamentals of business process management*, volume 1. Springer, 2013.

[15] Christian W Gunther and HMW Verbeek. Xes-standard definition. 2014.

[16] W.M.P. Van der Aalst. *Process mining: data science in action*. Springer, 2016.

[17] Wil MP van der Aalst. What makes a good process model? lessons learned from process mining. *Software & Systems Modeling*, 11(4):557–569, 2012.

[18] Mathias Weske. *Business process management architectures*. Springer, 2007.

[19] Ulf K Teichgräber and Maximilian De Bucourt. Applying value stream mapping techniques to eliminate non-value-added waste for the procurement of endovascular stents. *European journal of radiology*, 81(1):e47–e52, 2012.

[20] Philip Sedgwick and Nan Greenwood. Understanding the hawthorne effect. *Bmj*, 351, 2015.

[21] Soumen Chakrabarti, Martin Ester, Usama Fayyad, Johannes Gehrke, Jiawei Han, Shinichi Morishita, Gregory Piatetsky-Shapiro, and Wei Wang. Data mining curriculum: A proposal (version 1.0). *Intensive working group of ACM SIGKDD curriculum committee*, 140:1–10, 2006.

[22] Usama Fayyad and Paul Stolorz. Data mining and kdd: Promise and challenges. *Future generation computer systems*, 13(2-3):99–115, 1997.

[23] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996.

[24] Manpreet Kaur and Shivani Kang. Market basket analysis: Identify the changing trends of market data using association rule mining. *Procedia computer science*, 85:78–85, 2016.

[25] Ronny S Mans, MH Schonenberg, Minseok Song, W.M.P. Van der Aalst, and Piet JM Bakker. Application of process mining in healthcare–a case study in a dutch hospital. In *International joint conference on biomedical engineering systems and technologies*, pages 425–438. Springer, 2008.

[26] Álvaro Rebuge and Diogo R Ferreira. Business process analysis in healthcare environments: A methodology based on process mining. *Information systems*, 37(2):99–116, 2012.

[27] W.M.P. Van Der Aalst, Hajo A Reijers, Anton JMM Weijters, Boudewijn F van Dongen, AK Alves De Medeiros, Minseok Song, and HMW Verbeek. Business process mining: An industrial application. *Information Systems*, 32(5):713–732, 2007.

[28] Michael Werner and Nick Gehrke. Multilevel process mining for financial audits. *IEEE Transactions on Services Computing*, 8(6):820–832, 2015.

[29] Musie Kebede and M Dumas. Comparative evaluation of process mining tools. *University of Tartu*, 2015.

[30] Hannah Snyder. Literature review as a research methodology: An overview and guidelines. *Journal of business research*, 104:333–339, 2019.

[31] Seref Sagiroglu and Duygu Sinanc. Big data: A review. In *2013 international conference on collaboration technologies and systems (CTS)*, pages 42–47. IEEE, 2013.

[32] Thomas C Redman. *Data quality for the information age*. Artech House, Inc., 1997.

[33] Danette McGilvray. *Executing data quality projects: Ten steps to quality data and trusted information (TM)*. Academic Press, 2021.

[34] Mitra Heravizadeh, Jan Mendling, and Michael Rosemann. Dimensions of business processes quality (qobp). In *Business Process Management Workshops: BPM 2008 International Workshops, Milano, Italy, September 1-4, 2008. Revised Papers 6*, pages 80–91. Springer, 2009.

[35] Carlo Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino. Methodologies for data quality assessment and improvement. *ACM computing surveys (CSUR)*, 41(3):1–52, 2009.

[36] Shirlee-ann Knight and Janice Burn. Developing a framework for assessing information quality on the world wide web. *Informing Science*, 8:159–172, 2005.

[37] Richard Y Wang and Diane M Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4):5–33, 1996.

[38] Verónika Peralta. *Data quality evaluation in data integration systems*. PhD thesis, Université de Versailles-Saint Quentin en Yvelines; Université de la . . . , 2006.

[39] Sakda Loetpipatwanich and Preecha Vichitthamaros. Sakdas: a python package for data profiling and data quality auditing. In *2020 1st International Conference on Big Data Analytics and Practices (IBDAP)*, pages 1–4. IEEE, 2020.

[40] RP Jagadeesh Chandra Bose and Wil MP van der Aalst. Analysis of patient treatment procedures. In *Business Process Management Workshops (1)*, volume 99, pages 165–166, 2011.

[41] Arya Adriansyah and Joos CAM Buijs. Mining process performance from event logs. In *Business Process Management Workshops: BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers 10*, pages 217–218. Springer, 2013.

[42] Chang Jae Kang, Young Sik Kang, Yeong Shin Lee, Seonkyu Noh, Hyeong Cheol Kim, Woo Cheol Lim, Juhee Kim, and Regina Hong. Process mining-based understanding and analysis of volvo it's incident and problem management processes. *BPIC@ BPM*, 85, 2013.

Acronyms

BPIC Business Process Intelligence Challenge. iii, viii, 2, 83, 86, 87, 89, 91

BPM Business Process Management. xiii, 1, 5–7, 9, 34

BPMN Business Process Model and Notation. xiii, 6, 7

BPMS Business Process Management System. 7

CRM Customer Relationship Management. 34

DAMA Data Management Association. vii, xi, 23, 24, 43–45

DQ Data Quality. iii, 21–24, 27, 28, 31, 37–39

DQAF Data Quality Assessment Framework. iii, v, 3, 45, 46, 97

EDQD Event Data Quality Dashboard. i, iii, v, viii, 3, 57–63, 65, 67, 81, 83, 97, 98, 100, 107

ERP Enterprise Resource Planning. 15, 34

IMF International Monetary Fund. 45

ITSM Information Technology Service Management. 1, 93

KDD Knowledge Discovery in Databases. 9

XES eXtensible Event Stream. 17, 18, 50, 57–59, 61, 63, 79, 80, 93, 99

A. Appendix

A.1. Deep-dive into the Comparison between envisioned Measures and actually implemented Assessments

In this section, we will further elaborate on and describe the dependencies between the measures outlined in the framework chapter and the assessments presented in the dashboard chapter. We will also reference relevant literature that addresses the underlying data quality issues, providing a scholarly context for each measure. Additionally, this section will detail the rationale behind why certain assessments were implemented, while others were omitted, offering insights into the practical constraints and decisions that shaped the final implementation.

- **Challenges** faced during implementation, such as the difficulty of assessing specific data quality aspects in an automated manner, and why some envisioned measures could not be included in the final implementation.
- **Newly identified assessments** that were not originally part of the conceptual framework but were introduced during development to better address practical concerns and real-world scenarios.
- **Refinements and trade-offs** made during the development process, including why certain measures were simplified or redefined to fit within the practical constraints of the dashboard architecture.

This comparison will provide valuable information on the evolution of the Event Data Quality Dashboard (EDQD), highlighting the iterative nature of development and the balance between conceptual ideals and practical realities. By understanding the gap between what was originally planned and what was ultimately implemented, we can better appreciate the complexities involved in building data quality assessment tools, as well as the compromises that are sometimes necessary to deliver a functional, scalable and useful solution.

Completeness			
Measure from 4.2	**Assessment in 5.7**	**Reference**	**Rational**
Measure 1: Percentage of not recorded traces versus all recorded traces	C4 Unrecorded Traces	[7] in Table I, issue I1	Measure implemented as envisioned.
Measure 2: Percentage of missing events versus all documented events	C5 Unrecorded Events	[7] in Table I, issue I2	Measure implemented as envisioned.
Measure 3: Percentage of not associated events and traces versus all event - traces pairs that are associated	C6 Orphan Events	[7] in Table I, issue I3	Measure implemented as envisioned.
Measure 4: Percentage of missing trace attributes versus all recorded trace attributes	C1 Missing Values	[7] in Table I, issue I4	Measure implemented as envisioned.
Measure 5: Percentage of not ordered events in a trace versus all ordered events in all traces	C7 Disordered Traces	[7] in Table I, issue I5	Measure implemented as envisioned.
Measure 6: Percentage of missing activity names of events versus all recorded activity names of events	C1 Missing Values (missing concept:name in event attributes)	[7] in Table I, issue I6	Measure implemented as envisioned.
Measure 7: Percentage of missing timestamps versus all documented timestamps	C1 Missing Values (missing time:timestamp in event attributes)	[7] in Table I, issue I7	Measure implemented as envisioned.
Measure 8: Percentage of missing resources to activities versus all recorded resources	C1 Missing Values (missing org:resource in event attributes)	[7] in Table I, issue I8	Measure implemented as envisioned.

Measure 9: Percentage of not recorded event attributes versus all recorded event attributes	C1 Missing Values	[7] in Table I, issue I9	Measure implemented as envisioned.
Measure 10: Verify that each trace with a start event also possesses a corresponding end event	C2 Incomplete Traces	[8] in Chapter 3, Section 3.1, GP1 and Chapter 4, Section 4.1, C1	Measure implemented as envisioned.

Table A.1.: Comparison between envisioned Measures and actually implemented Assessments of Completeness

Accuracy			
Measure from 4.3	**Assessment in 5.6**	**Reference**	**Rational**
Measure 1: Number of traces that belong to a different process	A2 Trace belongs to a different process:	[7] in Table I, issue I10	Measure implemented as envisioned.
Measure 2: Number of fictitious events that were never actually executed	A3-2 Event Duplicates	[7] in Table I, issue I11	While this measure could be fully addressed by comparing the event log to an external, trusted documentation of real events, this would require additional user input and data, posing challenges in format compatibility and user effort. Therefore, the implemented assessment A3-2 Event Duplicates pragmatically captures the essence of this measure by identifying duplicate events within the log, which are indicative of events that never occurred in real life. While this approach may not cover every aspect of fictitious events, it provides a feasible and valuable solution without overburdening the user.

Measure 3: Number of wrong associations between traces and events	-	[7] in Table I, issue I12	This measure was not implemented due to the complexity it introduces for the user. To accurately assess this, the user would need to provide a detailed file containing the correct trace-to-event associations, which adds a significant burden in terms of both data preparation and format compatibility. Since this level of input goes beyond what most users are likely able or willing to provide, it was deemed impractical to include this measure in the implementation. As a result, it was excluded in favor of more feasible assessments that do not rely on external documentation.
Measure 4: Number of false trace attributes	-	[7] in Table I, issue I13	This measure was not implemented for similar reasons. In large event logs with thousands of traces and potentially dozens of attributes, requiring users to provide a detailed list of allowed attributes would be an overwhelming task, making this approach impractical and cumbersome for most users.
Measure 5: Amount of events in a mixed up order	C7 Disordered Traces	[7] in Table I, issue I14	Is actually assessed through C7 Disordered Traces in the completeness dimension. This decision was made because event order is closely tied to the structural integrity of traces, and addressing this issue within completeness better reflects the goal of ensuring that traces follow the correct sequence of events, rather than focusing on data accuracy alone.
Measure 6: Number of incorrect activity names	-	[7] in Table I, issue I15	This measure was not implemented due to the significant complexity and effort required from users to supply a comprehensive list of allowed activity names. Given the potential for large event logs with numerous unique activities, this would place an unreasonable burden on the user to maintain such a reference list.
Measure 7: Number of erroneous timestamps	A1 Unrealistic Timestamps	[7] in Table I, issue I16	Measure implemented as envisioned.

Measure 8: Number of invalid resource to event association	-	[7] in Table I, issue I17	This measure was not implemented due to the effort it introduces for the user. To accurately assess this, the user would need to provide a detailed file containing the correct and allowed resource-to-event associations.
Measure 9: Number of falsely recorded event attributes	-	[7] in Table I, issue I18	This measure was not implemented for the same reasons.
New Measure: Number of fictitious traces that were never actually executed	A3-1 Trace Duplicates	-	This measure was identified during implementation. It mirrors the logic of Measure 2: Number of fictitious events that were never actually executed, but instead focuses on entire traces. This measure addresses instances where entire traces appear in the event log but were never carried out in reality, enhancing the detection of inaccuracies at a broader level than individual events.

Table A.2.: Comparison between envisioned Measures and actually implemented Assessments of Accurcy

Interpretability			
Measure from 4.5	**Assessment in 5.8**	**Reference**	**Rational**
Measure 1: Number of events that can not be correlated to specific traces as they are documented to coarsely	-	[7] in Table I, issue I19	This measure was not implemented because no dataset was found that exhibited this specific issue. Without real-world examples, it was unclear how such an event log would manifest, making it challenging to design an accurate assessment for this measure.

Measure 2: Attributes (trace and event) that are too coarse	-	[7] in Table I, issue I20 and I25	This measure could not be implemented on a generic level due to the need for domain-specific knowledge. For instance, while country information may be too coarse for a logistics company delivering goods, it could suffice for an international marketing agency analyzing customer locations. Without understanding the specific context of each dataset, it's impossible to determine if an attribute's granularity is appropriate.
Measure 3: Concurrent events that have been totally ordered	A3-2 Event Duplicates and C7 Disordered Traces	[7] in Table I, issue I21	This measure is partly addressed by A3-2 Event Duplicates in the accuracy dimension and C7 Disordered Traces in the completeness dimension. These assessments already provide sufficient indications of potential concurrency issues, so no further specific measures were deemed necessary.
Measure 4: Activity names that have weak semantics (named too coarsely)	I1-3 Coarse Activity Names	[7] in Table I, issue I22 and [8] in Chapter 3, Section 3.1, GP1 and Chapter 4, Section 4.1, C1	Measure implemented as envisioned.
Measure 5: Timestamps that only capture the day an event occurred, without information of the hour or minute it happened	I1-1 Coarse Timestamps	[7] in Table I, issue I23 and [8] Chapter 4, Section 4.1, C1	easure implemented as envisioned.

Measure 6: Resources that just represent roles or departments instead of concrete persons or machines	I1-2 Coarse Resources	[7] in Table I, issue I24 and [8] Chapter 4, Section 4.1, C1	Measure implemented as envisioned.

Table A.3.: Comparison between envisioned Measures and actually implemented Assessments of Interpretability

Relevancy			
Measure from 4.6	**Assessment in 5.9**	**Reference**	**Rational**
Measure 1: Number of irrelevant traces within the log	-	[7] in Table I, issue I26	This measure was not implemented because we opted to focus on detecting noise events instead. Noise events, such as system information logs, provide insight into system status. If entire traces consist solely of such logs, they could be relevant in IT-specific event logs. Thus, assessing irrelevant traces was deemed unnecessary, as it might misinterpret valid industry-specific logs.
Measure 2: Number of irrelevant events, as they are duplicates or just system status information, within the log	R1-1 Noise Events and A3-2 Event Duplicates	[7] in Table I, issue I27 and [8] Chapter 4, Section 4.1, C1	This measure is partially assessed by A3-2 Event Duplicates in the accuracy dimension for duplicate events, while system status information is covered by R1-1 Noise Events.

Table A.4.: Comparison between envisioned Measures and actually implemented Assessments of Relevancy

Validity			
Measure from 4.4	**Assessment in 5.10**	**Reference**	**Rational**

Measure 1: Check all values of the event log against the standards specified by XES	V1-1 Valid Key Name and V1-2 Unique Attribute Keys and V3-1 Allowed Data Types	[15] Chapter 2, Section 2.2	Measure implemented as envisioned.
Measure 2: Check if predefined data types actually hold true	V2-1 Data Type to Value Conformance	-	Measure implemented as envisioned.
Measure 3: Check if the syntax of data elements does not change over time	V4-1 Consistent Timestamp Format	[1] in Table I, Consistent Representation	Measure implemented as envisioned.
New Measure: Check if timestamps follow logical constraints, such as valid days, hours, and months (e.g., no 25th hour or 45th month)	V4-2 Logically Valid Timestamps	-	This is a new measure discovered during implementation. While it is more of an edge case, it addresses scenarios where timestamps might be manually logged, potentially leading to logically invalid entries.

Table A.5.: Comparison between envisioned Measures and actually implemented Assessments of Validity