Foundations of Process Event Data

Jochen De Weerdt[1(✉)] and Moe Thandar Wynn[2]

[1] KU Leuven, Leuven, Belgium
jochen.deweerdt@kuleuven.be
[2] Queensland University of Technology, Brisbane, Australia

Abstract. Process event data is a fundamental building block for process mining as event logs portray the execution trails of business processes from which knowledge and insights can be extracted. In this Chapter, we discuss the core structure of event logs, in particular the three main requirements in the form of the presence of case IDs, activity labels, and timestamps. Moreover, we introduce fundamental concepts of event log processing and preparation, including data sources, extraction, correlation and abstraction techniques. The chapter is concluded with an imperative section on data quality, arguably the most important determinant of process mining project success.

1 Introduction

This chapter is devoted to a core building block of process mining, namely event data or event logs. The particularities of event logs in comparison to traditional attribute-value data sets used for non-process mining data science and analytics applications, make that dedicated analysis techniques become worthwhile. To put it more concretely, classical data science analyses, e.g. learning a decision tree or running a clustering algorithm, when straightforwardly applied to an event log, will not give you workable results. This is because events in an event log, which can be considered as the observations (rows) in our dataset, are related to each other in terms of time and by means of an overarching case dimension, which, when not taken into account via dedicated analysis techniques, results in useless or biased results. In this chapter, we will first explain and exemplify the fundamental structure of event logs. In addition, we will discuss the most common sources from which event logs can be obtained. Furthermore, we will dive into the data preprocessing pipeline, bringing in the perspectives of event extraction, correlation and abstraction. Finally, given the uphill battle in many organizations in terms of data availability and especially data quality, we close the chapter with a discussion of this theme.

2 The Fundamental Structure of Event Logs

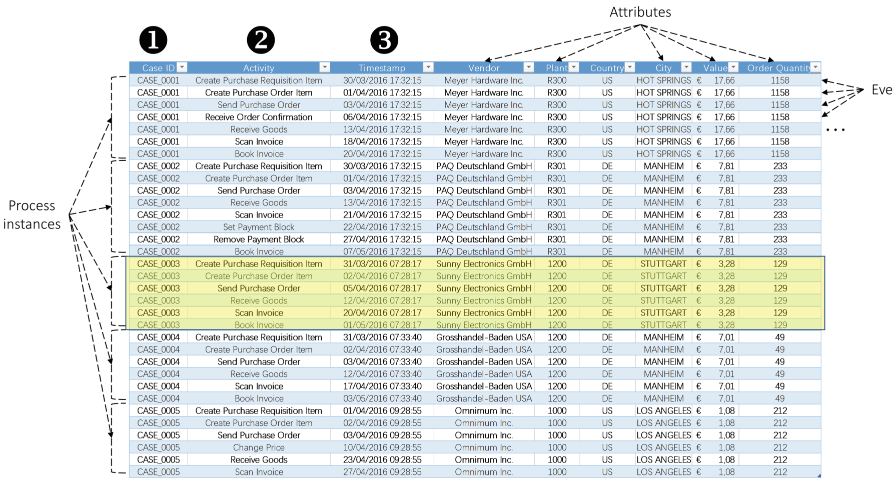
We refer to [3] for the conceptual definition of an event log. Here, we will complement the definition with a more practical view on the essential event log data requirements, an exploration on additional data attributes, an analysis of event types, as well as the link to the XES storage standard.

W. M. P. van der Aalst and J. Carmona (Eds.): Process Mining Handbook, LNBIP 448, pp. 193–211, 2022.
https://doi.org/10.1007/978-3-031-08848-3_6

2.1 Essential Event Log Data Requirements

Figure 1 illustrates an excerpt of an example event log related to a fictitious Purchase-to-Pay (P2P) process. This small excerpt can help to understand the three essential data requirements for event logs to be analysis-ready for process mining technique application. First, each event should be linked to a case or process instance, typically by using a *Case ID*. This is "Requirement 1". In the simple example of Fig. 1, each case or process instance will refer to one procurement of a product or service by an organization with one of its suppliers. Events will be collected for every process instance and will pertain to activities or steps executed within the different stages of the P2P process (e.g. requisitioning, invoicing, reception of goods, etc.).

We thus argue that the presence of a Case ID is an essential requirement for an event log. However, it should be pointed out that Case IDs are not always straightforwardly available. This problem has been addressed in both process mining literature, as well as in practice, and is often referred to as *event correlation*. This topic is addressed in Sect. 3. There also exists research on the direct application of process mining techniques on event data without Case IDs (e.g. [27]), however, this is a rather niche application. Nevertheless, it is important to point out that, in contrast to static event logs, an increasing number of process mining techniques are developed for streams of events. In such event streams, the notion of a CaseID is often even more complicated.



Case ID	Activity	Timestamp	Vendor	Plant	Country	City	Value	Order Quantity
CASE_0001	Create Purchase Requisition Item	30/03/2016 17:32:15	Meyer Hardware Inc.	R300	US	HOT SPRINGS	€ 17,66	1158
CASE_0001	Create Purchase Order Item	01/04/2016 17:32:15	Meyer Hardware Inc.	R300	US	HOT SPRINGS	€ 17,66	1158
CASE_0001	Send Purchase Order	03/04/2016 17:32:15	Meyer Hardware Inc.	R300	US	HOT SPRINGS	€ 17,66	1158
CASE_0001	Receive Order Confirmation	06/04/2016 17:32:15	Meyer Hardware Inc.	R300	US	HOT SPRINGS	€ 17,66	1158
CASE_0001	Receive Goods	13/04/2016 17:32:15	Meyer Hardware Inc.	R300	US	HOT SPRINGS	€ 17,66	1158
CASE_0001	Scan Invoice	18/04/2016 17:32:15	Meyer Hardware Inc.	R300	US	HOT SPRINGS	€ 17,66	1158
CASE_0001	Book Invoice	20/04/2016 17:32:15	Meyer Hardware Inc.	R300	US	HOT SPRINGS	€ 17,66	1158
CASE_0002	Create Purchase Requisition Item	30/03/2016 17:32:15	PAQ Deutschland GmbH	R301	DE	MANHEIM	€ 7,81	233
CASE_0002	Create Purchase Order Item	01/04/2016 17:32:15	PAQ Deutschland GmbH	R301	DE	MANHEIM	€ 7,81	233
CASE_0002	Send Purchase Order	03/04/2016 17:32:15	PAQ Deutschland GmbH	R301	DE	MANHEIM	€ 7,81	233
CASE_0002	Receive Goods	13/04/2016 17:32:15	PAQ Deutschland GmbH	R301	DE	MANHEIM	€ 7,81	233
CASE_0002	Scan Invoice	21/04/2016 17:32:15	PAQ Deutschland GmbH	R301	DE	MANHEIM	€ 7,81	233
CASE_0002	Set Payment Block	22/04/2016 17:32:15	PAQ Deutschland GmbH	R301	DE	MANHEIM	€ 7,81	233
CASE_0002	Remove Payment Block	27/04/2016 17:32:15	PAQ Deutschland GmbH	R301	DE	MANHEIM	€ 7,81	233
CASE_0002	Book Invoice	07/05/2016 17:32:15	PAQ Deutschland GmbH	R301	DE	MANHEIM	€ 7,81	233
CASE_0003	Create Purchase Requisition Item	31/03/2016 07:28:17	Sunny Electronics GmbH	1200	DE	STUTTGART	€ 3,28	129
CASE_0003	Create Purchase Order Item	02/04/2016 07:28:17	Sunny Electronics GmbH	1200	DE	STUTTGART	€ 3,28	129
CASE_0003	Send Purchase Order	05/04/2016 07:28:17	Sunny Electronics GmbH	1200	DE	STUTTGART	€ 3,28	129
CASE_0003	Receive Goods	12/04/2016 07:28:17	Sunny Electronics GmbH	1200	DE	STUTTGART	€ 3,28	129
CASE_0003	Scan Invoice	20/04/2016 07:28:17	Sunny Electronics GmbH	1200	DE	STUTTGART	€ 3,28	129
CASE_0003	Book Invoice	01/05/2016 07:28:17	Sunny Electronics GmbH	1200	DE	STUTTGART	€ 3,28	129
CASE_0004	Create Purchase Requisition Item	31/03/2016 07:33:40	Grosshandel-Baden USA	1200	DE	MANHEIM	€ 7,01	49
CASE_0004	Create Purchase Order Item	02/04/2016 07:33:40	Grosshandel-Baden USA	1200	DE	MANHEIM	€ 7,01	49
CASE_0004	Send Purchase Order	03/04/2016 07:33:40	Grosshandel-Baden USA	1200	DE	MANHEIM	€ 7,01	49
CASE_0004	Receive Goods	12/04/2016 07:33:40	Grosshandel-Baden USA	1200	DE	MANHEIM	€ 7,01	49
CASE_0004	Scan Invoice	17/04/2016 07:33:40	Grosshandel-Baden USA	1200	DE	MANHEIM	€ 7,01	49
CASE_0004	Book Invoice	03/05/2016 07:33:40	Grosshandel-Baden USA	1200	DE	MANHEIM	€ 7,01	49
CASE_0005	Create Purchase Requisition Item	01/04/2016 09:28:55	Omnimum Inc.	1000	US	LOS ANGELES	€ 1,08	212
CASE_0005	Create Purchase Order Item	02/04/2016 09:28:55	Omnimum Inc.	1000	US	LOS ANGELES	€ 1,08	212
CASE_0005	Send Purchase Order	03/04/2016 09:28:55	Omnimum Inc.	1000	US	LOS ANGELES	€ 1,08	212
CASE_0005	Change Price	10/04/2016 09:28:55	Omnimum Inc.	1000	US	LOS ANGELES	€ 1,08	212
CASE_0005	Receive Goods	23/04/2016 09:28:55	Omnimum Inc.	1000	US	LOS ANGELES	€ 1,08	212
CASE_0005	Scan Invoice	27/04/2016 09:28:55	Omnimum Inc.	1000	US	LOS ANGELES	€ 1,08	212

Fig. 1. Example event log from a fictitious P2P process, illustrating the three essential requirements: presence of a case ID, activity label, and timestamp per event.

The second key requirement ("Requirement 2") for event log data is the fact that each event should correspond to an activity executed within the process. More specifically, an assumption is made that there exists a restricted set of labels, reflecting the activities in the business process, to which each event is mapped. In Fig. 1, this is shown in the second column. Given that activity labels are simple strings, there is a lot of freedom to tailor the activity label for the right analysis viewpoint. However, oftentimes, natural log data is stored at lower levels of granularity than desired for analysis purposes. Typically, one would prefer that the granularity level of activities is such that they can be understood and interpreted by business experts. Nonetheless, a lot of event data exists for which the granularity level is much lower. In Sect. 3, we discuss the task of bringing lower level events to a better granularity level, which is referred to as *event abstraction*.

Finally, the last requirement ("Requirement 3") entails that there exists an ordering of the events pertaining to a case. As such, each case logically consists of a sequence of events. Most often, this ordering will be derived from a timestamp attribute. However, this is not strictly mandatory, given that the order could also be derived from the order in which events are recorded in a database or table, insofar this order in which events occurred matched with their factual execution order within a process.

It should be pointed out that, while a Case ID, Activity and Timestamp column are essential requirements in order to be able to conduct process mining analyses, their definition might not always be as clear cut as is the case for the illustrative example. For instance, for many real-life datasets, different choices can be made in terms of using one single or multiple columns to create the activity label, and as such provide a different perspective on the process. A similar effect can also occur for Case IDs, where for instance, with an example from a clinical pathway perspective, the use of a patient ID instead of an admission ID as case identifier, can yield a very different analysis.

2.2 Additional Data Attributes

In addition to the mandatory elements of a Case ID, Activity, and Timestamp, event logs will usually contain several or often many additional attributes (columns). In Fig. 1, the event log contains additional attributes including Vendor, Plan, Country, City, Value and Order Quantity. In our example, the values for these attributes remain constant within a single case, and accordingly can be considered as process instance-level attributes. However, this is not mandatory, as attributes can pertain to events or activities, and might be updated throughout the execution of a process instance. For instance, an item number or item type that is recorded when a purchase order item is created is an example of such an event-level attribute.

Additional data attributes can have many purposes, but typically the following three uses are most important. Foremost, these additional data attributes can help to filter cases and events in order to obtain a more focused analysis viewpoint or perform comparative analysis between subsets of process instances. Secondly, these additional data attributes might contain valuable context information, and can therefore be exploited to gain better insights into the process. For instance, a textual comment field in an incident management process could contain essential information regarding the problem at hand, which in turn might impact routing choices, timing, resource allocation, etc. Finally, the availability of additional data attributes, especially information on resources, costs, etc. opens up possibilities for the application of process mining techniques that go beyond process discovery and conformance checking. For instance, organizational mining techniques were developed to focus on resources employed within the process [53]. Moreover, these additional data attributes also play a fundamental role in decision mining [18,47] (see [17]) and predictive process monitoring [19] (see [20]).

2.3 Storing Event Data

Event data is intrinsically simple attribute value data, easily visualized in a two-dimensional table. Nonetheless, unstructured data formats including Excel-files or plain text files, without any form of underlying schema, fail to serve as a proper storage format. This is mainly due to the complex interactions between events, cases, and their attributes. This observation drove the development of the eXtensible Event Stream (XES) standard [1], an IEEE Standards Association-approved language to transport, store, and exchange event data. Its metadata structure is represented in Fig. 2. XES uses the W3C XML Schema definition language, guaranteeing interoperability between various systems. An IEEE XES instance corresponds to a file-based event log or formatted event stream that can be used to transfer event data in a unified manner. In IEEE XES, events are considered as an observed atomic granule of activity. Next to events, IEEE XES specifies the concept of a log, a trace, and an attribute component. Event and/or trace classifiers are used to assign an *identity* to traces and events. The standard does not define a specific set of attributes for events, traces or logs. However, it does allow for *extensions*. An extension can be used to define a set of attributes for events, traces and/or logs. For instance, a common set of attributes can be defined for event logs within a particular application domain. An overview of currently available standard extensions is available on the XES website[1].

2.4 Event Types

To conclude the section on the fundamental structure of event logs, it is important to point to the concept of event types or lifecycle transitions of activities. When sourcing events from many process-aware information systems, events oftentimes relate to the transactional lifecycle that activities undergo. One example of such a transactional lifecycle model is shown in Fig. 3a. This is the transition lifecycle model of the BPMN 2.0 standard[2]. Such a transactional lifecycle model describes the states and state transitions which an activity might take in its execution. Also in IEEE XES, a *lifecycle* extension has been approved, which specifies a default activity lifecycle[3]. This state machine is shown in Fig. 3b.

[1] http://www.xes-standard.org/.

[2] https://www.omg.org/spec/BPMN/2.0/.

[3] http://www.xes-standard.org/.

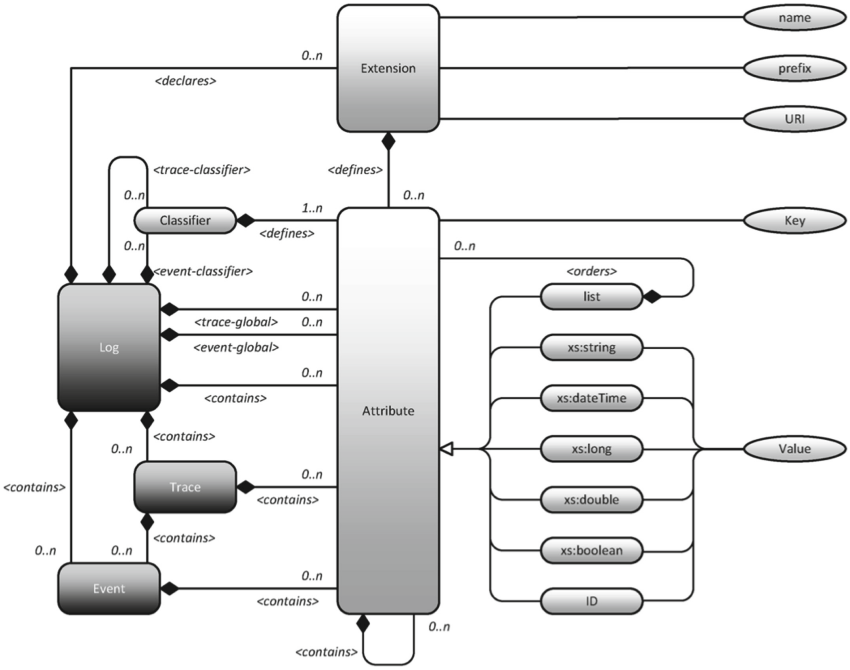
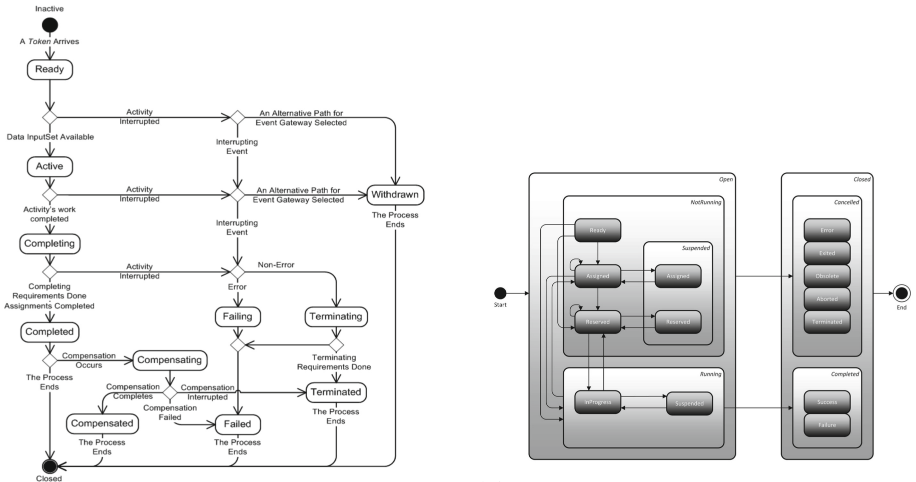


Fig. 2. The IEEE XES metadata structure



(a) The lifecycle of an activity as defined in BPMN 2.0

(b) State machine illustrating the most typical transitions in an activity's lifecycle, according to XES

Fig. 3. Two different activity life cycle models

When retrieving data from process-aware information system, especially from Business Process Management Systems (BPMS) [43], a large collection of event types might be readily available. This is oftentimes not the case in other environments, for instance for web data. In case there are no defined event types, one typically assumes that an event pertaining to the execution of an activity reflects the completion of the activity. In this case, every activity execution is represented by a single event. However, having only a single event per activity execution does not allow to make a distinction between waiting time and execution time of activities. As such, for more fine-grained performance analysis, one would typically prefer two events per activity execution, indicating its start and completion time.

3 Event Log Preprocessing

Data preprocessing is a fundamental part of any data science project. While not as attractive compared to model building or deployment, the preprocessing stage of a project is often most time and effort consuming. Estimates indicate that 80% of resources in typical data science projects is devoted to data preprocessing. One model illustrating the typical data analytics process is depicted in Fig. 4. This model, originally introduced in [25] as the Knowledge Discovery in Databases (KDD) process, reflects the main stages in the execution of a data analytics process. It should be pointed out that this model is an oversimplification of reality, given the frequent and unpredictable iterations that most often occur, rendering the management and completion of a typical data science project usually much more difficult. One notable complexity is the preprocessing of data, usually consisting of data selection, data cleaning, and data transformation.

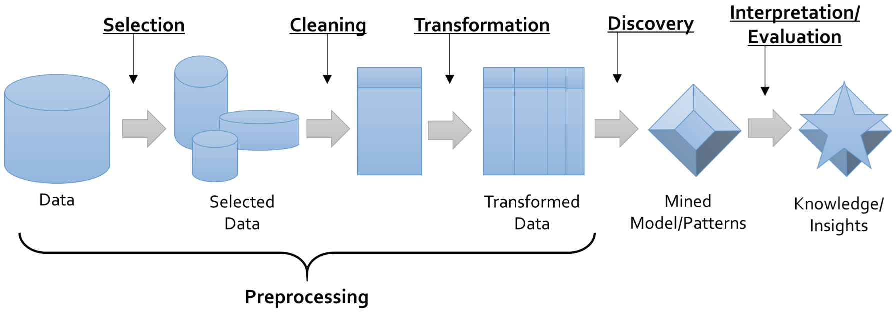


Fig. 4. A representation of the typical stages in a data analytics project [25]

In this part, we want to zoom in on a couple of aspects related to the different stages of a process mining-based analytics project. Most importantly, we want to elaborate on event log data sources, as well as the differences in terms of pipelines between classical data analytics projects and process mining projects.

3.1 Event Log Data Sources

Event data is rapidly becoming an almost untameable beast, given the widespread and drastic increase in availability of such kind of data. In application domains ranging from typical service sector companies including banks and insurers, over manufacturing, to healthcare and education. At system level, we identify the following categorization of most common and important sources for event data:

- **BPMS:** On a scale of most to least process-aware systems, BPMSs most likely rank on top. As such, without exception, event data obtained from these systems is readily available for process mining analysis. Very little or even no data integration is required, and logging is usually executed at the ideal level of granularity.
- **Case management and ticketing sytems:** In line with BPMSs, also case management and ticketing systems natively log timestamped data that is directly useful for process mining. Oftentimes, logs from case management and ticketing systems relate to status changes, so some additional preprocessing might be required to disentangle the true units of work or activity labels.
- **ERP/CRM:** Given their widespread adoption, these enterprise information systems are probably the most important source of event data for modern businesses and organizations. An ERP (Enterprise Resource Planning) system can be seen as a suite of integrated applications for supporting and managing the core business processes. CRM (Customer Relationship Management) systems on the other hand have a dedicated focus on managing all interactions and relationships with customers. By design, ERP systems use shared databases to store relevant business data. As such, and although sometimes a bit more arduous than expected, event log data can be sourced from ERP and CRM systems.
- **Operational databases:** Next to ERP and CRM systems, companies might employ other operational databases supporting their business processes. If these databases have some functionality to store historical data, they can often also serve as a valuable event data source.
- **Project management software:** Applications including popular Hive, Trello, ZOHO, and JIRA support many organizations with managing projects according to a scrum, agile, lean or other fancy project management methodology. When you take an interest into process mining analysis of project management and execution, these systems can provide valuable event data.
- **Data warehouses and data lakes:** Next to operational systems including ERP and CRM, many organizations have a dedicated stack of Business Intelligence (BI) systems and technologies in place. Classical data warehouses are oftentimes a goldmine for process miners. Their hype alternative, allowing for more flexible and unstructured data storage by shifting from schema-on-write to a schema-on-read data management, are referred to as data lakes.
- **Web data:** Website and apps data are another unmistakably important source of event data. From online shopping, gaming, investing, trading, media consumption, to social interaction, online platforms are the main driver of modern B2C business models. With the strong uptake in customer centricity

for business value and competitive advantage creation, customer-centric process mining analysis has strong potential. As such, in addition to CRM data, process mining has a strong interest into event data produced on these online platforms. Please note that, in many cases, including for instance learning environments such as MOOCs, a default standard for web-based platforms to store data is JSON (JavaScript Object Notation).

- **Internet of Things (IoT):** Finally, IoT systems also contain a high potential as source for event data. Sensors and actuators have been deployed widely for all kinds of purposes. Although the granularity gap between typical IoT data (sensor readings) and event data is sometimes challenging to bridge, IoT is becoming a hugely important source of even data in areas such as security, manufacturing, healthcare, and transportation.

It is pointed out that this is not a comprehensive list of all possible event log data sources. In an online survey with 289 participants spanning the roles of practitioners, researchers, software vendors, and end-users, SAP ECC (R/3), SAP S/4 HANA, and Salesforce are selected as the top three most analyzed source systems for process mining analysis [57].

3.2 A Comparison with Classical Analytics Data Preprocessing

While sourcing appropriate data is always the first step in any data preprocessing exercise, it seems reasonable to state that in many situations, analysts could rely on a vast amount of event data sources. This is in line with classical analytics tasks, for which a growth in available data has been observed as well. However, in comparison to classical data preprocessing stages within an analytics process, starker differences exist at the level of cleaning and transforming data.

With respect to data cleaning, where in classical setups, problems including missing values and outliers are a main focus, data cleaning of event logs has received much less scientific and practical attention. A more detailed discussion on data quality for process mining can be found below in Sect. 5. Other differences between a process mining project process and a classical data analytics process are even more notable.

First, at the selection stage, a typical procedure within classical data analytics is to, early-on in the process, divide obtained data into training and test data. Especially when considering predictive analytics, it is of crucial importance to evaluate the true predictive power of learned models by means of independent test data that was not used for training the model. This procedure is rarely seen in process mining, with the exception of some works on predictive process monitoring. One could claim that this is due to the more unsupervised nature of process discovery algorithm, nonetheless, the difference remains striking.

Another essential data preprocessing step for classical data analytics projects relates to transforming the features space such that more valuable features are provided to algorithms for training models. Feature transformation includes techniques such as normalization, grouping and binning. Moreover, advanced feature engineering is also an important but often neglected step to improve model performance. Feature engineering aims at crafting new features based on the original data. The typical data format of event logs, consisting of events pertaining to cases, make that the "rows" in event log are intrinsically correlated. This invalidates the assumption of data being independent and identically distributed (IID). This is a central assumption underpinning about every machine learning technique. However, for process mining, when considering events as the observation level, they are by definition not IID. As such, a large majority of techniques addressing data cleaning and feature transformation including advanced feature engineering, remain purposeless when applied to event data.

When making an assessment of one of the most recently introduced process mining methodologies, i.e. PM^2 [56], four event data preprocessing tasks are defined: (1) creating views, (2) filtering logs, (3) enriching logs, and (4) aggregating events. All these tasks are tailored to the process mining context, and have no immediate corresponding task in a classical data analytics pipeline. For instance, in CRISP-DM [52], data preparation includes selection, cleaning, construction, integration and formatting of data. Several process mining case studies such as the one presented in [6] adapted CRISP-DM to work with healthcare datasets.

In the next Section, we will dive deeper into the problem of event log preparation, which is often extensive and demanding, especially when data for process mining cannot be sourced from process-aware information systems.

4 Event Log Preparation

While possibly not perfectly disjoint, event log preparation often includes three types of techniques: extraction, correlation and abstraction [21]. Figure 5 illustrates the relationship between these types of techniques and fundamental process mining concepts.

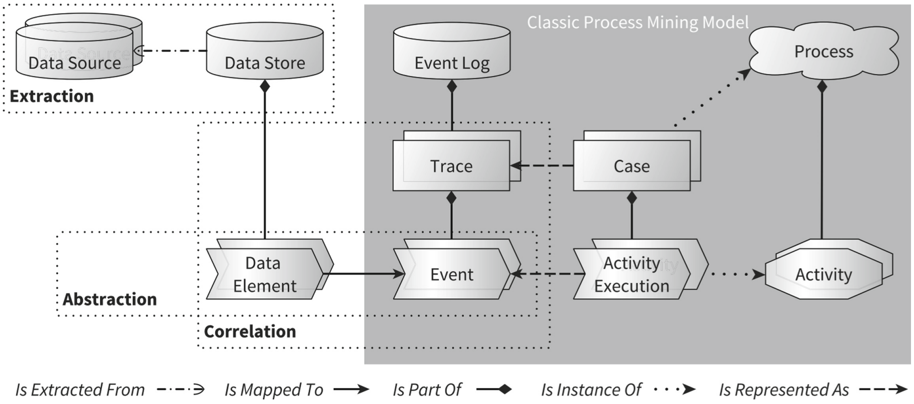


Fig. 5. Event log preparation techniques (extraction, correlation, and abstraction) and their relationship to key process mining concepts [21].

In what follows, we will provide a summary overview of reported tools and techniques for abstraction, correlation and abstraction of event data.

4.1 Extraction of Event Data

Extraction refers to obtaining event data from source systems, most often databases underlying a variety of information systems. Generally, data stored in such databases is not recorded with a process perspective in mind, and therefore will not automatically reflect essential concepts such as events and traces. Accordingly, identification of relevant event data is a primordial challenge. It often requires strong domain knowledge, and despite standardization efforts, often remains prone to ad-hoc solutions.

Two perspectives should be separated when investigating solutions for event data extraction. On the one hand, there is commercial process mining software, where vendors have adopted a clear strategic focus to address the challenges that come with extraction of event logs. Accordingly, a majority of commercial process mining tools comes with software solutions (connectors) that have been developed to allow tapping into all kinds of source systems and databases. Such connectors define how to extract relevant event data from particular source systems and which additional transformations should be applied. As such, these tools promise the holy grail of automating data extraction, a problem addressed in the academic community for over a decade.

One of the first tools stemming from scientific research was the ProM Import Framework [31]. Already in these early days, the idea of an extensible plug-in architecture allowing to develop adapters to hook into a large variety of systems was proposed and partially implemented. With the uptake of XES, XESame was developed as a more flexible successor to the ProM Import Framework. Other researchers have focused on extraction from ERP systems, e.g. the EVS Model Builder [33] and XTract [41], or other operational systems, e.g. Eventifier [46].

Another important stream of research within the realm of event extraction addresses object or artifact centricity. Many source systems, including popular ERP systems, store data at the logical level of objects instead of providing a true process perspective. Oftentimes, assumptions in terms of a desired perspective (definition of case id and activity) are required in order to flatten an object-centered database into a "flat" event log. One noteworthy scientific initiative in this context is ontology-based data access (ODBA) for event log extraction [13, 14]. The approach is based on an ontological view of the domain of interest and linking it as such to a database schema and has been implemented in the Onprom tool. Finally, the recently introduced OCEL standard[4] is another relevant piece of work, putting forward a general standard to interchange object-centric event data with multiple case notions.

The XES survey also uncovered the top tools that are currently being used by the process mining community for the preparing of event logs [57]. There is also ongoing work by the IEEE Task force on reinventing the IEEE XES standard

[4] http://ocel-standard.org/.

to address several identified data related challenges in the XES survey [57], in particular, to capture the semantics of event data and to support complex data structures.

4.2 Correlation of Event Data

Mapping event data extracted from source systems and databases to cases (instances of the business process under investigation) is denoted as correlation. In cases where event data is obtained but Case IDs are missing, a non-trivial process can be started to automatically or semi-automatically generate Case IDs. In a scientific context, several solutions have been proposed, most of them being focused on using additional event data attributes [12,15,42,44,48], sometimes aided by a conceptual model [9,40] or even a process model [8,37].

In practical situations, the problem of correlating event data is probably more related to a variety of non-integrated data sources, which all capture or support part of a business process. As such, an integration of these different sources should be achieved. Hereto, especially when an organizational data warehousing architecture is present, Extract-Transform-Load (ETL) processing would be a default technology to resort to. ETL tools are perfectly equipped to derive and deploy matching schemes to integrate data from non-integrated data sources. Nevertheless, an ETL-approach leading to a data consolidation integration pattern is not the sole option. Increasingly, companies start to focus on the introduction of data virtualization layers in order to realize a more federation-oriented data integration. Data federation can prevent the creation of yet another duplicated database or data store, but instead provides flexible querying and analysis tools for information from multiple source systems as if all data resides within a single integrated database.

4.3 Abstraction of Event Data

Next to extraction and correlation, abstraction is considered as the third prong of the process mining event data preparation trident. In many real-world scenarios, event data is stored at much more fine-grained granularity levels compared to a business-understandable process activity level. As such, abstraction techniques can be considered as mapping techniques that can translate one or more lower-level events into higher-level events pertaining to business process activities. For a detailed taxonomy of event abstraction, we refer the interested reader to [59].

IoT. One particular field of application in which event abstraction is becoming a crucial factor for success is IoT business processes [34]. In IoT, a wide variety of sensors and actuators record contextual observations of a physical environment. These sensor readings or measurements give rise to low-level events, which are intrinsically useful to derive activity-level events from. For instance, in [51], a technique for mapping location-based sensor data to process activities was proposed using so-called *interactions*. Another prominent work in this area is

[23], which relies on clustering of segmented continuous sensor data to derive higher-level activities.

Clustering. Given that event abstraction is a largely unsupervised learning problem in most cases (i.e. unless domain knowledge is used, there is no natural target available), a pretty intuitive way to map lower-level events to coarse-grained events is using clustering. The earliest proposed event abstraction techniques took this perspective, i.e. by clustering sets or sequences of lower-level events, abstraction into higher-level events can be obtained. For instance, in [32], coherent subsequences of events are learned via trace segmentation to create coarse-granular events. Also in [29,45], clustering techniques have been put forward for event abstraction.

Pattern-Based Approaches. Another frequently used paradigm to perform abstraction is pattern matching. The work by Bose and van der Aalst [11] can be considered as origination of pattern-based abstraction. Repeated local subsequence patterns, e.g. maximal repeats or tandem arrays are discovered and used as a basis for the creation of coarse-granular activities. In [38], a more advanced technique is proposed based on mining local process models.

Supervised Learning. Despite the unsupervised nature of the problem, abstraction techniques will often leverage additional domain knowledge, a process model, or other information to turn the problem into a more supervised approach. The technique in [7] relies on a predefined process model, an approach also followed by [26]. Other approaches expect supervision in the form of a set of annotated traces in which fine-granular event sets are matched with a higher-level activity [55], or in the form of timing information, e.g. for sessionization as in [36]. Another example of event abstraction from the healthcare domains was presented in [35], in which they rely on multi-level semantic abstraction using a combination of ontologies and dynamic programming. Also active learning is a promising pathway, bringing the expert in the learning loop to solve the supervision problem.

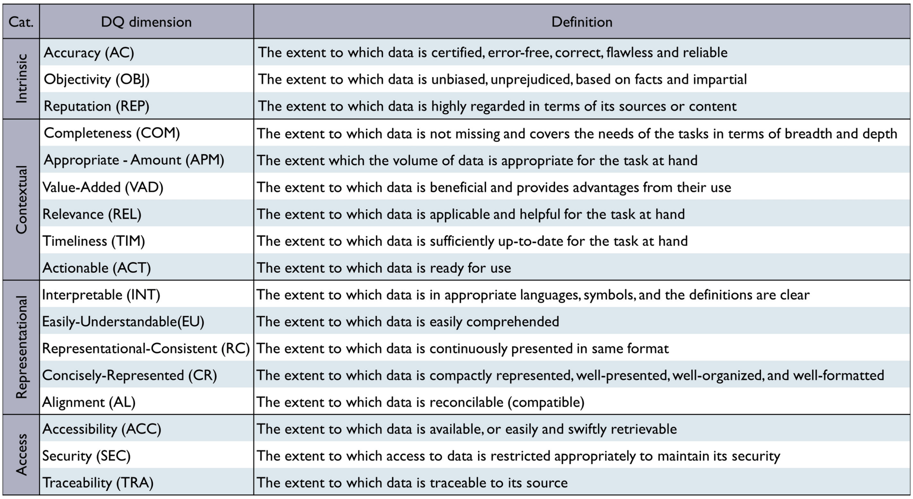
5 Process Mining Data Quality Considerations

"Garbage in, garbage out." It is by far the most mentioned quote in data science and far beyond. But it appears that the more the quote is used, the more relevant it becomes. In process mining, while the problem has been acknowledged in both scientific literature and in practice [57], there is still a need for further research into the development of a comprehensive framework to address the problem of bad quality data leading to incorrect analysis results [58]. We also need to have a better understanding of the root-causes of such data quality issues [5,24].

5.1 Data Quality Dimensions

Some typical data quality dimensions are shown in Fig. 6 [39]. Although there are some similarities between the data quality challenges encountered for event data and traditional data sets for data mining, a key distinguishing factor is our need for detailed correlated event data in their raw form, to capture the true behavior of processes.

In [10], four broad data quality dimensions are identified for event logs: missing data, incorrect data, imprecise data and irrelevant data. Among these four dimensions, incorrect data (where a data item is not recorded correctly) and imprecise data (where a recorded value is too coarse to be useful) for key event attributes such as activity labels and timestamps could have significant consequences for all forms of process mining techniques.



Cat.	DQ dimension	Definition
Intrinsic	Accuracy (AC)	The extent to which data is certified, error-free, correct, flawless and reliable
	Objectivity (OBJ)	The extent to which data is unbiased, unprejudiced, based on facts and impartial
	Reputation (REP)	The extent to which data is highly regarded in terms of its sources or content
Contextual	Completeness (COM)	The extent to which data is not missing and covers the needs of the tasks in terms of breadth and depth
	Appropriate - Amount (APM)	The extent which the volume of data is appropriate for the task at hand
	Value-Added (VAD)	The extent to which data is beneficial and provides advantages from their use
	Relevance (REL)	The extent to which data is applicable and helpful for the task at hand
	Timeliness (TIM)	The extent to which data is sufficiently up-to-date for the task at hand
	Actionable (ACT)	The extent to which data is ready for use
Representational	Interpretable (INT)	The extent to which data is in appropriate languages, symbols, and the definitions are clear
	Easily-Understandable(EU)	The extent to which data is easily comprehended
	Representational-Consistent (RC)	The extent to which data is continuously presented in same format
	Concisely-Represented (CR)	The extent to which data is compactly represented, well-presented, well-organized, and well-formatted
	Alignment (AL)	The extent to which data is reconcilable (compatible)
Access	Accessibility (ACC)	The extent to which data is available, or easily and swiftly retrievable
	Security (SEC)	The extent to which access to data is restricted appropriately to maintain its security
	Traceability (TRA)	The extent to which data is traceable to its source

Fig. 6. An overview of some of the most common data quality dimensions, taken from [39].

5.2 Detection and Repair

The process mining manifesto [2] categorizes the quality of event data from one star to five stars; while most real-life event logs are found to be in-between these two extremes of the scale with many quality issues [58]. Some advocate for repairing or fixing the erroneous data, while others argue that the data should be left alone as it is meant to reflect reality. Regardless of your personal view, it is unavoidable that these data quality issues are dealt with in one way or another. As a process mining professional, it is imperative that we measure the quality of an event log respective to the type of process mining analysis being considered [58]. The data pre-processing task is recognized to be one of the most time-consuming aspects of a process mining study with many spending 60–80% of their efforts while some spending up to 90% of their total efforts on this step [57].

Suriadi et al. [54] identified eleven event log imperfection patterns based on their experience with over 20 Australian industry data sets. The eleven patterns include form-based event capture, inadvertent time travel, unanchored event, scattered event, elusive case, scattered case, collateral event, polluted label, distorted label, synonymous labels and homonymous labels. These event log patterns have been used as a starting point for detection and repair of quality issues in event logs.

There is a growing body of work focusing on the detection and repair of data quality issues associated with activity labels, timestamps, and event orderings. In [49], crowdsourcing and gamification approaches are being proposed to solicit domain expert knowledge for the detection and repair of activity labels while [50] proposes an automated context-aware approach to detecting synonymous and polluted activity labels in an event log. In [28], the authors described a framework to detect timestamp quality issues in an event log and proposed measures to quantify the extent of these data quality issues as a way to measure the quality of an overall event log. In [16], an approach to automatically repairing same-timestamp errors in an event log is presented. In [22], an interactive approach to detect and repair event order imperfections in an event log is presented.

5.3 Quality-Informed Process Mining

Although data quality issues are well-acknowledged in the process mining community by now, most of the existing process mining algorithms do not explicitly take the potential presence of data quality issues. A notable exception is the removal of infrequent behaviors or noises from discovered process models. The algorithms also typically treat an event log as the "whole truth" without considering the potential effects of data-preprocessing on the reliability of the results [58]. This could lead to misleading or inaccurate conclusions about the process under investigation. In [30], the authors proposed a range of quality annotations at event, trace and log levels to keep track of the data quality issues founded in an event log and also to record the extent of repairs are made to the event log as a result. Such metadata about data quality can assist in undertaking quality-informed process mining. One such algorithm is presented as the 'Quality-Informed visual Miner'plug-in' which demonstrates the use of these data quality annotations for conformance checking and performance analysis purposes.

Alternatively, it is possible to determine whether certain data attributes are of high-quality (i.e., fit-for-purpose) before incorporating them into an event log and then into the process mining analysis. In the Process Mining in Practice book[5], checklists are provided to detect a range of data quality issues and suggestions are provided on how to potentially correct them. The quality issues covered

[5] https://fluxicon.com/book/.

include formatting errors, missing data (event, attribute values, case IDs, activities, timestamps, attribute history, timestamps for activity repetition) as well as zero timestamps, wrong timestamps, same timestamps for multiple activities and different timestamp granularity. In [4], a data-quality informed approach is proposed where data attributes from a relational database are evaluated on their quality across a range of data quality measures before generating an event log.

References

1. IEEE Standard for eXtensible Event Stream (XES) for achieving interoperability in event logs and event streams. IEEE Std 1849–2016, pp. 1–50 (2016). https://doi.org/10.1109/IEEESTD.2016.7740858
2. van der Aalst, W., et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBIP, vol. 99, pp. 169–194. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_19
3. Aalst, W.: Process mining: a 360 degrees overview. In: van der Aalst,W.M.P., Carmona, J. (eds.) Process Mining Handbook. LNBIP, vol. 448, pp. 3–34. Springer, Cham (2022)
4. Andrews, R., van Dun, C.G.J., Wynn, M.T., Kratsch, W., Röglinger, M., ter Hofstede, A.H.M.: Quality-informed semi-automated event log generation for process mining. Decis. Support Syst. **132**, 113265 (2020). https://doi.org/10.1016/j.dss.2020.113265
5. Andrews, R., Emamjome, F., ter Hofstede, A.H.M., Reijers, H.A.: An expert lens on data quality in process mining. In: van Dongen, B.F., Montali, M., Wynn, M.T. (eds.) 2nd International Conference on Process Mining, ICPM 2020, Padua, Italy, 4–9 October 2020, pp. 49–56. IEEE (2020). https://doi.org/10.1109/ICPM49681.2020.00018
6. Andrews, R., Wynn, M.T., Vallmuur, K., Ter Hofstede, A.H., Bosley, E., Elcock, M., Rashford, S.: Leveraging data quality to better prepare for process mining: an approach illustrated through analysing road trauma pre-hospital retrieval and transport processes in Queensland. Int. J. Environ. Res. Public Health **16**(7), 1138 (2019)
7. Baier, T., Mendling, J., Weske, M.: Bridging abstraction layers in process mining. Inf. Syst. **46**, 123–139 (2014)
8. Bayomie, D., Helal, I.M.A., Awad, A., Ezat, E., ElBastawissi, A.: Deducing case ids for unlabeled event logs. In: Reichert, M., Reijers, H.A. (eds.) BPM 2015. LNBIP, vol. 256, pp. 242–254. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42887-1_20
9. Beheshti, S.-M.-R., Benatallah, B., Motahari-Nezhad, H.R., Sakr, S.: A query language for analyzing business processes execution. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 281–297. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23059-2_22
10. Bose, J.C., Mans, R., van der Aalst, W.M.P.: Wanna improve process mining results - it's high time we consider data quality issues seriously. In: IEEE Symposium on Computational Intelligence and Data Mining. pp. 127–134. IEEE (2013). https://doi.org/10.1109/CIDM.2013.6597227
11. Jagadeesh Chandra Bose, R.P., van der Aalst, W.M.P.: Abstractions in process mining: a taxonomy of patterns. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 159–175. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03848-8_12

12. Burattin, A., Vigo, R.: A framework for semi-automated process instance discovery from decorative attributes. In: 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), pp. 176–183. IEEE (2011)
13. Calvanese, D., Kalayci, T.E., Montali, M., Tinella, S.: Ontology-based data access for extracting event logs from legacy data: the onprom tool and methodology. In: Abramowicz, W. (ed.) BIS 2017. LNBIP, vol. 288, pp. 220–236. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59336-4_16
14. Calvanese, D., Montali, M., Syamsiyah, A., van der Aalst, W.M.P.: Ontology-driven extraction of event logs from relational databases. In: Reichert, M., Reijers, H.A. (eds.) BPM 2015. LNBIP, vol. 256, pp. 140–153. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42887-1_12
15. Cheng, L., Van Dongen, B.F., Van Der Aalst, W.M.: Efficient event correlation over distributed systems. In: 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), pp. 1–10. IEEE (2017)
16. Conforti, R., Rosa, M.L., ter Hofstede, A.H.M., Augusto, A.: Automatic repair of same-timestamp errors in business process event logs. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) Business Process Management - 18th International Conference, BPM 2020, Seville, Spain, September 13–18, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12168, pp. 327–345. Springer (2020). https://doi.org/10.1007/978-3-030-58666-9_19
17. de Leoni, M.: Foundations of Process Enhancement. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. LNBIP, vol. 448, pp. 243–273. Springer, Cham (2022)
18. De Smedt, J., Hasić, F., vanden Broucke, S.K., Vanthienen, J.: Holistic discovery of decision models from process execution data. Knowl.-Based Syst. **183**, 104866 (2019)
19. Di Francescomarino, C., Dumas, M., Maggi, F.M., Teinemaa, I.: Clustering-based predictive process monitoring. IEEE Trans. Serv. Comput. **12**(6), 896–909 (2016)
20. Di Francescomarino, C., Ghidini, C.: Predictive process monitoring. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. LNBIP, vol. 448, pp. 320–346. Springer, Cham (2022)
21. Diba, K., Batoulis, K., Weidlich, M., Weske, M.: Extraction, correlation, and abstraction of event data for process mining. WIREs Data Mining Knowl. Discov. **10**(3), e1346 (2020). https://doi.org/10.1002/widm.1346
22. Dixit, P.M., et al.: Detection and interactive repair of event ordering imperfection in process logs. In: Krogstie, J., Reijers, H.A. (eds.) Advanced Information Systems Engineering - 30th International Conference, CAiSE 2018, Tallinn, Estonia, 11–15 June 2018, LNCS, vol. 10816, pp. 274–290. Springer, Berlin (2018). https://doi.org/10.1007/978-3-319-91563-0_17
23. van Eck, M.L., Sidorova, N., van der Aalst, W.M.: Enabling process mining on sensor data from smart products. In: 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS), pp. 1–12. IEEE (2016)
24. Emamjome, F., Andrews, R., ter Hofstede, A.H.M., Reijers, H.A.: Signpost - a semiotics-based process mining methodology. In: Rowe, F., et al. (eds.) 28th European Conference on Information Systems - Liberty, Equality, and Fraternity in a Digitizing World, ECIS 2020, Marrakech, Morocco, 15–17 June 2020 (2020), https://aisel.aisnet.org/ecis2020_rip/50
25. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI Mag. **17**(3), 37 (1996)
26. Fazzinga, B., Flesca, S., Furfaro, F., Masciari, E., Pontieri, L.: Efficiently interpreting traces of low level events in business process logs. Inf. Syst. **73**, 1–24 (2018)

27. Ferreira, D.R., Gillblad, D.: Discovering process models from unlabelled event logs. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 143–158. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03848-8_11
28. Fischer, D.A., Goel, K., Andrews, R., van Dun, C.G.J., Wynn, M.T., Röglinger, M.: Enhancing event log quality: detecting and quantifying timestamp imperfections. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) BPM 2020. LNCS, vol. 12168, pp. 309–326. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58666-9_18
29. Folino, F., Guarascio, M., Pontieri, L.: Mining multi-variant process models from low-level logs. In: Abramowicz, W. (ed.) BIS 2015. LNBIP, vol. 208, pp. 165–177. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19027-3_14
30. Goel, K., Leemans, S.J., Martin, N., Wynn, M.T.: Quality-informed process mining: a case for standardised data quality annotations. ACM Trans. Knowl. Discov. Data **16**, 1–47 (2022)
31. Günther, C.W., van der Aalst, W.M.: Mining activity clusters from low-level event logs. Beta, Research School for Operations Management and Logistics (2006)
32. Günther, C.W., Rozinat, A., van der Aalst, W.M.P.: Activity mining by global trace segmentation. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 128–139. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_13
33. Ingvaldsen, J.E., Gulla, J.A.: Preprocessing support for large scale process mining of SAP transactions. In: ter Hofstede, A., Benatallah, B., Paik, H.-Y. (eds.) BPM 2007. LNCS, vol. 4928, pp. 30–41. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78238-4_5
34. Janiesch, C., et al.: The internet of things meets business process management: a manifesto. IEEE Syst. Man Cybern. Mag. **6**(4), 34–44 (2020). https://doi.org/10.1109/MSMC.2020.3003135
35. Leonardi, G., Striani, M., Quaglini, S., Cavallini, A., Montani, S.: Leveraging semantic labels for multi-level abstraction in medical process mining and trace comparison. J. Biomed. Inform. **83**, 10–24 (2018)
36. de Leoni, M., Dündar, S.: Event-log abstraction using batch session identification and clustering. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing, pp. 36–44 (2020)
37. Mannhardt, F., de Leoni, M., Reijers, H.A.: Extending process logs with events from supplementary sources. In: Fournier, F., Mendling, J. (eds.) BPM 2014. LNBIP, vol. 202, pp. 235–247. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15895-2_21
38. Mannhardt, F., Tax, N.: Unsupervised event abstraction using pattern abstraction and local process models. arXiv preprint arXiv:1704.03520 (2017)
39. Moges, H.T., Dejaeger, K., Lemahieu, W., Baesens, B.: A multidimensional analysis of data quality for credit risk management: new insights and challenges. Inf. Manag. **50**(1), 43–58 (2013)
40. Motahari-Nezhad, H.R., Saint-Paul, R., Casati, F., Benatallah, B.: Event correlation for process discovery from web service interaction logs. VLDB J. **20**(3), 417–444 (2011)
41. Nooijen, E.H.J., van Dongen, B.F., Fahland, D.: Automatic discovery of data-centric and artifact-centric processes. In: La Rosa, M., Soffer, P. (eds.) BPM 2012. LNBIP, vol. 132, pp. 316–327. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36285-9_36

42. Pérez-Castillo, R., Weber, B., de Guzmán, I.G.-R., Piattini, M., Pinggera, J.: Assessing event correlation in non-process-aware information systems. Softw. Syst. Model. **13**(3), 1117–1139 (2012). https://doi.org/10.1007/s10270-012-0285-5
43. Pourmirza, S., Peters, S., Dijkman, R., Grefen, P.: BPMS-RA: a novel reference architecture for business process management systems. ACM Trans. Internet Technol. **19**(1), 1–23 (2019)
44. Reguieg, H., Benatallah, B., Nezhad, H.R.M., Toumani, F.: Event correlation analytics: scaling process mining using Mapreduce-aware event correlation discovery techniques. IEEE Trans. Serv. Comput. **8**(6), 847–860 (2015)
45. Rehse, J.-R., Fettke, P.: Clustering business process activities for identifying reference model components. In: Daniel, F., Sheng, Q.Z., Motahari, H. (eds.) BPM 2018. LNBIP, vol. 342, pp. 5–17. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11641-5_1
46. Rodrıguez, C., Engel, R., Kostoska, G., Daniel, F., Casati, F., Aimar, M.: Eventifier: extracting process execution logs from operational databases. Proc. Demonstr. Track BPM **940**, 17–22 (2012)
47. Rozinat, A., van der Aalst, W.M.P.: Decision mining in ProM. In: Dustdar, S., Fiadeiro, J., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 420–425. Springer, Heidelberg (2006). https://doi.org/10.1007/11841760_33
48. Rozsnyai, S., Slominski, A., Lakshmanan, G.T.: Discovering event correlation rules for semi-structured business processes. In: Proceedings of the 5th ACM International Conference on Distributed Event-Based System, pp. 75–86 (2011)
49. Sadeghianasl, S., ter Hofstede, A.H.M., Suriadi, S., Turkay, S.: Collaborative and interactive detection and repair of activity labels in process event logs. In: van Dongen, B.F., Montali, M., Wynn, M.T. (eds.) 2nd International Conference on Process Mining, ICPM 2020, Padua, Italy, 4–9 October 2020, pp. 41–48. IEEE (2020). https://doi.org/10.1109/ICPM49681.2020.00017
50. Sadeghianasl, S., ter Hofstede, A.H.M., Wynn, M.T., Suriadi, S.: A contextual approach to detecting synonymous and polluted activity labels in process event logs. In: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (eds.) On the Move to Meaningful Internet Systems: OTM 2019 Conferences - Confederated International Conferences: CoopIS, ODBASE, C&TC 2019, Rhodes, Greece, 21–25 October 2019, LNCS, vol. 11877, pp. 76–94. Springer, Berlin (2019). https://doi.org/10.1007/978-3-030-33246-4_5
51. Senderovich, A., Rogge-Solti, A., Gal, A., Mendling, J., Mandelbaum, A.: The ROAD from sensor data to process instances via interaction mining. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) CAiSE 2016. LNCS, vol. 9694, pp. 257–273. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39696-5_16
52. Shearer, C.: The CRISP-DM model: the new blueprint for data mining. J. Data Warehousing **5**(4), 13–22 (2000)
53. Song, M., Van der Aalst, W.M.: Towards comprehensive support for organizational mining. Decisi. Support Syst. **46**(1), 300–317 (2008)
54. Suriadi, S., Andrews, R., ter Hofstede, A.H.M., Wynn, M.T.: Event log imperfection patterns for process mining: towards a systematic approach to cleaning event logs. Inf. Syst. **64**, 132–150 (2017). https://doi.org/10.1016/j.is.2016.07.011
55. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.: Mining process model descriptions of daily life through event abstraction. In: Bi, Y., Kapoor, S., Bhatia, R. (eds.) IntelliSys 2016. SCI, vol. 751, pp. 83–104. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-69266-1_5

56. van Eck, M.L., Lu, X., Leemans, S.J.J., van der Aalst, W.M.P.: PM^2: a process mining project methodology. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) CAiSE 2015. LNCS, vol. 9097, pp. 297–313. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19069-3_19
57. Wynn, M.T., et al.: Rethinking the input for process mining: Insights from the XES survey and workshop. In: International Conference on Process Mining: Workshop Proceedings. LNBIP, Springer, Cham (2021). https://doi.org/10.1007/978-3-030-98581-3_1
58. Wynn, M.T., Sadiq, S.: Responsible process mining - a data quality perspective. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNCS, vol. 11675, pp. 10–15. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_2
59. van Zelst, S.J., Mannhardt, F., de Leoni, M., Koschmider, A.: Event abstraction in process mining: literature review and taxonomy. Granular Comput. **6**, 719–736 (2020)