Towards the Detection of Promising Processes by Analysing the Relational Data

Belén Ramos-Gutiérrez[1(✉)], Luisa Parody[2], and María Teresa Gómez-López[1]

[1] Universidad de Sevilla, Sevilla, Spain
{brgutierrez,maytegomez}@us.es
[2] Universidad Loyola Andalucía, Sevilla, Spain
mlparody@uloyola.es
http://www.idea.us.es/

Abstract. Business process discovery provides mechanisms to extract the general process behaviour from event observations. However, not always the logs are available and must be extracted from repositories, such as relational databases. Derived from the references that exist between the relational tables, several are the possible combinations of traces of events that can be extracted from a relational database. Different traces can be extracted depending on which attribute represents the $case\_id$, what are the attributes that represent the execution of an activity, or how to obtain the timestamp to define the order of the events. This paper proposes a method to analyse a wide range of possible traces that could be extracted from a relational database, based on measuring the level of interest of extracting a trace log, later used for a discovery process. The analysis is done by means of a set of proposed metrics before the traces are generated and the process is discovered. This analysis helps to reduce the computational cost of process discovery. For a possible $case\_id$ every possible traces are analysed and measured. To validate our proposal, we have used a real relational database, where the detection of processes (most and least promising) are compared to rely on our proposal.

Keywords: Process discovery · Promising process · Measures · Relational databases

1 Introduction

Business Process Management (BPM) facilities the modelling and deployment of the process with a high level of automation [15]. BPM is centred on the optimization of the processes, based on a described model and the observation of the real actions executed in the organizational daily activities. The irruption of

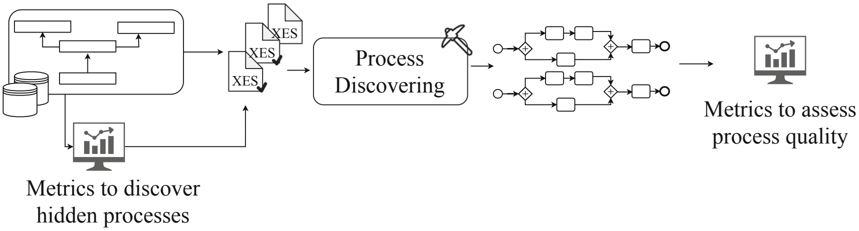

Fig. 1. Measuring the promising processes

BPM in industrial scenarios [20] has provoked to tackled cases where processes are not modelled or executed by Business Process Management Systems but implemented as specific applications for each organization. These applications generate an important quantity of data that is stored as evidence of the processes executed. This data could be used in Process Mining [5] to discover the processes and to ascertain if the organization is working as expected. However, one of the challenges to applying process mining techniques is to generate the event logs used as input of the discovering algorithms. An event log can be defined as an ordered list of activities executed in accordance with a *case_id* that differentiates each trace. There are several papers and tools that facilitate the extraction of the traces from relational databases for later discovery, as detailed in Related Work section (Sect. 2). The problem is that derived from the relationships that exist between the data of the relational tables, a huge number of different traces can be obtained from a relational database. The knowledge of the expert is a key aspect to ascertain which attributes are relevant to extract promising business model in a discovery process. This implies to ascertain which attributes will represent: the *case_id*, the *activity name* and the *timestamp*.

In order to guide the expert about the feasibility of detecting a promising process from a possible trace extracted from a relational database, we propose a method to measure the potential traces that can be extracted. As shown in Fig. 1, several are the potential traces (represented by the standard XES [2]) that can be extracted from a relational database. To propose a ranking for the most and least promising traces for later discovery, we propose a set of metrics for detecting hiding processes. To validate our proposal, we have included another metric to measure the quality of the discovered processes, to verify if the metrics for detecting hiding processes are valid.

The paper is organized as follows: Sect. 2 summarises the most relevant related proposals; Sect. 3 presents the real example where our proposal has been applied; Sect. 4 details the method proposed to extract the most promising traces; in Sect. 5 the evaluation of our proposal in a real example is shown; and finally the paper is concluded and future work is explained.

2 Related Work

The relevance of the extraction of event data from databases is widely known [4,10], and it is an important mechanism to enrich the process mining [22]. For a general point of view, the analysis of the evolution of the data also represents the activities executed [21]. Previous proposals have analysed the possible relations between the stored data and the business processes [11,16,17,25]. Especially, relational databases have been used as a source of analysis to extract log traces for a later process discovery [22]. In [12], Dijkman et al. apply relational algebra to query the database and extract the log traces. In [8], Berti and var der Aalst include the discovery of Multiple Viewpoint models annotated with performance and frequently information from relational databases. However, their proposal is based on the analysis of the attributes but not in the values of the attributes. In addition, different tools have been implemented to support the trace generation [18,27] and to retrieve event data from databases, such as OpenSLEX [23].

Not only relational databases have been the analysed source of traces, but also the problem of log extraction from semi-structured sources has been addressed [26]. There are also some studies to ascertain the $case\_id$ from unstructured data. Bayomie et al. in [7] infer the $case\_id$ for the unlabeled event logs, and Helal et al. in [19] establish a ranking of possible $case\_id$ from unlabeled event logs. Nevertheless, none of them infers the $case\_id$ ranking the information from a relational database.

The necessity to measure the quality of the process is a known problem, and the application of discovery techniques to incorrect or inaccurate data log will generate incorrect or inaccurate business process models [28]. There exist several criteria to assess the data quality in general [6,24], but centred on data log quality, the Process Mining Manifesto [3] develops a deeper analysis, including safety, completeness, correctness or trustworthiness. Nevertheless, this analysis only includes the quality of the log, but not the evaluation of the quality of the knowledge acquired from the log. In [3] the quality is measured quantitatively. These maturity levels assign the lowest quality when the recorded events do not correspond with the reality, for example, when they are recorded manually. Whereas high quality describes an automatic and complete recovery, reaching the highest quality when every event recorded, and its attributes, have a known semantic meaning about the Business Process Model (BPM). However, this way to measure the quality is not related to the type of processes that could be discovered after the application of a discovery process, the reason why we have defined different metrics to guide in this issue.

To the best of our knowledge, this is the first proposal that tackles the problem of detecting the most promising traces for a later discovery process.

3 Case Study

As outlined in Sect. 1, one of the biggest challenges of the discovery task in process mining is the automatic creation of an event log from a data source. The extraction of these logs from relational databases is an area in which numerous works have been developed as it is analysed in Sect. 2.

However, all these solutions have a common factor: they all require expert knowledge about the domain and the data in order to perform an extraction that has the potential to be used in subsequent discovery tasks. This fact represents a major problem when we are facing with a large volume of data, which is very common in digitized organizations nowadays. Moreover, this problem seems even more interesting to address when is taken into account that the organization's data expert does not have to be a business process management expert, which means that the data extracted by the data expert could be of little use for process discovery. In the case of the business process management expert, it may take a long time to understand the distribution and semantics of information in order to extract the most relevant in the correct way.

As an example of this scenario, we present our case study: a relational database with more than 300 tables containing the daily operation of assembly and testing processes on aircraft in one of the *Airbus Space & Defence* factories[1], that will serve us to evaluate our proposal to automate the detection of promising processes. More specifically, for our proof of concept, we have focused the experiments on the four fundamental database tables shown in Fig. 2:

- **Aircraft** represents the different aircraft that are assembled and tested. The table contains, among other things, information such as unique identification of the aircraft, its type and version, a serial number and registration, and modification dates and registrars.
- **Functional tests** stores every data related to ground test instructions, which are a representation of the functional tests carried out on aircraft during their assembly process. This table has more than 45 attributes, containing the information as relevant as the unique identifier of the test, its title, code and version, the reasons why it exists, the type of test, various dates and user identifiers related to its creation and modification, subsystem in which it is carried out, different natural text fields describing test specifications, etc.
- **Test execution** keeps the information concerning the execution of the tests described above. So that each record in this table will be associated with a particular functional test and aircraft, as well as a workstation where the test has been executed.
- **Incidents** saves the incidents that occur during each running test, using more than 30 attributes, which indicate the type of incident and its severity, dates and users associated with its registration, modification or cancellation, and various attributes related to observations made in natural text.

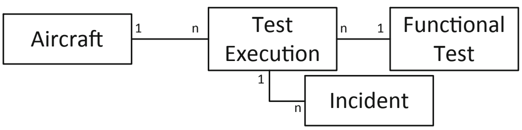

Fig. 2. Case study class diagram.

[1] The data cannot be published due to a confidentiality agreement.

With only these four tables, it is possible to obtain several interpretations of the assembly and testing process, and understand the high number of possible traces that can be created, since it provides us with enough information to be able to discover a wide variety of processes using different *case_id* and *events*. For example, the evolution in the assembly of the aircraft can be analysed according to the workstations in which the tests are executed, the tests can be studied for the type of the incidents that occur during its execution. or the incidents can be analysed based on the functional tests with which they are associated.

To be more precise, with only this small sample of four tables of the database, at least 240 different processes could be discovered. Moreover, the combinatorial nature of the problem has been reduced with: (1) only the primary keys of the tables can be selected as *case_id* since the *case_id* must be a unique value, and; (2) float numbers, dates and Boolean attributes cannot be taken as events since this type of data cannot help to determine the execution of activity of a change of state in a system. This fact highlights the need to find out a way to automate the search for information that can be extracted to create promising processes.

4 Proposal: Method to Detect Promising Processes

The main objective of the proposal consists of analysing the data of the relational database in order to detect hiding business processes that can be promising to know how our system works. The method comprises the following steps:

1. **Analysis of Promising Traces** is the first step to decide which attributes of the database are more suitable to be the *case_id* and to represent the *events* of the promising business processes. In this step, various metrics are necessary to evaluate the traces and select which ones will compose the event logs, before the traces are generated or the processes are discovered. Each part of this analysis is detailed in Subsect. 4.1.
2. **Generation of traces** is executed once the *case_id* and the *events* are selected according to the previous step. The traces are extracted from the relational database by using some of the existing solutions, such as XESame [18] plugin of ProM.
3. **Discovery of Business Process and Assessment of Process Quality** algorithms are applied to the selected traces to discover the promising processes. These business processes should be analyzed to ascertain if they are promising or not. This step is analysis in deep in Subsect. 4.2.

4.1 Analysis of Promising Traces

Many times, the information generated during the execution of a process is stored in a scattered manner in the databases. For example, a test run of an aircraft includes new tuples in a table but can also update values from existing tuples in other tables. This implies that to know how an aircraft evolves can be relevant to analyse the attributes of Table Aircraft but also their tests, execution of incidents. But hundred are the possible combinations according to the different attributes of each table.

The first step is the analysis of a possible *case_id*, that must be a unique value that comes together with a set of events that represents the execution of an instance of a process. Thus, the primary key of the tables is a unique identifier, as well. In our approach, we consider the primary keys as a good candidate to represent the *case_id*. Besides, since the relationship between two tables in a database is established by using a pair of primary-foreign keys, *case_id* will allow the information of related tables to be included in the case.

On the other hand, an event in a log trace represents the execution of a task in a process instance. In our proposal, the possible events are the values of the attributes that are related to a specific *case_id*. It is worthwhile prioritizing among the attributes that give the most relevant information related to the selected *case_id*. Therefore, those attributes whose values are Float numbers, Dates, or Boolean, can be discarded at the outset since they cannot be understood as a type of task involved in a business model. For example, which activity would represent the *true* value or the date 05/05/2020?. To know what are the possible attributes, the relational database can be represented as a graph (as shown in Fig. 3) where each node represents an attribute, and the edges relate the attributes with its primary key or the relation primary-foreign key. From a primary key attribute, every reachable node can represent the execution of a task (representation of an event), whose timestamp is extracted from the redo log files of the database. To carry out the analysis of which attribute is the most appropriated to represent an event, we analyse the specific values stored in each attribute and its relation with the *case_id*. For the example, if the primary key of Aircraft Table is the *case_id*, and the events are the types of incidents of the Table Incidents, we could know (i) the number of traces (different values of the primary key in aircraft); (ii) the events of each trace (the types of incidences of every text related to each aircraft); and (iii) the different number of events (the different values of the types of incidents). Without creating the trace, we obtain important information about the potential traces, and the possible process discovered later. To measure this information, we present the following set of metrics to know the complexity, diversity, and noise of the different attributes as hypothetical events. The metrics are:

Complexity: the complexity (C) of a process model is understood as the average number of events per trace. A very high number of event per trace will generate too complex processes. However, a very low number of events will produce too simple processes. In order to obtain values between 0 and 1 that allows us to compare all the candidates, it is necessary to normalise (C_n) the values computed in the previous arithmetic operation. To do so, the formula below has been used to penalise both excessively simple and extremely difficult processes and favour those in-between. In the formula, C_i represents the complexity value of a candidate (attribute), C_{q1} and C_{q3} the values of the first and third quartile

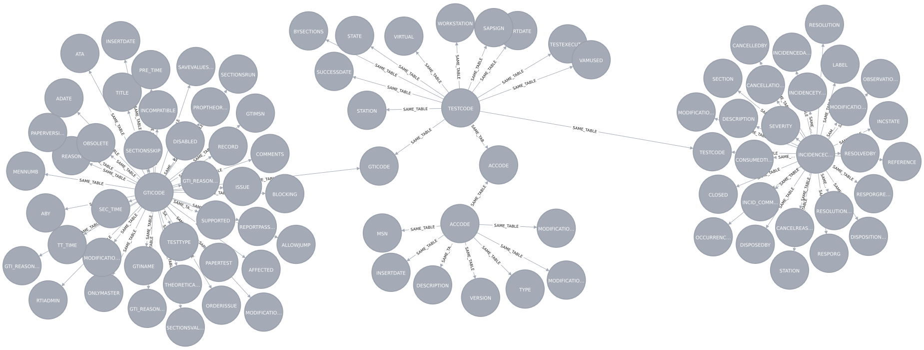

Fig. 3. Relational Database as a graph.

respectively of all the complexities of the chosen sample and C_{max} the maximum of the complexities of the selected sample.

$$C_n(C_i) = \begin{cases} C_i < C_{q1} & (\frac{1}{C_{q1}}) \cdot C_i \\ C_{q1} < C_i < C_{q3} & 1 \\ C_i > C_{q3} & (\frac{-C_i + C_{q3}}{C_{max} - C_{q3}}) + 1 \end{cases} \tag{1}$$

Diversity: the diversity (D) of a process model is the value that represents the density of different events that occur in all the traces among all the events that are presented in the log. This value should also be standardized (D_n) to allow comparison of the quality of different candidates. There is also necessary to penalise proportionally both processes, those in which there is little variety of events and those where is so much variety that is difficult to extract frequent behaviour. The normalisation function is as follows, where D_i represents the diversity of a candidate and D_{mean}, the mean of the diversities of the candidates of the selected sample:

$$D_n(D_i) = \begin{cases} D_i < D_{mean} & (\frac{1}{D_{mean}}) \cdot D_i \\ D_i > D_{mean} & (\frac{-D_i + D_{mean}}{1 - D_{mean}}) + 1 \end{cases} \tag{2}$$

Noise: the noise (N) of a process model means the average of events that only occur once in the whole log among all the events inside of it. The normalised (N_n) value should be measured between [0..1] according to the following function, which rewards candidates in which the presence of noise is minimal:

$$N_n(N_i) = -N_i + 1 \tag{3}$$

4.2 Discovery of Business Processes and Assessment of Process Quality

Once the XES traces are created, different algorithms can be used to discover the process model. In our case, we have applied Inductive Miner techniques using ProM [1,14]. The noise threshold used in the process discovery is 0.2.

In order to know if the selected traces are promising, we propose to (i) measure the quality of the business processes, (ii) verify the proper functioning of the metrics, and (iii) corroborate that those are promising indeed. To this end, we have defined the **Will Level** metric, as the mean of possible tasks that can be selected in each step according to the process model, divided into the number of total tasks of the process. It is applied to the discovered process model to know how the general it is. For example, in a flowering process, in each step of the instance, any task can be executed, then the metric will level informs about the low use of a process that does not restrict what activity can be executed. The range of the metric is between [0..1], where 0 represents a very restricted process (a sequence of tasks) and 1 a process with XOR-gates that include every task among their branches. The calculation of the Will Level is based on the analysis of the process as a graph, to analyse the possible next activity to execute analysing the possible paths. In order to do that, to obtain the value of the metric the following phases are required:

1. Translation of the process model (i.e. a BPM modelled using the standard BPMN [9] in our case) to a directed labelled graph, where events, gateways and tasks are nodes. The weight of the edges will be 1, if the target of the edge is a task since it means that we have an option, 0 otherwise.
2. Execution of the Dijkstra algorithm [13] to find out all the shortest paths in the graph from the start event and a task to the others.
3. Calculate the will level of each task and the start event as, the number of possible next activities, that are those paths of length equal to 1, divided into the total number of tasks.
4. The calculation of the total Will Level is the arithmetic mean of the Will Level value of each activity node.

5 Evaluation

In order to evaluate our proposal, we utilize the data described in Sect. 3 applying the metrics presented in Sect. 4. More specifically, we have made all the possible combinations in which the unique identifier of Table Aircraft acts as $case\_id$, while its attributes and those belonging to the other three tables represent the events. Thus, we analysed 58 possible attributes as potential events, before creating the traces used for discovering the processes. The use of our proposal reduces the analysis of those 58 traces, focused on discovering only the most promising according to the metrics.

Table 1 shows the results of the normalised metrics for a selection of 13 for the primary key of the aircraft table as the $case\_id$. These 13 represent the 9 best traces (good values for the three metrics), and the 4 worst (bad values for the three metrics).

Some of the discovered processes are shown in Fig. 4. Although the details of the processes cannot be seen, it is possible to observe which are the best rated since

Table 1. Results of the metrics for the detection of hiding processes.

Table	Attribute	Complexity	Diversity	Noise
Incident	Station	0.971	0.777	0.999
Functional test	Name	1	0.970	0.986
Functional test	Reason	1	0.860	0.986
Functional test	Affected	1	0.916	0.999
Functional test	Title	1	0.816	0.998
Functional test	Comments	0.970	0.973	0.985
Functional test	Aby	1	0.893	0.999
Functional test	Modification user	1	0.875	0.999
Functional test	Supported	1	0.988	0.996
Aircraft	Serial Number	0.005	0.094	0
Aircraft	Description	0.005	0.094	0
Incident	Comment	0.176	0.131	0.221
Functional test	Reason reference	0.015	0.142	0.666

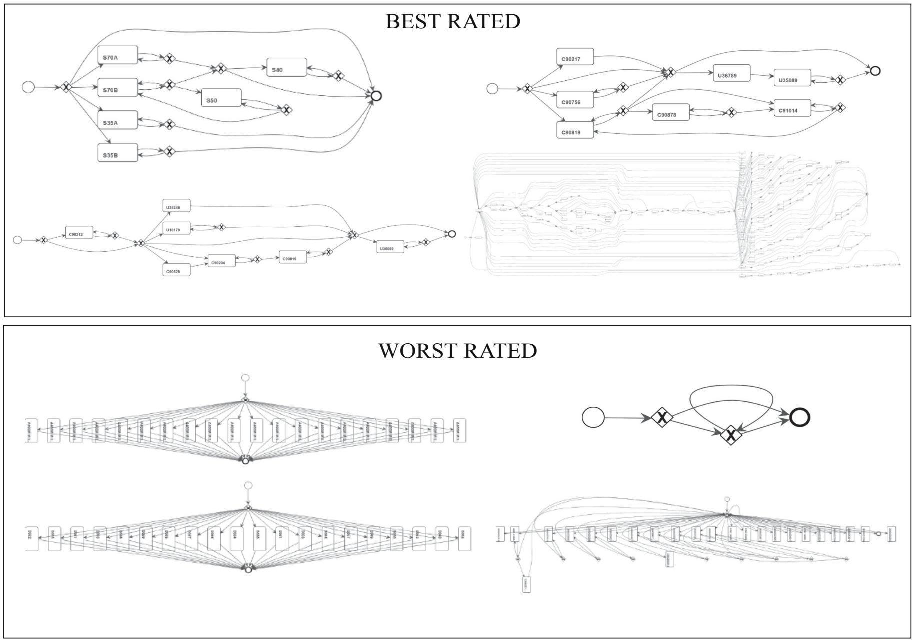

Fig. 4. Examples of discovered processes.

they are more understandable and relevant processes than the others. The worst-rated are processes with some XOR-gateways with several branches, that represent that any task can be executed.

By measuring the quality of these processes through the will level metric defined previously, we obtain Table 2 that shows the results.

Table 2. Sample of Will Level metric results.

Table	Attribute	Will Level
Incident	Station	0.28
Functional test	Name	0.02
Functional test	Reason	0.06
Functional test	Affected	0.76
Functional test	Title	0.06
Functional test	Comments	0.21
Functional test	Aby	0.21
Functional test	Modification user	0.28
Functional test	Supported	0.05
Aircraft	Serial Number	0.95
Aircraft	Description	0.95
Incident	Comment	-
Functional test	Reason reference	0.81

To assess the results obtained with the proposed methodology, we will rely on the values of the metrics depicted in Tables 1 and 2. In both, the candidates with the best results are placed above the double horizontal line, while those with the worst results are placed below. As can be seen, the candidates rated as best or worst in Table 1 have continued to be classified in the same way with the metrics to discover hidden processes.

In Table 1, it can be seen that all the best-rated attributes have a complexity close to 1, which is positive, as it implies that they have a well-balanced number of events per trace, meaning that they are not excessively simple or too large. Concerning diversity, we found that the best candidates present values greater than 0.75, while the worst are always below 0.2. For the last ones, we can observe that most of the worst-rated candidates have a diversity close to 0, meaning that either they are excessively repetitive traces, or there is so much variety in the distribution of events (infrequent behaviour). Regarding the noise, event logs that have a very low noise level are benefited, and this evidences that the best-rated candidates, that have a noise level very close to 1, have hardly any instances that represent an outlier and can alter the results in the discovery.

The results must be interpreted oppositely in Table 2, the candidates will be better, the lower their value in this metric, since this will indicate high levels of sequentiality. Special cases, such as *'Affected'* or *'Reason reference'*, whose values are slightly close to 1, might indicate that this type of event log would offer better results if declarative rather than imperative models are used. This conclusion comes since their traces reflect that they are very permissive processes but have certain restrictions. The attribute *Comment* does not have will level defined since the level of noise is so high that no activities are discovered by using Inductive Miner. Once the most promising processes have been obtained, the next step would be to ask the business experts which of them could really be used because they are really useful for the business. This step is outside the scope of this article but will be a future work.

6 Conclusion and Future Work

This paper presents a method to guide in the detection of hiding processes by analysing the information of the relational database that contains the data produced by a process. To extract the most promising processes, hiding into the data, some metrics have been proposed based on the number of traces, events, and frequency of them, aligned with the metrics of complexity, diversity, and noise. The analysis of these metrics provides a ranking to ascertain, for each $case\_id$, what are the possible event logs that could be interesting to participate in a discovery process. The validation of our proposal is focused on the analysis of the relevance of the obtained processes, using the evaluation of an expert and measuring the level of will that represents the discovered process. According to these metrics, our proposal has been ratified. For the future, we consider extending the types of metrics both before and after the processes discovery. Moreover, analysing the database structure to infer new possible indicators that help to infer the most promising processes.

Acknowledgement. This research was partially supported by Ministry of Science and Technology of Spain with projects ECLIPSE (RTI2018-094283-B-C33) and by Junta de Andalucía with METAMORFOSIS projects; and by European Regional Development Fund (ERDF/FEDER).

References

1. Prom tool. http://www.promtools.org/doku.php
2. IEEE standard for extensible event stream (XES) for achieving interoperability in event logs and event streams. IEEE Std 1849–2016, pp. 1–50 (2016)
3. 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
4. van der Aalst, W.M.P.: Extracting event data from databases to unleash process mining. In: BPM - Driving Innovation in a Digital World, pp. 105–128 (2015)
5. Aalst, W.: Data science in action. Process Mining, pp. 3–23. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4_1

6. Batini, C.: Data quality assessment. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems, 2nd edn. Springer, New York (2018). https://doi.org/10.1007/978-0-387-39940-9_107
7. 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.) Business Process Management Workshops, pp. 242–254. Springer, Cham (2016)
8. Berti, A., van der Aalst, W.M.P.: Extracting multiple viewpoint models from relational databases. CoRR abs/2001.02562 (2020)
9. Business process model and notation (BPMN) version 2.0.2. Standard, Object Management Group Standard (2014)
10. Calvanese, D., Kalayci, T.E., Montali, M., Santoso, A.: OBDA for log extraction in process mining. In: Reasoning Web. Semantic Interoperability on the Web - 13th International Summer School 2017, London, UK, 7–11 July 2017, Tutorial Lectures, pp. 292–345 (2017)
11. 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
12. Dijkman, R., Gao, J., Syamsiyah, A., van Dongen, B., Grefen, P., ter Hofstede, A.: Enabling efficient process mining on large data sets: realizing an in-database process mining operator. Distributed and Parallel Databases **38**(1), 227–253 (2019). https://doi.org/10.1007/s10619-019-07270-1
13. Dijkstra, E.W., et al.: A note on two problems in connexion with graphs. Numerische mathematik **1**(1), 269–271 (1959)
14. van Dongen, B.F., de Medeiros, A.K.A., Verbeek, H.M.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: The ProM framework: a new era in process mining tool support. In: Ciardo, G., Darondeau, P. (eds.) ICATPN 2005. LNCS, vol. 3536, pp. 444–454. Springer, Heidelberg (2005). https://doi.org/10.1007/11494744_25
15. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-33143-5
16. Gómez-López, M.T., Borrego, D., Gasca, R.M.: Data state description for the migration to activity-centric business process model maintaining legacy databases. In: BIS, pp. 86–97 (2014)
17. Gómez-López, M.T., Reina Quintero, A.M., Parody, L., Pérez Álvarez, J.M., Reichert, M.: An architecture for querying business process, business process instances, and business data models. In: Teniente, E., Weidlich, M. (eds.) BPM 2017. LNBIP, vol. 308, pp. 757–769. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74030-0_60
18. Günther, C.W., van der Aalst, W.M.P.: A generic import framework for process event logs. In: Eder, J., Dustdar, S. (eds.) BPM 2006. LNCS, vol. 4103, pp. 81–92. Springer, Heidelberg (2006). https://doi.org/10.1007/11837862_10
19. Helal, I.M.A., Awad, A., El Bastawissi, A.: Runtime deduction of case id for unlabeled business process execution events. In: 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), pp. 1–8 (2015)
20. Kalpic, B., Bernus, P.: Business process modelling in industry - the powerful tool in enterprise management. Comput. Ind. **47**(3), 299–318 (2002)
21. Li, G., de Murillas, E.G.L., de Carvalho, R.M., van der Aalst, W.M.P.: Extracting object-centric event logs to support process mining on databases. In: Mendling, J., Mouratidis, H. (eds.) CAiSE 2018. LNBIP, vol. 317, pp. 182–199. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92901-9_16

22. de Murillas, E.G.L., Reijers, H.A., van der Aalst, W.M.P.: Connecting databases with process mining: a meta model and toolset. Software & Systems Modeling (2018)
23. González López de Murillas, E., Reijers, H.A., van der Aalst, W.M.P.: Connecting databases with process mining: a meta model and toolset. In: Schmidt, R., Guédria, W., Bider, I., Guerreiro, S. (eds.) BPMDS/EMMSAD -2016. LNBIP, vol. 248, pp. 231–249. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39429-9_15
24. Otto, B., Lee, Y.W., Caballero, I.: Information and data quality in networked business. Electron. Markets **21**(2), 79–81 (2011). https://doi.org/10.1007/s12525-011-0062-2
25. Pérez-Alvarez, J., Gómez-López, M., Eshuis, R., Montali, M., Gasca, R.: Verifying the manipulation of data objects according to business process and data models, January 2020
26. Valencia-Parra, Á., Ramos-Gutiérrez, B., Varela-Vaca, A.J., Gómez-López, M.T., Bernal, A.G.: Enabling process mining in aircraft manufactures: extracting event logs and discovering processes from complex data. In: Proceedings of the Industry Forum at BPM 2019, Vienna, Austria, September 1–6, 2019, pp. 166–177 (2019)
27. Verbeek, H.M.W., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: XES, XESame, and ProM 6. In: Information Systems Evolution - CAiSE Forum 2010, Hammamet, Tunisia, June 7–9, 2010, Selected Extended Papers, pp. 60–75 (2010)
28. Wynn, M.T., Sadiq, S.: Responsible process mining - a data quality perspective. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) Business Process Management, pp. 10–15. Springer, Cham (2019)