



HAL
open science

Standardizing Process-Data Exploitation by Means of a Process-Instance Metamodel

Antonio Cancela, Antonia Quintero, María Teresa Gómez-López, Alejandro García-García

► **To cite this version:**

Antonio Cancela, Antonia Quintero, María Teresa Gómez-López, Alejandro García-García. Standardizing Process-Data Exploitation by Means of a Process-Instance Metamodel. 8th International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA), Dec 2018, Seville, Spain. pp.52-66, 10.1007/978-3-030-46633-6_3. hal-03188593

HAL Id: hal-03188593

<https://inria.hal.science/hal-03188593v1>

Submitted on 2 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Standardizing Process-Data Exploitation by means of a Process-Instance Metamodel

Antonio Cancela¹[0000-0003-0421-4412], Antonia M. Reina Quintero¹[0000-0003-3698-6302], María Teresa Gómez-López¹[0000-0002-3562-875X], and Alejandro García-García

Universidad de Sevilla, Sevilla, Spain
{acancela,reinaqu,maytegoz}@us.es, algarcia@emergy.com
<http://www.idea.us.es>

Abstract. The analysis of data produced by enterprises during business-process executions is crucial in ascertaining how these processes work and how they can be optimized, despite heterogeneous nature of these data structures. This data may also be used for various types of analysis, such as reasoning, process querying and process mining, which consume different data formats. However, all these structures and formats share a common ground: the business-process model and its instantiation are in each of their kernels. In this paper, we propose the use of a Business-Process Instance Metamodel, which serves as a common interface to perform an independent exploitation of data from the applications that produce the data and those which consume the data. A tool has been implemented as a proof of concept to illustrate the ease of matching the data with the proposed metamodel.

Keywords: Process-Instance Metamodel, Data Model, Model Mapping, Domain Knowledge, Business-Process Data Exploitation

1 Introduction

Companies today produce a great amount of data in a daily basis that accurately reflects their business processes. This data holds special interest for the study and optimization of these business processes. However, increasingly, the data comes from heterogeneous sources with different formats and structures (relational databases, NoSQL, APIs, data warehouses...). This causes the analysis and exploitation of data to become highly time-consuming, since many solutions are ad-hoc solutions, and, as a consequence, they have to be adapted depending on the techniques to be applied. This data-preparation process constitutes the most significant barrier to be improved and one of the highest time-consuming tasks in data-analysis projects [38].

In this paper, we propose the use of a Business-Process Instance Metamodel as an intermediate layer to specify the relation between the domain-specific data produced and its meaning in a business process, thereby facilitating how it can be

exploited by business analysis techniques. Our research goal is *the simplification of the data analysis by making independent the structures of data production from data consumption*. The approach is based on the definition of mappings between data sources and the business process concepts specified in the Business-Process Instance Metamodel. The benefits obtained by using an intermediate metamodel include the reduction of the analysis time and the exploitation of data in a more appropriate way [24]. In fact, the use of the intermediate metamodel is a benefit itself, since it provides a standard way to access business-process data and also improves the interoperability among organizations.

The paper is organized as follows: Firstly, Section 2 gives a general overview of the approach and introduces how the proposed Business Instance Metamodel may be employed in different contexts. Secondly, Section 3 describes a case study that has been used to test the approach. Thirdly, the metamodel is detailed in Section 4. Section 5 presents the tool implemented as a proof of concept. The tool helps to define the matching between an OracleTM database and the Process Instance Metamodel. Section 6 then surveys other existing approaches that exploit data in different contexts. Finally, conclusions are drawn and further work is outlined in Section 7.

2 Approach overview and contributions

Business process data exploitation depends highly on the technology that supports the business data storage as well as how the data is structured. As a consequence, no standard approach exists that can exploit any type of source, and it is therefore necessary to develop ad-hoc data analysis mechanisms adapted to both data technology and model. Thus, for example, the necessary data preparation to generate an event log to be employed by a process-mining tool is totally different depending on whether data is stored in a relational database or whether it comes from a cloud data source or a data warehouse. Moreover, this generation process also depends on the specific data model.

In order to render data exploitation as technology-agnostic regarding its data structure [31], our approach is inspired by the guidelines provided by Model-Driven Architecture in such a way that we propose a Business Process Instance Metamodel that allows us to separate produced data structure from data analysis solutions. In other words, the Business-Process Instance Metamodel can be seen as an intermediate artifact that allows applications that produce business process data to become independent from those applications that consume such data [15]. Figure 1 depicts how the process instance metamodel acts as an interface between data producers and consumers.

The following subsections describe how the metamodel proposed in the paper could be used in different contexts under the previous viewpoints.

2.1 Data production viewpoint

This viewpoint represents those contexts in which business-process data is produced. This data is mapped into the metamodel in order to be analysed. The

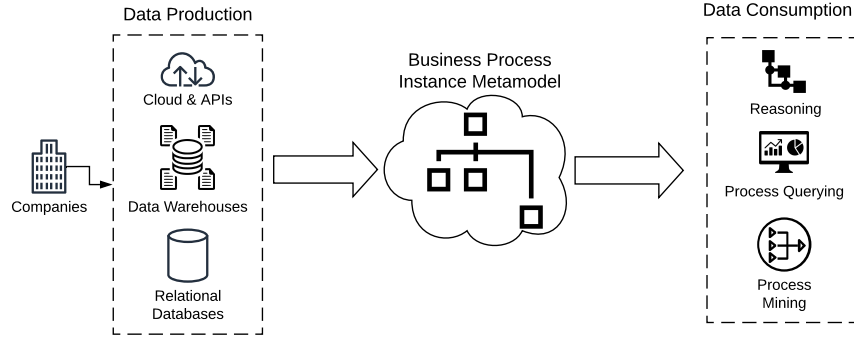


Fig. 1. Approach overview

left-hand side of Figure 1 depicts three different contexts of use related to formatting the produced data: APIs located in cloud systems; data warehouses; and relational databases.

Regarding the context of cloud systems and APIs, it should be borne in mind that companies use more and more cloud data sources which rely on complex structures such as JSON, whose objects might have different properties. This data usually complements the specific company data with data from payments, geolocation, etc. Furthermore, APIs can be used to cross-reference information (such as weather and macroeconomics) and cloud systems usually perform some computation over data, which results in new data sources.

Data warehouse based applications are an especially important context of use, above all when the business process data produced is used as input for process-mining and process-discovery techniques, since data warehouses commonly store historical information of the companies as well as many details regarding the timing of that information. Note that process-mining techniques need historical information in order to rebuild a consistent process model.

Finally, a common context of use from the data production viewpoint is related to applications that use relational databases. In fact, the case study introduced in this paper is based on a relational database. Relational databases also provide one of the most widely used scenarios for process querying, as detailed in Section 6.

2.2 Data consumption viewpoint

This viewpoint represents those contexts in which data obtained from business processes is exploited. The right-hand side of Figure 1 depicts three different contexts of use related to data exploitation: reasoning by using defined ontologies; process queries for the creation of dashboards that improve decision-making; and event log generation for process mining.

In the reasoning context of use, applications use semantization techniques, which are based on an ontology as a formal specification. Many business process

semantization approaches link concepts from domain ontologies with business process elements that are grounded in a business process ontology [19]. Thus, reasoning is used to derive facts from the ontology, which are not expressed explicitly. The elements of this business process ontology can be mapped to the concepts defined in our metamodel, since ontologies and metamodels are closely related [29].

Process queries improve decision-making, for example, by creating dashboards to exploit the business process data [35]. As far as our metamodel covers information related to process definitions, process instances, activities and activity instances and their attributes, we can ask for durations, sequences of activities, frequencies of executions, and can identify bottleneck activities, study deviated instances of activities/processes, etc. As a consequence, this information can be used to infer Key Performance Indicators which facilitate the monitoring of the process [32].

Finally, in the context of event logs for process-mining techniques, applications may not be able to produce event logs or may fail to produce them in the correct format [2]. Obtaining logs from the instances of our metamodel implies listing the activity instances ordered in terms of execution time, grouping them by process instance, and producing files with XES-formatted data. Note that the process to perform this transformation must be adapted to the data source in order to obtain the correct data output for processing. Moreover, it must be considered that the company systems work with various data sources at the same time. From the consumer viewpoint, all these details must be transparent by means of an appropriate transformation.

3 Case Study

This section presents the case study carried out to test the validity of the proposal. Data has been obtained from the execution of a business process within a prominent aerospace company. Although the company has no Business Process Management System, it does have a proprietary system that is supported by a relational database. The core business of the company consists of the assembly of aircraft and their modules. An aircraft undergoes a huge process of engineering, components designing, components construction that must be followed to assemble the final product to be ready to fly. When a new aircraft is about to be released, it must be tested several times. Figure 2 depicts the testing process of the aircraft modules.

When the aircraft testing process starts, the *New aircraft order arrives* activity begins, and, as a consequence, its data is introduced in the AIRCRAFT table (Figure 3). Bear in mind that an aircraft passes through different stations, and that in each station, the aircraft modules must pass a set of tests that are composed of different sections. Thereby the execution of each test brings about the execution of every section in that test.

The *Configuration of the test sections that the aircraft must pass* activity consists of scheduling the different test sections that must be executed on the

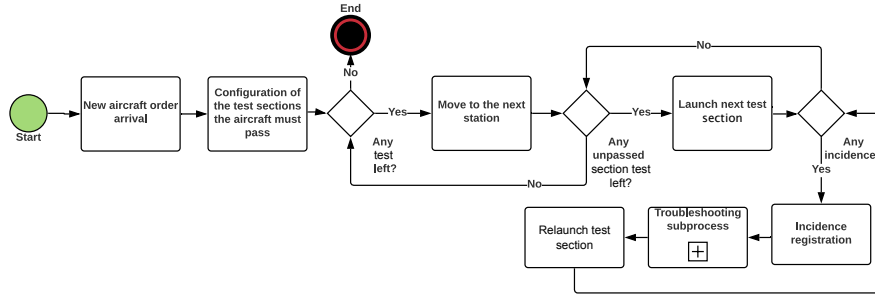


Fig. 2. Aircraft testing process

modules of each aircraft. This information is stored in the TEST_SECTIONS table. After the test configuration, the aircraft is driven to the first station to start the test execution (*Move to the next station* activity) and the set of tests are launched (*Launch next test* activity).

Every time a test is launched, a row is inserted in the TEST_EXECUTION table and another row is inserted in TEST_SECTION_EXECUTIONS (one for each test section executed). Furthermore, if the test fails, usually due to some kind of incidence, the *Incidence Registration* activity starts and the *Troubleshooting subprocess* is triggered. As a consequence: first, the original row is modified in order to register both the moment when the test failed and the status of the test after being executed; and second, a new row is inserted in the TEST_INCIDENCES table. Note that if an incidence appears during the execution of a section, it must be solved successfully before the airplane is released. As a consequence, the full test needs to be repeated, regardless of which section the error appeared, since the success of some parts of a test may depend on other parts of the test. Thus, when the *Troubleshooting subprocess* finishes, then the whole test is relaunched (*Relaunch test* activity).

Due to the lack of a Business Process Management System, every test execution is stored in detail in the database, whereby information related to aircraft, tests, stations, etc. is held. Thus, each time a new test is launched, the data involved is stored, such as timestamps related to every action, the status of the test, when the test has finished, and which sections were executed. The data model which supports this process is composed of the following tables (Figure 3):

- AIRCRAFT: This stores information about the tested aircraft. As a consequence, a row is inserted in this table each time an airplane is going to be tested. The table stores: the type of the airplane, the model of the airplane, the name, the start date, and the end date scheduled.
- TESTS: This stores information about the collection of tests defined in the system: the name of the test, the creation date, and its type.
- TEST_SECTIONS: This stores the sections of each test that each airplane should pass and the order in which the sections should be executed.
- TEST_EXECUTIONS: This table stores information about a test execution. Thus, a row is inserted in this table each time a test is launched. The table

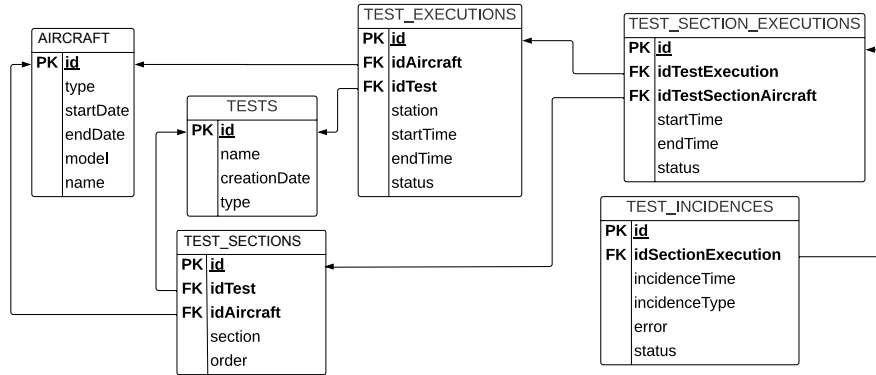


Fig. 3. Relational Model of the Aircraft Assembly Process

stores: the station in which the test was executed, the time when the test execution started, the time when the test execution ended, and the test status after finishing.

- TEST_SECTION_EXECUTIONS: This stores information about the execution of a test section. Note that each test is split into different sections that are in charge of preparing the execution or checking certain variables. The table stores the timestamp when the test section execution started, the timestamp when the test section execution ended, and the final status.
- TEST_INCIDENCES: This stores information about the incidences produced during test executions. As a consequence, a row is inserted in this table when an error appears while running a test. The table stores the time when the incidence appeared, the incidence type, the status of the incidence, and the error that caused the incidence.

4 Process Instance Metamodel

The Business Process Instance Metamodel is detailed in Figure 4. The metamodel has been specified with EMF [37]. Note that it is a very simple model which is mainly centred on the most basic entities related to business process instances together with their attributes. A previous extension of this metamodel was published in [15].

The root of the metamodel is the **ProcessEngine** metaclass and represents the BPMS or software application that is in charge of process execution. The process engine can be in charge of several processes. The **ProcessDefinition** metaclass represents the formal definition of the process, that is, what we call the Business Process Model. The attributes are:

- *id*. Key identifier of the process.
- *name*. Name of the process model.
- *description*. Description of the process model.
- *suspended*. This attribute represents whether a process is suspended (temporarily disabled). While it is suspended, the process is not instantiated.

A business process is composed of different activities and the **Activity** meta-class models these activities. The attributes are:

- *id*. Key identifier of the activity.
- *name*. Name of the activity.
- *description*. Description of the activity.

One business process can be executed many times and the **ProcessInstance** metaclass models these executions or instances. The attributes are:

- *id*. Key identifier of the process instance.
- *ended*. A flag (Boolean) indicating whether the instance is still running.
- *suspended*. A flag (Boolean) indicating whether the instance is suspended.
- *startUser*. The user who started the instance process.
- *duration*. Time spent on process execution. This information is recovered when the process has ended.
- *startTime*. This represents when the instance process started.
- *endTime*. This represents when the instance process ended.

Finally, the **ActivityInstance** metaclass represents the execution of an activity and is related to the **Activity** metaclass (note that an activity may be executed many times) and to the **ProcessInstance** metaclass (an activity may be executed in the context of different business processes). The attributes are:

- *id*. Key identifier of the activity instance.
- *startTime*. This represents when the instance activity started.
- *endTime*. This represents when the instance activity ended.
- *duration*. Time spent on activity execution. This information is recovered when the activity ends.
- *cancelled*. A flag (Boolean) indicating whether the instance is cancelled.
- *assignee*. The user assigned to the execution of the activity.

Note that this metamodel allows us to exploit business data in different contexts, independently of the storage technology and how the information is structured. We only need to define mappings from the concrete technology to the Process Instance Metamodel. Therefore, the information stored as instances of the Business Process Instance Metamodel may be used to generate event log traces (both in XES or MXML format), to be queried for decision-making or to be semantized thereby enabling the application of reasoning techniques.

4.1 Mapping the metamodel and the case study models

This section explains how the Process Instance Metamodel is used, from the data production viewpoint, in the context of the case study introduced in Section 3.

The **Process Definition** metaclass is related to the *Testing Aircraft Process* shown in Figure 2. Each instance of that process is mapped into the **Process Instance** metaclass (see Figure 5). As a consequence, when a new row is inserted into the TEST_EXECUTIONS table, then an instance of the **Process Instance**

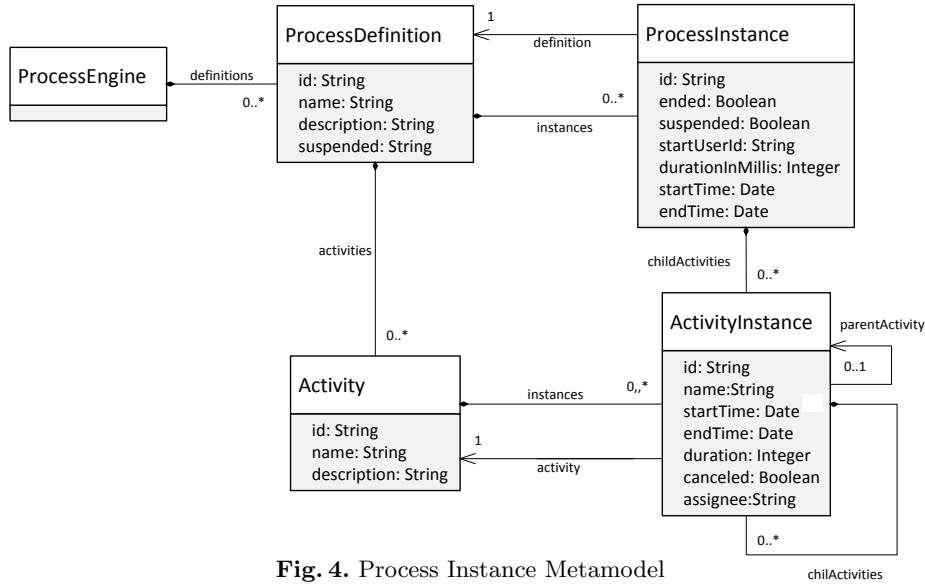


Fig. 4. Process Instance Metamodel

metaclass is created. It should be also noted that the *startTime* and *endTime* attributes are mapped to the *startTime* and *endTime* fields of that table.

Since there are different activities, such as the *Launch next test* or the *Incidence registration* activities, there are mappings between the **Activity Instance** metaclass and different tables (see Figure 5). Thus, an instance of the **Activity Instance** metaclass is created each time a new row is inserted into the `TEST_SECTION_EXECUTIONS`, `TEST_INCIDENCES`, or `TEST_EXECUTIONS` tables. The expected *startDate* of an assembly process of an airplane is stored in the `AIRCRAFT` table. However, the real start time is represented by the oldest *startTime* of the `TEST_EXECUTION` related to a specific *idAircraft* that indicates the true beginning of the process.

Finally note that, although in this case study every mapping is related to the insertion of a row into a table, this is not the only possible scenario. The mappings of the *Activity Instance* metaclass could also be related to editions of rows. Thus, for example, the *incidenceType* field could be mapped to different activities if the various types of incidences lead to different subprocess executions.

5 Proof of Concept Implementation

In order to support our proposal, a proof-of-concept has been implemented to illustrate the mapping process between the business data repository and our Process Instance Metamodel. One of the main benefits of our proposal is that the mapping itself is that which remains after performing the matching between data repositories and the metamodel, instead of the mapped data, as in other approaches [7]. Thus, every new item of data registered in the repository is automatically available in the Process Instance Metamodel, and it is able to perform

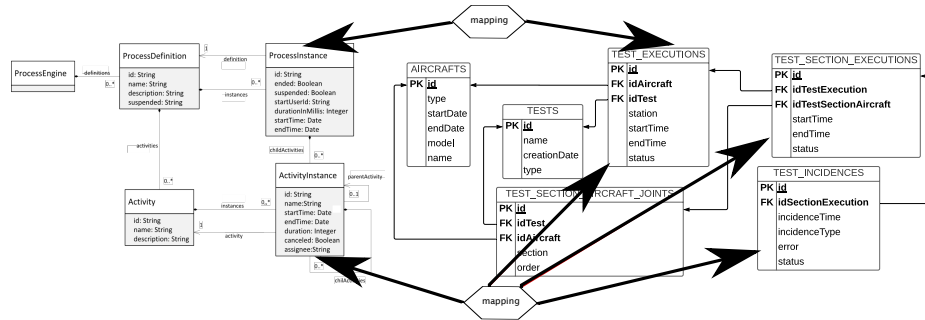


Fig. 5. Mappings between metamodel and produced data

business process analysis not just after the process execution, but also whilst the execution is happening, which is key in some cases. This provides agility and the opportunity of making decisions during the business process instance. Another considerable benefit of this approach, since it is not tied to any specific data consumption context (process mining, process querying, reasoning over processes...), is the ability to exploit the business process data simultaneously in different contexts. We could use it to generate event logs while visualizing statistic data on a dashboard, as is showed in the demo video recorded using the proof-of-concept tool. This provides versatility to the way business process data can be used by the companies without the necessity of performing specific ad-hoc applications or data transformations for each context or goal. The proof of concept has been developed as a web application and implements a simple dashboard where we can compare visually different instances, cross-check statistics information related to our instances, and watch the evolution of our process data over time. Furthermore, a video demo shows how the tool is able to automatically analyse the structure of the data repository and how the mapping process can be executed in an easy way. Figure 6 shows a screenshot that captures the mapping definition process. The proof-of-concept has been developed as a result of collaboration with a company whose data could never be publicly available. However, the software is available for application to other cases. Moreover, a video demo using the tool has been recorded to facilitate the use of the tool. For any further details regarding the tool, check the website <http://www.idea.us.es/portfolio-item/process-data-matching-tool/>.

6 Related Work

We will limit the scope of this section to the approaches related to business processes whose focus is on the exploitation of data generated during process execution. The approaches can be classified into: approaches whose goal is the semantization of process data in order to use ontology-based reasoning; approaches whose goal involves the querying of process data to aid in decision-making in business process scenarios; and approaches whose goal is the creation of exe-

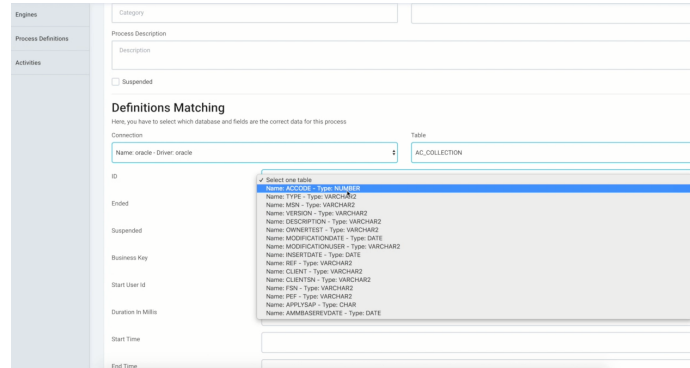


Fig. 6. Developed tool as proof of concept

cution traces that are used as input for process discovery algorithms. Bear in mind that these different scenarios consume data in different formats, and certain conversion and formatting tasks can be tedious and complex since data can be stored in heterogeneous repositories [7]. The following subsections give a general overview of the state-of-the-art of the aforementioned contexts.

6.1 Approaches that consume data for reasoning

This group introduces the incorporation of data ontology in order to support functionalities of a more intelligent nature, such as process reasoning. In general, these approaches augment existing processes with semantic annotations, so that formal reasoning techniques can be applied. There are several techniques of semantization of Business Processes [20]:

- The SUPER project [40] formally represents business process concepts by means of a stack of five ontologies and provides a modelling environment for the enrichment of existing processes with semantic annotations.
- The SAP AG system [6] integrates semantic descriptions and business process artifacts by linking concepts from an ontology and elements of business process models.
- The Prosecco project [28] provides a unified dictionary of business concepts to help with the systems integration and takes into account semantic dependencies between business process models and rule models.
- Finally, there is also a group of techniques that could be used for semantization of business processes that are not process specific, for example, since many business process execution environments use REST interfaces, certain techniques for semantization of REST interfaces could be used. However, these kinds of techniques remain out of the scope of this study.

The approach that is most closely related to ours is the SAP AG system in the sense that the domain ontology and the business process model are integrated by means of links; however, that system is focused on semantic sources.

6.2 Approaches that consume data for querying

The approaches in this group query process data to help in decision-making in business process scenarios [35]. There are many different approaches to query process data. According to [33], these approaches can be classified depending on the type of behaviour models they can take as input:

- **Methods that operate over event logs.** This group includes approaches such as CRG [23], eCRG [21], DAPOQ-Lang [27], FPSPARQL [5], and PIQL [30]. The approach most closely related to ours is DAPOQ-Lang because it is built on top of the metamodel proposed in [26]. The main difference is that their metamodel subsumes two different viewpoints, (process, and data), while in our approach the viewpoints are defined with different metamodels, in such a way that we have applied the principle of the separation of concerns.
- **Methods that operate over process model specifications.** This group includes a set of approaches that were originally conceived for querying conceptual models, and, as a consequence, they are also useful for querying process models, and another set of approaches that were originally conceived for querying process model collections. The first subgroup includes approaches such as DMQL [11], GMQL [10], and VMQL [39]. The second subgroup includes approaches such as BPMN-Q [3], BPMN VQL [12], BPSL [22], CRL [13], Descriptive PQL [18], IPM-PQL [9], and PPSL [14]. The approaches most closely related to ours are DMQL and GMQL in the sense that they define a generic metamodel to cover all types of modelling languages. This metamodel can be seen as a way to decouple query languages from modelling languages.
- **Methods that operate over behaviours encoded in process models.** This group includes approaches such as APQL [16], BQL [17], QuBPAL [36], and PQL [34]. All of these approaches are based on the definition of semantic relations between tasks. The most closely related to our approach is that of APQL, in the sense that the proposed language is independent of the notation used to specify process models.
- **Methods that operate over collections that may include process models and/or event logs.** This group includes approaches such as BPQL [25] and NP-QL [4]. This group is the least related to our approach.

6.3 Approaches that consume data for the creation of execution traces

There are several approaches that consume data to create execution traces. Thus, in [7], a conversion from a data source in table format to an event log is proposed. The approach is tested by means of two case studies: an SAP system, and a set of CSV files that are the result of exporting a database.

In [8], a framework to extract XES event log information from legacy relational databases is proposed. The extraction is made by defining two ontologies,

one that represents the domain of interest and another one that represents event logs. The domain ontology is linked to the legacy data by using the ontology-based data access paradigm (OBDA), and the concepts defined in the event log ontology are mapped into the concepts defined in the domain ontology by means of annotations.

In [1], a framework to unify existing approaches of process discovery from event logs is introduced. The framework is based on event log and process model abstractions, and, as a consequence, it only includes concepts from event log and process viewpoints.

In [26], a metamodel is proposed to query the data from different sources in a standardized way. Thus, the metamodel allows the decoupling of the application of the data analysis techniques. The proposed metamodel includes concepts related to two different viewpoints: process and data. Furthermore, in order to be compatible with the XES metamodel, the proposed metamodel also includes events and cases. Mappings from data sources of three different scenarios (database redo logs, in-table version storage, and SAP-style change tables) to the proposed metamodel are formalized.

Note that all these approaches share at least one of the following two weak points covered by our approach: 1) different data sources are considered, but relational databases and/or tabular formats are taken for granted [7] [26]; and 2) the focus is on the results (event logs) instead of on the means (relations between stored data and events data), which forces the process of mapping to be repeated each time a new log needs to be generated [7] [8] [1].

7 Conclusions and further work

Due to the existence of multiple techniques based on Business Process Analysis, this paper introduces the necessity of the utilization of a Business Process Instance Metamodel as a bridge between data sources and data exploitation techniques.

As we have seen, this metamodel provides the first step towards isolating the process data produced and the objective of its analysis. This is especially relevant in scenarios where different types of business process data exploitation are going to be applied and/or scenarios where different data sources with various formats are working together. Thereby, this paper shows how the use of an intermediate metamodel can help to standardize the exploitation of business process data by defining a common infrastructure that may be used in various contexts of business process analytics.

In terms of further work, how to query the metamodel in order to extract the required information in the correct format constitutes the next challenge to tackle. This challenge brings about the extension of the metamodel to encapsulate other existing proposals of consumers and producers, while it maintains the abstraction level to ensure adaptability to any business regardless of its sector or domain knowledge.

Finally, we consider it interesting to enrich the way of defining the matching, by making the tool more flexible and by allowing the building of processes of a more complex nature and the exploitation of more complex data sources.

Acknowledgements

This work has been partially funded by the Ministry of Science and Technology of Spain by the Project ECLIPSE (RTI2018-094283-B-C33 and TIN2016-75394-R) and the European Regional Development Fund (ERDF/FEDER).

References

1. van der Aalst, W.M.P.: Process discovery from event data: Relating models and logs through abstractions. *Wiley Interdisc. Rev.: Data Mining and Knowledge Discovery* 8(3) (2018), <https://doi.org/10.1002/widm.1244>
2. van der Aalst, W.M.P.: Process mining: The missing link, pp. 25–52. Springer Berlin Heidelberg, Berlin, Heidelberg (2016), https://doi.org/10.1007/978-3-662-49851-4_2
3. Awad, A.: BPMN-Q: a language to query business processes. In: *Enterprise modelling and information systems architectures*. pp. 115–128 (2007)
4. Beeri, C., Eyal, A., Kamenkovich, S., Milo, T.: Querying business processes. In: *Proceedings of the 32nd International Conference on Very Large Data Bases*. pp. 343–354. VLDB '06, VLDB Endowment (2006), <http://dl.acm.org/citation.cfm?id=1182635.1164158>
5. 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.) *Business Process Management*. pp. 281–297. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
6. Born, M., Dörr, F., Weber, I.: User-friendly semantic annotation in business process modeling. In: Weske, M., Hacid, M.S., Godart, C. (eds.) *Web Information Systems Engineering – WISE 2007 Workshops*. pp. 260–271. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
7. Buijs, J.: Mapping data sources to XES in a generic way. Department of Mathematics and Computer Science, Eindhoven University of Technology (2010)
8. Calvanese, D., Montali, M., Syamsiyah, A., van der Aalst, W.M.P.: Ontology-driven extraction of event logs from relational databases. In: *Business Process Management Workshops*. pp. 140–153 (2015), http://dx.doi.org/10.1007/978-3-319-42887-1_12
9. Choi, I., Kim, K., Jang, M.: An XML-based process repository and process query language for integrated process management. *Knowledge and Process Management* 14(4), 303–316, <https://onlinelibrary.wiley.com/doi/abs/10.1002/kpm.290>
10. Delfmann, P., Breuker, D., Matzner, M., Becker, J.: Supporting information systems analysis through conceptual model query. the diagramed model query language (DMQL). *Communications of the Association for Information Systems* 37 (2015)
11. Delfmann, P., Steinhorst, M., Dietrich, H.A., Becker, J.: The generic model query language GMQL. conceptual specification, implementation, and runtime evaluation. *Information Systems* 47, 129 – 177 (2015)

12. Di Francescomarino, C., Tonella, P.: Crosscutting concern documentation by visual query of business processes. In: Ardagna, D., Mecella, M., Yang, J. (eds.) *Business Process Management Workshops*. pp. 18–31. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
13. Elgammal, A., Turetken, O., van den Heuvel, W.J., Papazoglou, M.: Formalizing and applying compliance patterns for business compliance. *Software & Systems Modeling* 15(1), 119–146 (Feb 2016), <https://doi.org/10.1007/s10270-014-0395-3>
14. Foerster, A., Engels, G., Schattkowsky, T.: Activity diagram patterns for modeling quality constraints in business processes. In: Briand, L., Williams, C. (eds.) *Model Driven Engineering Languages and Systems*. pp. 2–16. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
15. 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.) *Business Process Management Workshops*. pp. 757–769. Springer International Publishing, Cham (2018)
16. ter Hofstede, A.H.M., Ouyang, C., La Rosa, M., Song, L., Wang, J., Polyvyanyy, A.: APQL: A process-model query language. In: Song, M., Wynn, M.T., Liu, J. (eds.) *Asia Pacific Business Process Management*. pp. 23–38. Springer International Publishing, Cham (2013)
17. Jin, T., Wang, J., Wen, L.: Querying business process models based on semantics. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) *Database Systems for Advanced Applications*. pp. 164–178. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
18. Kammerer, K., Kolb, J., Reichert, M.: PQL. a descriptive language for querying, abstracting and changing process models. In: Gaaloul, K., Schmidt, R., Nurcan, S., Guerreiro, S., Ma, Q. (eds.) *Enterprise, Business-Process and Information Systems Modeling*. pp. 135–150. Springer International Publishing, Cham (2015)
19. Kluza, K., Kaczor, K., Nalepa, G.J., layski, M.: Opportunities for business process semantization in open-source process execution environments. In: *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*. vol. 5, pp. 1307–1314 (Sep 2015)
20. Kluza, K., Nalepa, G.J., Ślęzyński, M., Kutt, K., Kucharska, E., Kaczor, K., Luszpaj, A.: Overview of Selected Business Process Semantization Techniques, pp. 45–64. Springer International Publishing, Cham (2017), https://doi.org/10.1007/978-3-319-47208-9_4
21. Knuplesch, D., Reichert, M.: A visual language for modeling multiple perspectives of business process compliance rules. *Software & Systems Modeling* 16(3), 715–736 (Jul 2017), <https://doi.org/10.1007/s10270-016-0526-0>
22. Liu, Y., Muller, S., Xu, K.: A static compliance-checking framework for business process models. *IBM Systems Journal* 46(2), 335–361 (2007)
23. Ly, L.T., Rinderle-Ma, S., Dadam, P.: Design and verification of instantiable compliance rule graphs in process-aware information systems. In: Pernici, B. (ed.) *Advanced Information Systems Engineering*. pp. 9–23. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
24. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P., Toussaint, P.J.: Guided process discovery - A pattern-based approach. *Inf. Syst.* 76, 1–18 (2018), <https://doi.org/10.1016/j.is.2018.01.009>
25. Momotko, M., Subieta, K.: Process query language: A way to make workflow processes more flexible. In: Benczúr, A., Demetrovics, J., Gottlob, G. (eds.) *Advances*

- in Databases and Information Systems. pp. 306–321. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
26. González López de Murillas, E., Reijers, H.A., van der Aalst, W.M.P.: Connecting databases with process mining: a meta model and toolset. *Software & Systems Modeling* (Feb 2018), <https://doi.org/10.1007/s10270-018-0664-7>
 27. de Murillas, E.G.L., Reijers, H.A., van der Aalst, W.M.P.: Everything you always wanted to know about your process, but did not know how to ask. In: *BPM Workshops 2016*. pp. 339–351 (2016)
 28. Nalepa, G.J., layski, M., Kutt, K., Kucharska, E., uszpaj, A.: Unifying business concepts for smes with prosecco ontology. In: *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*. pp. 1321–1326 (Sep 2015)
 29. Parreiras, F.S.: chap. *Marrying Ontology and Model-Driven Engineering*. Wiley and Sons (2012)
 30. Pérez-Álvarez, J.M., Gómez-López, M.T., Parody, L., Gasca, R.M.: Process instance query language to include process performance indicators in DMN. In: *2016 IEEE 20th International Enterprise Distributed Object Computing Workshop (EDOCW)*. pp. 1–8 (Sep 2016)
 31. Pérez-Álvarez, J.M., Gómez-López, M.T., Eshuis, R., Montali, M., Gasca, R.M.: Verifying the manipulation of data objects according to business process and data models. *Knowledge and Information Systems* (Jan 2020), <https://doi.org/10.1007/s10115-019-01431-5>
 32. Pérez-Álvarez, J.M., Maté, A., Gómez-López, M.T., Trujillo, J.: Tactical business-process-decision support based on KPIs monitoring and validation. *Computers in Industry* 102, 23–39 (2018), <https://doi.org/10.1016/j.compind.2018.08.001>
 33. Polyvyanyy, A.: *Business Process Querying*, pp. 1–9. Springer International Publishing, Cham (2018), https://doi.org/10.1007/978-3-319-63962-8_108-1
 34. Polyvyanyy, A., Corno, L., Conforti, R., Raboczi, S., Rosa, M.L., Fortino, G.: *Process querying in Apromore*. In: *BPM 2015 Demo Track* (2015)
 35. Polyvyanyy, A., Ouyang, C., Barros, A., van der Aalst, W.M.P.: Process querying: Enabling business intelligence through query-based analytics. *Decision Support Systems* 100, 41–56 (2017), <https://doi.org/10.1016/j.dss.2017.04.011>
 36. Smith, F., Missikoff, M., Proietti, M.: *Ontology-based querying of composite services*, pp. 159–180. Springer Berlin Heidelberg, Berlin, Heidelberg (2012), https://doi.org/10.1007/978-3-642-32439-0_10
 37. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: *EMF: Eclipse Modeling Framework 2.0*. Addison-Wesley Professional, 2nd edn. (2009)
 38. Stodder, D.: *Improving data preparation for business analytics. applying technologies and methods for establishing trusted data assets for more productive users*. Best Practices Report Q3 2016 pp. 19–21 (2016)
 39. Strrle, H.: VMQL: A visual language for ad-hoc model querying. *Journal of Visual Languages Computing* 22(1), 3 – 29 (2011), special Issue on Visual Languages and Logic
 40. Wetzstein, B., Ma, Z., Filipowska, A., Kaczmarek, M., Bhiri, S., Silvestre Losadaand José-Manuel López-Cobo, L.C.: *Semantic business process management: A lifecycle based requirements analysis*. In: *Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management*. vol. 251. CEUR Workshop Proceedings (2007)