Business Process and Organizational Data Quality Model (BPODQM) for Integrated Process and Data Mining

Francisco Betancor, Federico Pérez, Adriana Marotta, and Andrea Delgado(✉)

Instituto de Computación, Facultad de Ingeniería, Universidad de la República,
J. Herrera y Reissig 565, Montevideo, Uruguay
{francisco.betancor.pallas,federico.andres.perez,
amarotta,adelgado}@fing.edu.uy

Abstract. Data Quality (DQ) is a key element in any Data Science project to guarantee that its results provide consistent and reliable information. Both process mining and data mining, as part of Data Science, operate over large sets of data from the organization, carrying out the analysis effort. In the first case, data represent the daily execution of business processes (BPs) in the organization, such as sales process or health process, and in the second case, they correspond to organizational data regarding the organization's domain such as clients, sales, patients, among others. This separate view on the data prevents organizations from having a complete view of their daily operation and corresponding evaluation, probably hiding useful information to improve their processes. Although there are several DQ approaches and models for organizational data, and a few DQ proposals for business process data, none of them takes an integrated view over process and organizational data. In this paper we present a quality model named Business Process and Organizational Data Quality Model (BPODQM) defining specific dimensions, factors and metrics for quality evaluation of integrated process and organizational data, in order to detect key issues in datasets used for process and data mining efforts.

Keywords: Data quality model · Process mining and Data mining · Data science · Integrated process and organizational data

1 Introduction

In last years the need to exploit available data in organizations has increased considerably, due to the continuous generation of data from different and interconnected sources. The complexity of socio-technical systems connecting people, things, data and business processes (BPs), integrating heterogeneous technologies and elements, also increases the complexity of integrating and generating useful datasets, as well as their management. Both Process mining [2] and Data mining [15], as part of Data Science [12], operate over large datasets from the

© Springer Nature Switzerland AG 2021
A. C. R. Paiva et al. (Eds.): QUATIC 2021, CCIS 1439, pp. 431–445, 2021.
https://doi.org/10.1007/978-3-030-85347-1_31

organization carrying out the analysis effort. In the first case, data represent the daily execution of BPs in the organization, such as sales process, health process, and in the second case, they correspond to organizational data regarding the organization's domain such as clients, sales, patients, among others. Most organizations manage their process and organizational data separately. This separate view on the data prevents organizations from having a complete view of their daily operation and corresponding evaluation, hindering the recovery of useful information to improve their processes.

On the one hand, Data Mining [15] aims at analyzing large datasets in search for general rules, providing predictions and behavior patterns based on the input data. Data Mining techniques are often classified as descriptive or predictive. Descriptive techniques include clustering and association rules to characterize data sets, while Predictive techniques are classification and regression, in the first case mainly decision trees and neural networks, and in the second regression functions, among others. On the other hand, Process Mining [2] focus on large datasets that are specific from process execution, in order to provide insight on process execution. Process mining provides three main perspectives: i) discovering BP models from event logs i.e. generating process models from execution data; ii) process conformance by checking BP models against the real execution in event logs; and iii) enhancing BP models with extra information such as participating roles and resources. Also, performance execution analysis can be performed such as bottlenecks, duration of process cases, average, etc.

Data Quality (DQ) [5,18] is a key element in any Data Science project to guarantee that its results provide consistent and reliable information. A DQ model defines dimensions conceptualizing different aspects of quality, which are composed of factors defining a specific quality aspect within a dimension, and metrics to specify the way that a factor is measured. Very well-known dimensions of data quality include *Accuracy*, *Consistency*, *Completeness*, *Uniqueness*, but several others can be taken into account depending on the context and domain at hand. Although there are several DQ approaches and models for organizational data, and a few DQ proposals for BPs data, none of them takes an integrated view over process and organizational data.

Business Process Management (BPM) [3,11,19] provides organizations with the basis for managing their BPs, which can be supported by traditional Information Systems or with process platforms such as BPM Systems [6]. In most BPMS settings, the process engine is in charge of BP execution i.e. the control flow defined within the process model registering all events in a database (schema) of its own, and the organizational data that is managed within the BP cases (i.e. instances) are registered in another (or several) database/s (schema) where other systems also register and query common organizational data i.e. clients, patients, etc. A key step towards BPs continuous improvement [10] is to be able to proactively assess their real execution to provide business people with Business Intelligence support for evidence-based decision making. To provide a complete view on data, process and organizational data must be integrated and analyzed in an integrated manner. To generate such integrated datasets, for each activity (human or automated) executed within a BP case, the organizational data that was involved in the execution of the activity must be related back to the corresponding activity.

In previous works we have addressed this problem by defining an integrated framework [9] and working with integrated process and organizational data [8], as well as identifying the need for a DQ model to be applied over the data before the mining effort can be carried out. In this paper we present such DQ model that we named Business Process and Organizational Data Quality Model (BPODQM), which defines specific dimensions, factors and metrics for quality evaluation of integrated process and organizational data, in order to detect key issues in integrated data sets used for process and data mining efforts.

The rest of this document is organized as follows: in Sect. 2 we present concepts and definitions and in Sect. 3 we introduced an example and preliminaries. In Sect. 4 we present our proposal of a BPO Data Quality Model (BPODQM), and in Sect. 5 we show an application of our proposal as proof of concept. In Sect. 6 we discuss other approaches to DQ for process and organizational data, and finally in Sect. 7 we present some conclusions.

2 Background

The model we present in this work is supported by some basic concepts of the Data Quality (DQ) area, which are introduced in this section. We also present concepts and definitions regarding BP event logs that contain process data for process mining, and the extension we have made to integrate organizational data.

2.1 Data Quality

DQ is managed with a multi-faceted approach, where the notion of quality is represented through dimensions that conceptualize different aspects of quality [5, 18]. Therefore, DQ dimensions address potential data problems, for example, a mistyping error is a data problem that is addressed by the DQ dimension *accuracy*. Additionally, as DQ dimensions are very general aspects, a second level of detail is considered, so that a DQ dimension is decomposed in DQ factors, i.e. a DQ factor is a specific quality aspect of a DQ dimension. Continuing with the example above, a mistyping error would be addressed by the DQ factor *syntactic accuracy*, which corresponds to the DQ dimension *accuracy*.

An essential part of DQ management is DQ measurement and evaluation. In order to measure DQ of a dataset, it is necessary to define metrics, which specify the way a DQ factor is measured, as well as the range of the numerical result that is obtained and the data granularity over which it is applied. For example, *syntactic accuracy* may be measured through a *string-distance* metric that is applied between a value of the dataset and the values of a referential dictionary of terms. The granularity of this metric is *data value* and the result range is [0..1]. Alternatively, if the metric obtained the percentage of correct values of a column of a data table, the granularity would be *column*. In addition, for each metric, different aggregations may be defined. An aggregation states the way that a set of measures, obtained through the metric application, are aggregated for obtaining a summarized measure that corresponds to a coarser data granularity. Continuing with the example, after applying the string-distance metric, whose granularity is data value, to all the values of a column, we can calculate an aggregation for obtaining one DQ value for the entire column.

The DQ literature shows a huge amount of DQ dimensions, factors and properties, as well as many different approaches for modeling quality characteristics. However, there is consensus in some main DQ concepts, such as dimensions and metrics, and in a basic set of dimensions, which address typical DQ problems and are present in most cases. These dimensions, which are usually represented with the same terms and have the same general semantics, are: *accuracy*, *completeness*, *consistency*, *timeliness/currency*, *uniqueness*. In recent times, where big data characteristics are present in most data scenarios, DQ dimensions related to credibility, trustworthiness and reputation, have gained much attention and relevance [14].

2.2 Process Mining and Event Logs

Process Mining [2] is a discipline within Data Science that uses and extends data mining techniques to discover information from process execution data, instead of organizational data i.e. clients, sales, etc. as data mining does. Data from process execution also come from organization's systems, where events that happen within each process instance (case) are registered as traces in a so-called event log. As mentioned before, process mining provides three main perspectives to: discover process models, check process models conformance and enhance process models with execution data.

Within BPM process mining can be used in the late phase of Evaluation in the BPs lifecycle in order to evaluate process execution and discover information based on process data to improve BPs and the operative of the organization, as described above, or can also be used in the first phase of Analysis as another input for the requirements elicitation and BP modeling. Whether there is a BP model in place for the process or not, real process data execution will help to get insights into the organization's operation. The BP model corresponds to the template of the process, where for structure process all execution possibilities should be included i.e. in domains such as banks, management, etc., while this is extremely challenging for unstructured BPs i.e. in domains such as health, knowledge management, etc. Each execution of the process is a BP case (instance) in which values of the specific execution are handled.

The input for process mining efforts is an Event log [2], which corresponds to only one process, and contains BP cases (instances) which in turn contains events that happened within the execution of the case. An event correspond to only one case and refers to work that is carried out in the organization i.e. activity, including when it is carried out (time stamps) and by whom (role, people, system). Events in a trace are ordered by execution time i.e. time stamp and a transaction type indicating the lifecycle event that is being registered e.g.

start or complete of an activity. Events can also have attributes such as cost, etc. The most common format for event logs data is the XES (eXtensible Event Stream) format [1], which is an XML format supported by process mining tools such as ProM[1]. In next Sect. 3.2 we show an example of the XES format within our extension for data integration.

3 Preliminaries

We present preliminary work over a real BP from our university that we have been working with throughout the application of our complete approach [9]. The aim of this section is to present the context of our work and the settings we are considering, as well as the extended Event log we generate from process and organizational data execution to which apply our BPODQM proposal. Although the BP has a simple control flow, the execution data i.e. event log contains process and organizational data that present cases and elements that are commonly included, so the quality example we present based on this process is both specific enough to this context and general enough to be applied to other processes.

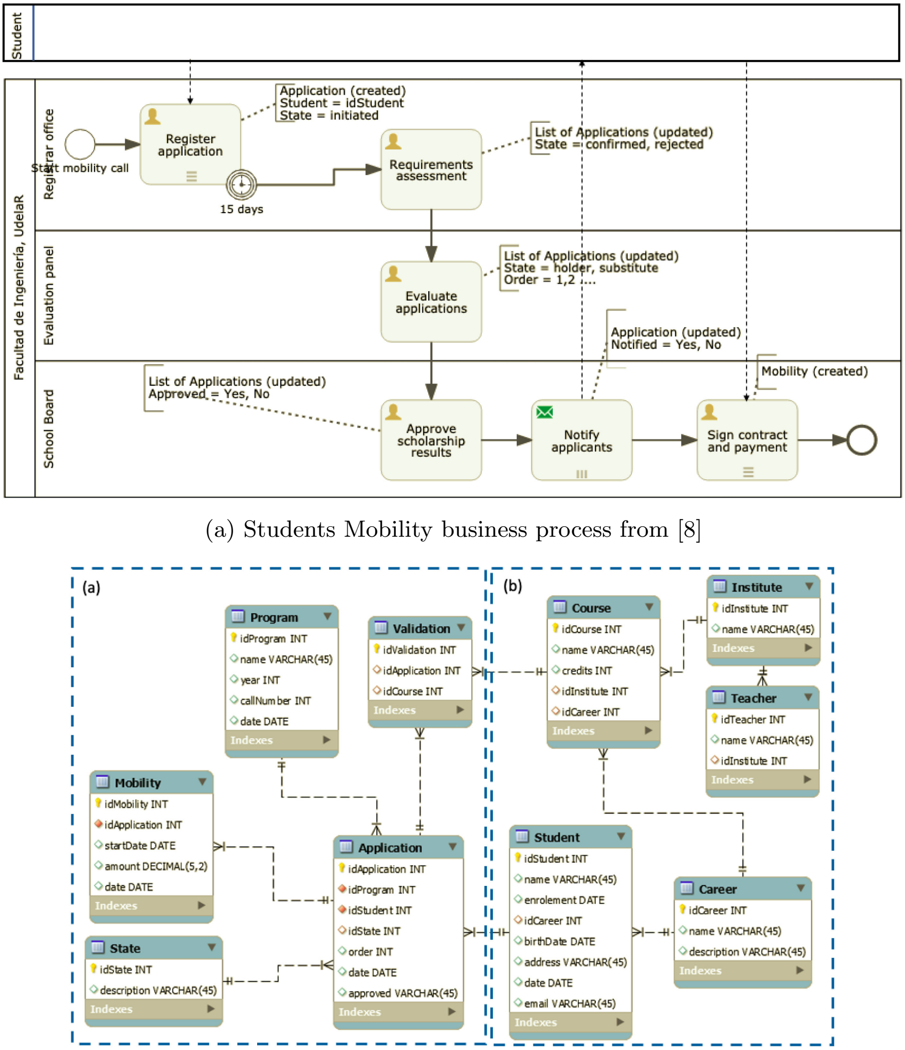
3.1 BP Model and Organizational Data Model

The BP is named "Students Mobility" [8] and deals with granting scholarships for students who apply to exchange programs to attend courses in others country's universities which participate in the mobility programs. In Fig. 1a the BP model for the Students Mobility process is shown specified in BPMN 2.0, and in Fig. 1b the data model supporting the BP, extended from [8].

The BP Students Mobility depicted in Fig. 1a shows a simple path through the complete BP, which starts when the Register office receives applications to open Mobility programs from Students, and registers them in the task "Register application". After the period for applications is closed, registered applications are checked to be compliant with the requirements from the mobility program, in the task "Requirements assessment". After that, the confirmed applications i.e. the ones that comply with the requirements of the mobility call goes through an evaluation process in the "Evaluate applicants" task, where applicants are ordered and holders are selected, as well as substitutes. Then, with the list of ordered applications the task "Approve scholarships results" approve the results, then applicants are notified in task "Notify applicants" and in task "Sign contract and payment" holders sign and get the money granted by the scholarship.

It can be seen that the organizational data (from the data model in Fig. 1b) that is managed within the process is shown as a text comment associated to the corresponding task, e.g. in the "Register application" task the table Application is accessed in order to insert a new application for the Student with identification idStudent in the State "initiated". Other values are not shown, as the validation of courses or the corresponding Program to which the application

[1] https://www.promtools.org/.

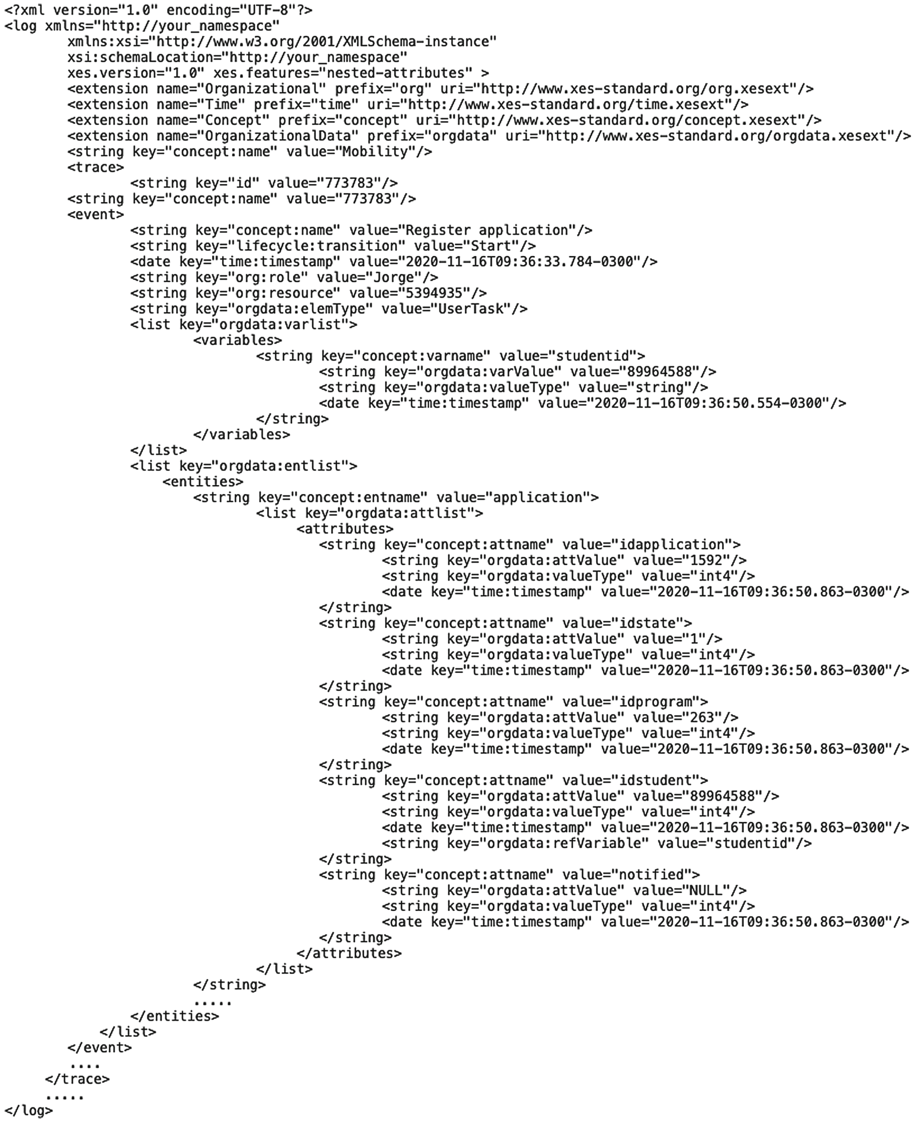


(a) Students Mobility business process from [8]

(b) Data model for the Students Mobility BP extended from [8]

Fig. 1. Students Mobility proof of concept

is being submitted. In the "Requirements assessment" task the list of applications is recovered from the organizational database, and it is updated with the confirmed or rejected result. In the subsequent tasks the list of applications is manipulated with corresponding updates and each application when the applicant is notified of the results. Finally, the last task creates a new record in the Mobility table including the start date of the mobility and the amount granted.



```
<?xml version="1.0" encoding="UTF-8"?>
<log xmlns="http://your_namespace"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://your_namespace"
        xes.version="1.0" xes.features="nested-attributes" >
        <extension name="Organizational" prefix="org" uri="http://www.xes-standard.org/org.xesext"/>
        <extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xesext"/>
        <extension name="Concept" prefix="concept" uri="http://www.xes-standard.org/concept.xesext"/>
        <extension name="OrganizationalData" prefix="orgdata" uri="http://www.xes-standard.org/orgdata.xesext"/>
        <string key="concept:name" value="Mobility"/>
        <trace>
                <string key="id" value="773783"/>
        <string key="concept:name" value="773783"/>
        <event>
                <string key="concept:name" value="Register application"/>
                <string key="lifecycle:transition" value="Start"/>
                <date key="time:timestamp" value="2020-11-16T09:36:33.784-0300"/>
                <string key="org:role" value="Jorge"/>
                <string key="org:resource" value="5394935"/>
                <string key="orgdata:elemType" value="UserTask"/>
                <list key="orgdata:varlist">
                        <variables>
                                <string key="concept:varname" value="studentid">
                                        <string key="orgdata:varValue" value="89964588"/>
                                        <string key="orgdata:valueType" value="string"/>
                                        <date key="time:timestamp" value="2020-11-16T09:36:50.554-0300"/>
                                </string>
                        </variables>
                </list>
                <list key="orgdata:entlist">
                    <entities>
                        <string key="concept:entname" value="application">
                                <list key="orgdata:attlist">
                                    <attributes>
                                        <string key="concept:attname" value="idapplication">
                                                <string key="orgdata:attValue" value="1592"/>
                                                <string key="orgdata:valueType" value="int4"/>
                                                <date key="time:timestamp" value="2020-11-16T09:36:50.863-0300"/>
                                        </string>
                                        <string key="concept:attname" value="idstate">
                                                <string key="orgdata:attValue" value="1"/>
                                                <string key="orgdata:valueType" value="int4"/>
                                                <date key="time:timestamp" value="2020-11-16T09:36:50.863-0300"/>
                                        </string>
                                        <string key="concept:attname" value="idprogram">
                                                <string key="orgdata:attValue" value="263"/>
                                                <string key="orgdata:valueType" value="int4"/>
                                                <date key="time:timestamp" value="2020-11-16T09:36:50.863-0300"/>
                                        </string>
                                        <string key="concept:attname" value="idstudent">
                                                <string key="orgdata:attValue" value="89964588"/>
                                                <string key="orgdata:valueType" value="int4"/>
                                                <date key="time:timestamp" value="2020-11-16T09:36:50.863-0300"/>
                                                <string key="orgdata:refVariable" value="studentid"/>
                                        </string>
                                        <string key="concept:attname" value="notified">
                                                <string key="orgdata:attValue" value="NULL"/>
                                                <string key="orgdata:valueType" value="int4"/>
                                                <date key="time:timestamp" value="2020-11-16T09:36:50.863-0300"/>
                                        </string>
                                    </attributes>
                                </list>
                        </string>
                        .....
                </entities>
            </list>
        </event>
        ....
     </trace>
     .....
</log>
```

Fig. 2. Excerpt of the extended event log in XES format

3.2 Extended Event Log

Within our framework proposal we defined a XES extension for the Event logs to include integrated data from process and organizational data execution. Although the XES format allows to add attributes to events, they are added as tags without a logical order or correspondence to the event, as we provide in ours. As mentioned before, we integrate process execution data with corresponding organizational data [8] handled by the process, by means of several steps of data ETL from BPM systems and organizational databases, prior to the generation of the Event log. In Fig. 2 we present an excerpt of the extended Event log for the Students Mobility BP.

The extension adds two lists of attributes to the Event element: i) a variables list and ii) an entities list, which in turn contains a list of attributes. This extension reflects the integration of process and organizational data, in the following way. First, variables correspond to process variables that are handled by that event i.e. activity within the BPMS execution of the process, thus adding information from the process execution side. Secondly, entities and attributes correspond to organizational data that are handled by that event i.e. activity, within an organizational database different from the process one, thus adding information from the organizational data execution.

In the example, the activity that is shown corresponds to the "Register application" task, the process variable handled by the process execution is the "studentid" variable, which matches with the organizational database attribute "idstudent" of the entity (table) application. It can be noticed that the value for both elements is "89964588". This corresponds to the fact presented above that the "Register application" task inserted a new record in the application table of the organizational database, for each student application received, for the mobility program call. Thus, the application table, as shown in Fig. 1b has several references to other existing tables such as the Students and Program tables. As this is an excerpt of the extended event log, other variables, entities and attributes related to the "Register application" task are omitted for simplicity.

4 BPO Data Quality Model (BPODQM)

In this section we present the DQ model we developed for managing the quality of the integrated event log and organizational data (extended event log). For this, we first must present the format in which data is obtained as well as the granularities that are considered for its manipulation.

4.1 Data Format and Granularities

The extended event log shown in Fig. 2 is represented through the standard data format XES [1], as mentioned. In order to define a DQ model for the log, it is necessary to define the different granularities for identifying portions of data. These granularities are the following:

- **attribute value**. This is a particular value of an attribute. For example, in Fig. 2, the value "Register application".
- **attribute**. It refers to the set of values corresponding to the same key. For example, in Fig. 2, all the values that appear for key "concept:name".
- **event**. It involves all data included in an event data. For example, in Fig. 2, all data included in the event named "Register application"
- **log**. This granularity is used for properties that refer to the whole log.

4.2 BPODQM

This section presents the general DQ model BPODQM (Business Process and Organizational Data Quality Model) we have defined, in which specific dimensions, factors, and metrics for integrated process and organizational data are provided. It is based on previous quality models we have defined for other contexts [7,16], and on [17] which we have adapted and extended. This model is intended to serve as a general data quality model for the domain of integrated process and organizational data, which may be instantiated for any extended event log. Table 1 presents DQ dimensions, factors and metrics defined.

Accuracy dimension is composed by the following factors: (i) *syntactic accuracy*, which focuses on how the data is written, and whose metric measures if the data fits with the required format for the attribute, (ii) *semantic accuracy*, which refers to the existence of the attributes of an event with respect to reality, and may be measured through two different metrics, where the first one verifies the attributes that are not event identifiers, and the second one verifies the event identifiers, and finally, (iii) *precision*, which captures the detail level of a data item, and its metric is applied to an attribute value, which is a timestamp.

Consistency dimension is composed by the following factors: (i) *domain consistency*, which has two metrics, such that the first one compares an attribute value to a set of values and the second one verifies if an attribute value satisfy a values set definition, (ii) *inter-element consistency*, whose metric verifies if two attribute values of different events satisfy a consistency rule, and (iii) *intra-element consistency*, whose metric verifies if two attribute values of an event satisfy a consistency rule.

Completeness dimension has two factors: (i) *coverage*, which measures the proportion of the quantity of events contained in a trace wrt the quantity of events that the trace should contain, and (ii) *density*, for which there are three metrics, the first one verifies if an attribute value is Null, the second if certain attribute does not appear in an event, and the third one measures the density of an event considering weights over the different attributes.

Uniqueness dimension is composed by two factors: (i) *duplication-free*, which verifies if an attribute has duplicated values and if a trace has duplicated events, through two different metrics, and (ii) *contradiction-free*, whose metric evaluate if a trace has two different events that correspond to the same event in reality and have contradictory information.

For *Freshness* dimension we define only the factor *timeliness*. This factor has three different metrics, each one measuring attribute, event and trace timeliness, respectively. They verify if the timestamp of the object (attribute, event or trace) belongs to the time range of the parent object (event, trace or log, respectively).

Credibility dimension is composed of two factors: (i) *provenance*, which may be measured by three metrics; *responsibility*, which gives a score to the credibility of the person who is responsible of the log data, *origin*, which measures the credibility of the event origin, and *reproducibility*, which verifies if a log is reproducible following workflow rules, and (ii) *trustworthiness*, which may be measured through three different metrics that are applied over attribute values;

Table 1. BPODQM dimensions, factors and metrics

Dimension	Factor	Metric	Granularity
Accuracy	Syntactic Accuracy	Format	Attribute value
	Semantic Accuracy	Weak Semantic Accuracy	Event
		Strong Semantic Accuracy	Event
	Precision	Timestamp precision	Attribute value
Consistency	Domain Consistency	Extensional Values	Attribute value
		Intensional Values	Attribute value
	Inter-element Consistency	Inter-event Rule	Activity
	Intra-element Consistency	Intra-event Rule	Event
Completenes	Coverage	Coverage Ratio	Trace
	Density	Not Null	Attribute value
		Inexistent Value	Event
		Weighted Density	Event
Uniqueness	Duplication-free	Duplicate Attribute	Attribute value
		Duplicate Event	Event
	Contradiction-free	Contradictory Event	Event
Freshness	Timeliness	Attribute Timeliness	Attribute value
		Event Timeliness	Event
		Trace Timeliness	Trace
Credibility	Provenance	Responsibility	Log
		Origin	Event
		Reproducibility	Log
	Trustworthiness	Believability	Attribute value
		Reputation	Attribute value
		Verifiability	Attribute value
Security	User Permissions	Authorized User	Event
	Encrypted Data	Encrypted Attribute	Attribute value
		Ratio Encrypted Att	Event
	Anonymity	Anonymous Attribute	Attribute value
		Ratio Anonymous Att	Event

believability, which measures the degree of believability of the veracity of the data value, *reputation*, which refers to the reputation of the data source, and *verifiability*, which indicates if a data value es verifiable or not.

Finally, *security* dimension is composed by three factors: (i) *user permissions*, which verifies for an event, if the users that participated in it have the necessary rights, (ii) *encrypted data*, which has two metrics, one that verifies if all the values of an attribute are encrypted and another one that calculates the ratio of encrypted attributes of an event, and (iii) anonymity, which may be measured through a metric that verifies if an attribute is anonymized or a metric that calculates the ratio of anonymized attributes of an event.

All the results of a metric can be aggregated to the following granularity, calculating the percentage of results "1" over the total. For example, the metric *format* can be aggregated from *attribute value* to *attribute* granularity, obtaining the percentage of values that satisfy the required format for a given attribute.

5 Example of Application

In this section we present an example of BPODQM application on the extended event log we generated for the "Students Mobility" BP introduced in Sect. 3. In the first place, we have to select the quality characteristics from the BPODQM model, to be checked over the data in the extended event log. We have selected some basic ones in order to show its evaluation and to provide a discussion on the integrated process and organizational data to which we applied the model:

- Dimension: *Accuracy*, Factor: *Syntactic correctness*, Metric: *Format*
- Dimension: *Consistency*, Factor: *Domain consistency*, Metric: *Extensional values*
- Dimension: *Completeness*, Factor: *Density*, Metric: *Not null*
- Dimension: *Freshness*, Factor: *Timeliness*, Metrics: *Attribute Opportunity*

Each one is applied over a specific element of the extended event log which is also defined when selecting the characteristics to be evaluated, and specific metrics are defined for it to be calculated over the element. In what follows we present the definitions for each of the selected Metrics.

5.1 Metric Format

For this example, the Metric Format is applied to the existing timestamps in all levels i.e. event, variables and attributes. We defined a specific Metric that takes as correct format for the timestamp the one specified by the event log: ***yyyy-MM-dd'T'HH:mm:ss***. We then defined a specific function in order to calculate the specific metric, with signature ***timestampFormat(date, format): bool***, which returns true if the timestamp of an element of the log is in the defined format, false if not.

In this example, we had 1980 timestamp tags corresponding to event timestamps and the integrated data in variables timestamps and attributes timestamps, as shown in Fig. 2. As we automatically generated the extended event log from the integrated data formatting the timestamps as required, they all returned true in the correct format by construction. This is a basic example of the format metric that tools as ProM or Disco already check when importing event logs, but it can be checked for other attribute values including organizational data attributes that are domain specific for which specific formats are defined.

5.2 Extensional Values

We defined to check this factor in the extended event log for the values corresponding to the attribute *idstate* of the entity *application* as shown in Fig. 2. As introduced before, in the Students Mobility a new application is registered in the state "Initiated" whenever a student submits one for a mobility program. As the process progresses the application status values are updated, following domain predefined values, which are referenced from the application table i.e. "Initiated" = 1, "Approved" = 2, an so on. So we defined the domain values by extension as idstate = {1, 2, 3, 4} and we checked them within the extended event log, and the specific function ***extensionalValueAttribute(attribute, dom): bool***, which returns true if the attribute value lies between the defined domain, false if not. In this case we did not find values off range.

5.3 Metric Not Null

We checked this factor at two levels: the process data and the organizational data, to ensure key elements for the mining analysis will not be null. At the process level we defined to check the *resource* attribute of the event, and at the organizational level the *notified* attribute of the application entity. Due to the specificity of the process implementation and supporting data model, the notified attributed will be (correctly) null when associated with the first four tasks of the model, until the process reaches the "Notify applicants" task where applicants are notified and the attribute is updated with values {Yes, No}. We defined the specific function ***NotNullAttribute(attribute): bool***, which returns true if the attribute value is not null, false if not.

Checking for the resource attribute is useful if you are going to use the resource values for analysis, since tools such as ProM allows event logs without resources for process model discovering purposes. In our case, all resources where included either people or system values. Regarding the notified attribute, an interesting discussion arises, since most values of the attribute returned as null. It would be interesting to be able to select not only the attribute to be checked but the event to which the attribute should be attached and not be null, which in this case will be only the "Notify applicants" task.

5.4 Metric Attribute Timeliness

This metric was checked for all organizational data attributes of each entity within the corresponding event i.e. task to which they are related. We define the function ***attributeTimelinessTimestamp(attribute):bool*** which returns true if the timestamp of the organizational data attribute is within the timestamp of the corresponding event, and false if not. We detected several timestamps of attributes which had values that do not correspond to the timestamps of the event, i.e. were prior to the event timestamp. Although the log was correctly imported in ProM, since the extended log attributes are not taken into account in the checks, when importing it in Disco we got a warning on this fact, and several registers were not imported correctly. This fact helped us to dig into the generation of the log, the integration of data, and the process and organizational data from the sources, in order to find the problem.

6 Related Work

Data quality dimensions have been defined in the last decades generating a wide set of dimensions with focus on organizational data from which to choose. This is both and advantage and a disadvantage since it can be overwhelming to define, integrate and organize dimensions, factors and metrics in a model for an organization. However, there is a sub-set agreed and used by most authors [13,14] that could help in this task. [5] propose an organization of quality dimensions providing six clusters with which to cover key dimensions such as accuracy, completeness, consistency, among others. Data quality evaluation and cleaning is a key step in data mining and other analysis approaches e.g. data warehouses.

Regarding process data quality evaluation for process mining, there are some recent proposals which defined specific dimensions with focus on the process data at the event log level such as [17], without dealing with organizational data. It takes into account the guidelines for process data quality in [2], but it is not clearly structured in dimensions, factors and metrics as ours, only defining dimensions and less than ours. Also, metrics are defined in a fixed way and with a predefined score. [4] presents an approach and application for the health area, defining some dimensions and metrics.

Differently to these works, our proposal defines a quality model for integrated process and organizational data, that are put together in an extended event log where both quality and analysis take into account the complete data set of each process. The model is instantiated for the extended event log of each process.

7 Conclusions

This paper presents a DQ model, called BPODQM, for evaluating DQ in the extended event log, which is a log that contains data from process execution and the related organizational data. The proposed DQ model is a general model that can be instantiated for any extended event log, by selecting the most important DQ dimensions, factors, metrics and data to evaluate, according to the characteristics and requirements of the particular case. BPODQM contains a set of dimensions, factors and metrics that were selected, adapted and extended from the literature, considering the particularities of the BP domain, the goals of process mining activities and the characteristics of the extended event log we manage. On the other hand, this model was refined from its successive application to particular cases of BP. We illustrated the model application through a small part of an experience of DQ evaluation over a BP that deals with students applications and granting for mobility programs.

We believe that this DQ model is rich enough, and at the same time concise and practical, to address the most important issues that can appear when working with event logs and associated organizational data in order to perform process and data mining.

Acknowledgments. This work was supported by project "Minería de procesos y datos para la mejora de procesos en las organizaciones" funded by Comisión Sectorial de Investigación Científica (CSIC), Universidad de la República (UdelaR), Uruguay.

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)
2. van der Aalst, W.M.P.: Process Mining - Data Science in Action, 2nd edn. Springer, Heidelberg (2016)
3. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business process management: a survey. In: van der Aalst, W.M.P., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-44895-0_1
4. Andrews, R., et al.: 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. Pub. Health **16**(7), 1138 (2019)
5. Batini, C., Scannapieco, M.: Data and Information Quality - Dimensions, Principles and Techniques. Data-Centric Systems and Applications, Springer, Heidelberg (2016)
6. Chang, J.: BPM Systems: Strategy and Implementation. CRC Press, Boca Raton (2016)
7. Cristalli, E., Serra, F., Marotta, A.: Data quality evaluation in document oriented data stores. In: Woo, C., Lu, J., Li, Z., Ling, T.W., Li, G., Lee, M.L. (eds.) ER 2018. LNCS, vol. 11158, pp. 309–318. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01391-2_35
8. Delgado, A., Calegari, D.: Towards a unified vision of business process and organizational data. In: XLVI Latin American Computing Conference, CLEI 2020. p. To appear. IEEE (2020)
9. Delgado, A., Marotta, A., González, L., Tansini, L., Calegari, D.: Towards a data science framework integrating process and data mining for organizational improvement. In: 15th International Conference on Software Technologies, ICSOFT 2020, pp. 492–500. ScitePress (2020)
10. Delgado, A., Weber, B., Ruiz, F., de Guzmán, I.G.R., Piattini, M.: An integrated approach based on execution measures for the continuous improvement of business processes realized by services. Inf. SW Technol. **56**(2), 134–162 (2014)
11. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of BPM, 2nd edn. Springer, Heidelberg (2018)
12. IEEE: Task Force on Data Science & Adv. Analytics (2020). http://www.dsaa.co/
13. Scannapieco, M., Catarci, T.: Data quality under a computer science perspective. Arch. Comput. **2**, 1–15 (2002)
14. Shankaranarayanan, G., Blake, R.: From content to context: the evolution and growth of data quality research. J. Data Inf. Qual. **8**(2), 1–28 (2017)
15. Sumathi, S., Sivanandam, S.N.: Introduction to Data Mining and its Applications, Studies in Computational Intelligence, vol. 29. Springer, Heidelberg (2006)

16. Valverde, M.C., Vallespir, D., Marotta, A., Panach, J.I.: Applying a data quality model to experiments in software engineering. In: Indulska, M., Purao, S. (eds.) ER 2014. LNCS, vol. 8823, pp. 168–177. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12256-4_18
17. Verhulst, R.: Evaluating quality of event data within event logs: an extensible framework (2016)
18. Wang, R.Y., Strong, D.M.: Beyond accuracy: what data quality means to data consumers. J. Manag. Inf. Syst. **12**(4), 5–33 (1996)
19. Weske, M.: BPM - Concepts, Languages, Architectures, 3rd edn. Springer, Heidelberg (2019)