Data Quality Management in Web Warehouses using BPM

ADRIANA MAROTTA, Universidad de la República, Uruguay
ANDREA DELGADO, Universidad de la República, Uruguay

Abstract. The increasing amount of data published on the Web poses the new challenge of making possible the exploitation of these data by different kinds of users and organizations. Additionally, the quality of published data is highly heterogeneous and the worst problem is that it is unknown for the data consumer. In this context, we consider Web Warehouses (WW) (Data Warehouses populated by web data sources) as a valuable tool for analysis and decision making based on open data. In previous work we proposed the construction of WW with BPMN 2.0 Business Processes, automating the construction process through a two-phases approach, system configuration and system feeding. In this paper, we focus on the problem of including data quality management in the WW system, based on data quality models definitions, and allowing data quality assessment and data quality aware integration. In order to achieve this, we extend previous work with the modeling of the extra activities for data quality management in BPMN 2.0 and its implementation in a BPMS.

• Information systems → Decision support systems• Information systems → Information Integration• Information systems → Web applications.

Additional Key Words and Phrases: Data Quality, Web Warehouse.

INTRODUCTION

The increasingly amount of data published on the Web poses the new challenge of making possible the exploitation of these data by different kinds of users and organizations, considering for example the area of open government data [Working Group 2016]. Additionally, the quality of published data is highly heterogeneous and the worst problem is that it is unknown for the data consumer, who sometimes receives these data after transformations and combinations.

In order to manage data quality (DQ) issues in an information system, a DQ model should be constructed. A DQ model establishes and formalizes the different DQ aspects that will be considered for the particular case, as well as the metrics that will be applied for DQ assessment. In the DQ model definition task, prioritization is vital, since all quality aspects cannot be measured for all data items. Therefore, the most relevant data items should be selected and for each one, the most relevant quality aspects should be chosen. Since data and DQ aspects relevance is different for each particular domain, it is imperative to define a suitable DQ model for each case. However, despite its importance, the step of defining a DQ model is very frequently ignored. On the other hand, many DQ metrics are commonly used and repeated in different data types and domains, so that reusing metrics definitions and implementations is highly desirable.

In this context, we consider Web Warehouses (WW) (Data Warehouses populated by web data sources) as a valuable tool for analysis and decision making based on open data. The goal of these systems is to act as an intermediary between data publications and the user, pre-processing data and adding value to them. This pre-processing involves data integration, aggregation, re-structuring and DQ measurement and improvement. DQ is taken into account when building the WW, such that the system is able to: (i) provide to the user quality information associated to the obtained data, and (ii) improve quality of data throughout the WW process. We define a flexible WW, which can be configured accordingly to different domains, selecting the corresponding web sources and defining data processing characteristics; a first proposal for this architecture was published in [Marotta et al. 2012].

To automate this process, we modeled it using business processes (BPs) [Weske 2007], which are defined as the sequence of activities that are carried out to provide value to business in an organizational and technical environment. Business Process

Management (BPM) [Weske 2007][van der Aalst et al. 2003][Dumas et al. 2013] provides the means to support the BPs lifecycle [Weske 2007], and Business Process Management Systems (BPMS) [Chang 2006] help carrying out the activities that are defined within each phase. The Business Process Model and Notation (BPMN 2.0) [OMG 2013] standard provides both a notation for modeling BPs and a defined semantic for its elements to be executed. It is also easily understandable by business people and it is nowadays widely used both in academia and industry.

In previous work [Delgado et al. 2014], [Delgado and Marotta 2015] we proposed a two BPs level vision to help define and automate the process of building flexible WW: at the first level a *configuration process* to support the selection of web sources and the definition of schemas and mappings, which is mostly carried out manually, and at the second level, a *feeding process* which takes the defined configuration and loads the data into the WW, which is performed mostly automatically. Both the configuration and the feeding processes are modeled in the BPMN 2.0 notation and executed in the Open Source platform Activiti BPMS (http://www.activiti.org).

In this paper, we focus on the problem of including DQ management in the WW system, extending those works. In the configuration process we add the definition of the DQ model for the newly extracted data as well as for the DW data, also including definitions to allow the use of quality information in the data integration tasks. In the feeding process, we include the DQ measurement tasks and we refine the integration tasks so that they take profit of DQ information. In addition, we define a quality metadata model, which will generate a quality database from the configuration results, in order to register the DQ values measured during the feeding process. This quality database will be queried by the feeding process during integration tasks, and will enrich the data obtained by final users with associated quality information. It also will be updated during the feeding process so that quality data is propagated throughout the whole process.

The main contributions of this work are: (i) a proposal for including DQ management in the WW building process, and (ii) the modeling of the processes for DQ management with BPMN 2.0 and its implementation through a BPMS.

The rest of the document is organized as follows: in section 2 we present the background, in section 3 we describe the proposal for adding quality aspects to the WW configuration process, in section 4 we present the proposal for DQ management during the WW feeding, in section 5 we discuss related work and finally in section 6 we present some conclusions and current and future work.

BACKGROUND

The WW we define is a system that consists of several components and is service oriented. Its architecture is shown in Fig. 1. The system components are: Extraction, Integration, Transformation & Loading, OLAP (On-Line Analytical Processing), DQ Measurement (DQM) and DW Quality Measurement (DWQM). The Service Infrastructure layer offers to the other components the different available specialized services, which include data services, DQ services and integration services.

The Extraction component must solve the problem of format heterogeneity, since data can be found in a variety of formats, such as csv, html, xml, rdf, etc. It must be able to extract from each source a pre-defined set of items and store them into a database, assuring that each data item goes to the right place. The Integration component must perform the integration of data coming from the different web sources, solving the problems of entity resolution and data conflicts, with the presence of semantic heterogeneity. The Transformation & Loading component

transforms data to the multidimensional model, preparing them to OLAP manipulation, allowing multidimensional data analysis through front-end utilities.

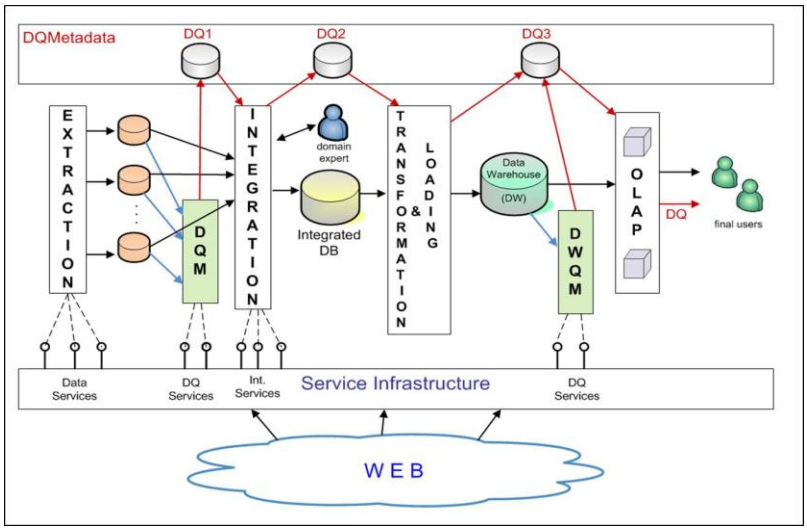

Fig. 1. Web Warehouse General Architecture

DQM component is in charge of measuring DQ of data that have been just extracted from the web, while DWQM component is in charge of measuring DQ of the DW. These components register DQ information in the databases DQ1, DQ2, and DQ3, which contain quality information (DQMetadata) about the WW data. DQMetadata is read by the WW processes in order to take quality issues into account when treating data. It is also written by them for enabling the propagation of the quality metadata throughout the different stages of the data transformation. This is necessary in order to maintain the correspondence between quality information and WW data, when WW data is combined and transformed.

The complete vision we take in our proposal to support the construction of these flexible WW consists of two main phases, the configuration phase and the feeding phase, and three main elements: (i) the configuration process, which generates the metadata that is needed to build the WW, (ii) the feeding process, which actually generates each specific WW from each defined configuration, and (iii) the DQ management, which can be allowed through the inclusion of DQ related definitions in the configuration process. The quality activities were not detailed in previous works since initial versions of the processes were modeled and implemented without DQ management. The configuration data that is gathered from the user along the configuration process is stored as metadata in the Configuration Database in our own defined data model. In the feeding process the configuration data will be read to automatically load and build the specific WW.

DATA QUALITY SUPPORT DURING THE WW CONFIGURATION

In this Section we present how DQ management is allowed in the WW, through the inclusion of the necessary definitions in the configuration phase.

DQ aspects must be configured for two different kinds of components of the WW: (i) for the DQ measurement components and (ii) for the integration component. The main configuration activities for (i) are the definitions of the DQ model for extracted data and the DQ model for DW data. We define these two activities separately

because, for the just extracted (from the Web) data, general DQ dimensions are considered for the specific domain, while for the obtained DW data (at the end of the WW process) DQ aspects specific to this stage are defined, considering: the multidimensional semantics of data, the analysis context and final-user domain knowledge. Configuration activities corresponding to (ii) make possible a data-integration process that is based on DQ. Finally, in the Configuration phase the DQ metadata model must be defined.

Fig. 2 shows the inclusion of DQ related definitions to the WW configuration process in BPMN 2.0. In the following sections we describe the corresponding tasks. Before, we present a simple running example to follow throughout the paper, in order to illustrate the different steps we propose.

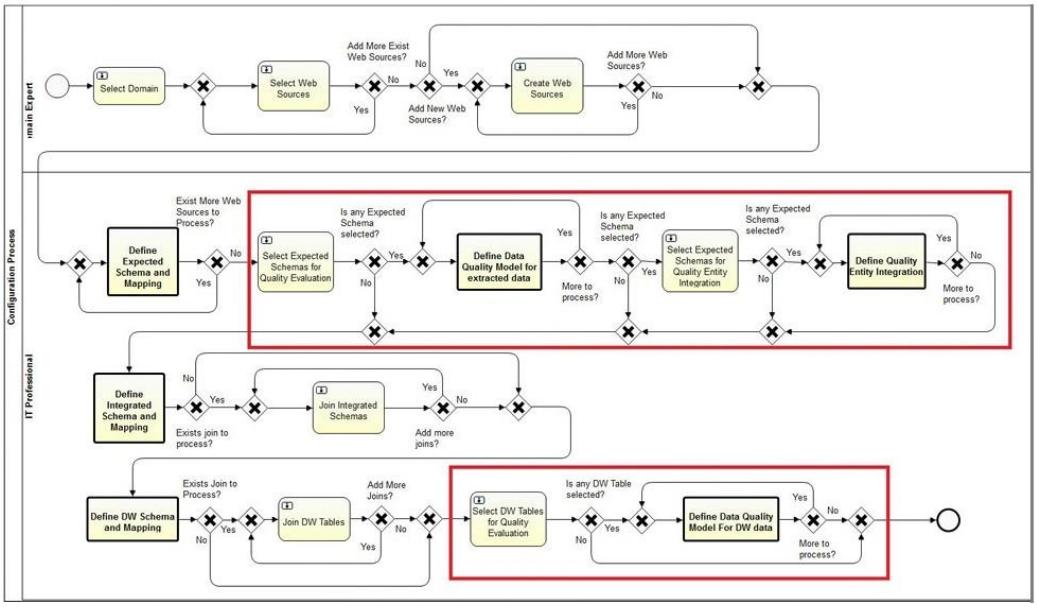

Fig. 2. DQ definitions included in the WW configuration process

Running Example:
Suppose we are building a WW about Tourism domain, for which we select several data sources, and two of them contain hotels data. We call them *S1* and *S2*. The configuration process will first execute activities to ask the user to select the domain and corresponding web sources ("Select domain" and "Select Web Sources" or "Create Web Sources") as shown in the first lane of the model in Fig. 2. During the sub-process "Define Expected Schema and Mapping" we define the expected schema *Hotels (name, stars, city, country)* and map its attributes to the corresponding data items of *S1* and *S2*.

Definition of the DQ Models

The first step to add DQ management to the WW is to define the DQ models. The task "Select Expected Schemas for Quality Evaluation" allows the user to select the expected schemas over which he wants to define quality metrics. In this way, he states that the quality of the data that will be loaded in these schemas (in the feeding process) will be measured. If he does not select any schema, the process directly goes to the integration tasks. The sub-process "Define DQ Model for Extracted Data" is shown in Fig. 3, and is repeated for each schema selected in the previous task.

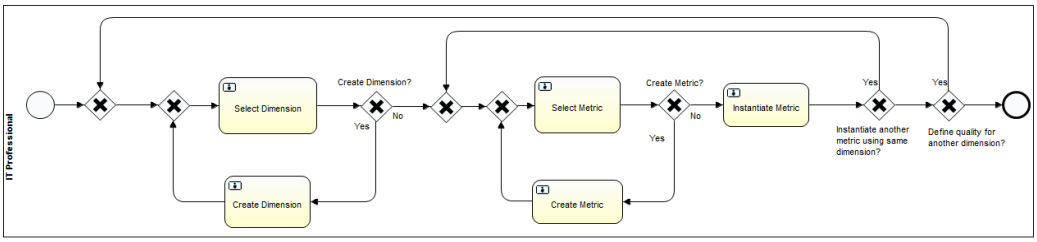

Fig. 3. Sub-process: Define DQ Model for Extracted Data

The "Select Dimension" task allows the user to select an existing DQ dimension (from a set of pre-defined dimensions available) or create a new one. If he decides to create a new one, the task "Create Dimension" is executed, where he must define a name for the new dimension and it is kept in the system for future configurations. Then, there are two tasks analogous to the previous ones: "Select Metric" and "Create Metric", which allows the selection and/or definition of the DQ metrics that correspond to the dimensions. In the "Create Metric" task the following aspects must be defined: a name, a description, the web service that will calculate the metric and the metric granularity. The possible granularities are: *cell*, *set of cells*, and *table*. Finally, each defined metric is instantiated, through the task "Instantiate Metric". In this task the user selects the table/attributes to which the metric will be applied. The process flow continues to the selection/creation of more metrics for the same dimension, or of more dimensions, or to the end of the sub-process. The configurations that are done in the tasks "Select DW Tables for Quality Evaluation" and "Define DQ Model for DW Data" are analogous to the ones for expected schemas, except that at the moment of instantiating the metrics, the user is informed about the multidimensional roles of the tables and attributes (i.e. if it is a fact table, if it is a measure attribute, etc.).

Running Example (cont.):

Returning to the Tourism WW with the expected schema *Hotels (name, stars, city, country)*, suppose that we want to define a DQ model so that its DQ is measured at the feeding stage. Then, we decide to measure the quality dimension *Accuracy* over attributes *name* and *city*, and the dimension *Freshness* over the table *Hotels* (globally). For this, we select the dimensions *Accuracy* and *Freshness* through the task "Select Dimension". Then, applying the tasks "Create Metric" and "Instantiate Metric", we obtain the configuration shown in Table 1. Fig. 4 shows the user screens in the Activiti BPMS portal for the first two activities.

Table I. DQ Model configured

Dimension	Metric	Web Service	Granularity	Instantiated over
Accuracy	Synt-Acc-Name	W1	cell	Hotels.name
Accuracy	Synt-Acc-City	W2	cell	Hotels.city
Freshness	Currency	W3	table	Hotels

Definition of the DQ Based Integration

The task "Select Expected Schemas for Quality Entity Integration" first presents to the user a list of expected schemas, where each one is associated to more than one web source. These are the cases where there are more than one data source that correspond to the same real-world concept (e.g., hotels data from two different web sites), so there will probably be more than one source tuple, provided from different data sources, that correspond to the same real-world entity (e.g. "Porto Ibis Hotel")

and must be integrated into one. The user must select the expected schemas where he wants the integration to be done based on their DQ information.

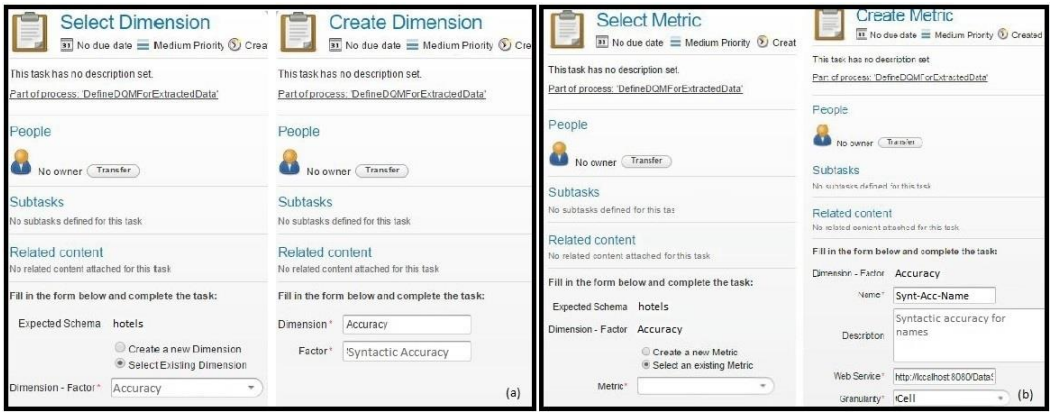

Fig. 4. User screens in Activiti portal to: (a) select or create dimension, (b) select or create metric

Finally, the sub-process "Define Quality Entity Integration" (see Fig. 5), which is repeated for each schema selected in the previous task, allows defining properties needed for the "entity resolution" and "entity fusion" activities. "Entity resolution" solves the problem of deciding if two or more tuples correspond to the same real-world entity, while "entity fusion" solves the problem of obtaining one tuple from several that correspond to the same real-world entity.

The service layer we have defined helps decoupling the implementation to solve these problems from the process, so we provide the user with existing web services implementations to choose from, but new ones can be also defined. This is allowed through the task "Select Entity Resolution and Entity fusion Services". After that, the task "Select Attributes for Entity Resolution Service" allows the user to choose the attributes of the expected schema that will be considered for solving the entity resolution problem. The following tasks allow defining properties for DQ consideration during entity fusion. If there are quality metrics associated to the current expected schema that have different granularities (e.g. two metrics with *cell* granularity and a metric with *table* granularity), the task "Select Granularity for Entity Fusion Service" is executed, allowing the user to choose one of them (e.g. *cell* granularity). When the entity fusion is executed (in the feeding stage), it will use the DQ measures of the metrics with the selected granularity. Finally in this sub-process, the task "Weigh Metrics for Entity Fusion Service" allows defining weights for the different DQ metrics involved, such that the entity fusion service can obtain one value from all the measures corresponding to the different metrics, and can compare the global quality of the tuples to fusion.

Running Example (cont.):
Continuing our example, where DQ metrics *Synt-Acc-Name*, *Synt-Acc-City* and *Currency* were defined for the exp. schema *Hotels*, suppose that the following configuration is done in the sub-process "Define Quality Entity Integration":

Atts for entity resolution: *name*, *country*
Granularity: cell
Metrics Weights: *Synt-Acc-Name*: 0.6 *Synt-Acc-City*: 0.4

This configuration says that: the entity resolution must be done considering the attributes *name* and *country*, the entity fusion must consider the metrics that have

cell granularity (the metric *Currency* will not be considered), and the DQ value for each tuple must be calculated considering the configured weights for each metric.

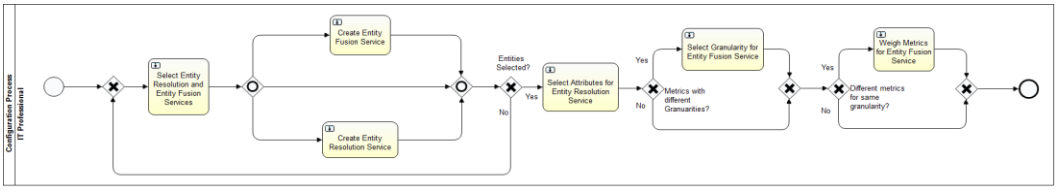

Fig. 5. Sub-process: Define Quality Entity Integration

DQMetadata Model

The DQMetadata is the database where the DQ measures will be stored. Part of this database is populated at the beginning of the Feeding Process, based on the DQ models defined in the Configuration Process. The other part is populated when the measurements processes are executed, during the Feeding Process. Due to space restrictions we show in Fig. 6 a simplified conceptual model, where only the measurements of cell-granularity metrics are represented.

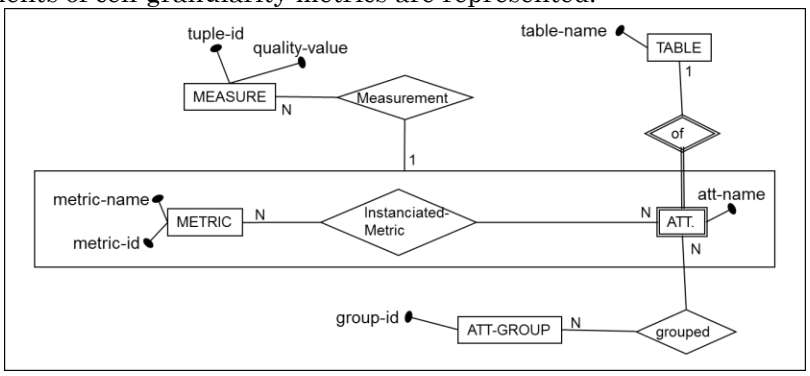

Fig. 6. Portion of the conceptual model for the DQMetadata database

DQ MANAGEMENT DURING THE WW FEEDING

In the WW feeding process, shown in Fig. 7, there are many DQ related tasks in different places of the process. Due to space restrictions we will focus only on describing how some of these tasks are carried out.

In Fig. 8 we show the portion of the process starting when data is extracted from the web sources until entity integration is finished. In the first task, which is automatic (ServiceTask with specific symbol at the left-top of the box), data is extracted from the sources and loaded into the "expected tables", whose schemas are the "expected schemas" defined in the configuration process.

Next, if required by the user, an error log generated during the extraction is shown. After that, the process verifies in the Configuration Database whether a DQ model was defined for the extracted data (i.e. for the expected schemas) and if so, the task "Measure Quality of Extracted Data" is executed. DQ metrics are applied through the execution of the DQ services configured and the DQ measures obtained are stored in the DQMetadata database. After that, the process verifies in the Configuration Database if there are expected tables that must be integrated (because they have the same expected schema), and for each one it asks if a quality-based entity integration was configured. If so, the sub-process "Do Quality Entity Integration", which we show and describe below, is executed. Otherwise, entity

integration is done with a by-default procedure that does not consider DQ aspects, and the data conflicts that arise are presented to the user in the following task, called "Show Conflicts of Entity Integration" so that the user can solve them.

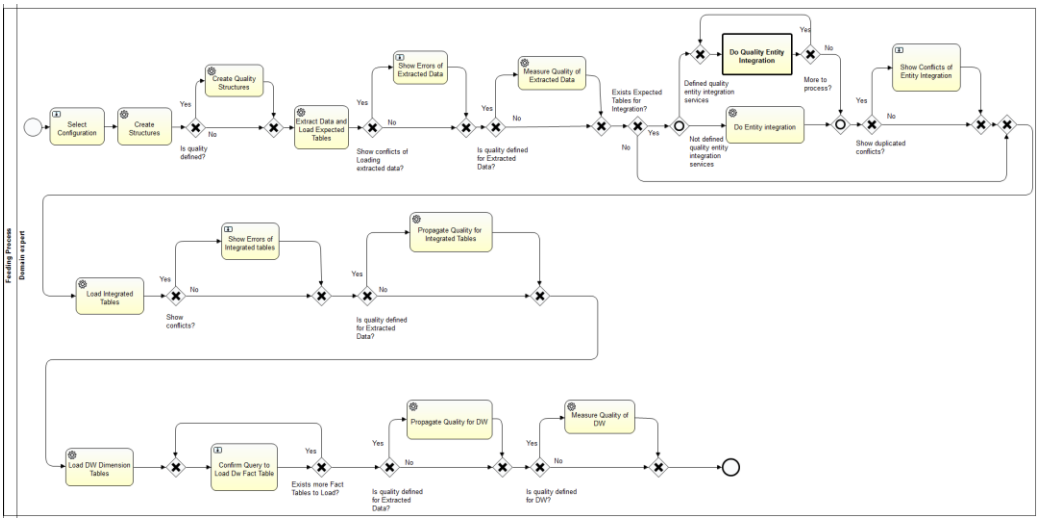

Fig. 7. Feeding Process

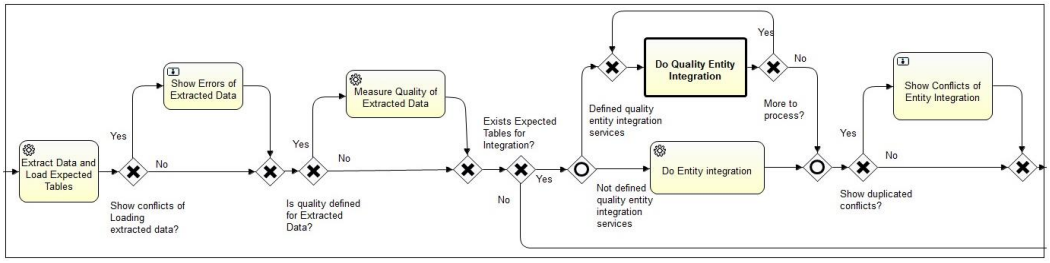

Fig. 8. Portion of the Feeding Process

The sub-process "Do Quality Entity Integration" is shown in Fig. 9. The task "Do Entity Resolution" identifies corresponding tuples from different expected tables, through the invocation to the entity resolution web service obtained from the Configuration Database. If this task identifies corresponding tuples, the task "Do Entity Fusion" is executed for each group of corresponding tuples, through the invocation of the entity fusion web service obtained from the Configuration Database. This service chooses the best tuple taking into account their associated quality values that were stored in the DQMetadata by the DQ measurement process.

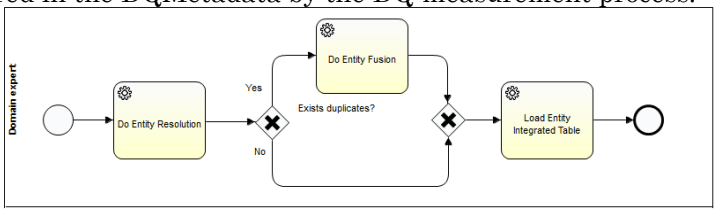

Fig. 9. Sub-process: Do Quality Entity Integration

Running Example (cont.):
Continuing with the running example, suppose that the expected schema *Hotels (name, stars, city, country)*, which is loaded from the web sources *S1* and *S2*,

generates two expected tables *T1* and *T2* that contain, respectively, the tuples: *t* <*'Four Seasons', 5, 'Mdeo.', 'Uruguay'*> and *t'* <*'4 Season', 4, 'Montevideo', 'Uruguay'*>. Suppose that the task "Measure Quality of Extracted Data" generates, and stores in the DQMetadata, the following quality measures:

Synt-Acc-Name (*T1.t*) = 1 / *Synt-Acc-Name* (*T2.t'*) = 0.2
Synt-Acc-City (*T1.t*) = 0.7 / *Synt-Acc-City* (*T2.t'*) = 1
Currency (*T1.t*) = 0.8 / *Currency* (*T2.t'*) = 1

The process "Do Entity Resolution", compares the tuples *T1.t* and *T2.t'*, considering the attributes *name* and *country*, as it was defined in the configuration, and concludes that they are corresponding tuples, i.e. they correspond to the same real-world entity. Then, the process "Do Entity Fusion" applies a formula that considers the quality measurements of the metrics of cell granularity, as set in the configuration, and the weights also defined in the configuration process. Then, the following DQ values are obtained for each tuple:

qv (T1,t) = *Synt-Acc-Name* (*T1.t*) * 0.6 + *Synt-Acc-City* (*T1.t*) * 0.4
= 1 * 0.6 + 0.7 * 0.4 = **0.5**
qv (T2,t) = *Synt-Acc-Name* (*T2.t'*) *0.6 + *Synt-Acc-City* (*T2.t'*) *0.4
= 0.2 * 0.6 + 1 * 0.4 = **0.4**

Finally, tuple *T1.t* is returned as result of the integration of *T1.t* and *T2.t'*.

RELATED WORK

Some work can be found that focuses on the ETL phase of the construction of a DW using BPMN for design purposes, not to automating the process. For example, in [Akkaoui et al. 2012], [Oliveira and Belo 2012] BPMN is used for conceptual modeling of ETL Processes. Also, in [Akkaoui et al. 2011] a model-driven framework for ETL process development is presented. The framework allows modeling ETL processes in a platform independent way, using a meta-model named BPMN4ETL, and generating vendor specific code from these models.

Regarding the representation of DQ issues in the context of processes modeling, IPMAP [Shankaranarayan et al. 2000] was proposed many years ago with the information product approach, where information is conceived as the result of a manufacturing process. This model allows representing several data processing tasks, in particular DQ checking. Based on this model, in [Sánchez-Serrano et al. 2009] the authors propose an extension to BPMN in order to allow the representation of DQ issues. Concretely, they add to the model a symbol that with the associated metadata allows representing the DQ dimensions to be controlled in the different tasks of the process. Later, motivated by the fact that DQ might affect business processes efficiency, in [Rodríguez et al. 2012] the authors propose an extension to BPMN 2.0 for modeling DQ requirements. They propose to mark the data-related BPMN elements with high and low-level DQ requirements so that the BP model can be improved with new activities for satisfying the DQ requirements. Additionally, in [Cappiello et al. 2013], a methodology for achieving this improvement is presented.

With the goal of automating ETL process design, the work presented in [Theodorou et al. 2015] focuses on modeling quality characteristics of ETL processes, including DQ related ones. The authors propose a set of DQ characteristics, measure indicators for them, and the inclusion of steps for DQ corrections.

Differently from previous works, we are not extending BPMN neither proposing specific DQ dimensions for ETL processes, but we are using the standard BPMN 2.0 for modeling a configurable WW, which is capable of managing DQ accordingly to the needs and characteristics of the particular constructed WW.

CONCLUSIONS

In this paper we have presented a proposal to add DQ management in the construction of a WW, which is automatically generated after a configuration phase. The whole process was modeled with BPMN 2.0 and implemented in a BPMS.

Providing support for considering DQ in the construction of the WW allows us to provide final users with improved data or, at least with information regarding DQ aspects which will help them to decide on data usefulness.

We believe that our proposal for constructing the WW with DQ management through the use of BPM tools, provides to the WW developer a strongly guided framework that helps him to consider and apply the main steps for obtaining a robust WW with quality data. With this approach, many repetitive tasks that are commonly done manually and remain embedded in the ETL code, such as DQ procedures and integration decisions, are explicitly and systematically managed.

As current and future work we are planning to carry out a case study in an organization which allows us to confirm the usefulness of the approach, and also to improve our definitions including the processes and the DQ management aspects.

ACKNOWLEDGMENTS

The authors would like to thank the students who worked in the development of the BPs with DQ management and their validation, Pablo Barceló and Diego Pérez.

REFERENCES

Cinzia Cappiello, Angelica Caro, Alfonso Rodríguez, and Ismael Caballero. 2013. “An Approach To Design Business Processes Addressing Data Quality Issues.” In 21st European Conference on Information Systems, ECIS 2013, Utrecht, The Netherlands, June 5-8, 2013, 216.

James F. Chang, BPM Systems: strategy and implementation. Auerbach Pubs, Taylor&Francis Gr., 2006.

Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo Reijers, Fundamentals of BPM, Springer, 2013

Andrea Delgado, Adriana Marotta, Laura González, "Towards the construction of quality-aware Web Warehouses with BPMN 2.0 Business Processes". In procs. IEEE 8th International Conference on Research Challenges in Information Science, RCIS 2014, Marrakech, Morocco, 2014.

Andrea Delgado and Adriana Marotta. 2015. “Automating the Process of Building Flexible Web Warehouses with BPM Systems.” In 2015 Latin American Computing Conference, CLEI 2015, Arequipa, Peru, October 19-23, 2015, 1–11. doi:10.1109/CLEI.2015.7360005.

Adriana Marotta, Laura González, Raúl Ruggia. 2012. A Quality Aware Service-oriented Web Warehouse Platform. In *Proc. of Business intelligencE and the WEB (BEWEB), EDBT*, Berlín, Germany, 2012.

Bruno Oliveira and Orlando Belo. 2012. “BPMN Patterns for ETL Conceptual Modelling and Validation.” In Foundations of Intelligent Systems - 20th International Symposium, ISMIS 2012, Macau, China, December 4-7, 2012. Proceedings, 445–54. doi:10.1007/978-3-642-34624-8_50.

Object Management Group, BP Model and Notation (BPMN 2.0), http://www.omg.org/spec/BPMN/2.0/

Alfonso Rodríguez, Angelica Caro, Cinzia Cappiello, and Ismael Caballero. 2012. “A BPMN Extension for Including Data Quality Requirements in Business Process Modeling.” In BPMN, 116–25.

Noelia Sánchez-Serrano, Ismael Caballero, and Félix García. 2009. “Extending BPMN to Support the Modeling of Data Quality Issues.” In , 46–60. In Proc. Int. Conf. on Information Quality (IQ 2009).

G. Shankaranarayan, R. Y. Wang and M. Ziad. 2000. “Modeling the Manufacture of an Information Product with IP-MAP.” In Proc. 5th International Conference on Information Quality (IQ 2000).

Vasileios Theodorou, Alberto Abelló, Wolfgang Lehner, and Maik Thiele. 2015. “Quality Measures for ETL Processes: From Goals to Implementation.” Concurrency and Computation: Practice and Experience, n/a – n/a. doi:10.1002/cpe.3729.

van der Aalst W.M.P.,ter Hofstede A., Weske M., "Business Process Management: A Sur-vey", In Proceedings Int. Conf. on Business Process Management (BPM), The Nether-lands, 2003.

M. Weske. 2007. Business Process Management: Concepts, Languages, Architectures, Springer, 2007.

Working Group on Open Government Data. Retrieved April 3, 2016 from http://opengovernmentdata.org/

Akkaoui Zineb El, Jose-Norberto Mazón, Alejandro Vaisman, and Esteban Zimányi. 2012. “BPMN-Based Conceptual Modeling of ETL Processes.”. In Proc. of DAWAK 2012, 1–14.

Akkaoui Zineb El, Esteban Zimányi, Jose-Norberto Mazón, and Juan Trujillo. 2011. “A Model-Driven Framework for ETL Process Development.” In DOLAP, 45–52.

Received Month YYYY; revised Month YYYY; accepted Month YYYY