Multidimensional Contexts for Data Quality Assessment[$\star$]

Aida Malaki, **Leopoldo Bertossi**[$\star\star$] and **Flavio Rizzolo**
Carleton University, School of Computer Science
Ottawa, Canada
{amaleki,bertossi,flavio}@scs.carleton.ca

Abstract. The notion of data quality cannot be separated from the context in which the data is produced or used. Recently, a conceptual framework for capturing context-dependent data quality assessment has been proposed. According to it, a database D is assessed wrt. a *context* which is modeled as an external system containing additional data, metadata, and definitions of quality predicates. The instance D is "put in context" via schema mappings; and after contextual processing of the data, a collection of alternative clean versions D' of D is produced. The quality of D is measured in terms of its distance to this class. In this work we extend contexts for data quality assessment by including multidimensional data, which allows to analyze data from multiple perspectives and different degrees of granularity. It is possible to navigate through dimensional hierarchies in order to go for the data that is needed for quality assessment. More precisely, we introduce *contextual hierarchies* as components of contexts for data quality assessment. The resulting contexts are later represented as ontologies written in *description logic*.

1 Introduction

In previous research we have proposed a model of context for data quality assessment [6, 7], with the goal of formalizing the empirical fact that data quality is context dependent. In that work we presented the context $\mathfrak{C}$ as essentially a logical theory, into which a database D under assessment can be mapped, for further processing and analysis. The resulting instance (or set thereof) at the contextual level can then be compared with the original one, through a distance measure between D and the resulting instance [6, 7].

More specifically, the contextual theories used in [6] appeared as a form of data integration system, with additional (possibly partial) data, metadata, and definitions of quality predicates. The latter are used to extract quality data from those in the external dirty instance. In particular, the problem of doing *clean query answering*, i.e. of obtaining clean answers to queries posed to D via the context $\mathfrak{C}$ was introduced and investigated.

An important element that was not included in those contexts is the one of *data dimension*. And this is necessary, because data for data quality analysis are usually of a dimensional or hierarchical nature. Furthermore, dimensions provide different perspectives or points of view from which data can be seen and analyzed.

[$\star$] **Extended abstract.**

[$\star\star$] Contact author.

In this work we enrich our context model by introducing dimensions and their categories. The main purpose of this paper is to convey the general ideas, problems, approach, and issues that appear when dimensions are introduced in combination with the other contextual elements mentioned above. For the same reason, we concentrate mainly on the introduction of dimensions, rather than on their use in data quality assessment.

More precisely, we show how to use dimensional navigation for accessing the data that is necessary to assess the external data. We also show how this navigation can be combined, for the same purpose, with general knowledge expressed by rules. We also show the relevance of introducing and having intra- and interdimensional semantic constrains. They restrict certain combinations of related values that may appear in (or are associated to) different categories in a dimension, or pair of them.

To this end, we introduce in the context model an extension of the model for multidimensional databases (MDDBs) in [21]. This model is extended with different kinds of relations associated to categories and dimensions (going far beyond fact tables); and also with the above mentioned dimensional constraints.

Finally, following [6, 7], and in the spirit of having a context as a theory, we provide an ontological representation of the contexts as enriched with dimensions. This is because ontologies written in, e.g. *description logic* (DL) become logical theories. An ontology provides language components and constructs for representing knowledge about a domain of interest, and also reasoning capabilities. More concretely, we represent contexts as ontologies written in a description logic of the *DL-Lite* family [12], actually, *DL-Lite*$_{Horn}^{(\mathcal{H}N)^+}$ [2].

The rest of this paper is structured as follows. In Section 2, we present and discuss contexts for data quality assessment. In Section 3, we introduce dimensions into our context framework. In Section 4, we show how to specify dimensional contexts by means of DL-ontologies. In Section 5, we discuss related work. Finally, in Section 6, we draw some conclusions and point to ongoing work.

2 Contexts for Data Quality Assessment

The *quality* of data is relative to their *intended and interpreted use* [22]. It is related to the possible differences between the *actual* stored values and the *real* values that were *expected* or *supposed* to be stored [6]. Furthermore, the notions of “good” and “poor” data quality are inseparable from the *context* in which the data is used or produced [6].

In this paper, data quality (DQ) is addressed from these points of view, i.e. in relation to *semantic discrepancy* [22] (as opposed to misspellings, for example), and as determined by a formal context that enables data quality analysis.

Example 1. Tom is a patient in a hospital. Several times a day different medical tests are performed on him, and test values are recorded by a nurse. His doctor, John, wants to see Tom’s test values every day, to follow his evolution.

The data that John needs about Tom appear, among other, in the *PatientValue* relation in Table 1 below. John has additional *quality* concerns. He only wants to see, for all his patients, test results that are taken with instruments of the brand B_1. On Sep/5, for morning tests, the nurse, Jane, performed the test on Lou with an instrument of brand B_2, and inserted it as the 6th tuple into the *PatientValue* relation.

Table 1: *PatientValue*

Patient	Value	Time
Tom Waits	38.5	11:45/5/Sep/2011
Tom Waits	38.2	12:10/5/Sep/2011
Tom Waits	38.1	11:50/6/Sep/2011
Tom Waits	38.0	12:15/7/Sep/2011
Tom Waits	110/70	11:45/8/Sep/2011
Lou Reed	37.9	12:10/5/Sep/2011

Based on John's quality concerns, this tuple should not appear in a *quality* relation, one that satisfies John's requirements. However, its does appear in the doctor's view of data (which is Table 1). In this case, there is a difference between the value which is recorded and the real value which was expected to be recorded (one measured with an instrument from the intended brand). This is an example of "semantic discrepancy" or "semantically inaccurate data" [3]. □

The quality of data depends on the context [6]. In this work, we define a context for the quality assessment of a database instance D of schema $\mathcal{S}$ as a separate information system $\mathfrak{C}$. The latter may contain its own relational schema $\mathcal{C}$, a possibly partial (incomplete) instance I of $\mathcal{C}$, additional predicates with definitions in $\mathfrak{C}$ that can be used for D's quality assessment, etc.

The schema $\mathcal{C}$ could be an extension of $\mathcal{S}$, and I an extension of D. In order to assess the quality of D, the latter has to be *put in context* via schema mappings between $\mathcal{S}$ and $\mathcal{C}$. Different cases and situations can be accommodated in this framework [6]. A *quality database* instance D', as an alternative to the instance D at hand, could be a *footprint* of the contextual instance I after some additional processing via quality predicates at the contextual level. Depending on how much D departs from D', we can assign to the data in D a quality grade or measure. In other cases, instead of a single quality instance D', we can obtain a whole class $\mathcal{K}$ of quality instances, and D has to be assessed on the basis of its distance to $\mathcal{K}$ [6].

We can see that a context for data quality assessment can be conceived as a shared information space that is designed to serve a particular purpose [14].

Example 2. (example 1 cont.) We have a contextual relation *Measurement* (Table 2). It contains all the values of different tests that are performed on patients by using instruments of different brands at different times. From relation *Measurement* we obtain the values that are taken by instruments of brand B_1, satisfying the doctor's requirement.

Table 2: *Measurement*

Patient	Value	Hour	Brand
Tom Waits	38.5	11:45/5/Sep/2011	B_1
Tom Waits	38.2	12:10/5/Sep/2011	B_1
Tom Waits	38.1	11:50/6/Sep/2011	B_1
Tom Waits	38.0	12:15/7/Sep/2011	B_1
Tom Waits	110/70	11:45/8/Sep/2011	B_2
Lou Reed	37.9	12:10/5/Sep/2011	B_2

The quality version of *PatientValue* relation based on John's condition is (Table 3).

Table 3: *PatientValue'*

Patient	Value	Time
Tom Waits	38.5	11:45/5/Sep/2011
Tom Waits	38.2	12:10/5/Sep/2011
Tom Waits	38.1	11:50/6/Sep/2011
Tom Waits	38.0	12:15/7/Sep/2011

This new instance is obtained via a select-project view on *Measurement* at the contextual level. The quality of Table 1 is assessed by comparing it with the extension of *PatientValue'* in Table 3 by using some appropriate distance measure [6]. □

The context $\mathfrak{C}$ could also contain additional metadata or general knowledge that can be used in data quality assessment.

Example 3. (example 2 cont.) Let us assume that, instead of Table 2, we have the table *PatientWard* (Table 4). It contains all the patients and the wards they were staying on each day. Note that it does not explicitly contain information about the tests or the instruments used. However, the context has information about hospital guidelines:

Table 4: *PatientWard*

Patient	Date	Ward
Tom Waits	5/Sep/2011	W_1
Tom Waits	6/Sep/2011	W_1
Tom Waits	7/Sep/2011	W_1
Lou Reed	5/Sep/2011	W_2

Hospital Guideline 1: "Medical tests on patients in ward W_1 have to be performed by instruments of brand B_1".

A guideline like this can be used in different forms, e.g. as a hard rule, as a default rule, or as a semantic constraint at the contextual level. If we had the guideline instead of Table 2 with the detailed measurements, and assuming that the guideline is satisfied, we can use the contextual relation *PatientWard* in combination with the hospital guideline to conclude that all tests performed on Tom Waits between 5/Sep and 7/Sep (the dates in which he was in ward 1) were done with instruments of brand B_1. The quality version of *PatientValue* relation is again Table 3. □

The framework for data quality assessment proposed in [6] is depicted in Figure 1. It shows the relational schema $\mathcal{S}$ with is predicates $R_1, R_2, ..., R_n$. The instance D of $\mathcal{S}$ under quality assessment has extensions $R_1(D), R_n(D)$ for them. Context $\mathfrak{C}$ contains a *contextual schema*, $\mathcal{C}$, including a set $\mathcal{B}$ of built-in predicates, e.g. comparisons; and also a set, $\mathcal{P} = \{P_1, ..., P_k\}$, of *contextual quality predicates* (CQPs) with definitions over $\mathcal{C}$. The built-in predicates are used in the definitions of other predicates, like those in $\mathcal{P}$, so they are not explicitly shown in Figure 1.

The connection between the schemas in the framework is provided by *schema mappings* [4], α_i, like those found in *virtual data integration systems* (VDISs) [5, 23] or data exchange [1]. Now, schema $\mathcal{S}'$ is a copy of schema $\mathcal{S}$, with relational predicates $R_1', ..., R_n'$. Their extensions are the ideal, quality instances for the R_i's. Each of the R_i's is defined as a conjunctive view over the contextual schema, say $\forall \bar{x}(\alpha_i^{\mathcal{P}}(\bar{x}) \equiv R_i'(\bar{x}))$, where $\alpha_i^{\mathcal{P}}$ is a conjunction of atomic formulas with predicates in $\mathcal{C} \cup \mathcal{P}$. These views are computed on top of contextual instances I for schema $\mathcal{C}$, that are related to D in terms of contents by the mappings α_i. In this way, D is mapped into $\mathfrak{C}$, integrated into contextual instances I, and further qualified via the views R_i', to obtain quality data [6].

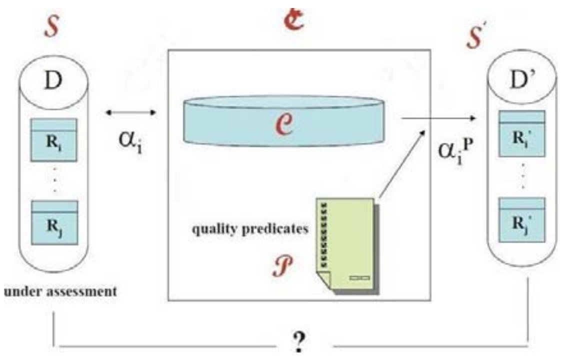


Fig. 1: Data quality assessment framework

3 Extending the Contextual Framework with Dimensions

In Example 2, the required data was explicitly stored in a relation that extends the table under assessment. However, in some other situations we may have to go outside a contextual table and navigate within the context, searching for the necessary data. This is particularly the case when the contextual data is of a *multidimensional* and *hierarchical* nature [8, 9].

Example 4. (example 3 cont.) John has a new quality requirement. He asks nurses to perform all the medical tests with instruments made by manufacturer M_1. Furthermore, there is a new hospital guideline in place:

Hospital Guideline 2: "Medical tests on patients in *standard care* and *intensive care* units have to be taken with instruments made by manufacturer M_1 and M_2, respectively".

The information explicitly provided by the contextual relation *Measurement* (Table 2) is about the *Ward* category, but data about the *units*, that could be used in combination with Guideline 2, belongs to a higher or more general category, *Unit*. This is illustrated in Figure 2.

Table 4: *PatientWard*

Patient	Date	Ward
Tom Waits	5/Sep/2011	W_1
Tom Waits	6/Sep/2011	W_1
Tom Waits	7/Sep/2011	W_1
Lou Reed	5/Sep/2011	W_2

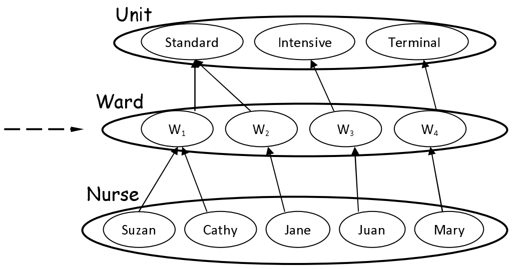


Fig. 2: *PatientWard* to *Location* mapping

In order to reach the relevant data, we have to navigate through the hierarchy. In this case, by *rolling up* from *Ward* to *Unit*, we can identify the wards that belong to *standard* *care units*, namely W_1 and W_2. This allows us to build a quality version of the original instance, the one shown in Table 5. □

Table 5: *PatientValue*”

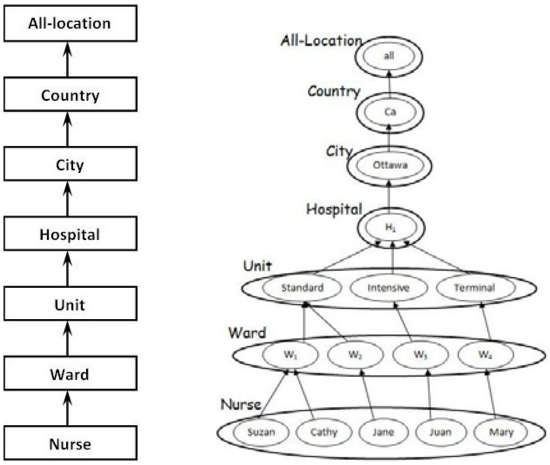
Patient	Value	Hour
Tom Waits	38.5	11:45/5/Sep/2011
Tom Waits	38.2	12:10/5/Sep/2011
Tom Waits	38.1	11:50/6/Sep/2011
Tom Waits	38.0	12:15/7/Sep/2011
Lou Reed	37.9	12:10/5/Sep/2011

Data in a contextual hierarchy are organized in categories that are (possibly partially) ordered according to the level of detail they provide. A contextual hierarchy provides information from one perspective according to different granularities. By having several hierarchies in a context, we will have multiple perspectives for data analysis and quality assessment.

Example 5. The hierarchy in Figure 2 is a portion of the *Location* hierarchy (or dimension) in Figure 3. The *dimension schema* in Figure 3a shows the *categories* and their relationships. Figure 3b shows a possible *dimension instance* for the *Location*; and a partial order relation between the elements of categories. □

Dimensions can be made part of a context by embedding in it a multidimensional database. For this purpose we can use an extension of the Hurtado-Mendelzon (HM) data model for multidimensional databases (MDDBs) [21]. Before discussing the possible extensions we may need, we briefly describe the HM model.

A *contextual dimension schema DS* is a pair $(CAT, \nearrow)$, where *CAT* is a set of named categories, and $\nearrow$ is a child/parent relation between categories [21]. The transitive closure of $\nearrow$ is denoted by $\nearrow^*$. There are no “shortcuts” between categories. There is also a distinguished *top category*, denoted with *All*, which is reachable from all other categories. The categories without an incoming $\nearrow$ are called *bottom* or *base* categories.



(a) Schema (b) Instance

Fig. 3: Location dimension schema and an instance

An *instance* of a contextual *DS* is a tuple $(M, <)$, where M is a finite collection of ground atoms of the form $C(a)$ with $C \in CAT$, and the data element a belongs to an underlying domain. We assume that different categories do not have elements in common. Relation $<$ stands for the partial order between elements of categories, and parallels the partial order $\nearrow$ between the corresponding categories. Category *All* has only one element, *all*. Categories are assumed to be disjoint. The transitive closure of $<$ is denoted with $<^*$.

Example 6. For the contextual dimension *Location* in Figure 3, we have the schema: $CAT = \{Nurse,\ Ward,\ Unit,\ Hospital,\ City,\ Country,\ All\}$, and $\nearrow = \{\langle Nurse, Ward\rangle, \langle\ Ward,\ Unit\ \rangle, \ldots\}$. Now, for the instance, $M = \{Nurse(Suzan), Nurse(Cathy),\ Ward(\mathrm{W}_1),\ \ldots\}$, and $< = \{\langle\ Suzan,\ \mathrm{W}_1\rangle,\ \langle Cathy,\ \mathrm{W}_1\rangle, \langle \mathrm{W}_1,\ Standard\ \rangle, \langle Intensive,\ \mathrm{H}_1\ \rangle, \ldots\}$ (shown also with arrows in Figure 3b). □

As is common in DWHs and MDDBs, in general, we can have *fact tables* associated to the base categories. However, using a multidimensional model and a multidimensional database within a context serves purposes not traditionally found in data warehousing, OLAP, or data analytics. We can, for example, extend the HM model with additional tables associated to the different level of the hierarchies or to categories other than base categories (to which the *fact tables* are usually associated).

Example 7. (example 3 cont.) In Figure 4 we have two kinds of tables associated to categories. Relation *HospitalDescription* (Table 7) is connected only to the *Hospital* category, providing descriptions for the elements of the category. In contrast, relation *UnitInst* (Table 6) represents *Guideline 2* and contains attributes connecting two categories, *Unit* and *Manufacturer*, in different dimensions, say *Location* and *Product*. It contains an extra, non-dimensional attribute *Origin*.

A table like *UnitInst* can be used in combination with dimensional navigation to obtain the required data; in this case about the instrument used with the patients of a given unit. Notice that the table might be incomplete: not all units or manufactures are necessarily related. □

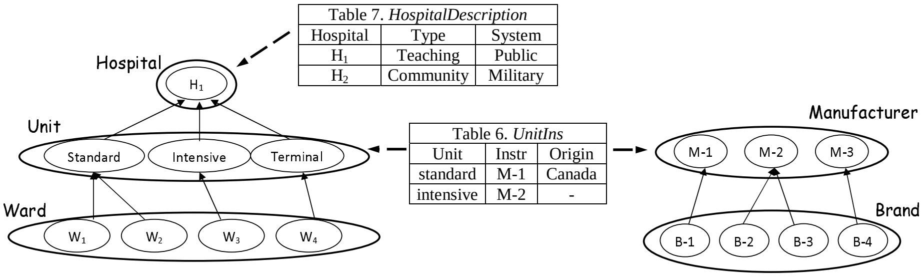


Table 7. *HospitalDescription*

Hospital	Type	System
H$_1$	Teaching	Public
H$_2$	Community	Military

Table 6. *UnitIns*

Unit	Instr	Origin
standard	M-1	Canada
intensive	M-2	-

Fig. 4: Attributive and categorial relations

In more general terms, we extend the multidimensional model (MDM) with *categorical relations* and *attributive relations*, both of them connected to categories of dimension hierarchies.

In a *categorical relation* (CR) some of the attributes are category names (or nicknames for them). In such a case, the attribute and the category share the underlying domain. In order to establish the connection between the attribute of a CR, R, and the corresponding category, we need *schema mappings*. More precisely, if the ith position of R corresponds to the category $C \in CAT$, then we have the following mapping:

$$\forall x_1 \ldots x_n(R(x_1, \ldots, x_i, \ldots, x_n) \rightarrow C(x_i)).$$

Actually, this formula acts as a sort of referential constraint.

UnitInst is an example of a CR. Another example is *Measurement* (Table 2), as shown in Figure 2. It contains the attribute *Ward*, that corresponds to a category in the *Location* dimension schema. Accordingly, we have the mapping:

$$\forall p \forall v \forall h \forall w(\mathit{Measurement}(p, v, h, w) \rightarrow \mathit{Ward}(w)).$$

We also have *attributive relations* (AR). They are connected to a single category in a single dimension schema. Each AR provides, through its attributes, description for the elements of the category. In terms of mappings between the relational and dimensional schemas, they may be as for CRs. However, we may also have constraints of the form

$$\forall x \exists y_1 \cdots y_n(C(x) \rightarrow R(y_1, \ldots, x, \ldots, y_n)).$$

HospitalDescription (Table 7) is an example of an AR that provides descriptions for the elements of the *Hospital* category of dimension *Location*. The following constraint should be satisfied:

$$\forall h \exists t \exists s(\mathit{Hospital}(h) \rightarrow \mathit{HospitalDescription}(h, t, s)).$$

Notice that from the formal point of view, a one-category categorical relation, i.e. with a single categorical attribute, does not differ from an attributive relation. However, the latter kind is always associated to a single category, whose elements are described through the other attributes. As opposed to CRs, an AR is not meant to support navigation via the categorical attributes (a process that could be realized with the machinery developed in [25, 26]).

Many other sensible elements could be added to this dimensional embedding. For example, *intradimensional* and *interdimensional* constraints. The former may restrict certain combinations of descriptive values (in attributive relations) associated to elements in different categories that are $<^*$-connected. The latter may prohibit combinations of values from several dimensions, as would be the case of categorical relations.

Notice that these constraints may have interesting logical interactions with the well-known *semantic constraints* of the HM models, e.g. about *homogeneity* (every element of a category rolls up to an element of a parent category) and *strictness* (rolls up to at most one) [21].

Example 8. Suppose we want to specify the commonsense assumption that no single measurement can be taken by more than one nurse. For this, we can use $\mathit{PatientValue}(\mathit{Patient}, \mathit{Value}, \mathit{Time})$, that can be seen as a CR linked to the *Time* dimension through its last attribute; and also $\mathit{PatientWard}(\mathit{Patient}, \mathit{Day}, \mathit{Ward})$, a CR linked to both the *Location* and *Time* dimensions. We also need to appeal to the *Nurse* category of the *Location* category. With all these elements, the above requirement can be expressed as an *interdimensional constraint*:

$$\neg\exists p\ v\ t\ d\ w\ n_1 n_2\ (\ \mathit{PatientValue}(p,v,t) \wedge \mathit{PatientWard}(p,d,w) \wedge T(t,d) \wedge L(n_1, w) \wedge L(n_2, w) \wedge n_1 \neq n_2\ ).$$

Here, T and L are (the extensions of) the partial orders ($<$) between elements of categories *Time* and *D*ay in the *Time* dimension, and *Nurse* and *W*ard in the *Location* dimension, respectively. This constraint takes the form of a *denial constraint*, that prohibits certain combinations of atoms. □

In this section we have shown how contexts can be enriched by the introduction of multidimensional models. The basis can be a model for MDDBs, like the one in [21]. However, it can be extended with additional data associated to category elements. On the other side, a given contextual relation can be linked to one or several dimension instances. Navigation through them allows us to find explicit additional information or implicit information, like the one provided by guidelines, as rules in the contextual framework.

4 Ontological Representation of Contextual Dimensions

A context can be represented as an ontology expressed in a logical language. As such, contexts will possibly admit several models in comparison to a context that is represented by a single relational database. As a consequence, logical reasoning, for example for checking inconsistencies, deriving implied relations, inferring instances of relationships and concepts, etc., becomes a new issue. From this point of view, a context becomes a knowledge base containing explicit data, e.g. in relational tables, explicit metadata, including data semantics, and also general rules that can be used for extracting implicit information. This would be the case, for example, of the hospital guidelines of the previous sections. An ontology would allow us to extract and analyze information via some sort of logical reasoning.

In this section, with the motivation of having a full-fledge ontology as a contextual framework for data quality assessment [6], we will show how the dimensions (and their extensions) introduced in the previous section can be expressed as a part of an ontology in a *description logic* (DL). In DL there is knowledge both at the intentional level, in a terminological TBox, $\mathcal{T}$, and at the extensional level, in an assertional ABox, $\mathcal{A}$. DL ontologies contain formulas expressed in terms of *concepts* and *roles*. Some of them are atomic and other can be defined.

Ontological, DL-based representation of dimensions have been previously proposed in [15, 16], as representations of data warehouse conceptual schemas. They start from description of DWH schemas as extended entity relationship (EER) diagrams, and provide the formal semantics for the EER representation via the $\mathcal{ALCFI}$ description logic [19].

Here, we sketch a DL-based representation of our extended contextual MD model in one of the members of *DL-Lite* family [12]. In general, *DL-Lite* and its extensions have a good balance of expressive power and good computational properties; and have found interesting applications in data management and semantics web [13].

The contextual relational schema and the contextual dimension schemas become part of the TBox $\mathcal{T}$. Since we may be dealing with large amounts of data, we keep the data at their sources, which are connected to the TBox, $\mathcal{T}$, via mappings. For that purpose we use functions symbols, as proposed in [27], for linking data sources to DL ontologies. Thus, instances of contextual dimensions, categorical and attributive relations will all be mapped to the corresponding constructs in $\mathcal{T}$ (concepts and roles).

Example 9. Consider the dimension schema for *Location* in Figure 3a. The categories (category names) and attribute domains for attributive relations are represented as concepts in $\mathcal{T}$: *Nurse, Ward, Unit,* $\ldots$*, Integer, String,* etc.

Two different category of the *Location* dimension are assumed to be disjoint, which is captured by axioms in $\mathcal{T}$, e.g. $Nurse \sqcap Ward \sqsubseteq \perp$, $Ward \sqcap Unit \sqsubseteq \perp$, etc.

Each dimension schema has a unique role, denoted with the same name as the corresponding dimension. It represents the child/parent relation between elements of categories ($<$), and we also make it transitive, using the axiom: *Tra(Location)*.

As shown in Figure 3a, for categories *Ward* and *Unit*, we have: $Ward \nearrow Unit$, which is represented by the following axiom: $Unit \sqsubseteq \exists Location^-.Ward$.

Attribute *HospitalType* in attributive relation *HospitalDescription* (cf. Table 7 in Example 7, providing descriptions for elements of the *Hospital* category, is represented as a role, *HospitalType*, between the *Hospital* category and the attribute domain. For *HospitalType*, the following axioms are introduced:

$$\exists HospitalType^- \equiv Hospital, \;\; \exists HospitalType \sqsubseteq String, \;\; \geq 2 HospitalType \sqsubseteq \perp.$$

Similarly for other attributes. The first two describe the kind of argument the role takes, and the third one that each hospital is at most of one type.

The ABox is not explicitly represented. Instead, the TBox goes for data (or facts) to the data sources via mappings. For example, assume that we have a relation *locationIns(Ward,Unit)*. It is a subrelation of the dimension instance; actually the one at the top of Figure 2. It has to be mapped into the *Location* role. More precisely, the (virtual) instances of the concept *Ward* and the role *Location* are, possibly only partially built from *locationIns* via:

$$\forall w \forall u (locationIns(w,u) \;\; \rightarrow \;\; Ward(f_{ward}(w)),$$
$$\forall w \forall u (locationIns(w,u) \;\; \rightarrow \;\; Location(f_{ward}(w), f_{unit}(u)).$$

The reason for having the functional terms on the right-hand-side is that the "abstract representation" of data values at the ontological level may differ from the actual values at the source level. Thus, the latter are mapped to abstract values through the representation functions [27]. The reader can safely assume for the rest of this section that each of those functions is the identify function.

The attribute *Type* in the attributive relation *HospitalDescription* (Table 7) is mapped to the role *HospitalType* through the mapping:

$$\forall h \forall t (HospitalDescription(h,t) \rightarrow HospitalType(f_{hospital}(h), t).$$

In this case, the hospital type (a string) is mapped as it is as a value at the ontological level.

Guidelines, e.g. *Hospital Guideline 2* stating that medical tests on patients in *standard care* units have to be taken with instruments made by manufacturer M_1, can be captured as axioms in the TBox $\mathcal{T}$. In order to do this, we first need to specify two concepts, *StandardCon* and $\mathrm{M}_1 Con$, respectively containing the element $standard$ from category *Unit*, and the element M_1 from category *Manufacturer*. That is, $\mathrm{M}_1 Con(\mathrm{M}_1)$

and $StandardCon(standard)$ are facts in the ontology.[1] These two new concepts satisfy the following conditions:

$$StandardCon \sqsubseteq Unit, \ \ \mathrm{M}_1Con \sqsubseteq Manufacturer.$$

We introduce a new concept *StandardRelate* that contains all the locations that have $standard$ as an ancestor in *Unit* category. Similarly, $\mathrm{M}_1Relate$ is a concept with instruments with M_1 as an ancestor in the *Manufacturer* category. The satisfy the conditions:

$$StandardRelate \equiv \exists Location.StandardCon, \ \ \mathrm{M}_1Relate \equiv \exists Instrument.\mathrm{M}_1Con.$$

Finally, *Guideline 2* can be expressed using the role *UnitInst* that represents the attributive relation in Table 6, by: $\exists UnitIns^-.StandardRelate \sqsubseteq \mathrm{M}_1Relate.$ □

More specifically, our ontological representation can be done in *DL-Lite*$^{(\mathcal{H}N)^+}_{Horn}$ [2]. *DL-Lite* contains names for atomic concepts, e.g. *Nurse*, and $\bot$, the *empty concept*, used, e.g. to say that *Nurse* and *Ward* are disjoint. In *DL-Lite* we also have atomic roles, R, and inverse role, R^-, which we used in the axiom $\exists HospitalType^- \equiv Hospital$, saying that the domain of role *HospitalType* coincides with the concept *Hospital*.

In an extension of *DL-Lite*, *DL-Lite*$^{\mathcal{N}}$, we find *number restrictions* of the form $\geq qR$, that we used in $\geq 2HospitalType \sqsubseteq \bot$, for expressing that concept *Hospital* is functional via attribute *HospitalType*. Now, *DL-lite*$^{\mathcal{N}}_{Horn}$ allows for concept inclusions of the form $B_1 \sqcap B_2 \sqcap \ldots \sqcap B_n \sqsubseteq B$, that we used in $Nurse \sqcap Ward \sqsubseteq \bot$.[2]

Next, with *DL-Lite*$^{\mathcal{H}N}_{Horn}$, we also have role hierarchies ($\mathcal{H}$), or inclusions of the form $R_1 \sqsubseteq R_2$, which is used as a basis for defining the property of *transitive* role, producing the extension *DL-Lite*$^{(\mathcal{H}N)^+}_{Horn}$ [2]. We used it with the axiom *Tra(Location)*.

DL-Lite$^{(\mathcal{H}N)^+}_{Horn}$ also offers the *qualified existential quantifier* $\geq qR.C$, that we used in $Unit \sqsubseteq \exists Location^-.Ward$ (with $q = 1$), for expressing that *Units* contains values that are connected to the elements of *Wards* through the role *Location*.

5 Related Work

A hierarchal framework for organizing data quality dimensions is proposed in [29], with a discussion of high-quality data as intrinsically good, contextually appropriate for the task, and clearly represented and accessible to the data consumer. Research on data quality problems are mainly based on the implicit assumption that data errors are mostly syntactic errors [3]. As discussed in [22], data quality may also be related to the semantic of data. For example, a data quality problem occurs when there is a difference between the *intended* meaning (according to its producer) and *interpreted* meaning (according to its consumer) of a data value. The discrepancy occurs because the communication between the data producer and costumer is not clear.

In [24], contexts are represented as first class, named objects in a logical theory, i.e. they appear in the form $ist(c, p)$, saying that proposition p is true in context (denoted by) c.

[1] Since these concepts reside at the ontological level, in order to be consistent with the our previous development, we should abstractly represent their elements by functions $f_{unit}(standard)$, $f_{manuf}(\mathrm{M}_1)$, with mappings from the sources, as above. For simplicity we omit doing this.

[2] Notice that in this extensions of *DL-Lite* we do not have explicit negation of concepts, as it is the case in its *krom* or *bool* extensions.

A context n be defined as a partial description of the world [17, 18]. Each context is represented in a formal language, and checking the satisfiability of a formula in that language is local and performed in its context. In addition, contexts mutually influence themselves and different forms of reasoning, in different contexts, are made compatible.

There are some previous proposals for using contexts in data management [6, 8, 10], including some dimensional aspects [9]. In [10], contextual commonsense assumptions are expressed as sets of variables that may be a point of interest for an agent. The values of those variables influence the agent's actions. The representation of context in [10, 9] is based on a tree-like context model, whose instances are sets of context elements, in essence, collections of attribute-value pairs, e.g. *role = 'CEO', situation = 'in-office' and location = 'city'*. Context-aware views are automatically or semi automatically generated from a given database and a contextual instance, allowing to see the data from different perspectives or points of view. The model also allows for the specification of constraints on a context instance, e.g. sating that when a role is 'CEO', a situation cannot be 'on-site'. In our work, this type of conditions is captured by interdimensional constraints.

In this work we represented contexts as extensions of MD ontologies written in DL. As previously discussed in this paper, there are other proposals for representing a data warehouse conceptual schema [15, 16, 28]. These use as a first modeling step an extended entity relationship (EER), whose formal semantics of EER is captured by means of an ontology written a DL in the $\mathcal{ALCFI}$ family.

6 Conclusions

In this paper we have pointed to issues, problems around, and sketched approaches to, *using and going for data* that can be used by a context to assess the quality of a given database instance. The latter is put in(to the) context via schema mappings [6, 7]. In this paper, instead of emphasizing the use of the contextual data for quality assessment, we have concentrated on extending the context model introduced in [6, 7], by adding rules, dimensions, dimensional data, and, finally putting all this together as a DL ontology. As a next step, (not developed in this paper, but see [6]), the ontological context is used for assessing the quality of a database instance through the use of quality properties. The assessment is done by comparing the database instance under assessment with a class of alternative, intended instances that are obtained by the interaction of the original data with the context. Investigating the quality query answering through the ontological context is also a part of our ongoing research [6]. In addition, we are currently exploring alternative representations for the contextual ontologies and the access to data through them, e.g. using $Datalog^{+}_{-}$ [11].

In this work, we have focused mainly on the introduction of contextual dimension for navigating in search for the data required to assess the quality of another set of data. This also allows doing data quality assessment from multiple views and level of granularity. In this regard, we can benefit from tools and methodologies developed in [25, 26] in the form of an extended relational algebra, the *contextual relational algebra*.

Our proposed multidimensional contexts allows for natural extensions, such as the introduction of explicit, lower-level data quality predicates [22], the access to external data sources at quality-assessment time, and also *intra*- and *interdimensional semantic* *constraints*. This latter addition is particularly interesting since they do not commonly appear in MDDBs, but they could interact with usual MD semantic constraints, like homogeneity and strictness [21]. Notice that the combination of role transitivity, e.g. of *Location* above, and functionality, e.g. $Nurse \sqcap \geq 2 Location.Ward \sqsubseteq \perp$ (of the roll-up relation from *Nurse* to *Ward*), is problematic [20]. All these issues are subject to our ongoing research.

Acknowledgments: Research funded by the NSERC Strategic Network on BI (BIN, ADC05 & ADC02) and NSERC/IBM CRDPJ/371084-2008. L. Bertossi is a Faculty Fellow of the IBM CAS.

References

[1] Arenas, M., Barcelo, P., Libkin, L. and Murlak, F. *Relational and XML Data Exchange*. Synthesis Lectures on Data Management, Morgan & Claypool Publishers, 2010.

[2] Artale, A., Calvanese, D., Kontchakov, R. and Zakharyaschev, M. The *DL-Lite* Family and Relations. J. Artif. Intell. Res., 36, 2009, pp. 1-69.

[3] Batini, C. and Scannapieco, M. *Data Quality: Concepts, Methodologies and Techniques*. Springer, 2006.

[4] Bernstein, Ph. and Melnik, S. Model Management 2.0: Manipulating Richer Mappings. Proc. SIGMOD 2007, pp. 1-12.

[5] Bertossi, L. and Bravo, L. Consistent Query Answers in Virtual Data Integration Systems. In *Inconsistency Tolerance*, Springer LNCS 3300, 2004, pp. 42-83.

[6] Bertossi, L., Rizzolo, F. and Lei, J. Data Quality is Context Dependent. Proc. WS. Enabling Real-Time Business Intelligence (BIRTE'10), Springer LNBIP 48, 2011, pp. 52-67.

[7] Bertossi, L., Rizzolo, F. Contexts and Data Quality Assessment. Journal subm., Feb. 2012.

[8] Bolchini, C., Curino, C., Quintarelli, E., Schreiber, F. and Tanca, L. A Data-Oriented Survey of Context Models. *SIGMOD Record*, 2007, 36(4):19-26.

[9] Bolchini, C., Curino, C., Quintarelli, E., Schreiber, F. and Tanca, L. Context Information for Knowledge Reshaping. *Int. J. Web Eng. Technol*, 2009, 5(1):88-103.

[10] C. Bolchini, C. Curino, G. Orsi, E. Quintarelli, R. Rossato, F. A. Schreiber, and L. Tanca. And What Can Context Do for Data? *Commun. ACM*, 52(11):136–140, 2009.

[11] Cali, A., Gottlob, G. and Lukasiewicz, T. Datalog Extensions for Tractable Query Answering over Ontologies. In *Semantic Web Information Management*, R. De Virgilio, F. Giunchiglia, L. Tanda (eds.), Springer, 2010, pp. 249-279.

[12] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M. and Rosati, R. Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. *J. Autom. Reasoning*, 2007, 39(3):385-429.

[13] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R. and Ruzzi, M. Using OWL in Data Integration. In *Semantic Web Information Management*, R. De Virgilio, F. Giunchiglia, L. Tanda (eds.), Springer, 2010, pp. 397–424.

[14] Coutaz, J., Crowley, J.L., Dobso, S. and Garlan, D. Context Is Key. *Commun. of ACM*, 2009, 48(3):49-53.

[15] Franconi, E. and E.Sattler. A Data Warehouse Conceptual Data Model for Multidimensional Aggregation. Proc. DMDW 1999, CEUR WS Proc., Vol. 19.

[16] Franconi, E. and Sattler, E. A Data Warehouse Conceptual Data Model For Multidimensional Aggregation: A Preliminary Report. *AI*IA Notizie*, 1999, 1:9-21.

[17] Giunchiglia, F. Contextual Reasoning. *Proc. IJCAI-WS on Using Knowledge in its Context*, 1993, pp. 39-49.

[18] Ghidini, C. and Giunchiglia, F. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. *Artificial Intelligence*, , 2001, 127:221-259.

[19] Horrocks, I. and Sattler, U. A Description Logic with Transitive and Inverse Roles and Role Hierarchies. *Journal of Logic and Computation*, 1999, 9(3):385-410.

[20] Horrocks, I., Sattler, U. and Tobies, S. Practical Reasoning for Very Expressive Description Logics. *J. of the Interest Group in Pure and Applied Logic*, 2000, 8(3):239-264.

[21] Hurtado, C.A., Gutierrez, C., and Mendelzon, A.O. Capturing Summarizability with Integrity Constraints in OLAP. ACM Trans. Database Syst. 30(3). (2005) 854-886

[22] Jiang, L., Borgida, A. and Mylopoulos, J. Towards a Compositional Semantic Account of Data Quality Attributes. Proc. ER 2008, Springer LNCS 5231, pp. 55-68.

[23] Lenzerini, M. Data Integration: A Theoretical Perspective. Proc. PODS 2002, pp. 233-246.

[24] McCarthy, J. Notes on Formalizing Context. Proc. IJCAI 1993, pp. 555-562.

[25] Martinenghi, D. and Torlone, R. Querying Context-Aware Databases. Proc. FQAS 2009, pp. 76-87.

[26] Martinenghi, D. and Torlone, R. Querying Databases with Taxonomies. Proc. ER 2010, pp. 377-390.

[27] Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M. and Rosati, R. Linking Data to Ontologies. *J. on Data Semantics*, 2008, 10:133-173.

[28] U. Sattler. Description Logics for the Representation of Aggregated Objects. In W.Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 239–243. IOS Press, Amsterdam, 2000.

[29] Wang, R. and Strong, D. Beyond Accuracy: What Data Quality Means to Data Consumers. *J. Management and Information Systems,*, 1996, 12(4):5-33.