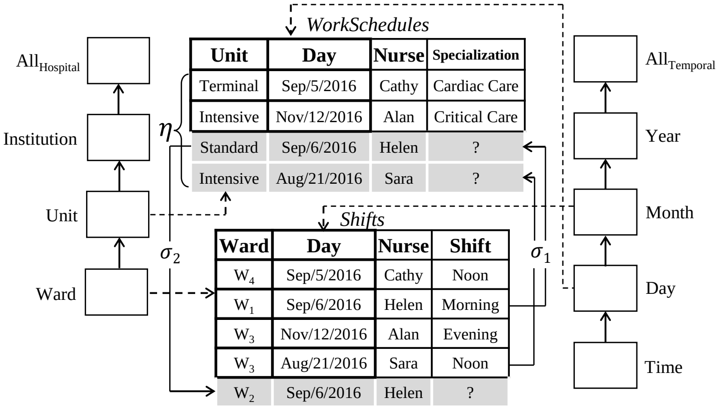
The Ontological Multidimensional Data Model in Quality Data Specification and Extraction

(extended abstract and progress report)

Leopoldo Bertossi[⋆] and **Mostafa Milani**[⋆⋆]

In this abstract we briefly present, using a running example : (a) the *Ontological Multidimensional Data Model* (OMD model) [3, 13] as an ontological, Datalog$^{\pm}$-based [6] extension of the Hurtado-Mendelzon (HM) model for multidimensional data [8]; (b) its use for quality data specification and extraction via query answering; and (c) some ongoing research.


$\mathrm{All}_{\mathrm{Hospital}}$
Institution
Unit
Ward
η
σ_2
σ_1
$\mathrm{All}_{\mathrm{Temporal}}$
Year
Month
Day
Time

WorkSchedules

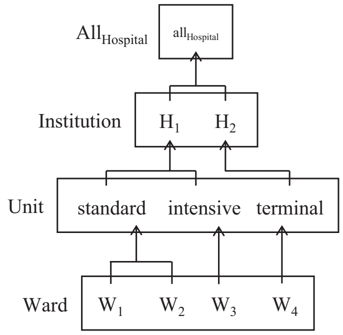
Unit	Day	Nurse	Specialization
Terminal	Sep/5/2016	Cathy	Cardiac Care
Intensive	Nov/12/2016	Alan	Critical Care
Standard	Sep/6/2016	Helen	?
Intensive	Aug/21/2016	Sara	?

Shifts

Ward	Day	Nurse	Shift
W_4	Sep/5/2016	Cathy	Noon
W_1	Sep/6/2016	Helen	Morning
W_3	Nov/12/2016	Alan	Evening
W_3	Aug/21/2016	Sara	Noon
W_2	Sep/6/2016	Helen	?

Fig. 1. An OMD model with categorical relations, dimensional rules, and constraints

An OMD model has a *database schema* $\mathcal{R}^{\mathcal{M}} = \mathcal{H} \cup \mathcal{R}^c$, where $\mathcal{H}$ is a relational schema with multiple dimensions, with a set $\mathcal{K}$ of unary category predicates, and sets $\mathcal{L}$ of binary, child-parent predicates; and $\mathcal{R}^c$ is a set of *categorical predicates.*



Example: Figure 1 shows Hospital and Temporal dimensions. The former's instance is here on the RHS. $\mathcal{K}$ contains predicates $\mathit{Ward}(\cdot)$, $\mathit{Unit}(\cdot)$, $\mathit{Institution}(\cdot)$, etc. Instance $D^{\mathcal{H}}$ gives them extensions, e.g. $\mathit{Ward} = \{\mathsf{W}_1, \mathsf{W}_2, \mathsf{W}_3, \mathsf{W}_4\}$. $\mathcal{L}$ contains, e.g. $\mathit{WardUnit}(\cdot,\cdot)$, with extension: $\{(\mathsf{W}_1$, standard), (W_2, standard), (W_3, intensive), (W_4, terminal)$\}$. In the middle of Figure 1, *categorical relations* are associated to dimension categories, e.g. $\mathit{WorkSchedules} \in \mathcal{R}^c$. □

Attributes of categorical relations are either *categorical*, whose values are members of dimension categories, or *non-categorical*, taking values from arbitrary domains. Categorical predicates are represented in the form $R(C_1, \ldots, C_m; N_1, \ldots, N_n)$, with categorical attributes before ";" and non-categorical after.

The extensional data, i.e the instance for the schema $\mathcal{R}^{\mathcal{M}}$, is $I^{\mathcal{M}} = D^{\mathcal{H}} \cup I^c$, where $D^{\mathcal{H}}$ is a complete instance for dimensional subschema $\mathcal{H}$ containing the category and child-parent predicates; and sub-instance I^c contains possibly

[⋆] Carleton Univ., School of Computer Science, Canada. bertossi@scs.carleton.ca
[⋆⋆] McMaster Univ., Dept. Computing and Software, Canada. mmilani@mcmaster.ca

partial, incomplete extensions for the categorical predicates, i.e. those in $\mathcal{R}^c$. Schema $\mathcal{R}^{\mathcal{M}}$ comes with a set Ω^M of basic, application-independent semantic constraints:

1. Dimensional child-parent predicates must take their values from categories. Accordingly, if child-parent predicate $P \in \mathcal{L}$ is associated to category predicates $K, K' \in \mathcal{K}$, in this order, we introduce inclusion dependencies (IDs) as Datalog$^\pm$ *negative constraints* (*ncs*): $P(x,x'),\ \neg K(x)\ \rightarrow\ \bot$, and $P(x,x'),\ \neg K'(x')\ \rightarrow\ \bot$. (The $\bot$ symbol is denotes an always false propositional atom.) We do not represent them as Datalog$^\pm$'s *tuple-generating dependencies* (*tgds*) $P(x,x') \rightarrow K(x)$, etc., because we reserve *tgds* for possibly incomplete predicates (in their RHSs).

2. Key constraints on dimensional child-parent predicates $P \in \mathcal{K}$, as *equality-generating dependencies* (*egds*): $P(x,x_1), P(x,x_2)\ \rightarrow\ x_1 = x_2$.

3. The connections between categorical attributes and the category predicates are specified by means of *ncs*. For categorical predicate R: $R(\bar{x};\bar{y}),\ \neg K(x)\ \rightarrow\ \bot$, where $x \in \bar{x}$ takes values in category K.

Example: Categorical predicate $\mathit{WorkSchedules}(\mathit{Unit},\mathit{Day};\mathit{Nurse},\mathit{Speciality})$ has categorical attributes Unit and Day connected to the $\mathsf{Hospital}$ and $\mathsf{Temporal}$ dimensions. E.g. the ID $\mathit{WorkSchedules}[1] \subseteq \mathit{Unit}[1]$ is written in Datalog$^+$ as $\mathit{WorkSchedules}(u,d;n,t), \neg\, \mathit{Unit}(u)\ \rightarrow\ \bot$. For the $\mathsf{Hospital}$ dimension, one of the two IDs for the child-parent predicate $\mathit{WardUnit}$ is $\mathit{WardUnit}[2] \subseteq \mathit{Unit}[1]$, which is expressed as a *nc*: $\mathit{WardUnit}(w,u), \neg\, \mathit{Unit}(u) \rightarrow \bot$. The key constraint on $\mathit{WardUnit}$ is the *egd*: $\mathit{WardUnit}(w,u),\ \mathit{WardUnit}(w,u')\ \rightarrow u = u'$. □

The OMD model allows us to build *multidimensional ontologies*, $\mathcal{O}^{\mathcal{M}}$, which contains -in addition to an instance $I^{\mathcal{M}}$ for schema $\mathcal{R}^{\mathcal{M}}$, and the set $\Omega^{\mathcal{M}}$ in **1.-3.** above- a set $\Sigma^{\mathcal{M}}$ of *dimensional rules* (those in **4.** below), and a set $\kappa^{\mathcal{M}}$ of *dimensional constraints* (in **5.** below); of all of them application-dependent and expressed in the Datalog$^\pm$ language associated to schema $\mathcal{R}^{\mathcal{M}}$.

4. *Dimensional rules* as Datalog$^+$ *tgds*: $R_1(\bar{x}_1;\bar{y}_1), ..., R_n(\bar{x}_n;\bar{y}_n), P_1(x_1,x_1'), ..., P_m(x_m,x_m')\ \rightarrow\ \exists \bar{y}'\ R_k(\bar{x}_k;\bar{y})$. Here, the $R_i(\bar{x}_i;\bar{y}_i))$ are categorical predicates, the P_i are child-parent predicates, $\bar{y}' \subseteq \bar{y}$, $\bar{x}_k \subseteq \bar{x}_1 \cup ... \cup \bar{x}_n \cup \{x_1, ..., x_m, x_1', ..., x_m'\}$, $\bar{y} \smallsetminus \bar{y}' \subseteq \bar{y}_1 \cup ... \cup \bar{y}_n$; repeated variables in bodies (join variables) appear only categorical positions in categorical relations and in child-parent predicates. Existential variables appear only in non-categorical attributes.

5. *Dimensional constraints*, as *egds* or *ncs*: $R_1(\bar{x}_1;\bar{y}_1), ..., R_n(\bar{x}_n;\bar{y}_n), P_1(x_1,x_1'), ..., P_m(x_m,x_m')\ \rightarrow\ z = z'$, and $R_1(\bar{x}_1;\bar{y}_1), ..., R_n(\bar{x}_n;\bar{y}_n), P_1(x_1,x_1'), ..., P_m(x_m,x_m') \rightarrow \bot$. Here, $R_i \in \mathcal{R}^c$, $P_j \in \mathcal{L}$, and $z, z' \in \bigcup \bar{x}_i \cup \bigcup \bar{y}_j$.

Example: Figure 1 shows a *dimensional constraint* η on categorical relation $\mathit{WorkSchedules}$, which is linked to the $\mathsf{Temporal}$ dimension via the Day category. It says: *"No personnel was working in the Intensive care unit in January"*: $\eta\colon\ \mathit{WorkSchedules}(\mathsf{intensive}, d; n, s), \mathit{DayMonth}(d, \mathsf{jan})\ \rightarrow\ \bot$.

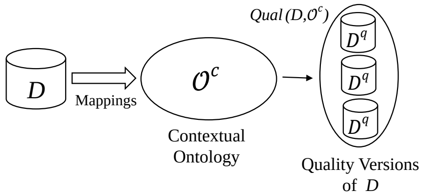
Figure 1 we also shows the dimensional *tgd* $\sigma_1\colon\ \mathit{Shifts}(w,d;n,s),\ \mathit{WardUnit}(w, u) \rightarrow \exists t\ \mathit{WorkSchedules}(u,d;n,t)$, saying that *"If a nurse has shifts in a ward on a specific day, he/she has a working schedule in the unit of that ward on the same day"*. The use of σ_1 generates, from the *Shifts* relation, new tuples for relation *WorkSchedules*, with *null values* for the *Specialization* attribute. Relation *Work Schedules* may be incomplete, and new -possibly virtual- entries can be inserted (the shaded ones showing *Helen* and *Sara* working at the *Standard* and *Intensive* units, resp.). This is done by *upward navigation and data propagation* through the dimension hierarchy.

Now, σ_2: $\mathit{WorkSchedules}(u, d; n, t)$, $\mathit{WardUnit}(w, u) \rightarrow \exists s \mathit{Shifts}(w, d; n, s)$ is a dimensional *tgd* that can be used with *WorkSchedules* to generate data for categorical relation *Shifts* (its shaded tuple is one of them). It reflects the guideline stating that *"If a nurse works in a unit on a specific day, he/she has shifts in every ward of that unit on the same day"*. σ_2 supports downward navigation and tuple generation, from the *Unit* category down to the *Ward* category.

If we have a categorical relation *Therm*(*Ward*, *Thertype*; *Nurse*), with *Ward* and *Thertype* categorical attributes (the latter for an Instrument dimension), the following is an *egd* saying that *"All thermometers in a unit are of the same type"*: $\mathit{Therm}(w, t; n), \mathit{Therm}(w', t'; n'), \mathit{WardUnit}(w, u), \mathit{WardUnit}(w', u) \rightarrow t = t'$. □

The OMD model goes far beyond classical MD data models. It enables *ontology-based data access* (OBDA) [10] and allows for a seamless integration of a logic-based conceptual model and a relational model, while representing dimensionally structured data. Our MD ontologies have good computational properties [3, 13]. Actually, they belong to the class of *weakly-sticky* Datalog$^\pm$ programs [7], for which conjunctive query answering (CQA) can be done in polynomial time in data [14].

With dimensions as fundamental elements of contexts, the OMD model can be use for contextual quality data specification and extraction [13]. The context $\mathcal{O}^c$ is represented as a ontology that contains an OMD model (as a sub-ontology), possibly extra not-necessarily dimensional data, and additional predicate definitions that are used as auxiliary tools for the specification of data quality concerns. A database instance D under quality assessment and quality data extraction is logically mapped into the $\mathcal{O}^c$, for further processing through the context, which provides the otherwise missing elements for addressing or imposing quality concerns on D. The mapping and processing of instance D into/in the context may give rise to alternative quality versions, D^q, of D. (Cf. the figure right below.)



The *quality data* are those shared by all the *quality instances*. Quality data is then extracted from the context through *certain* CQA from the collection, $\mathit{Qual}(D, \mathcal{O}^c)$, of quality instances.

There are several directions of interesting ongoing research in relation to the OMD model and its applications, in general and to data quality. We mention two. First, the interaction of *tgds* and constraints, specially *egds*, may lead to inconsistency. Under certain conditions, such as *separability* [7], the combination with *egds* is computationally easier to handle. In the general case, it may be necessary to apply a *repair semantics*, e.g. to obtain *inconsistency-tolerant* query answers. There are repair semantics for Datalog$^\pm$ ontologies [11] (and also for DL ontologies [9]). Most commonly, the extensional data are (minimally) repaired. In our case, this means repairing MD data, for which certain special MD repair semantics [5, 15] may be better than those applied to general relational data [4]. The OMD model allows to express typical MD constraints that guarantee correct *summarizability* (aggregation) [8]. We might also want to keep them satisfied.

Second, the *open-world assumption* on Datalog$^{\pm}$ (and DL) ontologies makes predicates incomplete (and completable by *tgd* enforcement). Sometimes, in particular with MD extensional data, it may make sense to consider existential variables on categorical attributes with closed domains (e.g. categories and child-parent relations). This creates new issues related to the meaning of existential quantifiers (non-deterministic choices from a fixed set of elements?), data generation, computational aspects, and dimensional navigation (mainly downward, because a parent may have several children). Cf. [2] for Datalog$^{\pm}$ with closed predicates ([1, 12] for the DL case).

References

[1] Ahmetaj, S., Ortiz, M. and Šimkus, M. Polynomial Datalog Rewritings for Expressive Description Logics with Closed Predicates. Proc. IJCAI 2016, pp. 878-885.

[2] Ahmetaj, S., Ortiz, M. and Šimkus, M.: Polynomial Datalog Rewritings for Ontology Mediated Queries with Closed Predicates. Proc. AMW 2016. CEUR 1644.

[3] Bertossi, L. and Milani, M. Ontological Multidimensional Data Models and Contextual Data Quality. Journal submission, 2017. Corr ArXiv paper cs.DB/1704.00115.

[4] Bertossi, L. *Database Repairing and Consistent Query Answering.* Morgan & Claypool, Synthesis Lectures on Data Management, 2011.

[5] Bertossi, L., Bravo, L. and Caniupan, M. Consistent Query Answering in Data Warehouses. Proc. AMW 2009. CEUR 450.

[6] A. Cali, G. Gottlob, and T. Lukasiewicz. Datalog±: A Unified Approach to Ontologies and Integrity Constraints. Proc. ICDT 2009, pp. 14-30.

[7] A. Cali, G. Gottlob, and A. Pieris. Towards more Expressive Ontology Languages: The Query Answering Problem. *Artificial Intelligence*, 2012, 193:87-128.

[8] Hurtado, C. and Mendelzon, A. OLAP Dimension Constraints. Proc. PODS 2002.

[9] Lembo, D., Lenzerini M., Rosati, R., Ruzzi, M. and Savo, F. Inconsistency-Tolerant Query Answering in Ontology-Based Data Access. *J. Web Semantics*, 2015, 3:3-29.

[10] M. Lenzerini. Ontology-Based Data Management. Proc. AMW 2012, CEUR 866.

[11] Lukasiewicz, T., Martinez, M., Pieris, A. and Simari, G. Inconsistency Handling in Datalog+/- Ontologies. Proc. ECAI 2012, pp. 558-563.

[12] Lutz, C., Seylan, I. and Wolter, F. Ontology-Based Data Access with Closed Predicates is Inherently Intractable (Sometimes). Proc. IJCAI 2013.

[13] Milani, M. and Bertossi, L. Ontology-Based Multidimensional Contexts with Applications to Quality Data Specification and Extraction. Proc. RuleML 2015.

[14] Milani, M. and Bertossi, L. Extending Weakly-Sticky Datalog$^{\pm}$: Query-Answering Tractability and Optimizations. Proc. RR 2016.

[15] Yaghmaie, M., Bertossi, L. and Ariyan, S. Repair-Oriented Relational Schemas for Multidimensional Databases. Proc. EDBT 2012.