Unifying Warehoused Data with Linked Open Data: A Conceptual Modeling Solution

Franck Ravat, Jiefu Song

To cite this version : Ravat, Franck and Song, Jiefu *Unifying Warehoused Data with Linked Open Data: A Conceptual Modeling Solution.* (2016) In: 6th International Conference Model and Data Engineering (MEDI 2016), 21 September 2016 - 23 September 2016 (Almeria, Spain).

Unifying Warehoused Data with Linked Open Data: A Conceptual Modeling Solution

Franck Ravat, Jiefu Song

IRIT - Université Toulouse I Capitole, 2 Rue du Doyen Gabriel Marty
F-31042 Toulouse Cedex 09
{ravat|song}@irit.fr

Abstract. Linked Open Data (LOD) become one of the most important sources of information allowing enhancing business analyses based on warehoused data with external data. However, Data Warehouses (DWs) do not directly cooperate with LOD datasets due to the differences between data models. In this paper, we describe a conceptual multidimensional model, named *Unified Cube*, which is generic enough to include both warehoused data and LOD. *Unified Cube*s provide a comprehensive representation of useful data and, more importantly, support well-informed decisions by including multiple data sources in one analysis. To demonstrate the feasibility of our proposal, we present an implementation framework for building *Unified Cube*s based on DWs and LOD datasets.

Keywords: Data Warehouse, Linked Open Data, Unified Conceptual Model

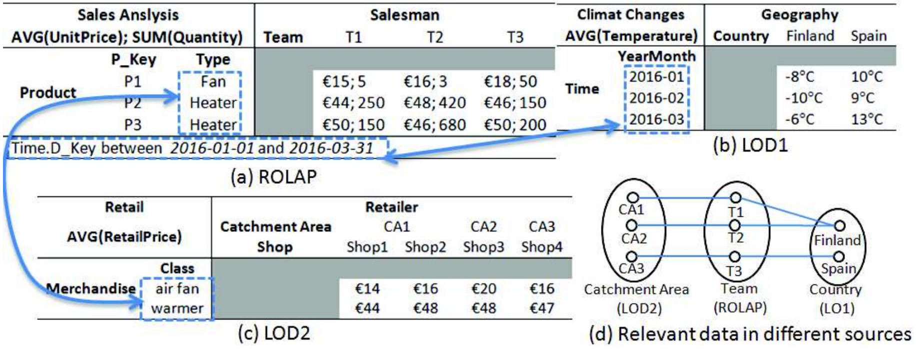
1 Introduction

In today's highly dynamic business context, decision-makers should access internal and external sources to obtain an overall perspective over an organization [2]. *Data Warehouse*s (DWs) have been widely used as internal sources to support online, interactive analyses, while *Linked Open Data* (LOD)[1] have become one of the most important external information sources allowing enhancing business analyses on a web scale [12]. However, warehoused data and LOD follow different models in each domain, which makes it difficult to analyze both types of data in a unified way. Moreover, dispersion of related data in different schemas results in repetitive searches for relevant information in different sources, which reduces the efficiency of analysis.

Motivating example. In a company selling home appliances, a decision-maker looks up in an internal R-OLAP DW to assess the performance of sales staff. The DW relates to an analysis subject (i.e. fact), named *Sales Analysis*, which contains a set of numeric indicators (i.e. measures), namely *unit price* and *quantity*. Each measure can be computed according to three analysis axes (i.e. dimensions): *salesman*, *product* and *time* (cf. figure 1(a)). The R-OLAP DW alone does not provide enough information to support effective and well-informed decisions. The decision-maker must

[1] http://linkeddata.org

search for additional information to obtain other complementary perspectives over the sales activities. Since the sales of some home appliances (e.g., heaters) are strongly influenced by the climate changes, the decision-maker browses in an online dataset denoted LOD1 revealing the *monthly* average temperature according to *countries*. The LOD are published in RDF *Data Cube Vocabulary* (QB)[2] format, which is the current W3C standard to publish multidimensional statistical data. Moreover, since retail sales may compete with the company's promotions in the same catchment area, the decision-maker consults another online dataset denoted LOD2 about the outlet prices offered by rival retailers. The LOD2 dataset is published in QB4OLAP, it involves the *retail price* for a *class* (i.e., type) of merchandise offered by a retailers' *shop*. Extracts of the LOD datasets in tabular form are available in figure 1(b) and (c).



Sales Anslysis			Salesman		
AVG(UnitPrice); SUM(Quantity)		Team	T1	T2	T3
	P_Key	Type			
Product	P1	Fan	€15; 5	€16; 3	€18; 50
	P2	Heater	€44; 250	€48; 420	€46; 150
	P3	Heater	€50; 150	€46; 680	€50; 200

Time.D_Key between *2016-01-01* and *2016-03-31*

(a) ROLAP

Climat Changes		Geography	
AVG(Temperature)	Country	Finland	Spain
	YearMonth		
Time	2016-01	-8°C	10°C
	2016-02	-10°C	9°C
	2016-03	-6°C	13°C

(b) LOD1

Retail		Retailer			
AVG(RetailPrice)	Catchment Area	CA1		CA2	CA3
	Shop	Shop1	Shop2	Shop3	Shop4
	Class				
Merchandise	air fan	€14	€16	€20	€16
	warmer	€44	€48	€48	€47

(c) LOD2

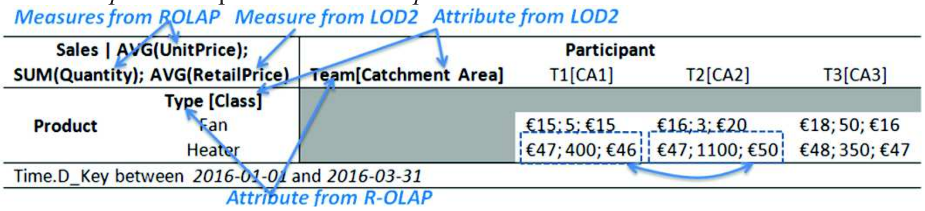
(d) Relevant data in different sources

Fig. 1. Extracts of a ROLAP DW and two LOD datasets.

Without a comprehensive representation of related data, analyses involving several sources are carried out in a sequential way. Decision-makers must explore all data sources one after another before obtaining an overall vision on an analysis subject. Carrying out such analyses is inefficient and difficult, because all schemas do not include the same information at the same analytical granularities: (a) the same analysis axes present in different sources may include different analytical granularities, e.g., for the temporal analysis axis, the source ROLAP contains three analytical granularities *Year-Month-Date*, whereas the sources LOD1 only includes one analytical granularity *YearMonth*; (b) the same data may have different labels in different sources e.g., *heater* is labeled as products' *type* in the source ROLAP and merchandises' *class* in the source LOD2; (c) a same analytical granularity may group several attributes from heterogeneous sources, e.g., since the decision-maker indicates each salesman's team competes with the retailers in one catchment area (cf. figure 1(d)), the attribute named *Team* from ROLAP and the one named *CatchmentArea* from LOD2 refer actually to the same analytical granularity; (d) an analytical granularity from one source may belong to a broader one from a different source, e.g., the decision-maker specifies that several salesman's teams are in charge of the sales in one country (cf. figure 1(d)), therefore the analytical granularity about *team* from ROLAP can be aggregated into the one related to *country* from LOD1; (e) some indicators

[2] http://www.w3.org/TR/vocab-data-cube

from different sources can be analyzed together starting from certain analytical granularities, e.g., figure 2 shows a dashboard including related measures sharing common analytical granularities. This dashboard allows better illustrating the sales' *quantity* is highly influenced by the price: higher sales *quantity* of a *type* of product is due to the lower *unit price* compared to the *retail price* in the *same catchment area*.



Sales | AVG(UnitPrice); SUM(Quantity); AVG(RetailPrice)			Participant		
		Team[Catchment Area]	T1[CA1]	T2[CA2]	T3[CA3]
	Type [Class]				
Product	Fan		€15; 5; €15	€16; 3; €20	€18; 50; €16
	Heater		€47; 400; €46	€47; 1100; €50	€48; 350; €47
Time.D_Key between *2016-01-01* and *2016-03-31*					

Measures from ROLAP
Measure from LOD2
Attribute from LOD2
Attribute from R-OLAP

Fig. 2. A dashboard built on a unified view.

Contribution. Our aim is to make full use of all relevant data in a decision-making context. To this end, we provide decision-makers with a unified view of both warehoused data and multidimensional LOD. To facilitate decision-making, the unified view should include in a single schema all the indicators along with all available analysis axes as well as all the attributes and hierarchies (coming from the heterogeneous sources). The unified view should be independent of the modeling solutions of the data sources. In the previous example, a unified view would enable decision-makers to more easily build the dashboard shown in figure 2.

In this paper, we describe a generic modeling solution for both warehoused data and multidimensional LOD. First, we discuss related work about a unified representation compatible with warehoused data and LOD (cf. section 2). Second, we present the conceptual definitions and graphical notations of *Unified Cube*s (cf. section 3). At last, we describe an implementation framework for *Unified Cube*s (cf. section 4).

2 Related Work

The classical method of analyzing data from multiple sources consists of combining data from several fact tables according to *conformed* dimensions under a DW Bus Architecture [6]. It has very limited practical utility since all dimensions must share the same structure and content across different sources. To overcome this drawback, many state-of-art papers [1, 2, 7] draw a roadmap enabling unified analyses to be carried out based on all kinds of dimensions. A key step towards such analyses consists of a generic multidimensional representation which is compatible with both warehoused data and LOD. Two approaches can be identified from the existing work.

The first approach aims at reusing classical DW models. The work [3, 9, 11] proposes mechanisms to *Extract*, *Transform* and *Load* (ETL) LOD into a local DW. However, warehousing real-time LOD in a stationary data repository is hardly practical [5]. According to the authors of [4], this approach is not recommended, since it collides with the distributed nature and the high volatility of LOD. In terms of analytical uses, the first approach only supports offline analyses of warehoused data with preprocessed LOD, which makes it difficult to guarantee the high freshness of the obtained information.

The second approach aims at publishing LOD according to a multidimensional structure. The RDF *Data Cube Vocabulary* (QB) is the current W3C standard to publish multidimensional statistical data. In [4], the authors propose the QB4OLAP vocabulary which adds more multidimensional characteristics to QB, like multiple analytical granularities within multiple aggregation paths and the specification of the aggregation functions associated to a measure. [8] proposes IGOLAP vocabulary allowing representing the correlation relationships between two different dimensions. QB, QB4OLAP and IGOLAP are logical models expressed in RDF vocabularies. No conceptual model independent of specific modeling languages has been proposed.

In this paper, we propose a generic multidimensional model which provides a uniform vision of both warehoused data and relevant multidimensional LOD. Unlike approaches involving ETL processes which collide with the dynamic nature of data, our unified data model supports on-the-fly analyses of data in the sources.

3 The *Unified Cube* Model

*Unified Cube*s provide a single, comprehensive representation of data from one or multiple sources. Within a *Unified Cube*, data are organized according to analysis axes (i.e., *dimension*s) and an analysis subject (i.e., fact). Concepts about *Unified Cube*s will be presented in the following sections.

3.1 Unified Cube

In a *Unified Cube*, a dimension is a union of relevant analytical granularities from several sources concerning the same analysis axis, while the fact includes all measures concerning the analysis subject. Each measure from one source may only be summarizable with regards to the set of analytical granularities from the same source. A generalization of the *dimension-measure* relationship is needed to associate subset of analytical granularities within a dimension with a measure in a *Unified Cube*.

Definition 1. A *Unified Cube* is a n-dimensional finite space describing a fact with some dimensions. A *Unified Cube* is denoted as UC={F; $\mathcal{D}$; $\mathcal{LM}$}, where

- F is a *fact* containing a set of *measure*s;
- $\mathcal{D}=\{D_1;\ldots;D_n\}$ is a finite set of *dimension*s;
- $\mathcal{LM}$: $2^{\mathcal{L}^1\backslash l_p \times \ldots \times \mathcal{L}^n\backslash l_q} \rightarrow m_e$ is a *level-measure* mapping which associates a subset of summarizable analytical granularities (i.e., *level*s) to a measure m_e of the fact, such as $\forall i \in [1..n]$, $\mathcal{L}^i\backslash_{l_s}$ ($l_s \in \mathcal{L}^i$) corresponds to a subset of levels on the dimension D_i ($D_i \in \mathcal{D}$) which starts from the level l_s.

We propose a graphical notation of *Unified Cube*s based on the *fact-dimension* model with minor modifications (cf. figure 3). The graphical notation aims at facilitating data exploitation at the schema level for non-expert users. For readability purposes, concepts involving data instances are not included in the graphical notation.

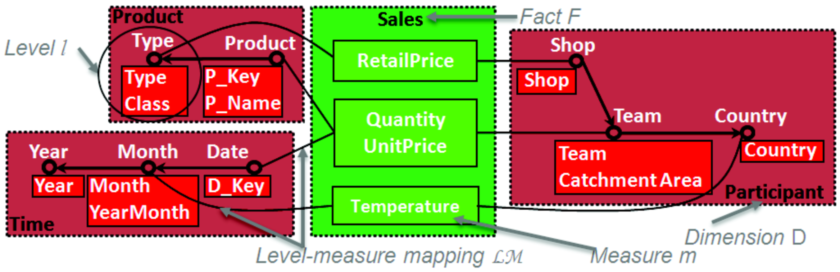


Fig. 3. Graphical notation of *Unified Cubes*.

Example. Figure 3 shows a *Unified Cube* which is built upon the warehoused data and the two LOD datasets in the motivating example. It contains three dimensions $\mathcal{D}$={$D_{Participant}$, D_{Time} , $D_{Product}$}. Each measure is associated to its related levels. For the sake of simplicity, in the graphical notation the *level-measure* mappings are represented only between the lowest levels of sub-dimension and related measures. i.e., $\mathcal{LM}$: {{ $\mathcal{L}^{Product}\backslash l_{Type}$; $\mathcal{L}^{Participant}$ }→{$m_{RetailPrice}$}; { $\mathcal{L}^{Product}$; $\mathcal{L}^{Participant}\backslash l_{Team}$ $\mathcal{L}^{Time}$ }→{$m_{Qunatity}$; $m_{UnitPrice}$}; {$\mathcal{L}^{Time}\backslash l_{Month}$; $\mathcal{L}^{Participant}\backslash l_{Country}$}→{$m_{Temperature}$}}.

3.2 Analysis Subject: Fact

A fact models an analysis subject composed of a set of *measure*s. Since the fact of a *Unified Cube* may include measures from DWs and LOD datasets, we should explicitly indicate how the values of a measure can be accessed from data sources.

Definition 2. A *Fact* corresponds to an analysis subject composed of a set of *measure*s. A fact is denoted as F={n^F; $\mathcal{M}^F$} where

- n^F is the name of the fact;
- $\mathcal{M}^F$={m_1;…; m_p} is a finite set of numeric indicators called *measure*s. Each measure m_e ($m_e \in \mathcal{M}^F$) is a pair ⟨n^{m_e}, E^{m_e}⟩, where n^{m_e} is the name of the measure, E^{m_e} is an *extraction formula* defined through query algebra (e.g., *relational* algebra and *SPARQL* algebra[3]).

Remark. *Extraction formulae* enable on-the-fly querying of measures' values during analyses. The algebraic representation of *extraction formula* makes sure its compatibility with specific implementation environments of data source. Note that although the *SPARQL* algebra is not yet a W3C standard, it has already been integrated within several RDF querying framework. Each algebraic SPARQL expression is translated into one SPARQL query which is generic enough to work with all types of LOD datasets. Table 1 shows the algebraic form of commonly used SPARQL queries.

Table 1. SPARQL queries and their algebraic representation

SPARQL query	**SPARQL Algebra**
SELECT * WHERE { ?s ?p ?o}	(BGP (TRIPLE ?s ?p ?o))

[3] https://www.w3.org/2001/sw/DataAccess/rq23/rq24-algebra.html

SELECT ?s ?p WHERE {?s ?p ?o}	(PROJECT(?s ?p) (BGP (TRIPLE ?s ?p ?o)))
SELECT ?o1 ?o2 WHERE{?s ?p ?o1. FILTER (?o1 < 5) OPTIONAL {?s ?p2 ?o2 . FILTER (?o2 > 10) }}	(PROJECT(?o1 ?o2) (FILTER (< ?o1 5) (LEFTJOIN(BGP (TRIPLE ?s ?p ?o1)) (BGP (TRIPLE ?s ?p2 ?o2)) (> ?o2 10))))
SELECT ?s (COUNT(?o) as ?nb) WHERE {?s ?p ?o} GROUP BY ?s HAVING (COUNT(?o) > 10)	(PROJECT(?s ?nb) (FILTER (> ?.0 10) (EXTEND((?nb ?.0)) (GROUP (?s) ((?.0 (COUNT ?o))) (BGP (TRIPLE ?s ?p ?o)))))

Example. The fact named *Sales* contains four measures, namely $m_{\text{RetailPrice}}$, m_{Quantity}, $m_{\text{UnitPrice}}$ and $m_{\text{Temperature}}$. The measure m_{Quantity} has an *extraction formula* $E^{m_{\text{Quantity}}}={}_{\text{P\_Key, D\_Key, Team}}\mathcal{F}_{\text{sum}}$(SalesAnalysis.Quantity). The *extraction formula* of the measure $m_{\text{RetailPrice}}$ is defined upon SPARQL algebra, such as:

$E^{m_{\text{RetailPrice}}}$ = (project (?retailer ?merchandise ?avgPrice)
(extend ((?avgPrice ?.0)) (group (?retailer ?merchandise) ((?.0 (avg ?Prive)))
(bgp (triple ?ob qb:dataSet eg:RETAIL) (triple ?ob eg:M_RETAILPRICE ?Price)
(triple ?ob eg:MERCHANDISE ?merchandise)(triple ?ob eg:RETAILER ?retailer)))))

3.3 Analysis Axis: Dimension

A dimension may include a single analytical granularity (e.g., dimensions in a QB dataset) or multiple analytical granularities. If several analytical granularities are defined, we can find one or several aggregation paths (i.e. hierarchies). Two hierarchies from different sources do not always share a common lowest analytical granularities. Therefore, we remove the constraint of unique root level in the following definition.

Definition 3. A *dimension* corresponds to a one-dimensional space regrouping the analytical granularities related to an analysis axis. A dimension is denoted as $D_i=\{n^{D_i}; \mathcal{L}^{D_i}; \preccurlyeq^{D_i}\}$, where:

- n^{D_i} is the dimension name;
- $\mathcal{L}^{D_i}=\{l_1;\ldots; l_k\}$ is a set of levels characterizing the dimension, each level models a distinct analytical granularity;
- $\preccurlyeq^{D_i}$ is a reflexive *binary* relation which associates a level l_a ($l_a\in\mathcal{L}^{D_i}$) with its parent level l_b ($l_b\in\mathcal{L}^{D_i}$), such as $l_a\preccurlyeq^{D_i}l_b$. The reflexivity means each level can be seen as a parent level of itself, i.e., $\forall l_c\in\mathcal{L}^{D_i}$, $l_c\preccurlyeq^{D_i}l_c$.

Example. We identify a dimension named *Participant* which groups all analytical granularities related to the participants of the sales activities (i.e., *salesmen* of the organization and their rival *retailers*). The dimension $D_{\text{Participant}}$ includes three levels, such as $\mathcal{L}^{\text{Participant}}=\{l_{\text{Shop}}; l_{\text{Team}}; l_{\text{Country}}\}$. The *binary* relation $\preccurlyeq^{\text{Participant}}$ reveals the aggregation paths (i.e., hierarchies) such as $l_{\text{Shop}}\preccurlyeq^{\text{Participant}}l_{\text{Team}}\preccurlyeq^{\text{Participant}}l_{\text{Counrty}}$.

Our definition of dimension is generic enough to model a non-hierarchical dimension as well. A non-hierarchical dimension (e.g. D_{QB}) is defined with only one level (e.g., $\mathcal{L}^{\text{QB}}=\{l_1\}$) including all the attributes of the dimension.

Without the constraint of unique root level (i.e., $\exists_{=1}l_p\in\mathcal{L}^{D_i}$, $\forall l_q\in\mathcal{L}^{D_i}$: $l_p\preccurlyeq^{D_i}l_q$[4]), a dimension may start at any level. This is an important property of a dimension re-

[4] $\exists_{=1}$represents the *unique existential quantification* meaning "there exists only one

grouping levels from multiple sources, since the measures from one source may only be analyzed according to a subset of levels coming from the same source. We define a *sub-dimension* as a part of dimension along which a measure can be summarized.

Definition 4. A *sub-dimension* of $\mathrm{D_i}$, denoted $\mathrm{D}_{\mathrm{i}\backslash l_{\mathrm{s}}}=\{\mathrm{n}^{\mathrm{D}_{\mathrm{i}\backslash l_\mathrm{s}}}; \mathcal{L}^{\mathrm{D}_{\mathrm{i}\backslash l_\mathrm{s}}}; \preccurlyeq^{\mathrm{D_i}}\}$, corresponds to the part of the dimension $\mathrm{D_i}$ starting with the level l_s, where

- $\mathrm{n}^{\mathrm{D}_{\mathrm{i}\backslash l_\mathrm{s}}}$ is the name of the sub-dimension;
- $\mathcal{L}^{\mathrm{D_i}}{}_{\backslash l_\mathrm{s}}$ is the subset of levels, $\mathcal{L}^{\mathrm{D_i}}{}_{\backslash l_\mathrm{s}} \subseteq \mathcal{L}^{\mathrm{D_i}}$, $\forall l_\mathrm{i} \in \mathcal{L}^{\mathrm{D_i}}{}_{\backslash l_\mathrm{s}}$, $l_\mathrm{s} \preccurlyeq^{\mathrm{D_i}} l_\mathrm{i}$;
- $\preccurlyeq^{\mathrm{D_i}}$ is the same binary relation of the one on the dimension $\mathrm{D_i}$.

Example. A sub-dimension of the dimension named *Time* may be $\mathrm{D}_{\mathrm{Time}\backslash l_{\mathrm{Month}}}$ named *Time-Month* with $\mathcal{L}_{\mathrm{Time}\backslash l_{\mathrm{Month}}}=\{l_{\mathrm{Month}}; l_{\mathrm{Year}}\}$, which represent the subpart of the dimension $\mathrm{D_{Time}}$ that measure from the LOD1 dataset can be calculated along.

3.4 Analytical Granularity: Level

A level includes a set of attributes describing a distinct analytical granularity. We present the definition of a level which (a) indicates the source of each attribute, (b) manages heterogeneous representations of attribute instances referring to the same concepts and (c) implements the *binary* relation at the attribute instance level.

Definition 5. A *level* represents a distinct analytical granularity on a dimension. A level is denoted as $l_\mathrm{d}= \{\mathrm{n}^{l_\mathrm{d}}; \mathcal{A}^{l_\mathrm{d}}; \mathcal{C}^{l_\mathrm{d}}; \mathcal{R}^{l_\mathrm{d}}\}$, where:

- $\mathrm{n}^{l_\mathrm{d}}$ is the name of level;
- $\mathcal{A}^{l_\mathrm{d}} =\{\mathrm{a_1};\ldots; \mathrm{a_e}\}$ is a finite set of *attribute*s. Each attribute $\mathrm{a_x}$ ($\mathrm{a_x} \in \mathcal{A}^{l_\mathrm{d}}$) is a pair $\langle \mathrm{n}^{\mathrm{a_x}}, E^{\mathrm{a_x}}\rangle$, where $\mathrm{n}^{\mathrm{a_x}}$ is the name of the attribute and $E^{\mathrm{a_x}}$ is an *extraction formula* indicating the instances of $\mathrm{a_x}$. The domain of an attribute is denoted as $dom(\mathrm{a_x})$.
- $\mathcal{C}^{l_\mathrm{d}}$: $dom(\mathrm{a_x}) \rightarrow 2^{dom(\mathrm{a_y})}$ ($\mathrm{a_x} \in \mathcal{A}^{l_\mathrm{d}}$, $\mathrm{a_y} \in \mathcal{A}^{l_\mathrm{d}} \backslash \mathrm{a_x}$) is a symmetric *correlative* mapping which associates an attribute $\mathrm{a_x}$ with its related ones at the same level.
- $\mathcal{R}^{l_\mathrm{d}}$: $2^{dom(\mathrm{a_y})} \rightarrow dom(\mathrm{a_z})$ ($\mathrm{a_x} \in \mathcal{A}^{l_\mathrm{d}}$, $\mathrm{a_z} \in \mathcal{A}^{l_\mathrm{e}}$ and $l_\mathrm{d} \preccurlyeq l_\mathrm{e}$.) is a *rollup* mapping implementing the *binary* relation between two levels. It connects the instances of child attributes with the instances of a parent attribute at an adjacent level.

Example. The level l_{Team} on the dimension $\mathrm{D_{Participant}}$ contains a finite set of attributes $\mathcal{A}^{l\mathrm{Team}}=\{\mathrm{a_{Team}}; \mathrm{a_{CatchmentArea}}\}$ from the ROLAP DW and the LOD2 dataset. To indicate attribute instances in data sources, two *extraction formulae* are defined within the level l_{Team}: $E^{\mathrm{a_{Team}}}=\pi_{\mathrm{Team}}$(SalesAnalysis.Salesman) is associated to the attribute $\mathrm{a_{Team}}$, while the attribute $\mathrm{a_{CatchmentArea}}$ is connected with an *extraction formula*:

$E^{\mathrm{a_{CatchmentArea}}}$=
```
(distinct (project (?CAName)
        (bgp (triple ?ob qb:dataSet eg:RETAIL) (triple ?ob eg:RETAILER ?retailer)
            (triple ?retailer skos:broader ?CA) (triple ?CA qb4o:inLevel eg:CATCHMENTAREA)
            (triple ?CA rdfs:label ?CAName)))
```

The *correlative* mapping $\mathcal{C}^{l\mathrm{Team}}$ associates the instances of the attribute $\mathrm{a_{Team}}$ to its related instances of the attribute $\mathrm{a_{CatchmentArea}}$, e.g., $\mathcal{C}^{l\mathrm{Team}}$: {{T1}→{CA1}; {T2}→{CA2}; {T3}→ {CA3}}. The *rollup* mapping $\mathcal{R}^{l\mathrm{Team}}$ aggregates the instances of $\mathrm{a_{Team}}$ and $\mathrm{a_{CatchmentArea}}$ at the level l_{Team} to the ones of $\mathrm{a_{Country}}$ at the level l_{Country}, such as:$\mathcal{R}^{l\mathrm{Team}}$: {{{T1; CA1}; {T2; CA2}}→{Finland}; {T3; CA3} →{Spain}}.

4 Implementation of *Unified Cube*s

In this section, we present an implementation framework for *Unified Cube*s. By building a *Unified Cube* based on the ROLAP DW and the two LOD datasets of the motivating example, we show the feasibility of our proposal.

4.1 Architecture

The implementation framework aims at enabling unified analyses of data from DWs and multidimensional LOD sources. Two modules are defined within the framework, namely *Schema* and *Instance* (cf. figure 4). The first module named *Schema* aims at revealing the internal structure of data from multiple sources. It includes a non-materialized *Unified Cube* schema with a set of extraction formulae allowing querying sources on-the-fly. The second module, named *Instance*, is devoted to managing related attribute instances scattered in different sources. It contains (a) a toolkit identifying the *correlative* and *rollup* relations between attribute instances and (b) a set of tables of correspondences materializing the identified relations.

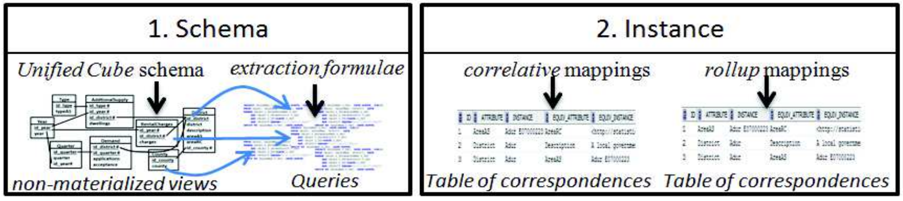


Fig. 4. Implementation framework for *Unified Cube*s

4.2 *Schema* Module

The implementation of a conceptual *Unified Cube* schema can take several forms. Due to the wide use of relational databases in current information management, in this paper we focus on one implementation alternative based on relational views.

Within our framework, the *Schema* module hosts (a) a set of non-materialized views implementing the components of a schema of *Unified Cube* and (b) a set of queries implementing the *extraction formulae* of attribute and measure. Specifically, each dimension is transformed into a set of views. Each view represents a level with a synthetic primary key and the set of attributes of the level. The *extraction formula* of each attribute is translated into an executable query to indicate how attribute instances can be accessed from sources. For a pair of views implementing two levels associated together through a *binary* relation, the view of the lower level includes a foreign key pointing to the view of the higher level. The fact is also implemented with a set of views: each one regroups a set of measures sharing the same sub-dimensions. A set of foreign keys pointing to the starting levels of the related sub-dimensions is included in each view of fact. We propose the following algorithm to automate the implementation of *Unified Cube* schema.

```
Algorithm Unified Cube Schema Implementation
Input: A Unified Cube={F; $\mathcal{D}$; $\mathcal{LM}$}.
Output: A set of non-materialized views implementing the Unified Cube schema
Begin
For each dimension $D_i \in \mathcal{D}$
    For each level $l_d \in \mathcal{L}^{D_i}$
        Create a view $V_d$ named $n^{l_d}$ with a key $id^{V_d}$;
        For each attribute $a_x \in \mathcal{A}^{l_d}$
            Add an attribute named $n^{a_x}$ in the view $V_d$, associate the attribute with a
            query $Q_{a_x}$ obtained by translating the query algebra of the formula $map^{a_x}$;
        End for
    End for
    For each pair of views $V_d$ and $V_e$ of the levels $l_d$ and $l_e$ ($l_d$, $l_e \in \mathcal{L}^{D_i} \wedge l_d \preccurlyeq^{D_i} l_e$)
        Add a foreign key $id^{V_e}$ in the view $V_d$ pointing to the view $V_e$;
    End for
End for
For each subset of measures $\mathcal{M}_{sub} \subseteq \mathcal{M}^F$ sharing the same related levels, such as
$\forall m_e \in \mathcal{M}_{sub}$, $\exists \mathcal{L}^r{}_{\backslash l_h} \times \ldots \times \mathcal{L}^t{}_{\backslash l_k} \subseteq \mathcal{L}^1 \times \ldots \times \mathcal{L}^n$, $\mathcal{LM}$: $\mathcal{L}^r{}_{\backslash l_h} \times \ldots \times \mathcal{L}^t{}_{\backslash l_k} \rightarrow m_e$ ($r,t \in [1..n] \wedge r \leq t$)
    Create a view $V_{sub}$;
    Add a measure named $n^{m_e}$ ($m_e \in \mathcal{M}_{sub}$) in $V_{sub}$, associate the measure with a
    query $Q_{m_e}$ obtained by translating the query algebra of the formula $map^{m_e}$;
    For each set of levels $\mathcal{L}^s{}_{\backslash l_j}$ ($\mathcal{L}^s{}_{\backslash l_j} \subseteq \mathcal{L}^r{}_{\backslash l_h} \times \ldots \times \mathcal{L}^t{}_{\backslash l_k}$) of the subdimension $D^s{}_{\backslash l_j}$
        Add a foreign key $id^{V_j}$ in $V_{sub}$ pointing to the view $V_j$ of the level $l_j$;
    End for
End for
End
```

4.3 *Instance* Module

Related data are scattered in multiple sources and represented according to different modeling vocabularies. The framework must provide methods identifying related data from DWs and/or LOD datasets. Once identified, related warehoused data and LOD should be kept in a generic, coherent environment to avoid repetitive relevance processing during analyses. To this end, the *Instance* module (a) pre-processes related attribute instances involved in *correlative* and *rollup* mappings before analyses and (b) materializes related data in tables of correspondences for future uses. At the beginning of an analysis process, the framework verifies the last changed date of each source to determine if materialized data in the *Instance* module should be updated.

Step I: identifying the relevance between data

In the context of *Unified Cube*s, the relevance between data from multiple sources can be divided into two types, namely *direct* relevance and *deductive* relevance. *Direct* relevance exists between two attribute instances which are already associated together in sources (e.g., the *correlative* mapping between the instances of a_{P_Key} and those of a_{P_Name} from ROLAP). *Deductive* relevance, on the other hand, is identified by using some processing methods. The *Instance* module contains a toolkit implementing some most effective methods to facilitate the identification of related data involved a *Unified Cube*. We describe three categories of processing methods that can be potentially included in the *Instance* module to identify related attribute instances.

Automatic processing methods for correlative mappings.

We identify two methods allowing automatically computing the *deductive* relevance between attribute instances involved in a *correlative* mapping. The first approach is applicable for data from different LOD datasets. It is based on an intermediate ontology with a comprehensive coverage of the common concepts in two LOD datasets (i.e., containing enough matches between equivalent entities). The second approach aims at identifying relevant instances sharing similar labels (e.g., a product's *type* from the ROLAP DW and a merchandise's class from the LOD2 dataset). This approach consists of calculating the *string-based* similarity $\overline{\sigma}$ of two related attribute instances s_1 and s_2, such as $\overline{\sigma}(s_1, s_2) \in [0..1]$. Several amelioration techniques can help improving the obtained similarity, such as case normalization (e.g., converting s_1 and s_2 to lowercase) and synonym matching (e.g., using an external thesaurus to associate "heater" with its synonym "warmer").

Automatic processing methods for rollup mappings.

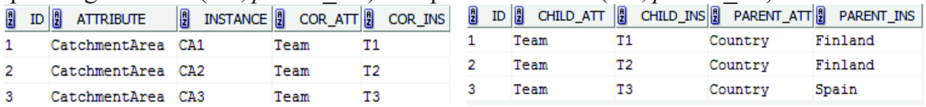
The DW domain mainly focuses on the multidimensional structure of data (i.e. schema), the *rollup* mappings between attributes from DWs do not need additional processing methods, since they can be directly derived by referring to the multidimensional schema of data source. In the domain of LOD, a dataset, especially real-world QB datasets, often only includes independent data instances without an explicitly defined schema. Discovering the *rollup* mapping in previously unknown LOD datasets is not a trivial task. The existence of various proposals in the scientific literature, such as some computer-assisted approaches presented in [5], shows there is no *one-size-fits-all* method for identifying *child-parent* relations between all types of LOD. The implementation framework should only include methods applicable to the hosted *Unified Cube*. With regard to the running example, we implement an automatic reasoning method based on existing *correlative* and/or direct *rollup* mappings. This approach is particularly useful to deduce the *rollup* mapping between an attribute from DWs and another one from LOD dataset, such as: let a_{iDW}, $a_{iLOD} \in \mathcal{A}^{l_i}$, a_{jDW}, $a_{jLOD} \in \mathcal{A}^{l_j}$ $(l_i \preccurlyeq l_j)$: $(\mathcal{C}^{l_i}: dom(a_{iDW}) \rightarrow dom(a_{iLOD})) \wedge (\mathcal{R}^{l_i}: dom(a_{iLOD}) \rightarrow dom(a_{jLOD})) \Rightarrow \exists \mathcal{R}^{l_i}: dom(a_{iDW}) \rightarrow dom(a_{jLOD})$, $(\mathcal{R}^{l_i}: dom(a_{iLOD}) \rightarrow dom(a_{jLOD})) \wedge (\mathcal{C}^{l_j}: dom(a_{jLOD}) \rightarrow dom(a_{jDW})) \Rightarrow \exists \mathcal{R}^{l_i}: dom(a_{iLOD}) \rightarrow dom(a_{jDW})$.

Semi-automatic processing methods for correlative mappings and rollup mappings.

Besides the automatic approaches, some semi-automatic approaches should also be adapted, especially for the relevance between attribute instances which holds only in a specific analysis context (e.g., the *correlative* mapping between a_{Team} and $a_{CatchmentArea}$ in figure 1(d) is valid only if a *catchment area* of retailer attracts the same clientele of a salesman's *team*). In this case, decision-makers should explicitly describe the correspondences between relevant attributes instances. Then the system checks the local and overall validity of *correlative* and *rollup* mappings in a *Unified Cube*. Due to limited space, more details can be found in our previous work [10].

Step II: materializing relevant data.

Direct relevance between attributes is already embedded in the sources thus does not need to be materialized in the *Instance* module. *Deductive* relevance between data, on the other hand, is identified after applying appropriate processing methods. To avoid repetitive relevance processing during analyses, *correlative* mappings and *rollup* mappings involving *deductive* relevance are materialized in tables of correspondences (cf. figure 5): the table of correspondences implementing the *correlative* mappings associates the instances (i.e., *instance*) of an attribute (i.e., *attribute*) to the related ones (i.e., *cor_ins*) of a correlative attribute (i.e., *cor_att*) within the same level, while the table of correspondences materializing the *rollup* mappings connects a set of instances (i.e., *child_ins*) of a child attribute (i.e., *child_att*) with a corresponding instance (i.e., *parent_ins*) of a parent attribute (i.e., *parent_att*).



ID	ATTRIBUTE	INSTANCE	COR_ATT	COR_INS
1	CatchmentArea	CA1	Team	T1
2	CatchmentArea	CA2	Team	T2
3	CatchmentArea	CA3	Team	T3

(a) Materialized *correlative* mappings

ID	CHILD_ATT	CHILD_INS	PARENT_ATT	PARENT_INS
1	Team	T1	Country	Finland
2	Team	T2	Country	Finland
3	Team	T3	Country	Spain

(b) Materialized *rollup* mappings

Fig. 5. Extract of tables of correspondences

4.4 Experimental Assessments

The experimental assessments aim at showing the feasibility of our proposal. Specifically, the internal structure of data from DWs and LOD datasets should be correctly managed by the *Schema* module, while related data from different sources must be identified and materialized within the *Instance* module within a reasonable time.

Protocol.

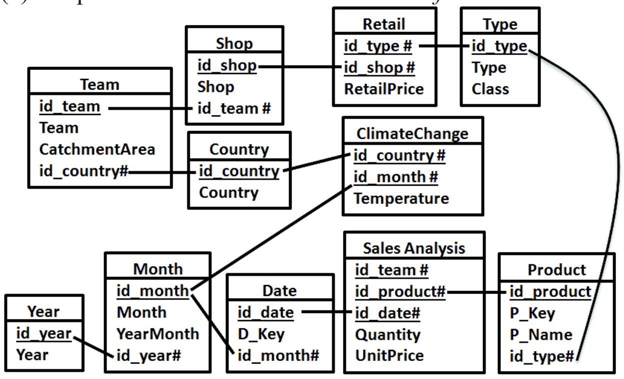
The data sources are hosted in a Microsoft Windows 7 work stations (Interl(R) i7-4510U 2GHz CPU, 8GB RAM, SSD 500GB disk). Each source is populated with a reasonable amount of synthetic data to avoid timeout during the experimental assessments: the ROLAP contains about 18 million pre-aggregated data in the fact table, while the LOD1 and LOD2 datasets respectively include 7240 and 840 observation (cf. table 2).

Table 2. Data Collection

Source	Dimensions
ROLAP	40 teams × 40 products (20 types) × 7300 days (360 months, 20 years)
LOD1	20 countries ×360 months
LOD2	40 shops (20 catchment area) ×20 merchandise classes

***Schema* module.**

We firstly implement the schema of the *Unified Cube* and the *extraction formulae* in the *Schema* module. After applying the algorithm described in section 4.2, we obtain (a) 8 views implementing the dimensions, (b) 3 views implementing the fact and (c) 16 queries translated from *extraction formulae*.



(a) non-materialized views

extraction formula of $a_{CatchmentArea}$

```
SELECT ?retailer ?merchandise
        (avg(?Price) as ?avgPrice)
WHERE {
  ?ob qb:dataSet eg:RETAIL.
  ?ob eg:M_RETAILPRICE ?Price.
  ?ob eg:MERCHANDISE ?merchandise.
  ?ob eg:RETAILER ?retailer.
}
```

extraction formula of $m_{RetailPrice}$

```
SELECT DISTINCT ?CAName
WHERE {
?ob qb:dataSet eg:RETAIL.
?ob eg:RETAILER ?retailer.
?retailer skos:broader ?CA.
?CA qb4o:inLevel eg:l_CATCHMENTAREA.
?CA  rdfs:label ?CAName
}
```

(b) examples of query translated from *extraction formulae*

Fig. 6. Content of the *Schema* module after implementation

The non-materialized *Unified Cube* schema along with extraction queries enables on-the-fly analyses of warehoused data and LOD to be carried out in a unified way. Details and examples of such analyses are presented in our previous work [10].

***Instance* module.**

Identifying relevant data.

As shown in table 3, four *correlative* mappings are identified. The *correlative* mapping between a_{P_Key} and a_{P_Name} are obtained by directly referring to the ROLAP DW, while the other *correlative* mappings require additional processing methods.

Table 3. List of *correlative* mappings

Type	Mapping	Processing method
direct	$dom(a_{P\_Key})\rightarrow dom(a_{P\_Name})$	n/a
deductive	$dom(a_{Type})\rightarrow dom(a_{Class})$	Automatic: *string-based* similarity
	$dom(a_{Type})\rightarrow dom(a_{CatchmentArea})$	Semi-automatic: declarative operator
	$dom(a_{Month})\rightarrow dom(a_{YearMont})$	Automatic: *string-based* similarity

Table 4 shows the *rollup* mappings in the implemented *Unified Cube*. To identify relevant data involved in *rollup* mappings, we firstly search for *child-parent* relations embedded in data sources. Four *rollup* mappings are found by directly referring to the ROLAP DW and the LOD2 dataset which both have a well-defined multidimensional schema. The other five *rollup* mappings involving *deductive* relevance are obtained by (a) executing the declarative operator which associates relevant data together according to users' needs and (b) using reasoning techniques based on existing *rollup* mappings and *correlative* mappings.

Remark. Reasoning is an important means to identify *rollup* mappings in a *Unified Cube*, especially for two attributes from different sources without intermediate ontological resource. For instance, the *rollup* mapping $\mathcal{R}^{d\,\mathrm{Shop}}$: $dom(\mathrm{a}_{\mathrm{Shop}}) \rightarrow dom(\mathrm{a}_{\mathrm{Team}})$ is obtained by referring to the *rollup* mapping $\mathcal{R}^{d\,\mathrm{Shop}}$: $dom(\mathrm{a}_{\mathrm{Shop}}) \rightarrow dom(\mathrm{a}_{\mathrm{CatchmentArea}})$ and the *correlative* mapping $\mathcal{C}^{d\,\mathrm{Team}}$: $dom(\mathrm{a}_{\mathrm{Team}}) \rightarrow dom(\mathrm{a}_{\mathrm{CatchmentArea}})$.

Table 4. List of *rollup* mappings

Type	Mapping	Processing methods
direct	$dom(\mathrm{a}_{\mathrm{P\ Key}}) \rightarrow dom(\mathrm{a}_{\mathrm{Type}})$	n/a
	$dom(\mathrm{a}_{\mathrm{D\ Key}}) \rightarrow dom(\mathrm{a}_{\mathrm{Month}})$	n/a
	$dom(\mathrm{a}_{\mathrm{Month}}) \rightarrow dom(\mathrm{a}_{\mathrm{Year}})$	n/a
	$dom(\mathrm{a}_{\mathrm{Shop}}) \rightarrow dom(\mathrm{a}_{\mathrm{CatchmentArea}})$	n/a
deductive	$dom(\mathrm{a}_{\mathrm{P\ Key}}) \rightarrow dom(\mathrm{a}_{\mathrm{Class}})$	Automatic: reasoning
	$dom(\mathrm{a}_{\mathrm{Shop}}) \rightarrow dom(\mathrm{a}_{\mathrm{Team}})$	Automatic: reasoning
	$dom(\mathrm{a}_{\mathrm{CatchmentArea}}) \rightarrow dom(\mathrm{a}_{\mathrm{Country}})$	Automatic: reasoning
	$dom(\mathrm{a}_{\mathrm{YearMonth}}) \rightarrow dom(\mathrm{a}_{\mathrm{Year}})$	Automatic: reasoning
	$dom(\mathrm{a}_{\mathrm{Team}}) \rightarrow dom(\mathrm{a}_{\mathrm{Country}})$	Semi-automatic: declarative operator

Without any optimization technique (e.g., parallel computing), defining all *deductive* mappings takes about 200 seconds. The execution time remains reasonable in consideration of the laptop-level configuration of the working station.

Materializing relevant data.

Deductive *correlative* mappings and *rollup* mappings are materialized in the *Instance* module. After the implementation, we obtain two tables of correspondences containing 400 tuples and 480 tuples for *correlative* mappings and *rollup* mappings respectively. Comparing to the large volume of data in the sources (cf. table 2), a *Unified Cube* only materializes a relatively small amount of data from different sources. Advantage of the partial materialization is twofold: it (a) avoids repetitive computing of *deductive* relevance during analyses and (b) minimizes the cost of updating the materialized data at the beginning of an analysis process.

5 Conclusion

Our aim is to make full use of all relevant data to support effective and well-informed decisions. To this end, we define a generic conceptual multidimensional model, named *Unified Cube*, which includes data coming from DW and LOD domains.

A *Unified Cube* organizes data coming from different sources according to analysis axes (i.e., *dimension*s) and an analysis subject (i.e., *fact*). A dimension is composed of levels which are associated together through a reflexive *binary* relation. The definition of dimension is generic enough to model several aggregation paths (i.e., *hierarchies*) sharing no common lowest level as well as a non-hierarchical dimension composed of only one level. A level groups a set of attributes from multiple sources. Each attribute is associated with an extraction formula, so that attribute instances that can be directly obtained from data sources are not materialized in a *Unified Cube*. *Correlative* mappings are defined between related attributes, while *rollup* mappings manage *child-parent* relations among attributes. A fact represents the analysis subject containing numeric indicators (i.e., *measure*s). Through a *level-measure* mapping, a measure can be associated with its related dimensions starting from any level. We also propose an implementation framework compatible with *Unified Cube*s. By describing how a *Unified Cube* is built from the DW and the LOD datasets of the motivating example, we show the feasibility of our proposal.

In the future, we intend to build *Unified Cube*s from other LOD formats (e.g. RDF) that do not necessarily have a multidimensional structure. The scalability of the proposed implementation framework and the precision of the obtained mapping will also be studied with real-world data. A more long-term objective is to study the influences of the materialization of source data over analysis efficiency in *Unified Cube*s.

6 Reference

1. Abelló A, Darmont J, Etcheverry L, Golfarelli M, Mazón J-N, Naumann F, Pedersen T, Rizzi SB, Trujillo J, Vassiliadis P, Vossen G (2013) Fusion Cubes: Towards Self-Service Business Intelligence. Int J Data Warehous Min 9:66–88.
2. Abelló A, Romero O, Pedersen TB, Berlanga R, Nebot V, Aramburu MJ, Simitsis A (2015) Using Semantic Web Technologies for Exploratory OLAP: A Survey. IEEE Trans Knowl Data Eng 27:571–588.
3. Deb Nath RP, Hose K, Pedersen TB (2015) Towards a Programmable Semantic Extract-Transform-Load Framework for Semantic Data Warehouses. ACM Press, pp 15–24
4. Etcheverry L, Vaisman A, Zimányi E (2014) Modeling and Querying Data Warehouses on the Semantic Web Using QB4OLAP. In: Data Warehous. Knowl. Discov. Springer International Publishing, Cham, pp 45–56
5. Ibragimov D, Hose K, Pedersen TB, Zimányi E (2014) Towards Exploratory OLAP over Linked Open Data–A Case Study. HangZhou, pp 1–18
6. Kimball R (1998) The data warehouse lifecycle toolkit: expert methods for designing, developing, and deploying data warehouses. Wiley, New York
7. Laborie S, Ravat F, Song J, Teste O (2015) Combining Business Intelligence with Semantic Web: Overview and Challenges. Inform. Organ. Syst. Inf. Decis. INFORSID 2015
8. Matei A, Chao K-M, Godwin N (2015) OLAP for Multidimensional Semantic Web Databases. In: Enabling Real-Time Bus. Intell. Springer Berlin Heidelberg, pp 81–96
9. Nebot V, Berlanga R, Pérez JM, Aramburu MJ, Pedersen TB (2009) Multidimensional Integrated Ontologies: A Framework for Designing Semantic Data Warehouses. In: J. Data Semant. XIII. Springer Berlin Heidelberg, pp 1–36
10. Ravat F, Song J, Teste O (2016) Designing Multidimensional Cubes from Warehoused Data and Linked Open Data. In: IEEE Int. Conf. Res. Chall. Inf. Sci. Grenoble, France, pp 171–182
11. Romero O, Abelló A (2007) Automating multidimensional design from ontologies. In: Int. Workshop Data Warehous. OLAP. ACM Press, pp 1–8
12. Zorrilla ME, Mazón J-N, Ferrández Ó, Garrigós I, Daniel F, Trujillo J (2012) Business Intelligence Applications and the Web: Models, Systems and Technologies. IGI Global.