OLAP Manipulations on RDF Data following a Constellation Model

Rafik Saad, Olivier Teste, and Cássia Trojahn

IRIT (UMR5505) & Universite Toulouse 2 Le Mirail (UTM2), France
srf.rafik@gmail.com,{olivier.teste,cassia.trojahn}@irit.fr

Abstract. Multidimensional analysis is an alternative way for summarising, aggregating and viewing RDF data on different axes (dimensions) and subjects of analysis (facts). From a RDF data collection conforming to the W3C Data Cube specification, we formalise a multidimensional model in terms of RDF data structures following a conceptual constellation model. This model regroups facts, which are studied according to several dimensions possibly shared between facts, with dimensions relating multi-hierarchies. We show how elementary OLAP operations can be translated into SPARQL queries using an OLAP algebra that is compliant to the constellation model. This algebra is based on a multidimensional table which displays data from one fact and two of its linked dimensions. Initial experiments have been carried out using both synthetic data sets and real data sets.

1 Introduction

The Linked Open Data (LOD) movement has promoting the publication of large interlinked collections of data, represented as RDF graphs. Following this initiative, many organisations currently publish statistical data in RDF format (e.g., Eurostat[1], European Central Bank[2], UK COINS[3], to cite a few examples). The need for exploiting these data for analytical and decision-making purposes becomes rapidly evident. On the one hand, one promising way of analysing these numeric data is by means of OLAP (Online Analytical Processing) analysis [1, 2]. This technique allows for summarising, filtering, aggregating, and viewing data on different axes and subjects of analysis. On the other hand, OLAP treatments require data to be structured following a specific model, i.e., the multidimensional model, which organises data on a set of facts (subjects of analysis), and dimensions and hierarchies (axes of analysis).

A first category of approaches for manipulating RDF data following a multidimensional model considers an ETL process for extracting and transforming these data into a specific structure, usually the star relational model, before using standard OLAP systems [6]. Another category of approaches aims at manipulating OLAP operations directly on RDF data collections without using an

[1] http://eurostat.linked-statistics.org
[2] http://ecb.270a.info/.html
[3] http://data.gov.uk/resources/coins

ETL process [8, 3]. While the first approach requires the ETL process to be repeated for propagating the data evolutions in the sources, the second approach requires a multidimensional modelling of RDF data and a dynamic translation of OLAP operations into SPARQL queries.

In this paper, we present an approach for OLAP manipulations on RDF data which falls into the second category of approaches. To that extent, we consider that RDF data are modelled according to the RDF Data Cube Vocabulary[4], a vocabulary to model multidimensional data, such as statistics, in RDF. First, we propose a formalisation of a multidimensional structure based on RDF format following a constellation model of facts and dimensions composed of multi-hierarchies. This model has been introduced by Ravat and colleagues in [12], extending star schemes [9], which are commonly used in the multidimensional modelling. Second, based on this formalisation, we show how the main OLAP operations (DRILLDOWN, ROLLUP, SELECT and ROTATE) can be translated into SPARQL queries using an OLAP algebra that is compliant to the constellation model. This algebra is based on a multidimensional table which displays data from one fact and two of its linked dimensions. It defines a set of elementary operators from which more complex OLAP operations can be defined. Our proposal has been implemented and experiments were carried out using both synthetic data sets and real data sets. The main contributions of the paper can be outlined as follows :

- we provide an efficient RDF constellation model that is intimately related to the multidimensional data model. This generalised model supports multiple facts, multiple dimensions and multiple hierarchies;
- the proposed modelling supports complex hierarchies to represent several real world data organisations and covers the case of *non-covering* hierarchies [11, 10], where instances can not strictly follow the hierarchical specifications by allowing values of a child level to jump the intermediate levels along the hierarchy. An example of this kind of hierarchy is a company having customers into different cities of several countries. American cities can be regrouped into parent levels such as states whereas French cities jump these intermediate levels.

The remainder of the paper is organised as follows. §2 introduces the conceptual constellation model we proposed for the multidimensional modelling of RDF data. Based on this model, §3 presents how OLAP operations are translated into SPARQL queries. Then, §4 describes the prototype developed to validate our approach. §5 discusses related work and §6 concludes the paper and discusses future work.

2 Constellation Model on RDF Data

The multidimensional manipulation of RDF data requires the definition of a conceptual model from which the OLAP operations will be specified. This section

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

formally defines the elements of a multidimensional model, in terms of RDF data described using RDF Data Cube[5], SKOS[6] and RDFS[7] vocabularies. This formalisation is based on a conceptual model that defines a constellation of facts and dimensions, which are composed of multi-hierarhies. A constellation regroups several facts, which are studied according to several dimensions possibly shared between facts [12]. Before introducing the model, we consider an example of a constellation schema that will be used throughout the remainder of the paper. This schema is composed by the fact *Sales* that has two measures, *quantity* and *amount*, and which can be analysed throughout three dimensions, *Time*, *Product*, and *Geography*. The *Geography* dimension is composed of two hierarchies, H_{Geo}, a three-level hierarchy (composed by the attributes *City*, *Region* and *Country*) and H_{Area}, a two-level hierarchy (*City*, *Area*). Figure 1 depicts the example of a multidimensional schema using graphical notations. Note that H_{Geo} is a non-covering hierarchy because some cities do not belong to regions, whereas each region as well as each city belongs to one country.

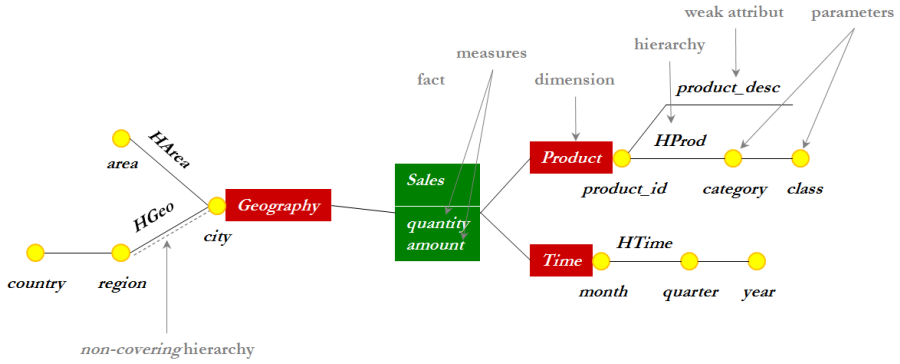

Fig. 1. Example of a multidimensional schema.

A dimension models an analysis axis and is composed of attributes (also called parameters or levels). These attributes are organized according to one or several hierarchies within a dimension :

Definition 1. *A dimension D is defined as 4-tuple (?d,H^D,A^D,I^D), where*

- *?d a qb:DimensionProperty*
- *H^D=$\{?h_1^D,...,?h_v^D\}$ is a set of hierarchies, where ?h a skos:ConceptScheme ∧ ?d qb:codeList ?h*
- *A^D=$\{?a_1^D,...,?a_u^D\}$ is a set of attributes, where ?a a rdfs:Class ∧ ?a rdfs:subClassOf skos:Concept ∧ (∃ ?h ∈ H^D : ?a skos:inScheme ?h)*

[5]http://www.w3.org/TR/2012/WD-vocab-data-cube-20120405/
[6]http://www.w3.org/2004/02/skos/
[7]http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

– I^D=$\{I_1^D,...,I_p^D\}$ *is a set of dimension instances, where* I_j^D=$\{?i_{j_1}^D,...,?i_{j_k}^D\}$ *is a set of attribute instances, where* ?*i a* ?*a* $\wedge$?$a \in A^D \wedge k <= |A^D|$

Each hierarchy relates to a skos:ConceptScheme and attributes of dimensions are modelled as subclasses of *skos:Concept* and instances of *rdfs:Class*. In fact, attributes represent levels of granularity within a dimension. As hierarchies in the RDF Data Cube Vocabulary version we use are defined as SKOS hierarchies, each hierarchical level refers to a *skos:Concept* in the SKOS hierarchy. In order to link the instances of attributes to each hierarchical level, defining them as instances of *rdfs:Class* allows for using the property *rdf:type* to state that instances refer to a specific level of hierarchy[8].

As stated above, hierarchies represent a particular vision (perspective) of a dimension where each attribute represents one data granularity according to which measures could be analysed. Following the constellation model in [12], weak attributes (attributive properties) complete the parameter semantics. As one dimension may have multiple hierarchies of attributes, this means that an attribute can have several direct ancestors, each one belonging to a specific hierarchy :

Definition 2. *A hierarchy H of a dimension D is defined as 3-tuple (*?*h,*P^H*,*$Weak^H$*), where*

– ?*h a skos:ConceptScheme*
– P^H = <?p_1^H,...,?p_v^H > *is an ordered set of attributes (parameters), representing levels of granularity within a dimension, where* ?*p a rdfs:Class* $\wedge$?*p rdfs:subClassOf skos:Concept* $\wedge$?*p skos:inScheme* ?*h* $\wedge$?*h skos:hasTopConcept* ?*p, where* ?*h skos:hasTopConcept* ?x_{top} $\wedge$ $\exists$ *C=*<?x_1...?x_n> : $n < |P_H| \wedge$?x_1 *skos:broader* ?x_2 $\wedge$?x_2 *skos:broader* ?x_3...?x_n $\wedge$?x_n *skos:broader* ?x_{top}
– $Weak^H$: $P^H \rightarrow 2^{A^D - P^H}$ *is a function associating each parameter to one or more weak attributes, where* $Weak^H$*(*?*p)=*{?a_1...?a_n}, *where* ?*a a rdfs:Class* $\wedge$?*a rdfs:subClassOf skos:Concept* $\wedge$?*a skos:related* ?*p*

A fact reflects the information that has to be analysed according to dimensions and that is modelled through one or several indicators (measures):

Definition 3. *A fact F is defined as 3-tuple (*?*ds,*M^F*,*I^F*), where*

– ?*ds a qd:DataSet*
– M^F=$\{f_1(?m_1^F),...,f_w(?m_w^F)$ *is a set of measures associated with an aggregate function f, where* ?*m a db:MeasureProperty*
– I^F=$\{?i_i^F,...,?i_q^F\}$ *is a set of fact instances, where* ?*i is a 3-tuple (*?*o,(*?dv_1, ?dv_2,...?dv_n),(?mv_1,?mv_2,...?mv_m)), where* ?*o a qb:Observation* $\wedge$?*o qb:DataSet* ?*ds :* $\forall i \in [1..n]$, $\exists ?d \in Star(F)$:?*o*?*d*?dv_i $\wedge$ $\forall i \in [1..m]$, $\exists ?m \in M^F$:?*o*?*m*?mv_j

[8] An alternative to associating instances to SKOS concepts could consider XKOS [4], a recently proposed extension of SKOS that takes into account the representation of levels in concept schemes, semantic relations like *isPartOf* and instantiation of concepts, between other features. This will be further investigated in future work.

A constellation regroups several facts, which are studied according to several dimensions, possibly shared between facts :

Definition 4. *A Constellation Cs is defined as a 3-tuple (F^{Cs},D^{Cs},$Star^{Cs}$), where*

- F^{Cs}=$\{?f_1,...,?f_m\}$ *is a set of facts, where ?f a qb:DataSet*
- D^{Cs}=$\{?d_1,...,?d_n\}$ *is a set of dimensions, where ?d a qb:DimensionProperty*
- $Star^{Cs} : F^{Cs} \rightarrow 2^{DCs}$ *associates each fact to its linked dimensions, where ?f qb:Structure ?cube* $\wedge$ *?cube a qb:DataStructureDefinition* $\wedge$ *?cube qb:component ?compD* $\wedge$ *?compD qb:dimension ?d* $\wedge$ *(*$\forall ?m \in M^f$, $\exists ?compM$ *: ?cube qb:component ?compM* $\wedge$ *?compM qb:measure ?m).*

3 Translating OLAP Operations into SPARQL

Based on the constellation model presented above, we define a SPARQL query mechanism for performing OLAP operations directly on the RDF data. This mechanism is based on the OLAP algebra defined in [12]. This algebra is a procedural query language that provides a set of elementary operators from which more complex operations can be specified. It is based on a *multidimensional table* (MT) which displays data from one fact and two of its linked dimensions

Definition 5 (Multidimensional table (MT) [12]). *A MT is as 4-tuple (S,L,C,R) where S=(F^S,M^S) represents the analysed subjects through a fact* $F^S \in F^{Cs}$ *and a set of projected measures* M^S*, L=(DL,HL,PL) represents the horizontal analysis axis where PL=*$<p^{HL}_{max},...p^{HL}_{min}>$*, HL*$\in H^{DL}$ *and DL*$\in Star^{Cs}(F^S)$*, HL is the current hierarchy of DL, C=(DC,HC,PC) represents the vertical analysis axis where PC=*$<p^{HC}_{max},...p^{HC}_{min}>$*, HC*$\in H^{DC}$ *and DC*$\in Star^{Cs}(F^S)$*, HC is the current hierarchy of DC, R=*$pred_1 \wedge ... \wedge pred_t$ *is a normalised conjunction of predicates (restrictions of dimension data and fact data).*

The algebraic operators take as input a source MT, noted MT_{SRC}=(S_{SRC}, L_{SRC},C_{SRC},R_{SRC}), and produces an output MT, noted MT_{RES}=(S_{RES}, L_{RES}, C_{RES},R_{RES}). Each MT_{RES} can further be manipulated using operators of the same algebra. In the scope of this paper, we focus on the minimal core of operators, namely DISPLAY (for defining a first MT), DRILLDOWN and ROLLUP (for moving the analysis details along a hierarchy), SELECT (for selecting data of a multidimensional schema), and ROTATE (for replacing an analysis axis by another one). We assume querying a single data set. Each OLAP operation has an input MT_{SRC} and an output MT_{RES}. For the formal definition of each OLAP operator the reader can refer to [12]. Here, we define how each operation has been defined in terms of SPARQL queries. The aggregations and optimisations are out of the scope of this paper.

As Figure 2 depicts, an initial multidimensional table MT is built from a constellation Cs, using the operator DISPLAY, where DISPLAY (F^S,M^S, DL,HL,DC,HC)=MT_{RES}, with MT_{RES}=(S_{RES},L_{RES},C_{RES},R_{RES}). This operation displays the root parameters of each hierarchy (where all observations

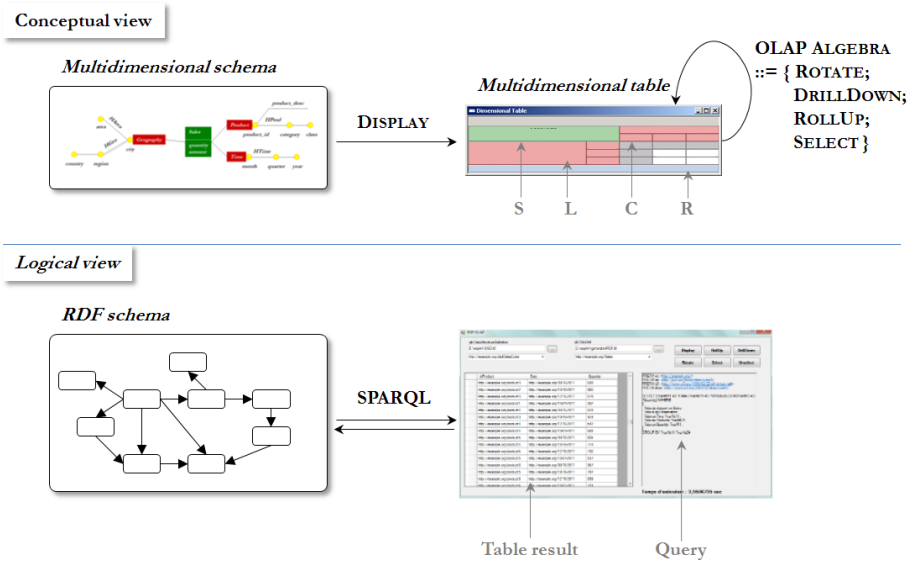

Fig. 2. Principle of translating OLAP operations into SPARQL queries

refer to the lowest level of the dimension hierarchy, i.e., root level or level 1). The process of generating a SPARQL query (Table 1) corresponding to DISPLAY considers the following set of nested operations :

1. Identify the instances of the fact F^S to be displayed;
2. Retrieve the values of the root parameters $PL1$ (attribute in the horizontal analysis axis) and $PC1$ (attribute in the vertical analysis axis) of the dimensions DL and DC, respectively;
3. Retrieve the value mv_i of each measure m_i;
4. Group the measure values mv_i by PL_1 and PC_1;
5. Calculate the aggregations by applying on the measure values mv_i the corresponding aggregation functions Agg_i.

SELECT ?PL_1 ?PC_1 ($Agg_i(mv_i)$ AS ?mes_i)	SELECT ?prodId ?city (SUM(?qty) AS qtySales)
WHERE	WHERE
{	{
?obs rdf:type qb:Observation.	?obs rdf:type qb:Observation.
?obs qb:dataset IRI(F^S).	?obs qb:dataset ex:Sales.
?obs IRI(DL) ?PL_1.	?obs ex:Products ?prodId.
?obs IRI(DC) ?PC_1.	?obs ex:Geography ?city.
?obs IRI(m_i) ?mv_i.	?obs ex:quantity ?qty.
}	}
GROUP BY ?PL_1 ?PC_1	GROUP BY ?prodId ?city

Table 1. DISPLAY SPARQL query and example.

From a MT_{RES} resulting from a DISPLAY operation, the operations ROLLUP and DRILLDOWN modify the analysis precision by manipulating the hierarchical levels of the dimensions. In ROLLUP (MT_{SRC},D,Lvl_{sup})=MT_{RES}, $D \in$

{DL,DC} is the dimension on which the operation is applied, Lvl_{sup} = is a coarser-graduation level used in MT_{RES}, where MT_{RES}=(S_{SRC},L_{RES},C_{RES}, R_{SRC}). Inversely, for DRILLDOWN (MT_{SRC},D,Lvl_{inf})=MT_{RES}, D is the dimension on which the operation is applied, Lvl_{inf} = is a lower attribute in the current hierarchy H of D, and MT_{RES}=(S_{SRC},L_{RES},C_{RES},R_{SRC}). For the SPARQL query generation, the operation consists of positioning the hierarchical level of the current hierarchies under a parameter of level $n \geq 1$ (by means of *skos:broader*). As an example, consider the ROLLUP (MT_{SRC},DL,Lvl_{sup}), supposing that for the dimension in $\text{DC}^{MT_{SRC}}$, the hierarchy $HC^{MT_{SRC}}$ is positioned at the root parameter. The operations to be performed are the following (Table 2) :

1. Identify the instances of the fact $\text{F}^{MT_{SRC}}$ in MT_{SRC} to be displayed;
2. Retrieve the values of the root parameters PL_1 and PC_1 of the dimensions $DL^{MT_{SRC}}$ and $DC^{MT_{SRC}}$;
3. Retrieve the value mv_i of each measure $m_i^{MT_{SRC}}$;
4. From PL_1, drill upward in the hierarchy $HL^{MT_{SRC}}$ until getting up the level Lvl_{sup} (level n) through the predicate *skos:broader* and by ensuring that the new parameter accessed is part of the hierarchy $HL^{MT_{MT_SRC}}$ (via the predicate *skos:inScheme*. This allows for navigating through the good hierarchy, in the case where the dimension has multiple hierarchies. Hence, for getting the parameter of level n requires navigating through $n-1$ parameters;
5. Retrieve the value mv_i of each measure $m_i^{MT_{SRC}}$;
6. Group the measure values mv_i by PC_1 and Lvl_{sup};
7. Calculate the aggregations by applying on the measure values mv_i the corresponding aggregation functions Agg_i;
8. Display the aggregated measure values with the values of PC_1 and Lvl_{sup}.

SELECT ?PC1?$PSup$($\text{Agg}_i(mv_i)$) AS ?mes_i	SELECT ?prodId ?region (SUM(?qty) AS qtySales)
WHERE	WHERE
{	{
?obs rdf:type qb:Observation.	?obs rdf:type qb:Observation.
?obs qb:dataset IRI($\text{F}^{MT_{SRC}}$).	?obs qb:dataset ex:Sales.
?obs IRI($DC^{MT_{SRC}}$) ?PC_1.	?obs ex:Products ?prodId.
?obs IRI($DL^{MT_{SRC}}$) ?PL_1.	?obs ex:Geography ?city.
?PL_1 skos:broader ?PL_2.	?city skos:broader ?region.
?PL_2 skos:inScheme IRI($HL^{MT_{SRC}}$).	?region skos:inScheme ex:HGeo.
...	?region rdf:type ex:region.
?PL_{n-1} skos:broader ?PSup.	
?PSup skos:inScheme $HL^{MT_{SRC}}$.	
?PSup rdf:type IRI(Lvl_{sup}).	
?obs IRI($m_{i_{MT_{SRC}}}$) ?mv_i.	?obs ex:quantity ?qty.
}	}
GROUP BY ?PC_1 ?PSup	GROUP BY ?prodId ?region

Table 2. ROLLUP SPARQL query and example.

Note that for the DRILLDOWN operation, the evaluation principle is similar to the evaluation of ROLLUP. For manipulating the hierarchies within dimensions, it is important to take into account a special case where, at instances level, some hierarchical levels have not any associated instance. It is the case of *non-covering* hierarchies. For example, in the case of the geographical hierarchy H_{Geo},

data at the instance level may contain some cities which are not associated to any region or state (for instance, Vatican City is considered as a city-state which is not associated to a region). Suppose that H_{Geo} is a non-covering hierarchy, and one wants to analyse product sales by countries. Hence, the aggregations have to be done by country regardless of the *real* hierarchical level on which they are positioned, i.e, (a) countries positioned at the third level (case of instances that respect the hierarchy specification in the scheme); (b) countries positioned at the second level (countries which have a state or a city but not both), (c) countries positioned at the first level (countries without states and cities, like Monaco). Hence, we combine graph patterns resulting from the UNION SPARQL operator. An example of query that takes into account non-covering hierarchies, from the previous ROLLUP query (ROLLUP (MT_{SRC},DL,Lvl_{sup}), where Lvl_{sup} is the parameter of level n in the scheme, is presented in Table 3.

SELECT ?PC_1 ?PSup ($Agg_i(mv_i)$) AS ?mes_i	SELECT ?prodId ?country (SUM(?qty) AS ?SalesQty)
WHERE	WHERE
{	{
?obs rdf:type qb:Observation.	?obs rdf:type qb:Observation.
?obs qb:dataset IRI($\text{F}^{MT_{SRC}}$).	?obs qb:dataset ex:Sales.
?obs IRI($m_{i_{MT_{SRC}}}$) ?mv_i.	?obs ex:quantity ?qty.
?PC1 rdf:type IRI(PC_1).	?obs ex:Products ?prodId.
?obs ($DC^{MT_{SRC}}$) ?PC_1.	?country rdf:type ex:Country.
?PSup rdf:type IRI(Lvl_{sup}).	{
{	?obs ex:Geography ?geo1.
?obs ($DL^{MT_{SRC}}$) ?PL_1.	?geo1 skos:broader ?geo2.
?PL_1 skos:broader ?PL_2.	?geo2 skos:inScheme ex:HGeo.
?PL_2 skos:inScheme IRI($HL^{MT_{SRC}}$).	?geo2 skos:broader ?country.
...	}
?PL_{n-1} skos:broader ?PSup.	UNION
}	{
UNION	?obs ex:Geography ?geo1.
{	?geo1 skos:broader ?country.
?obs ($DL^{MT_{SRC}}$?PL_1.	}
?PL1 skos:broader ?PL_2.	UNION
?PL2 skos:inScheme IRI($HL^{MT_{SRC}}$).	{
...	?obs ex:Geography ?country.
?PL_{n-2} skos:broader ?PSup.	}
}	GROUP BY ?prodId ?country
...	
UNION	
{	
?obs ($DL^{MT_{SRC}}$) ?PSup.	
}	
GROUP BY ?PC_1 ?PSup	

Table 3. Example of SPARQL query on non-covering hierarchy.

For changing the analysis criteria, the ROTATE operation allows changing one analysis axis by another or the current hierarchy by another in a same dimension, ROTATE (MT_{SRC},D_{old},D_{new},H_k^{Dnew})=MT_{RES}, where $D_{old} \in$ {DL,DC} is the dimension on which the operation is applied, D_{new} is the dimension replacing D_{old}, H_k^{Dnew} is the current hiearchy of D_{new}, and MT_{RES}=(S_{SRC},L_{RES},C_{RES}, R_{SRC}). D_{old}=D_{new} where only the current hierarchy is to be replaced. The hierarchical level of the modified axis corresponds to the root parameter of this hierarchy. The process of generating the SPARQL query ensuring these oper-

ations is similar to the operation DISPLAY, but accessing the level of detail specified for the unmodified axis. In case of performing hierarchy rotations, the corresponding axis is positioned on the root level of the new hierarchy.

Finally, the SELECT operation (i.e., SLICE and DICE in the common OLAP terminology) removes the data which do not satisfy a condition. A condition can be applied on the dimension attribute values or fact measure values: SELECT (MT_{SRC},pred)=MT_{RES}, where $pred$=$pred_1 \wedge pred_t$ is a selection predicate on facts F^S and/or its linked dimensions $D_i \in Star^{Cs}(F^S)$. The SPARQL query implementing this operation can be obtained by integrating the restriction condition in the query producing the initial multidimensional table (MT_{SRC}). This can be achieved using SPARQL FILTER operator which can apply a logical condition to filter the results of queries.

4 Prototype and experiments

In this section, we present the prototype we have implemented in order to validate the conceptual approach and the experiments carried out using this prototype. The prototype has been implemented using Microsoft .NET Framework and dotNetRDF API[9]. To allow the system to load the multidimensional schema modelling the analysis needs, it is necessary to specify the data set structure definitions (*qb:DataStructureDefinition*) and the hierarchical specification of dimensions according to the model described above. This operation (step 1) is assured by the first principal module, the *Multidimensional Model Loader*. Once the schema loaded, the manipulation of OLAP queries becomes possible (step 2). The second module is the *SPARQL Generator*. It is responsible for generating SPARQL queries from OLAP manipulations specified as input (step 3). Using the dotNetRDF API, the generated SPARQL queries run on the data set (step 4). Query results are then presented to the user (step 5). RDF/XML, Turtle, and N3 RDF syntaxes are supported by the prototype. Figure 3 shows a screenshot of our prototype. The user interface allows to specify the analysis needs and shows the results returned after querying the data sources. The generated SPARQL queries are also given to the user together with their processing time.

We have conducted our experiments on a RDF data set about *Annual producer price of industrial products from CA 1996 Statistical Office of the Republic of Serbia*[10] having with 789 instances of attributes and 156 observations. It contains one temporal and one geographical (about Serbia) dimensions. Hierarchical structure of data is given in associated dictionaries providing a temporal hierarchy and information on regions and municipalities of Serbia. Initially, observation values (measures) are given according to the Year level for the temporal dimension and the country level for the geographical dimension (only one country: Serbia). We have modified the observation values according to different levels of

[9] http://www.dotnetrdf.org/

[10] http://wiki.planet-data.eu/web/Annual_producer_price_indicatores_of_industrial_products_CA_1996_from_Statistical_Office_of_the_Republic_of_Serbia

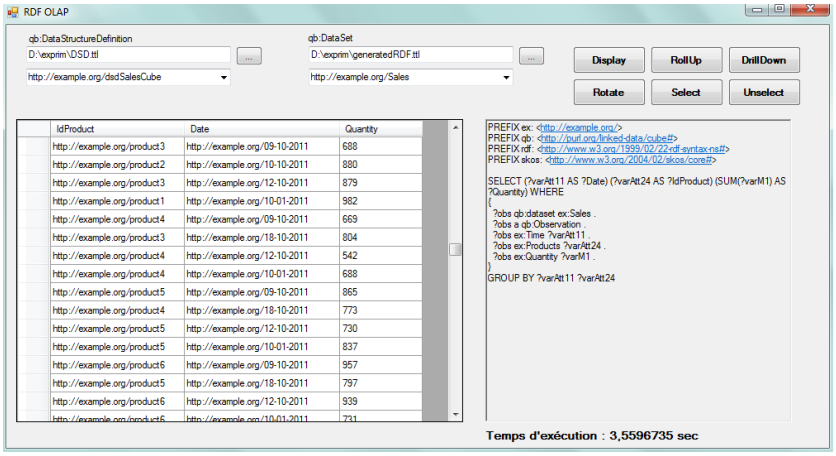

Fig. 3. Screen-shot of our prototype.

the dimensions hierarchies in order to carry out our experiments on manipulating hierarchies and make some non-covering data available. This first data set contains only two dimensions, however, OLAP operations such as ROTATE requires more than two dimensions for performing dimension rotations, and more than one hierarchy per dimension for performing hierarchy rotations. Hence, a second data set has been used. It has 69888 observations and 1191 instances of attributes. This data set have been obtained by converting a synthetic multidimensional database generated in our team to RDF format. The conversion tool is actually limited to this specific database. We are planning to develop a generic version of it to handle with any multidimensional data set.

The evaluation of our approach has been limited to the manipulations of OLAP operations on the two data sets described above. More specifically, we evaluated the adequacy and the correctness of the results from a sequence of applied operations. From a DISPLAY operation, we are able to correctly aggregate data according to a specific level of granularity, rotate dimensions and hierarchies or select parts of a multidimensional table. We do not evaluate runtime for executing the operations, with respect to the number of observations in the data sets. Further evaluation on different data sets have to be carried out.

5 Related work

Traditionally, OLAP analysis operates on data obtained from multiple and heterogeneous sources. In order to organise data coming from these sources in a multidimensional structure, a pipeline of extraction, transformation and loading (ETL) is usually carried out. In [6], an ETL module transforms RDF data (described using RDF Data Cube Vocabulary) into a Multidimensional Model. The resulting structure is further manipulated using Mondrian OLAP system and MDX queries. Hence, the OLAP manipulations are performed on the multidimensional data source and not directly on the RDF data collection. The inconve-

nient of this approach is the ETL process has to repeated in order to propagate changes in the raw data. In order to overcome this drawback, further proposals [14, 3, 8] deal with the direct manipulation of RDF data via SPARQL queries. In [14], an approach for online analysis of RDF triples has been proposed, where a specific storage system has been designed. The aim is to efficiently manage large collections of RDF data. A specific query mechanism extends SPARQL and multidimensional modelling of RDF data has not been considered in this solution. In [3], the authors define an RDF vocabulary called *Open Cubes* (OC) for the multidimensional modelling of RDF data. OC provides a set of classes and properties to model the different structures of the multidimensional model (dimensions, attributes, measures), including hierarchical relationships between attributes of dimensions. From RDF collections described using OC, different OLAP manipulations can be performed directly via queries expressed in SPARQL. Although this solution is based on a multidimensional modelling of RDF data and allows expressing OLAP operations in terms of SPARQL, its main limitation is the use of a specific and non-standard vocabulary. RDF Data Cube Vocabulary, meanwhile, is supported by the W3C, that justifies its use by several collections of published statistical data for subsequent analysis. Tools for supporting the publication of these statistical data have been proposed. This is the case of *OLAP2Cube* and *CSV2Cube* presented in [13], which allow the generation of RDF collections using the vocabulary RDF Data Cube from multidimensional databases implemented on relational data. In [15], a process of identifying data sources for publishing statistical linked data following the RDF Data Cube vocabulary is also presented. Domain ontologies are used to provide a semantic meaning to the data cube.

Comparing the two main categories of approaches presented above, the approaches based on direct manipulation of RDF data ([14, 3, 8]) are more advantageous in terms of flexibility and adaptation to the specificities of published data on the Web. However, the main drawback in [14] and [3] is related to the use of non-standard vocabularies. With regards to the work described in [8], although it is based on RDF Data Cube Vocabulary, it does not take into account the hierarchical structure at multiple levels neither the multiples hierarchies in a dimension. Our approach supports these hierarchical notions. Contrary to [7], we do not implement any kind of aggregation for optimising query execution.

6 Final remarks and future work

This paper has presented a formalisation of a multidimensional model in terms of RDF data described following RDF Data Cube, SKOS and RDFS vocabularies. On the basis of this model, we defined a mechanism for translating common OLAP operations into SPARQL queries. Two important aspects addressed in this paper are the ability of representing multiple hierarchies in a dimension and the ability of handling the cases where the hierarchical structures are not fully covered at the instance level (a common case in real data). We implemented a prototype in order to experiment and validate our proposal. A weak point of

our work is evaluation, which has been mainly based on the correctness of query results, from a sequence of applied OLAP operations. We have several opportunities for future work. First, we plan to study the ability to express, through SPARQL queries, more advanced OLAP manipulations using the operators described in [5]. Second, we intend to focus on performance and optimisation of query execution by means of pre-aggregations. Third, RDF data represent basically resources referenced by links on the Web. A point to study, would be the ability to integrate these interconnections between resources in order to associate automatically more data and extend information available in initial data sources. Finally, XKOS could be further investigated in our formalisation.

References

1. S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *SIGMOD Rec.*, 26(1):65–74, Mar. 1997.
2. E. F. Codd, S. B. Codd, and C. T. Salley. Providing OLAP (On-Line Analytical Processing) to User-Analysis: An IT Mandate, 1993.
3. L. Etcheverry and A. A. Vaisman. Enhancing OLAP Analysis with Web Cubes. In *ESWC*, pages 469–483, 2012.
4. D. Gillman, F. Cotton, and Y. Jaques. eXtended Knowledge Organization System (XKOS). METIS, Work Session on Statistical Metadata, Geneva, May 2013.
5. J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Min. Knowl. Discov.*, 1(1):29–53, 1997.
6. B. Kämpgen and A. Harth. Transforming statistical linked data for use in olap systems. In *I-SEMANTICS*, pages 33–40, 2011.
7. B. Kämpgen and A. Harth. No Size Fits All ? Running the Star Schema Benchmark with SPARQL and RDF Aggregate Views. In *ESWC*, pages 290–304, 2013.
8. B. Kämpgen, S. O'Riain, and A. Harth. Interacting with Statistical Linked Data via OLAP Operations. In *Proceedings of Interacting with Linked Data (ILD 2012), Workshop co-located with the 9th ESWC*, pages 36–49, 2012.
9. R. Kimball. *The data warehouse toolkit: practical techniques for building dimensional data warehouses.* John Wiley & Sons, Inc., New York, NY, USA, 1996.
10. E. Malinowski and E. Zimnyi. OLAP Hierarchies: A Conceptual Perspective. In *Advanced Information Systems Engineering*, pages 477–491. 2004.
11. T. B. Pedersen, C. S. Jensen, and C. E. Dyreson. A foundation for capturing and querying complex multidimensional data. *Inf. Syst.*, 26(5):383–423, July 2001.
12. F. Ravat, O. Teste, R. Tournier, and G. Zurfluh. Algebraic and graphic languages for olap manipulations. *IJDWM*, 4(1):17–46, 2008.
13. P. E. R. Salas, F. M. D. Mota, K. K. Breitman, M. A. Casanova, M. Martin, and S. Auer. Publishing statistical data on the web. *Int. J. Semantic Computing*, 6(4):373–388, 2012.
14. B. Wu, D. Wu, H. Zhong, H. Jin, P. Yuan, and P. Liu. Scalable online analysis of semantic web data. Semantic Web Challenge, 2010.
15. A. Zancanaro, L. D. Pizzol, R. de Moura Speroni, J. L. Todesco, and F. O. Gauthier. Publishing multidimensional statistical linked data. In *Proceedings of the Fifth International Conference on Information, Process, and Knowledge Management*, pages 290–304, 2013.