computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

ANNOTATING DATA WITH MULTIDIMENSIONAL PROPERTIES

Yassine Laadidi, Mohamed Bahaj

Faculty of Sciences and Technologies, University Hassan 1st, Settat, Morocco
yassine.laadidi@gmail.com, mohamedbahaj@gmail.com

Paper history:
Received 17 February 2019
Received in revised form 2 May 2019
Accepted 20 August 2019
Available online 30 September 2019

Keywords:
Data warehousing;
Multidimensional modeling;
Web semantic;
Data Integration;
Business Intelligence.

Abstract: The evolution of web technologies and the data we are manipulating announce profound changes on Business Intelligence (BI) systems and open up important researches and innovations particularly in multidimensional data modeling and data integration. The emergence of the semantic Web highlights the need of including external data sources in the BI system. The semantic web came with Resource Description Framework (RDF) model to describe data over the Web by annotating resources with semantics and properties and consequently establishing reasoning mechanisms. However, integrating and/or analyzing information from Wide World Sources still a very challenging process because of their "unpredictability" and heterogeneity. Consequently, the transition to an open BI/SW system is required to handle automatic alteration on structures and enabling discovery of multidimensional entities over multiple Web sources. In this paper, we introduce our prospective approach and architecture for including external data sources in an open BI/SW system and we provide an automatic method aimed to define multidimensional entities and properties over different sources for data acquisition and data analysis requests.

1. INTRODUCTION

The evolution of Web technologies and the data we are manipulating announce profound changes in Business Intelligence (BI) systems particularly in multidimensional data modeling and data discovery.

In general, multidimensional data modeling involves an initial study of key-business indicators and the identification of all possible data sources and data flows of the company. Consequently, the resulted model is adapted by the availability of data sources (what he has as a possible view) and user's analytical requirements (what he wants to view).

The semantic Web (SW) came with the ability to describe and link data over the Web using ontologies. The big idea behind SW is to automate intelligent programs to process data without human's interference and access easily to different sources of data through well-presented vocabularies and accurate declarations of information using RDF/OWL ontologies. Therefore, we would be capable to include external data sources from the Web in the BI system and provide additional information.

In the last years, many works have been proposed to exploit RDF/OWL ontologies in data warehousing, particularly, to generate multidimensional schemas. However, we found no publication presenting an inclusive solution of issues facing data integration or data discovery regarding the unpredictability of open Web sources.

In this paper, we introduce our new open BI/SW approach for data integration from semantic data sources by keeping the traditional BI features such as historical tractability and homogeneity. We also introduce a new method for automatic identification of multidimensional entities over RDF/OWL sources for both data integration and data analysis.

The rest of this paper is organized as follows: we introduce an overview of the problem context we are dealing with in section 2. In section 3, we introduce current proposals in the field. Afterward, we explain our transposition method to adopt SW in the BI system in section 4. Finally, a discussion about impacts and results of this method is presented to conclude the paper.

2. PROBLEM CONTEXT & MOTIVATION

2.1 OVERVIEW

The BI architecture introduces the data warehouse (DW) as the key component not only as a system of consolidation and storage of data but also because it is the result of a multidimensional modeling process.

The main aim of multidimensional modeling is to present data in an optimized form universally called the star schema. The star schema provides a better query performance compared to the entity-relationship model particularly when the execution plan is complex and the size of data is too big [1, 26].

Classically, the star-schema is the well-known schema adopted in data warehousing. It consists of a large table of measures, which are subjects of analysis known as a fact table. Descriptive tables called dimensions (e.g., sales revenue by year and by product) surround the fact table. A dimension presents a description of an analysis axis using attributes (dimension properties). Hierarchical dimensions are those dimensions that have a parent/child relationship, for example, one possible hierarchy in the date dimension is Year > Quarter > Month > Day.

A fact is modelled through one or several measures. Measures that can be added to all dimensions are called additives. Sometimes, it required more than one fact to respond analytical needs and link between all available dimensions especially when there are independent indicators, which form a group of star-schemas commonly known as a constellation schema.

The data warehouse, as the result of a multidimensional model, aims to organize and store subject-oriented, integrated, time variant and non-volatile collection of data [1]. Integrated collection of data means that data collected from several sources must be integrated in order to homogenize and give them a unique sense [2].

On-line analytical processing (OLAP) is applied to create multidimensional views from the data warehouse called OLAP cubes (views). In fact, a data warehouse characterizes a complete view in which users can filter and access to a large amount of information according to many analysis axis and therefore evaluate business indicators.

The semantic Web (or Web 3.0) is introduced as an extension of the actual Web 2.0 to enable a more intelligent interchange of information between machines by describing published data and enabling an automatic access and link of information sources. The Resource Description Framework (RDF) is the standard model for describing in a formal modus resources in the semantic Web. A RDF document is composed of a set of triples, each triple is an association between three elements: {*subject*, *predicate*, *object*}. The *subject* represents concept or resource described (e.g., a person), the *predicate* represents the type of property applied to the subject. It could be a *datatype* property (e.g., *hasName*) or an *object* property (e.g., *hasCar*). Finally, the object node, which correspond to related resource (e.g., Car) or a value (e.g., 'Sami) of the object/data property applied.

Sometimes, a subject can be related to many other resources that involves the use of blank node (abbreviated *b-node*) to serves as a grouping node. The subject and the object, if it is a resource, can be identified by a URI or be blank nodes, though, the predicate must be identified by a URI.

RDF Schema (RDFS) provides a vocabulary for RDF data and present taxonomies of classes and properties, for example, *rdfs:subClassOf*, *rdfs:range* and *rdfs:domain*. Sub-class properties are used to categorize classes.

The Web Ontology Language (OWL) is a W3C recommendation that provides much better integration, development, sharing and of ontologies regarding the basic layer provided by RDFS, OWL offer also a better reasoning capacity and provides a much larger vocabulary.

OWL-DL is an extension of OWL language based on Description Logic (DL) to supports expressiveness while retaining computational completeness and decidability [3]. OWL classes provide an abstraction mechanism for grouping resources with similar characteristics. RDF data can be found in several serializations and formats (e.g., XML, N-Triples, N3, etc.) or stored in particular databases systems optimized for RDF statements called triple stores or SPARQL *endpoints*.

SPARQL (adopted also by W3C) is a declarative query language (like SQL) and a web protocol designed to perform data manipulation on RDF graphs over the Web. The SPARQL query language is closely related to the following specifications.

2.2 RELATED WORKS

In an open-world context like the Web, data changes are unpredictable and users need to be aware of new external workflows that will fulfill their analysis requirements, therefore, an identification process of new multidirectional patterns must be applied using ontologies.

Several works are proposed to manipulate ontologies sources and supporting multidimensional data modeling, for example, in [4], [5] and [6] authors present semi-automatable methods aimed to generate multidimensional schemas by identifying multidimensional concepts (i.e., facts, measures, dimensions, and attributes) from an ontology along a set of criteria applied during the process.

Based-on OWL Lite vocabulary, authors in [7] proposed a method of transforming ontology structure into a star schema, the user choose in beginning point an object property as the subject of analysis and then create a dependency graph [8] before reorganizing the corresponding star schema.

So far, most of these works deal with a single/specific ontology source and assumes (implicitly) that the dimensions and the related levels within the multidimensional model are entirely static. In addition, they do not consider the possibility of performing OLAP operations directly over RDF sources.

Some works choose the opposite way to establish the link between multidimensional data modeling and OWL ontologies. For example, authors in [9] propose to transform multidimensional models into ontologies. The method is based-on the RDF Data Cube vocabulary [10] to generate the correspondent ontology using multidimensional concepts in order to perform OLAP operations and measures summarization directly over the graph.

Another comparable alternative proposed in [11], introduce QB4OLAP vocabulary as in extension of the RDF data cube vocabulary [3], the QB4OLAP engine aim to transform data stored in relational data warehouse into RDF triples (stored in a triple store) including dimension levels, measures, hierarchies within dimensions and the parent-child relationships among levels. Accordingly, OLAP operations in this case will be performed in SPARQL queries not SQL queries. The main aim of these approaches is to publish and share statistical information using data warehouses as alimentation sources [11, 17, 18, 23, 24, and 25].

Normally, data sources are most of the time well-known, structured and already identified, so ETL workflows could be easily automated to satisfy data warehouse needs. However, dealing with RDF data sources and open-world scenarios is quite different because of it require an identification process against the RDF graph to enclose and determine the business domain in the first place, and in the second place, to handle matching and merging tasks between different ontologies and testing data accessibility. Therefore, the changing nature of RDF data require a new type of ETL workflows conforming to the QoS-based characteristics [12].

The Extract, Transform, Query (ETQ) process was one of the proposals to handle on-demand user requests over the Web and to deal with semi-structured and streamed data under the Web. ETQ processes can be performed to directly respond user analysis requests without the need of loading the data into a data warehouse. However, this type of implementation require huge degree of automation tasks [2] and does not provide any data traceability, which is the great aim of data warehousing.

Other approaches have been focused to address the issue of data integration against RDF sources. A proposed survey [13] define three different type of ontology-based integration approaches. The single-ontology-based where sources are described to one general ontology with the aim of supplying the same vocabulary. The multiple-ontologies-based approach where each single source is described by its own ontology and managed independently. Finally, the hybrid approach where a global ontology is created from each local data source, as a result, local sources share the common vocabulary of the global ontology.

For example, ontology-based approach are presented in [2, 14], classified as hybrid approaches the purpose of these works is to populate an existent data warehouse from ontologies sources using OWL-DL and proceeding by a set of mapping tasks. Until now, this type of approaches require re-definition of local ontologies according to the common vocabulary when ETL plans are remade.

A semi-automatic ETL approach for integrating open-data sources is presented in [15, 24]. The process aims to construct a multidimensional view according to re-organization of data as rows, columns and values, it begin by identifying and classifying data (for each data source) according to its main type (i.e., verifying if data is a value type or structure type), based-on that, a graph representing the relationship between these two types of data is generated. However, classification techniques and underlying constraints cannot be fully automated during this process.

2.3 MOTIVATION

In actual BI systems, the data integration model is most of time applied under a well-controlled context. In consequence, data is available to be periodically loaded into a designed data warehouse through ETL processing.

The greatest advantage of this approach is that user's analytic requests are centralized and directed to a unique destination (i.e., DW) with no need for wrappers or other mapping operations, moreover, the historical aspect of data warehousing is preserved. Consequently, enabling a homogeneous system for users. However, even under a closed-world scenario the effectiveness of this system depend strictly on data sources that populate the centric data repository. For example, if one of the data sources (e.g., an XML file) changed its schema then it is necessary to include these metadata changes in the ETL process and rebuild the whole integration model.

In the opposite side, another approach is also available and consists of carry out users requests directly from sources. This approach give the impression to users that they interact with a centralized system while in fact their requests are dispatched overs distributed, autonomous and heterogeneous data sources.

The idea is based on schema mapping between the global schema used in the application layer and local schemas of data sources. In general, two methods are known to build the mapping between the global schema and data sources schemas. Firstly, the Global as View (*GaV*) method which consists of defining the global schema according to the schemas of the data sources (i.e., starting from the sources to produce the global schema). The Local as View (*LaV*) method in which the mapping is based on specifying the information content of every source in terms of a view over the global schema via dedicated queries.

Indeed, in any of these approaches, the system need to be aware automatically of any potential modifications or changes applied on the global schema (when using *GaV*) or on the sources schemas (when using *LaV*) in order to respond effectively to users request. The process is very delicate even under a well-controlled situation (i.e., the number of data sources is fixed and data structure changes are known in advance).

In a context like the Web, dealing with data is more complicated compared to a closed-world scenario due to three main factors:

a. The diversity and heterogeneity of distributed information sources,
b. The lack of unified vocabulary: an information could be expressed in different manner, thus, with no unified vocabulary it influence badly the accurateness and the quality of the information (e.g., dealing with similarities), and
c. The problematic of detecting and applying unpredictable schema modifications (like the former example).

Consequently, the evolution to an open BI/SW system is conditioned by the ability to handle automatic alteration on schemas (schemas of sources or schema of the data warehouse) and enabling discovery of multidimensional entities over multiple (especially non-relational) Web sources.

Certainly, by resolving some issues, analysts may have then the choice to populate an existing data warehouse with dynamic ETL (extract-transform-load) workflows, or even to perform analysis queries directly over a set of heterogeneous data source to extract the multidimensional view desired.

We assume that capabilities that present semantic Web technologies such as RDF, OWL and SPARQL are the solution to overcome the lack in the actual BI architecture by:

– Supporting well-presented declaration based on OWL ontologies.
– Resolving data conflicts and similarities issues during matching and merging data sources.
– Exploring data in a precise manner and avoiding irrelevant data using the SPARQL language.
– Enabling analytical queries based on reasoning and inference capabilities of the OWL language on its Description Logics' version (OWL-DL).

Moreover, the assumption has been a subject of many recent surveys. Authors in [16] introduced exploratory BI systems as the appropriate alternative for discovering and integrating semi-structured and/or unstructured data over the Web and discussed used-case possibilities according to a set of criteria (e.g. extensibility, structure, materialization, transformation, etc.).

3. METHOD FOUNDATION

3.1 ARCHITECTURE OF SW/BI SYSTEM

The method proposed in this paper is situated as a part of many workflows that represent our global BI/SW system (see Fig. 1). We introduce in this subsection main components of this system.

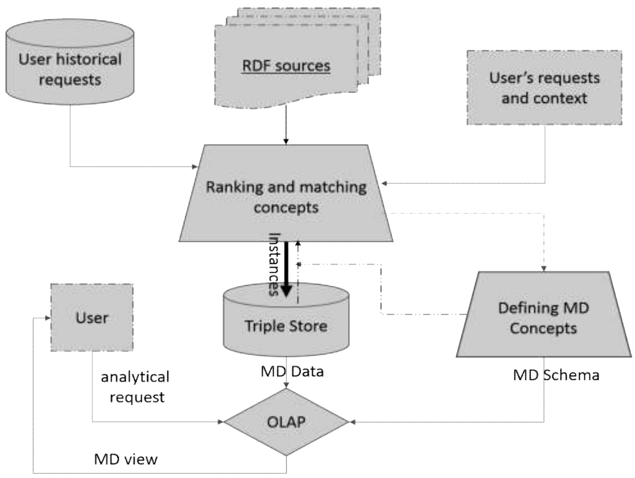


Figure 1 – Main workflows of our BI/SW system

It is difficult to set a mediation system to define mapping between the global vocabulary used by users and schemas of each data source because of heterogeneity of sematic Web sources.

In addition, as we focus essentially to surround a particular business domain (i.e., OWL concepts and relationships related to a specific domain), the user should be able to create his requests based on selecting necessary OWL classes from all available sources based-on classes' names or/and URIs. The main components are described as followed:

a. RDF sources: There are a many languages for building and exchanging RDF-structured information over the web. Each data source is flagged by the name of the related business domain.

b. User's requests and context: The user choose/ensign the business domain aimed (e.g., Car Sales, Astronomical Science, etc.) by identifying related concepts (i.e., RDF/OWL classes). As inputs, concepts selected are used to determine a temporary schema (t-schema) of related classes, object properties, data properties and literals.

c. User's historical requests: user's requests are saved as two logs: domain log and concepts log.

d. Ranking and matching concepts: data is gathered into the staging area source with a unified format (the local format used is TURTLE). A merging process of triples is established to connect RDF objects from diverse sources and produce a unified RDF graph. The matching phase of classes and relationships depend of user's selected concepts web-standards vocabularies used by those data sources.

e. Defining multidimensional concepts: establishing a superior layer of multidimensional properties and objects over OWL concepts and properties. It is also the multidimensional vocabulary (inspired from the *GaV* approach) of all user's analysis requests.

f. OLAP: performing multidimensional queries against the local triplestore using the resulted multidimensional scheme and according to user's analytical requests.

The improvement of this architecture is to optimize as much as possible time processing of RDF graphs by focusing on multidimensional data segments extracted using the present method.

3.2 META-MODEL IDENTIFICATION

According to the system above, data is loaded from sources into a triple store. The triple store will play the same role as a data warehouse to centralize and provide the historical mechanism needed. From this point, our aim is to transform the triple store into a multidimensional storage by annotating provided data with multidimensional properties. Therefore, our method is performed on available concepts in the staging area (see phase d. in Fig. 1) according to a set of predefined rules.

Based on selected concepts (OWL classes), a temporary ontology (t-ontology) describing the meta-model of related instances is created. The idea behind the use of a t-ontology is to focus on a particular area of metadata needed from the global graph for multidimensional annotation processing. The algorithm is divided into three phases as described in Fig. 2.

The first step consists of extracting triples for each OWL class from the selected list in which the class is a *domain*. All returned nodes are added to the t-ontology.

The initial phase begin by a massive extraction of objects that their domain classes are located in the selected user's list. The extraction is performed over all sources and the result is added into the t-ontology.

Secondly, the process identify and extract all objects linked with t-ontology subjects according to data type properties. Resulted triples are added to the temporary schema.

Finally, the t-ontology is enriched by adding all properties of collected objects.

The composition path $C_{unique}(s,p,o)$ relating a subject to an object is unique and inverse properties are ignored to avoid recursion within the t-ontology.

As a result, the t-ontology is more specific and provide the basic schema to identify potential measures, potential dimensions and eventual hierarchies. Steps of this phase are illustrated in Fig. 2.

```
Result: t-schema
Creating the list of concepts subjects and initialization of the t-schema
based-on selected ;
get all object properties and datatype properties of the domain and range
class of the user concepts list an put them to the rdf_subjects_inputlist.
subjects ←rdf_subjects_input;
tschema←null;
while subj in subjects do
    Get all domain classes of all objects and datatype properties from put
    them into the tschema
    if subj is a domain then
        tschema ←s ;
    end
end
Get all object properties existed between collected objects from sources and
which were not added during previous phases in order to complete the
current schema.
op ←rdf_op_input;
while p in op do
    if domain(p) ∈tschema and range(p) ∈tschema then
        tschema ←p ;
    end
end
Get all range class for each dataproperty that has the domain class in the
tschema.   dp ←rdf_dp_input;
while s in tschema do
    while p in dp do
        o ← rang(p);
        if p is a datatype property (dp) of s then
            tschema ←p ;
            tschema ←o ;
        end
    end
end
```

Figure 2 – Pseudo code used for building the t-ontology

Two nested loops in the third part of this algorithm are illustrated to explain how range classes are obtained, however, they are implemented as a single loop since all data properties are already selected and loaded into the t-ontology. Consequently, The worst case time complexity for this algorithm is O(N),

In our method, we consider a star join schema that is optimized for large sets of data such as RDF triples. Correspondingly, the constellation schema could be created from a set of available star-schemas that share some dimensions (conformed dimensions) which is helpful for aggregating fact tables.

The snowflake schema is not considered is this method because of its design's complexity that require a lots of joins to extract data compared to one single join using the star schema. Also, the star schema give the same level of details as the snowflake but with much more velocity and simplicity since sub-dimensions used in this last is transformed into full dimensions that are directly connected to the fact table in the star schema.

3.3 IDENTIFICATION OF MULTIDIMENSIONAL ENTITIES

We consider some rules to define multidimensional concepts over the resulted t-ontology. The first element to identify is the OWL class that describe the entity measure and enable us to construct the related fact.

Definition 1. Let F_m be a fact:

$$F_m = (m, I_F, D_F), \quad (1)$$

where m – measure, $I_F = I_F^1 ... I_F^j$ – Fact instances, $D_F = D_1 ... D_n$ – is a set of dimensions.

From the definition above, a fact is defined as the association of one measure and related dimensions. In the traditional relational model, a fact is defined with multiple measures, however, we assume that establishing the same analogy on RDF graphs will generate incoherence (e.g., recurrences, infinite cycles) and increase complexity during data manipulation. To overcome this lack, we defined the *sharedFact* concept. *sharedFact* is defined as a group of one-measure facts sharing same dimensions. In our model, a fact can be affected to none or several *sharedFacts*.

Definition 2. Let S_f be a *sharedFact* concept, D_i shared dimensions:

$$S_f = (M_F, D_F^{\cap}), D_F^{\cap} \neq \emptyset, \quad (1)$$

where $D_F^{\cap} = (D_1 \cap .. \cap D_n)$ – shared dimensions in a set of facts, $M_F = (m_1 \cup .. \cup m_n)$ – union of metrics in a set of facts (according to definition 1).

From this point, we introduce a set of rules to identify all these entities regarding the multidimensional terminology:

Rule 1. An OWL class is a potential measure concept if it has one numerical value as literal range.

Forcing numerical type for literal ranges is not recommended because measures could be expressed as string of characters.

Rule 2. A dimension class is defined when an OWL class is directly linked to a measure class.

Class expressions are not part of this rule, even though they are inserted during the loading process.

Rule 3. The range class of a dimension class is a hierarchy when the cardinality of correspondent object property is greater than 1.

Rule 4. The range class of a dimension class is a dimension attribute when the cardinality of correspondent object property is equal to 1.

Rule 5. When a *subClass* property describe a dimension class then the range object is a hierarchy class.

Rule 6. Only first level subclasses are considered in the t-ontology to represent details about the super class and avoid eventual loops.

We assume hierarchies are complement nodes in multidimensional nodes for a more general description of a dimension. This supposition came from the fact that hierarchical views could be created from details provided by a dimension. For example, if the measure *price* is described by dimension *city* then implicitly *price* is described according to *country* hierarchy.

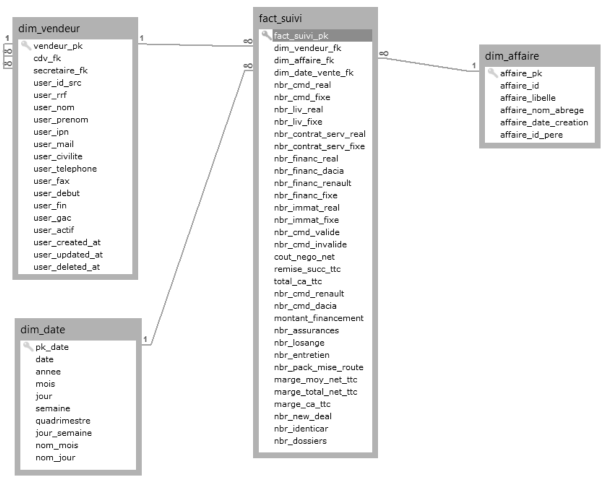
OWL description of the multidimensional schema within the t-ontology is based on the Data Cube vocabulary (QB) [10]. However, we do not consider grouping subsets of observations within a dataset as potential inputs because data should be extracted in its brut form without any grouping or slicing. Therefore, the vocabulary adopted will support only description of multidimensional data sets according to our definitions.

4. IMPLEMENTATION AND RESULTS

In order to test our method, we considered data sources from an existing relational BI system as inputs. In other word, we compared between multidimensional entities annotated during the process and predefined entities modeled in the data warehouse schema as illustrated in Fig. 3.

This type of scenarios will allow us to evaluate the number of facts extracted with the number of measures presented in the fact table. Our method is implemented using Java programming language and Jena APIs (https://jena.apache.org).

We begin by listing available sources (all sources are in this example are from relational tables) used to populate the existing data warehouse and we generate for each data source the correspondent OWL-DL ontology. Thus, we use Virtuoso (https://virtuoso.openlinksw.com/rdf/) to generate RDF-based ontologies from relational schemas. From this point, we provide the list of OWL classes needed as input for the execution.




```
Sfact_suivi ={
        Measures={
            [nbr_cmd_real],[nbr_cmd_fixe],[nbr_liv_real],[nbr_liv_fixe],[nbr_contrat_serv_real],
            [nbr_contrat_serv_fixe],[nbr_financ_real],[nbr_financ_dacia],[nbr_financ_renault],
            [nbr_financ_fixe],[nbr_immat_real],[nbr_immat_fixe],[nbr_cmd_valide],
            [nbr_cmd_invalide],[cout_nego_net],[remise_succ_ttc],[total_ca_ttc],[nbr_cmd_renault],
            [nbr_cmd_dacia],[montant_financement],[nbr_assurances],[nbr_losange],
            [nbr_entretien],[nbr_pack_mise_route],[marge_moy_net_ttc],[marge_total_net_ttc],
            [marge_ca_ttc],[nbr_new_deal],[nbr_identicar],[nbr_dossiers]
        }
        Dimensions={
            [dim_date],[dim_affaire],[dim_vendeur]
        }
    }
```

Figure 3 – Relational representation of multidimensional entities in this example

Ontologies generated are not edited or modified and thus they are used as potential sources for our test. The idea is keep the same nomenclature from the original sources and try to evaluate facts created by our method. The fact table is considered as a sharedFact according to our definition. Consequently, these generated facts are described as part of a parent-entity when measures share the same dimensions.

We run the program to annotate gathered RDF triples from generated ontologies using the local QB vocabulary. The result set shows that 26 measures were identified and described out of a total of 30 (87% of entities detected).

The same experience was performed, in one side, for six scenarios of detached fact-tables with related dimensions and original sources (like the first example), and in the other side, for the complete schema including all sources used in those scenarios. Table (1) and (2) show respectively for each scenario the percentage of detected measures and time consuming with the calculated and the real average.

Table 1. Percentage of detection for each scenario and the average.

SCENARIOS	FACT SCHEMA	TOTAL MEASURES	% DETECTION
S1	FACT_SUIVI	30	87%
S2	FACT_PILOTAGE	24	83%
S3	FACT_DOC_FORCE	1	100%
S4	FACT_COUT_NEGO	7	86%
S5	FACT_ANALYSE_DP	3	100%
S6	FACT_AIDE_POOL	6	100%
AVERAGE	ALL_FACTS	71	85%
S7	ALL_FACTS_REAL	71	80%

Table 2. Time consuming for each scenario and the average.

SCENARIOS	FACT SCHEMA	TIME (Seconds)
S1	FACT_SUIVI	0.554
S2	FACT_PILOTAGE	0.523
S3	FACT_DOC_FORCE	0.520
S4	FACT_COUT_NEGO	0.589
S5	FACT_ANALYSE_DP	0.596
S6	FACT_AIDE_POOL	0.552
AVERAGE	ALL_FACTS	0,556
S7	ALL_FACTS_REAL	0.543

It is obvious that the gap between the total of facts detected during the seventh scenario and the average of the six scenarios is too small which is very good and logic since reducing inputs (ontologies) will inevitably reduce time complexity of the gathered graph and the number of no-detected metrics.

5. CONCLUSION

Motivated by the need for a more compatible BI system with new external data sources from the semantic, we introduce in this paper our architecture of open BI system regarding the nature of workflows needed to integrate RDF data. The idea was to transform a triple store to a multidimensional repository capable to respond analysis queries with SPARQL.

To do that, we present in this paper our method to identify and annotate RDF data extracted from ontologies sources with multidimensional description based-on a local version of the RDF Cube vocabulary. The processes is begin by a meta-model identification in order to reduce the size of the targeted graph and provide less computations. During this phase a temporary ontology (t-ontology) is created from all ontologies sources involved in order to create a single graph for the next phase. Multidimensional entities are identified from the resulted schema and annotated according to a set of predefined rules and transformations. The evaluation showed that the method is efficient and allows us to more precisely annotate the necessary nodes and that its performance also depends on the quality of the ontologies describing the sources of data.

Interesting directions for future works include the optimization of the actual algorithm such as implementing path-unicity-hosted indexes and materializing multidimensional views from SPARQL result set.

6. REFERENCES

[1] R. Kimball, M. Ross, *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, John Wiley & Sons, 2011.

[2] Y. Laadidi, M. Bahaj, "Data integration system for RDF Data Sources," *Journal of Theoretical & Applied Information Technology*, vol. 83, no. 2, pp. 195-200, 2016.

[3] OBITKO, Web Ontology Language OWL, [Online]. Available at: https://www.obitko.com/tutorials/ontologies-semantic-web/web-ontology-language-owl.html, last accessed 2017/08/01.

[4] V. Nebot, R. Berlanga, "Building data warehouses with semantic web data," *Decision Support Systems*, vol. 52, issue 4, pp. 853-868, 2012.

[5] O. Romero, & A. Abelló, "A framework for multidimensional design of data warehouses from ontologies," *Data & Knowledge Engineering*, vol. 69, issue 11, pp. 1138-1157, 2010.

[6] M. Gulić, "Transformation of OWL ontology sources into data warehouse," *Proceedings of the IEEE 36th International Convention on Information and Communication Technology Electronics and Microelectronics (MIPRO)*, 2013, pp. 1143-1148.

[7] H. Wache, T. Voegele, U. Visser, et al., "Ontology-based integration of information-a survey of existing approaches," *Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing*, 2001, pp. 108-117.

[8] B. Vrdoljak, M. Banek, S. Rizzi, "Designing web warehouses from XML schemas," *Proceedings of the International Conference on Data Warehousing and Knowledge Discovery*, Springer, Berlin, Heidelberg, September 2003, pp. 89-98.

[9] L.T.T. Ho, C.P.T. Tran, Q. Hoang, "An approach of transforming ontologies into relational databases," *Proceedings of the Asian Conference on Intelligent Information and Database Systems*, Springer, Cham, 2015, pp. 149-158.

[10] The RDF Data Cube Vocabulary, [Online]. Available at: https://www.w3.org/TR/vocab-data-cube/, W3C Recommendation 16 January 2014, last accessed 2017/08/01.

[11] M. Bouza, B. Elliot, L. Etcheverry, et al., "Publishing and querying government multidimensional data using QB4OLAP," *Proceedings of the IEEE 9th Latin American Web Congress (LA-WEB-2014)*, 2014, pp. 82-90.

[12] O. Romero, A. Abelló, "Open access semantic aware business intelligence," *European Business Intelligence Summer School*, Springer, Cham, 2013, pp. 121-149.

[13] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner, "Ontology-based integration of information – A survey of existing approaches," *Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing*, Seattle, WA, 2001, pp. 108-117.

[14] W.H. Inmon, "What is a data warehouse," Prism Tech Topic, vol. 1, no 1, 1995.

[15] R. J. Santos, J. Bernardino, "Real-time data warehouse loading methodology," *Proceedings of the 2008 ACM International Symposium on Database Engineering & Applications*, September 2008, pp. 49-58.

[16] A. Abello, et al., "Using semantic web technologies for exploratory OLAP: a survey," *IEEE Trans. Knowl. Data Eng.*, vol. 27, issue 2, pp. 571–588, 2015.

[17] N. Prat, I. Megdiche, J, Akoka, "Multidimensional models meet the semantic web: defining and reasoning on OWL-DL ontologies for OLAP," *Proceedings of the*

ACM Fifteenth International Workshop on Data Warehousing and OLAP, ACM, 2012, pp. 17-24.

[18] B. Neumayr, C. Schütz, M. Schrefl, "Semantic enrichment of OLAP cubes: multi-dimensional ontologies and their representation in SQL and OWL," *Proceedings of the OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, Springer, Berlin, Heidelberg, 2013, pp. 624-641.

[19] I. Astrova, N. Korda, Ahto Kalja, "Storing OWL ontologies in SQL relational databases," *International Journal of Electrical, Computer and Systems Engineering*, vol. 1, issue 4, pp. 242-247, 2007.

[20] E. Vysniauskas, L. Nemuraite, "Mapping of OWL ontology concepts to RDB schemas. Information Technologies," *Proceedings of the 15th International Conference on Information and Software Technologies, IT 2009*, Kaunas, Lithuania, April 23-24, 2009, pp. 317-327.

[21] X. Liu, *Data Warehousing Technologies for Large-scale and Right-time Data*, PhD Thesis, Aalborg University, 2012.

[22] K. Selma, et al., "Ontology-based structured web data warehouses for sustainable interoperability: requirement modeling, design methodology and tool," *Computers in Industry*, vol. 63, issue 8, pp. 799-812, 2012.

[23] S. Khouri, B. Ladjel, "A methodology and tool for conceptual designing a data warehouse from ontology-based sources," *Proceedings of the ACM 13th International Workshop on Data Warehousing and OLAP*, 2010, pp. 19-24.

[24] M. Niinimäki, T. Niemi, "An ETL process for OLAP using RDF/OWL ontologies," *Journal on Data Semantics XIII, Lecture Notes in Computer Science*, vol 5530. Springer, Berlin, Heidelberg, pp. 97-119, 2009.

[25] A. Berro, I. Megdiche-Bousarsar, O. Teste, "Transformer les open data brutes en graphes enrichis en vue d'une intégration dans les systèmes OLAP," *INFormatique des Organisations et Systemes d'Information et de Decision (INFORSID)*, 2014, pp. 1-16. (in French)

[26] C. Ballard, D. M. Farrell, A. Gupta, et al., *Dimensional Modeling: In a Business Intelligence Environment*, IBM Redbooks, 2012.

***Yassine Laadidi**, He is PhD student (final year) and member of Laboratory of Innovation Technology in New Energy and Nano-materials (LITEN), University Hassan 1st Settat Morocco. He is a Senior Business Intelligence Analyst & database architect working on different projects for Automotive groups and multinational financial companies. He is interested in: DBA, OLTP/OLAP analysis, Data integration, Semantic Web technologies, data mining and Data warehousing.*

***Mohamed Bahaj**, Director of Laboratory of Innovation Technology in New Energy & Nano-materials (LITEN), University Hassan 1 st Settat Morocco. Dr. Mohamed Bahaj obtained his Ph.D. in Applied Mathematics from the University UPPA France. He is a Full Professor in Department of Mathematics and Computer Sciences from the University Hassan 1st Faculty of Sciences & Technologies Settat Morocco. He has published over 100 peer-reviewed papers. His research interests are Intelligent Systems, Ontologies Engineering, Software Engineering, Numerical Analysis and Scientific Computing. He has reviewed several papers in various journals. He has supervised several PhD theses in Computers Sciences and in Mathematics. He also attended a series of workshops, seminars and discussion forums for Academic Development on teaching and research.*