MMBR: a report-driven approach for the design of multidimensional models

Antonia Azzini[1], Stefania Marrara[1], Andrea Maurino[2], and Amir Topalović[1]

[1] Consorzio per il Trasferimento Tecnologico, C2T, Milano, Italy
name.surname@consorzioc2t.it,
[2] Universitá degli studi di Milano Bicocca
Dipartiment of Informatics, Systemistics and Communication
Milano, Italy
maurino@disco.unimib.it

Abstract. Nowadays, large organizations and regulated markets are subject to the control activity of external audit associations that require huge amounts of information to be submitted in the form of predefined and rigidly structured reports. Compiling these reports requires one to extract, transform and integrate data from several heterogeneous operational databases. This task is usually performed by developing a different ad-hoc and complex software for each report. Another solution involves the adoption of a data warehouse and related tools, which are today well-established technologies. Unfortunately, the data warehousing process is notoriously long and error-prone, therefore it is particularly inefficient when the output of the data warehouse is a limited number of reports. This article presents MMBR, an approach able to generate a multidimensional model starting from the structure of the reports expected as output of the data warehouse. The approach is able to generate the multidimensional model, and to populate the data warehouse by defining a domain-specific knowledge base. Even if using semantic information in data warehousing is not new, the novel contribution of our approach is the idea to simplify the design phase of the data warehouse, and make it more efficient, by using a domain specific knowledge base and a report-driven approach.

Keywords: multidimensional design, knowledge base, report driven methodology

1 Introduction

Reporting is a fundamental part of the business intelligence and knowledge management activity and it is strongly required by audit organizations. Reporting activity can be realized in an ad hoc way by means of specific and complex softwares, or by involving typical operations of extracting, transforming, and loading (ETL) procedures in coordination with a data warehouse. A data warehouse essentially combines information from several heterogeneous sources into one comprehensive database. By combining all of this information in one place,

a company can analyze its data in a more holistic way, ensuring that it has considered all the information available. At the basis of a data warehouse lies the concept of *multidimensional (MD) conceptual view of data*. The main characteristics of the multidimensional conceptual view of data is the fact/dimension dichotomy, which represents the data in an n-dimensional space. This representation facilitates the data interpretation and analysis in terms of *facts* (the subjects of analysis and related measures) and *dimensions* that represent the different perspectives from which a certain object can be analyzed.

Even if data warehousing benefits are well recognized by enterprises, it is well known that the warehousing process is time consuming, complex and error prone. Today the increasing reduction of the time-to-market of products forces enterprises to dramatically cut down the time devoted to the design ad the development of MD models that support the evaluation of the key performance indicators of services and products. Securitization is known by the literature as the financial practice of pooling various types of contractual debt such as residential mortgages, commercial mortgages, auto loans or credit card debt obligations (or other non-debt assets which generate receivables) and selling their related cash flows to third party investors as securities [1]. Mortgage-backed securities, which are the case study presented in this paper, are a perfect example of securitization.

Reports for auditing are often very specific, and their structure is usually imposed by the supervising organizations (e.g. Europan Central Bank, or the rating agency Moodys). The data included in the report are, in most cases, not useful for decision making activities due to the "control" nature of these reports. As a consequence, companies are forced to develop complex systems to compute data that are not useful for their business activities. In this situation, there is the need to develop a new approach able to support, in a fast and efficient way, the generation of reports. In this scenario we propose to adopt a data warehouse as storage system for data, but we introduce a new approach aimed at designing the multidimensional models on the basis of the structure of the report itself in a (semi-)automatic way, in order to significantly reduce the time needed to produce the report.

The MMBR (multi dimensional model by report) approach is able to automatically create the structure of a multi dimensional model (*MD* in the follow) and fill it on the basis of a knowledge base enriched with mapping information that depend on the specific application context. The preprocessing phase of the report (often a raw Excel file) is based on a table identification algorithm, which is able to extract the information needed to define the MD structure of the data warehouse. The approach has been tested in the context of financial data with the aim to automatically create the reports required by the Italian National Bank and by the European central bank. The methodology supports the creation of multidimensional model able to produce a given (set of) report(s). The term "by report" refers to the capability of our solution to create a multidimensional model starting from a given report that must to be filled with real data. MMBR is also able to generate the relational data structure related to the created Md

and it is also in charge of filling both fact and dimensional tables thanks to the use of domain ontologies enriched with mapping information to the operational sources. In the literature there are many methodologies for creating MDs starting by requirements, but this is the first attempt to define an approach for creating a MD model starting directly from the structure of the final reports only.

The remaining of the paper is organized as follows: Section 2 introduces the state of the art. Section 3 presents the proposed approach, while Section 4 describes the knowledge base that is a key element in the MMBR methodology. In Section 5, the table identification algorithm is presented, while Section 6 describes the creation of the MD models. A real example taken from the financial domain is then reported in Section 7. Conclusions and final remarks are reported in Section 8.

2 Related Work

In the literature several approaches for creating conceptual MD schema from heterogeneous data sources have been presented. According to [2], these approaches can be classified into three broad groups:

- *Supply-driven*: starting from a detailed analysis of the data sources these techniques try to determine the MD concepts. By this way there is the risk to waste resources by specifying unnecessary information structures, and by not being able to really involve data warehouse users. See for instance [3–5].
- *Demand-driven*: These approaches focus on determining the MD requirements based on an end-user point of view (as typically performed by other information systems), and mapping them to data sources in a subsequent step (see for example [6, 7]).
- *Hybrid approaches*: Some authors (see for example [8–10]) propose to combine the two previously presented approaches in order to harmonize, in the design of the data warehouse, the data sources information with the end-user requirements.

All the methodologies available in literature, however, have the goal to create a MD model as general as possible in order to allow the generation of any report. This assumption requires a lot of effort in both the warehouse conceptualization phase and in the ETL procedure design and development. In several industrial contexts, there is the need to produce a limited number of reports only and, sometimes, with a very strict and well defined structure due to auditing rules or for specific business requirements. In the finance domain, for example, banks are required by central authorities and rating agencies to produce very specific reports related to the securization activities they perform. In the field of the Semantic Web, Bontcheva and colleague [11] present an approach for the automatic generation of reports from domain ontologies encoded in Semantic Web standards like OWL. The novel aspects of their so-called "MIAKT generator" are in the use of the ontology, mainly the property hierarchy, in order to make it easier to connect a generator to a new domain ontology.

In [12] Nebot and colleagues propose an approach in which a Semantic Data Warehouse is considered as a repository of ontologies and other semantically annotated data resources. Then, they propose an ontology-driven framework to design multidimensional analysis models for Semantic Data Warehouses. This framework provides means for building an integrated ontology, called the Multidimensional Integrated Ontology (MIO), including the classes, relationships and instances representing the analysis developed over dimensions and measures. Romero and colleague [13] introduce a user-centered approach to support the end-user requirements elicitation and the data warehouse multidimensional design tasks. The authors explain how the feedback of a user is needed to filter and shape results obtained from analyzing the sources, and eventually produce the desired conceptual schema. In this scenario, they define the AMDO (Automating Multidimensional Design from Ontologies) method, aimed at discovering the multidimensional knowledge contained in the data sources regardless of the users requirements. Another work aimed at supporting the multidimensional schema design is given by [14], in which the authors propose an extension of their previous work [15]. They follow a hybrid methodology where the data source and the end-user requirements are conciliated at the early stage of the design process, by deriving only the entities that are of interest for the analysis. The requirements are converted from natural language text into a logical format. The concepts in each requirement are matched to the source ontology and tagged. Then, the multidimensional elements such as fact and dimensions are automatically derived using reasoning.

On the other hand, Benslimane and colleague [16] define a contextual ontology as an explicit specification of a conceptualization, while Barkat [17] proposes a complete and comprehensive methodology to design multi-contextual semantic data warehouses. This contribution is aimed to provide a context meta model (language) that unifies the definitions provided in Database literature. This language is considered as an extension of OWL, which is the standard proposed by the W3C Consortium [18] to define ontologies. It is defined by the authors in order to provide a contextual definition of the used concepts, by offering an externalization of the context from the ontology side.

Pardillo and colleagues [19] present an interesting approach aimed at describing several shortcomings of the current data warehouse design approaches, showing the benefits of using ontologies to overcome them. This work is a starting point for discussing the convenience of using ontologies in the data warehouse design. In particular the authors present a set of situations in which ontologies may help data warehouse designers with respect to some critical aspects.

As also considered in this approach, it is important to underline that a domain specific ontological knowledge allows to enrich a multidimensional model in aspects that have not been taken into account during the requirement analysis or data-source alignment phases, as well as other aspects, like for example the application of statistic functions in order to aggregate data.

3 Description of the approach and outline of the architecture

The MMBR approach main phases are shown in Figure 1: 1) Table Processing (TP), 2) Row and Column Header Identification and Extraction (RCHIE), 3) Ontology Annotation (OA), 4) Management of Non-Identified labels (MNL), 5) creation of the MD model, and 6) ETL Schema Generation (ETL). The input of the TP phase is the template file that has to be filled with the data extracted from an Operational Data Base (ODB). In the TP phase the preprocessing of the template is performed by removing icons and other figures, moreover all terms in the schema are lowered and comment and description fields are removed.

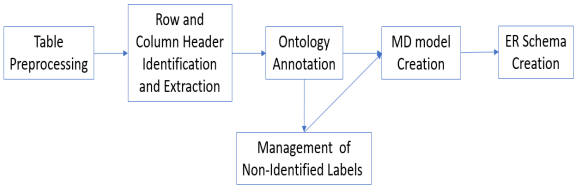

Fig. 1. Overall representation of the approach.

The RCHIE phase is based on the *table identification algorithm* aimed at identifying and extracting the row and column headers in the template. The details of the table identification algorithm are presented in Section 5.

The list of terms recognized in the reports by the table identification algorithm is then annotated on the basis of a knowledge base (see Section 4). This phase produces two lists; the first one is the list of identified terms annotated w.r.t. the knowledge base, the second one is the list of terms that are not annotated. There are several possible reasons of failure for the annotation activity. The most frequent reason is that a given term may be not included in the knowledge base because it is not relevant to the domain (e.g "Total"). It is also possible that a term is not annotated because it is a composition of different terms (such as "MortageLoan" or "DelinquentLoan")[3]. Moreover some terms are written in a language different from English (e.g. "garantito" that means guaranteeded in Italian). In all these cases, not annotated terms are manually checked and, if relevant, added to the ontology by defining the corresponding rdf:label property. The annotated list of terms is the input for the creation of the dimensional fact model (see Section 6). This logical model is finally translated into a relational star schema. In this phase the relational database is filled with data coming from the ODB. This activity is performed on the basis of the mapping rules included in the knowledge base. This activity is fully described in Section 4.

The architecture supporting the MMBR approach is represented in Figure 2.

[3] the description of these terms is reported in the case study section

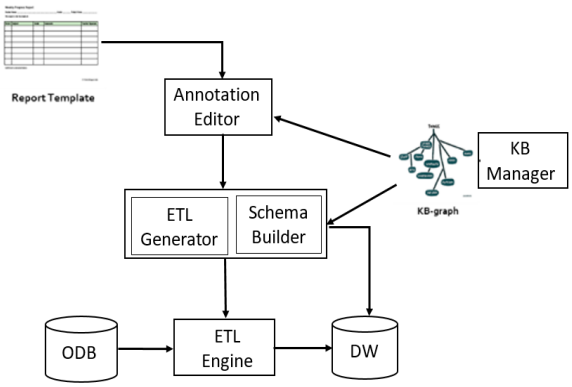

Fig. 2. Representation of the Overall architecture.

The Annotation Editor is in charge of the first three phases of the MMBR approach, by removing non relevant strings and images from the input file (e.g. logo, comments), and by identifying the terms that are annotated w.r.t. the KB. The Schema builder is the software component aimed at creating the logical relational description of the MD model. The ETL generator is in charge of extracting, on the basis of the knowledge base, the information necessary to create the extraction-transformation-load data from the ODB to the data warehouse. Then, the Knowledge base manager is in charge of managing and evolving the knowledge base. Any popular tool as, for instance, Protege[4] may be used for the KB creation.

4 MMBR Knowledge base

At the core of the proposed approach lies the creation of the knowledge base KB, which includes:

- the set of MD concepts and relations (fact, dimensions, measures, attributes);
- the list of terms adopted in the specific application domain (eg. ecommerce, bank securitization,...);
- the ODB schema.

In order to create a sharable knowledge base we started by using existing ontological description and only when no ontology is available we created new concepts. A simplified version of the *data cube vocabulary*[5], i.e., a W3C recommendation for modeling multidimensional data, is used to define the MD concepts. The top level representation of the defined KB is shown in Figure 3.

[4] `https://protege.stanford.edu/`
[5] `https://www.w3.org/TR/vocab-data-cube/`

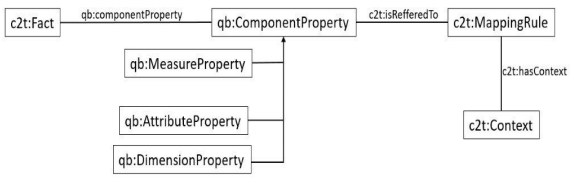

Fig. 3. Top level representation of the Knowledge Base.

The MD concepts are organized as follows. A fact (the event that is the target of a report, e.g., a sell in a e-commerce domain, a loan in the bank domain) is described by a set of measures and can by analyzed by considering its dimension and descriptive attributes. In the data cube vocabulary dimensions, measures and descriptive attributes are described by the concept *component properties instances*. Dimensions, measures and descriptive attributes are *terms* of the application domain and they are defined by the human (domain) expert trough the Knowledge Base. In fact, the KB annotation specifies if a KB component refers to a fact, a measure or to a dimension. Such elements are then compared with each label extracted from the Excel file in order to define fact, measures and dimensions of the corresponding model. In order to build a KB related to the e-commerce domain, it is possible, for example, to use concepts described in the *good relation* section of the vocabulary [6]. In this scenario instances of *DimensionProperties* are *gr:ProductOrService, gr:Brand*, while instances of *dq:Measure* are *gr:UnitPriceSpecification, gr:amountOfThisGood.* If no vocabulary is available, a new, ad-hoc vocabulary, has to be defined as first (as also reported in Section 7).

Concept *qd:ComponentProperty* can have one or more *rdf:label properties* associated to, that represent the references to the instances of the target concept. For example the dimension *gr:Brand* may be labeled as *"NameOfProduct"* or *"BrandName"*. During the annotation phase, labels are used to associate terms of the report to the application domain concepts.

In order to populate the MD model it is necessary to know how the *qb:componentProperties* are described in the operational DB. This mapping is described in the KB itself, by means of the *c2t:mappingRule* concept, which associates a *c2t:mappingFormula* related to a given instance of the *qb:ComponentProperties*. The *c2t:mappingFormula* contains a reference to some tables of the ODB and a query predicate over their tuples.

For example, in a bank scenario we can assume that the TLoan table of the ODB contains all information related to loans. A loan with a *fixed rate* (i.e., a loan where the interest rate on the note remains the same through the term of the loan) can be represented in the ODB by the predicate *InterestRate=1*, while a floating rate can be described by the predicate *InterestRate>1*. The

[6] `http://www.heppnetz.de/projects/goodrelations/`

Time	Thailand		Japan		Total
	Food	NonFood	Food	NonFood	
2006	2400	2200	11000	5000	20600
2007	4500	3200	12000	6000	25700
2008	5600	2900	10000	5500	24000
Total	12500	8300	33000	16500	70300

Fig. 4. Example of report

formula *c2t:mappingFormula* includes the references to the TLoan table and the predicate regarding *InterestRate*.

The concept *c2t:context* in Figure 3 has value when reports provided by different audit authorities have different mapping formulas for the dimension *dq:componentProperties*. For example, a given audit authority may classify a company as "small" if the employee number does not reach 10, while for another authority a company is small if it has less than 15 people employed. In this case we will have two different *c2t:MappingFormula*.

5 Table Identification

Reports are usually represented by tables that can be divided into different areas, according to their structure. Thus, being able to identify the inner structure of the table is important to find the concepts relevant to the MD models generation. As discussed in the introduction, the multidimensional model represents the data into a n-dimensional space; under this perspective each report can be considered as one of the possible hyperplane slicing the n-dimensional cube of data. To represent this hyperplane into a bi dimensional table it is necessary to reduce the dimensions. In figure 4 the MD is composed by three dimensions (time, nations and type of sold goods) that are "flattened" into a bi-dimensional space by associating the values of type of sold goods (Food and non Food) to the nation dimension. According to this assumption row and columns header may contain dimensions, values of dimensions and measures of the MD.

In the RCHIE phase a table was assumed as composed by three types of cell; respectively textual, data and schema ones. Figure 5 shows the general schema. The cell identifiers are represented by the couple $< X, Y >$, as reported in the table shown in the figure.

The table may contain several types of cells, as defined in the following:

- **textual-cell**: this cell is not used for table annotation, these cells are shown in grey in Figure 5, and they may contain simple text.
- **data-cell**: it contains data that are computed on the basis of the MD model. These cells are shown in white in the figure.
- **schema-cell**: it specifies properties over a set of data-cells. It is shown in dark grey in the figure. This cell defines the header $h =< x, y >$ of a set of

<1,1>	abc	abc				<1,n>
			C_Header	C_Header	...	C_Header
abc			C_Header	C_Header	...	C_Header
abc	R_Header	R_Header	<4,4>			
	R_Header	R_Header			...	
	...	...				
<m,1>	R_Header	R_Header				<m,n>

Fig. 5. Example of table used by the Table Identification algorithm

data cells, by specifying some semantic aspects (i.e., the measure or a value on a dimension).

Rows and columns are identified in order to extract the labels corresponding, respectively, to measures, dimensions, instances of the dimensions, etc. (for instance not relevant information as the $TOTAL$ value shown in Figure 6). These labels represent the input of the annotation phase, which produces the annotated list of terms as output.

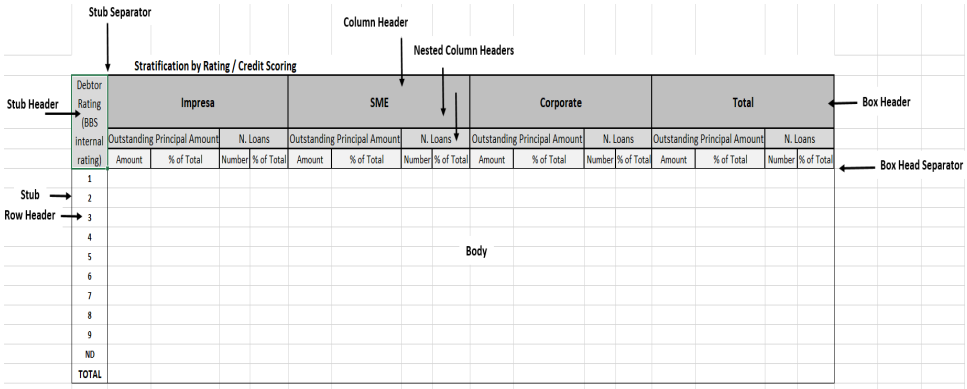

Fig. 6. Example of Table

In the literature different table identification algorithms aimed at handling the tables structure have been proposed [20]; in our work the focus is identifying and removing multi spanning cells. An example is reported in Figure 6, where one of the reports related to securitization is shown. The *Stub Header* details information w.r.t. the measures *Loan* and *Oustanding Principal* of different types of companies, as Corporate, SME and "Impresa" (it refers to retail companies in the Italian jargon). Measures, names and instances of dimensions are placed in the *Box Header* and/or the *Stub* areas as headers, and they are used to index the elements located in the *Body* area of the table. The *Stub Header* may also contain

a header naming or describing the dimensions located in the stub. Results of table identification algorithm is shown in figure 7 where all data-cell are semantically associated to their row and column headers.

	Impresa	Impresa	Impresa	Impresa	SME	SME	SME	SME	Corporate	Corporate	Corporate	Corporate	Total	Total	Total	Total
	Outstanding Principal Amount	Outstanding Principal Amount	N. Loans	N. Loans	Outstanding Principal Amount	Outstanding Principal Amount	N. Loans	N. Loans	Outstanding Principal Amount	Outstanding Principal Amount	N. Loans	N. Loans	Outstanding Principal Amount	Outstanding Principal Amount	N. Loans	N. Loans
Debtor Rating (BBS internal rating)	Amount	% of Total	Number	% of Total	Amount	% of Total	Number	% of Total	Amount	% of Total	Number	% of Total	Amount	% of Total	Number	% of Total
1	posx=3, posy=6	posx=4, posy=6	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2	posx=3, posy=7	posx=4, posy=7	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3	posx=3, posy=8	posx=4, posy=8	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4	posx=3, posy=9	posx=4, posy=9	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
ND	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
TOTAL	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Body

Fig. 7. Example of flattened table

Finally, the RCHIE phase extracts a list of unique terms that are in the column and row headers. These terms are then annotated by means of the knowledge base, by evaluating the labels related to the application domain concepts and the terms extracted from the report table.

6 MD creation and population

The list of annotated terms and the KB are the only two elements needed to design and populate the MD. The Dimensional Fact Model (DFM)[3] approach is used to describe the MD model. Each annotated term of the list is enriched by its type or subclass in order to understand if it is a measure, a dimension or an instance of dimension. This can be realized by means of a set of SPARQL[7] queries over the KB (an example of query is shown in Section 7). With this information is possible to create the DFM and the corresponding logical relational schema by means of the original methodology proposed in [3]. The relational schema is then populated according to the mapping information defined in the knowledge base.

All dimensional tables are populated with the instances defined in the KB, while the fact table is defined in a two steps procedure. In the first step all instances of the facts (e.g. sell or loan) are selected from the ODB by taking into account only the measures available in the annotated list. The second step is in charge of connecting the fact table with the dimensional tables. An Update

[7] https://www.w3.org/TR/rdf-sparql-query/

query is executed to associate each instance of the fact table with the instances of the dimensions tables. Even in this case the KB plays a strategic role since it allows to extract the mapping formula at the basis of the SPARQL queries (see section 7).

7 Case Study

The scenario motivating the definition of a report driven approach for the design of multidimensional models is related to the financial domain. In particular, the reporting activity of securitization was analyzed.

Applying the MMBR approach in this context, the first activity performed is the generation of the domain KB and vocabulary. The Literature has proposed two different vocabularies that partially describe the loan domain: FIBO [8] and Schema.org [9]. FIBO, a Financial Industry Business Ontology, contains the loan terms definitions without any further specification. Schema.org, does not contain a full exhaustive specification of the securitization domain, but it includes the LoanOrCredit concepts[10] only. The KB defined in this work to describe the securitization domain is an ontology called OntoLoan. During the KB definition, domain experts were in charge to define the main terms and concepts. OntoLoan ontology is not freely available, since it is covered by the company's intellectual properties. However, the top level of OntoLoan is shown in Figure 8.

Figure 9 shows an example of securization report. Note that all personal data related to the bank owning the report are removed for privacy issues, while the values for different kinds of loans are reported.

The term *Performing Loans* refers to all those loans with no overdue interest payments, or with unpaid installments due, even if under the limit on the delay of days outstanding (which changes according to the securitization contract terms). *Delinquent Loan* refers to the loans close to default, i.e., to unpaid installments due to a delay in payments close to the limit on the delay of days overdue. *Defaulted Loans* refers then to loans with significant delays in payments.

Any kind of loan is further divided according to other features, generating the definition of *Mortgage Loan*, *Guaranteed Loan*, i.e. loans insured not by mortgages but by other guarantees (e.g., pledges), and, finally, *Unguaranteed Loan*, i.e. not insured.

The first phase of the MMBR approach removes text fields that do not carry relevant information from the report. An example of removed test is the string "A. PORTFOLIO OUTSTANDING BALANCE". The annotation tool removes the cell spanning starting from the table of Figure 9, arriving to the table structure shown in Figure 10. Data-cell in position $< 3, 3 >$ represents the aggregation of the values of *Outstanding Principal* of loans that are both performing and able to pay off the loan even in case of default of the borrower. The value in the cell

[8] `https://www.edmcouncil.org/financialbusiness`
[9] `https://schema.org`
[10] `https://schema.org/LoanOrCredit`

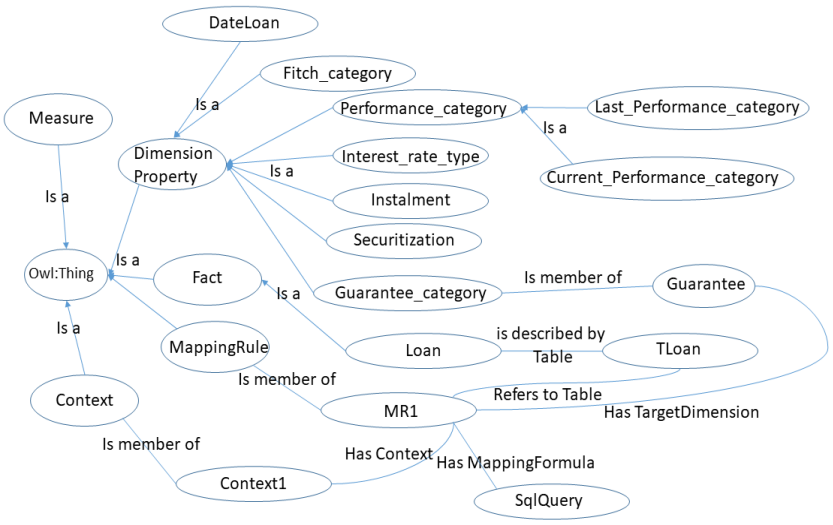

Fig. 8. The top level representation of OntoLoan

				Delinquent instalments			Total
		Outstanding Principal	Accrued Interest	Principal inst.	Interest inst.	Total	
Performing Loans	Mortgage Loans						
	Guaranteed Loans						
	Unguaranteed Loans						
	Total Unsecured						
Delinquent Loans	Mortgage Loans						
	Guaranteed Loans						
	Unguaranteed Loans						
	Total Unsecured						
Defaulted Loans	Mortgage Loans						
	Guaranteed Loans						
	Unguaranteed Loans						
	Total Unsecured						
	TOTAL						

Fig. 9. An example of report template

				Delinquent instalments	Delinquent instalments	Delinquent instalments	Total
		Outstanding Principal	Accrued Interest	Principal inst.	Interest inst.	Total	
Performing Loans	Mortgage Loans	pos=3, pos=3					
Performing Loans	Guaranteed Loans	pos=3, pos=4					
Performing Loans	Unguaranteed Loans	pos=3, pos=5					
Performing Loans	Total Unsecured						
Delinquent Loans	Mortgage Loans						
Delinquent Loans	Guaranteed Loans						
Delinquent Loans	Unguaranteed Loans						
Delinquent Loans	Total Unsecured						
Defaulted Loans	Mortgage Loans						
Defaulted Loans	Guaranteed Loans						
Defaulted Loans	Unguaranteed Loans						
Defaulted Loans	Total Unsecured						
	TOTAL						

Fig. 10. An example of flattened report

with position $< 3, 4 >$ represents the aggregation of the *Outstanding Principal* of loans that are both performing and guaranteed.

With this first activity the following list of terms related to the domain is extracted *loan:Performing, loan:Mortgage, loan:Guaranteed, loan:Unguaranteed, loan:Delinquent, loan:Defaulted, loan:DelinquentInstalments, loan:OutstandingPrincipal, loan:AccruedInterest, loan:PrincipalInstalment, loan:InterestInstalment.*. For each element of such list, MMBR retrieves from the KB the name of the dimensions or measures related to it, by means of Sparql queries. An example of query is the following.

```
SELECT distinct  ?x, ?p
WHERE {
loan:Guarantee rdf:type ?x.
?x rdfs:subClassOf ?p
 }
```

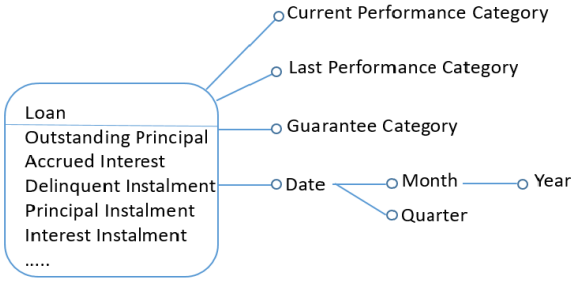

Fig. 11. Dimensional Fact Model Schema Example.

The example query is able to recognize, as shown in Figure 8 that *loan:Guarantee* is member of an entity named *Guarantee_Category* that is a subclass of *qb:DimensionProperty*. Figure 8 also shows the query properties. After the identification of measures and dimension the DFM is designed as shown in figure 11, according to the approaches already published in the literature [21] for the schema definition.

The DFM is then translated into a relational schema, whose instance is created in a relational dbms as described in section 6.

In order to update the fact table, it is possible to retrieve the mapping formula in the KB, by means of a SPARQL query. For example to update the *guaranteed loan*, first we recover from the KB the corresponding mapping formula by using the following query:

```
SELECT ?table, ?rule
WHERE {
?s rdf:type loan:MappingRule.
?s loan:hasContext loan:context1.
?s  loan:hasTargetDimension loan:Guarantee.
?s  loan:refersToTable ?table.
    ?s  loan:hasMappingFormula ?rule.
}
```

The result is the following predicate:

```
TLoan
VAL_IPOTECA = 0 and (flag_garanzia_confidi='Y' or
(importo_pegno + importo_garan_pers) > 0)^^string
```

The corresponding update query using IBM DB2 SQL is:

```
UPDATE Fact
SET id_Guarantee_category=
(SELECT Guarantee
     FROM fact join odb.TLoan
     WHERE fact.id=obd.TLoan.id and
     VAL_IPOTECA = 0 and
     (flag_garanzia_confidi='Y' or (importo_pegno + importo_garan_pers) > 0)
)
```

8 Conclusion and Future Work

This work presents a "multidimensional model by report" (MMBR) approach supporting the creation of multidimensional models able to produce a given (set of) report(s). The term "by report" refers to the ability to create a multidimensional (MD) model starting from a given report (typically expressed as Microsoft Excel file) that has to be filled with data extracted from a set of heterogeneous sources. Important contributions are the automatic generation of the relational data structure correlated to the MD models generated by the approach, and the ability to fill both fact and dimensional tables on the basis of domain ontologies enriched with mapping information related to the data sources. There may be several future directions of research. The first one is related to the definition of an approach for the automatic computation of aggregates of data according to the topological position of the cells that contain them, by taking into account row and column headers. Another interesting research activity will study how to enrich the table identification algorithm. The aim is to allow the management of a larger (w.r.t., the actual algorithm) number of types of report, improving the efficiency of the presented approach.

References

1. Simkovic, M.: Competition and crisis in mortgage securitization
2. Winter, R., Strauch, B.: A method for demand-driven information requirements analysis in data warehousing projects. In: 36th Hawaii International Conference on System Sciences (HICSS-36 2003), CD-ROM / Abstracts Proceedings, January 6-9, 2003, Big Island, HI, USA, IEEE Computer Society (2003) 231
3. Golfarelli, M., Maio, D., Rizzi, S.: The dimensional fact model: A conceptual model for data warehouses. Int. J. Cooperative Inf. Syst. **7**(2-3) (1998) 215–247
4. Golfarelli, M., Graziani, S., Rizzi, S.: Starry vault: Automating multidimensional modeling from data vaults. In Pokorný, J., Ivanovic, M., Thalheim, B., Saloun, P., eds.: Advances in Databases and Information Systems - 20th East European Conference, ADBIS 2016, Prague, Czech Republic, August 28-31, 2016, Proceedings. Volume 9809 of Lecture Notes in Computer Science., Springer (2016) 137–151
5. Blanco, C., de Guzmán, I.G.R., Fernández-Medina, E., Trujillo, J.: An architecture for automatically developing secure OLAP applications from models. Information & Software Technology **59** (2015) 1–16

6. Jovanovic, P., Romero, O., Simitsis, A., Abelló, A., Mayorova, D.: A requirement-driven approach to the design and evolution of data warehouses. Inf. Syst. **44** (2014) 94–119
7. Prat, N., Akoka, J., Comyn-Wattiau, I.: A uml-based data warehouse design method. Decision Support Systems **42**(3) (2006) 1449–1473
8. Nabli, A., Feki, J., Gargouri, F.: Automatic construction of multidimensional schema from OLAP requirements. In: 2005 ACS / IEEE International Conference on Computer Systems and Applications (AICCSA 2005), January 3-6, 2005, Cairo, Egypt, IEEE Computer Society (2005) 28
9. Giorgini, P., Rizzi, S., Garzetti, M.: Grand: A goal-oriented approach to requirement analysis in data warehouses. Decision Support Systems **45**(1) (2008) 4–21
10. Blanco, C., de Guzmán, I.G.R., Fernández-Medina, E., Trujillo, J.: An MDA approach for developing secure OLAP applications: Metamodels and transformations. Comput. Sci. Inf. Syst. **12**(2) (2015) 541–565
11. Bontcheva, K., Wilks, Y. In: Automatic Report Generation from Ontologies: The MIAKT Approach. Springer Berlin Heidelberg, Berlin, Heidelberg (2004) 324–335
12. Nebot, V., Berlanga, R., Pérez, J., Aramburu, M., Pedersen, T.: Multidimensional integrated ontologies: A framework for designing semantic data warehouses. Journal on Data Semantics XIII (2009) 1–36
13. Romero, O., Abelló, A.: A framework for multidimensional design of data warehouses from ontologies. Data & Knowledge Engineering **69**(11) (2010) 1138–1157
14. Thenmozhi, M., Vivekanandan, K.: An ontology based hybrid approach to derive multidimensional schema for data warehouse. International Journal of Computer Applications **54**(8) (2012)
15. Thenmozhi, M., Vivekanandan, K.: A framework to derive multidimensional schema for data warehouse using ontology. In: Proceedings of National Conference on Internet and WebSevice Computing, NCIWSC. (2012)
16. Benslimane, D., Arara, A., Falquet, G., Maamar, Z., Thiran, P., Gargouri, F. In: Contextual Ontologies. Springer Berlin Heidelberg, Berlin, Heidelberg (2006) 168–176
17. Barkat, O., Khouri, S., Bellatreche, L., Boustia, N.: Bridging context and data warehouses through ontologies. In: Proceedings of the Symposium on Applied Computing, ACM (2017) 336–341
18. : W3C Standard Consortium, howpublished = `http://www.w3.org`
19. Pardillo, J., Mazón, J.N.: Using ontologies for the design of data warehouses. International Journal of Database Management Systems, (IJDMS) **3**(2) (2011) 73–87
20. Zanibbi, R., Blostein, D., Cordy, J.R.: A survey of table recognition. Document Analysis and Recognition, Models, observations, transformations , and inferences **7**(1) (Mar 2004) 1–16
21. Sugumaran, V., Storey, V.C.: Ontologies for conceptual modeling: their creation, use, and management. Data & Knowledge Engineering **42**(3) (2002) 251–271