International Journal of Computer Applications (0975 – 8887)
Volume 54– No.8, September 2012

An Ontology based Hybrid Approach to Derive Multidimensional Schema for Data warehouse

M.Thenmozhi
Assistant Professor
Department of Computer Science and Engineering
Pondicherry Engineering College
Puducherry

K.Vivekanandan
Professor
Department of Computer Science and Engineering
Pondicherry Engineering College
Puducherry

ABSTRACT
Due to the diversity of data source data integration has become a challenging task. Data warehouse system plays a vital role to integrate the data for making important business decisions. Data within the data warehouse is arranged as multidimensional schema. In past many works exist to carry out the design of the multidimensional schema for data warehouse from either requirements and/or data sources. These approaches are either manual or automated which work with only relational sources. But as today the data warehouse system needs to deal with semi-structured and unstructured sources, the design task becomes much tedious. Recently, ontology has been very useful for different data integration projects. The use of ontology could solve the syntactic and semantic conflicts that arise from heterogeneous sources. It also provides a way for automating the design of multidimensional schema and populating the data warehouse in a more meaningful way. This paper proposes a framework using ontology for the design of multidimensional schema. Our framework uses a hybrid approach where the reconciliation of requirements and data source are done at the early stage of design. We adopt ontology reasoning in order to automatically derive multidimensional elements such as facts and dimensions.

Keywords
Data Modelling, Multidimensional Schema, Data warehouse, Ontology

1. INTRODUCTION
A data warehouse (DW) provides subject oriented, integrated, time-variant and non-volatile collection of data for strategic decision making. The data warehouse has been extensively used in past years for business analysis. It allows the top management to take critical decisions in order to improve their business in the competitive market world. The data warehouse process consists of three phases: extraction of data from distributed operational sources; integration and organization of data consistently into the DW; accessing the integrated data in an efficient and flexible fashion using OLAP or data mining tools [1]. In order to analyze a business in different perspective, the data within the data warehouse is organized as multidimensional schema. A multidimensional schema consists of fact and dimension tables. A fact is the subject by which a business is analyzed and dimensions are the different analysis perspective. For example, in a retail domain sales may become a fact and product, location, time etc., are the dimensions. A single dimension may be represented with different levels i.e., time dimension may have year, month, week as different levels. Numeric attributes are the measures of a fact through which a business is to be measured e.g., for fact sales, revenue may be treated as the measure. Figure 1 represents a sample star schema for a sales domain.

In past many works exist to carry out the conceptual design of multidimensional schema. Some of them are manual and few provide automated way to carry out the design task. As domain knowledge is crucial, the manual approaches place a heavy burden over the designer and the design outcome depends on his ability and expertise. The automated approaches available in the literature follow either a supply driven [1], [2], [3] or demand driven [4], [5], [6], [7], [8]. When the multidimensional schema is derived from data sources (supply driven) it may generate too many results which may not be of interest to the analyst. In the demand driven approach the design is carried out based on the requirements and hence it may miss some interesting concepts available in the data source. To overcome these drawbacks few approaches have been proposed which follows a hybrid methodology [6], [9], [10], [11], where the reconciliation of requirements with the data source is carried out at the early stage of design.

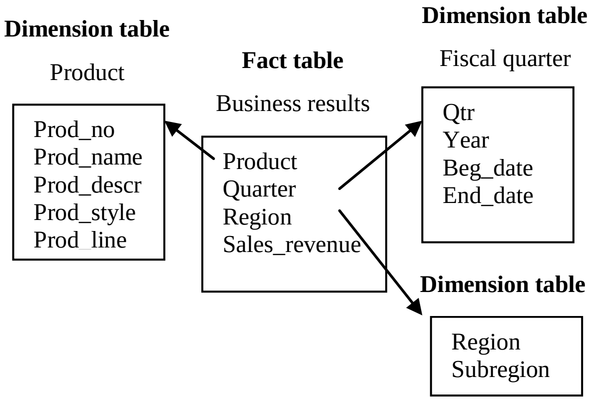

Figure 1 Star Schema for Sales Domain

The above approaches mainly work with relational sources hence they use set of heuristic to derive facts and dimensions. But today, the information systems are dealing with structured, semi-structured and unstructured sources. These sources may lead to heterogeneous problems which may affect the design output. The three heterogeneity issues that normally arise are syntactic, structural and semantic [12]. Resolving syntactic (e.g., product_id and product_no) and structural issues may be done irrespective of the context but semantic issues may be application dependent. In order to represent the concept of a domain irrespective of the application, ontology began to be used. Ontology is a formal specification of an agreed conceptualization of a domain in the context of knowledge description. The use of ontology for the data warehouse design helps to solve the heterogeneity issues that arise in the data sources [13]. The data sources can be represented by means of ontologies and mapping these ontologies can provide integrated view that could help to access and exchange information in a semantically sound manner [14]. Not only does ontology provides the conceptual representation of the domain but are machine processable. Hence it helps to derive the multidimensional schema elements automatically by reasoning. This paper proposes a comprehensive framework using a hybrid methodology to derive multidimensional schema from multiple ontology sources based on requirements. Our approach uses a set of ontology matching algorithms to map requirements with the source. Only the interesting concepts required for analysis are considered in the schema design. Using reasoning algorithms the facts and dimensions are derived automatically.

Section II discusses the related work for the design of the conceptual multidimensional model for the data warehouse. Section III presents the proposed framework Section IV presents the results and discussion V presents conclusion and future work.

2. RELATED WORK

Some of the ontology based automated approaches are discussed here. In [15] the authors have proposed a framework for designing semantic data warehouse. They represent the topic of analysis, measures and dimensions in the requirements. Based on this they derive the MIO (Multidimensional Integrated Ontologies) along with the knowledge from external ontology sources and domain ontologies. The scalability of this approach is that large sized ontologies could be managed. In [16] S^2WRC (Semantic Sources and Requirements driven tool for data Warehouse Conceptual design) a global ontology exists to represent the source. The requirements are represented as ontological query language (OntoQl) which is used to derive the data warehouse ontology from the global one. Here only if the requirements are stated clearly the design can be successful. In [17] AMDO (Automating Multidimensional Design from Ontologies) they use three criteria such as multidimensionality, the multidimensional space arrangement constraint and the summarization integrity constraint in order to carry out the design task from the source ontology. They use basic and generic reasoning algorithms to automatically derive facts and dimensions. Here they consider a single and rigid ontology for their design task. Moreover their approach generates too many results which need to be filtered according to the end-user requirements at posterior. In [18] AMDMM (Automatic method for data warehouse multi-dimension model) they use a hybrid approach for the conceptual design of the data warehouse. They develop an ontology meta-model bottom up from the source and extend the ontology relationships top down from the business requirements. From the meta-model facts and dimensions are derived. The method to derive facts and dimensions are not clearly stated. In [19] the GEM approach represents the requirements and source in xml format. For each requirement the concepts are mapped to the source and tagged. The tagged concepts are annotated with the multidimensional elements. Annotated ontology subset is derived by pruning and checking for path formation. Multidimensional validation is carried out to derive the data warehouse conceptual schema. It also performs ETL design in parallel. This approach achieves a good level of automation. But it requires that the requirements need to be represented in XML format along with concepts identified as facts and dimensions within the requirement.

3. PROPOSED FRAMEWORK

In this paper we propose an automated approach which is an extension of our previous work [20] for supporting multidimensional schema design. In our approach we follow the hybrid methodology where the data source and end-user requirements are conciliated at the early stage of design. This allows us to derive only the entities that are of interest for analysis. The requirements are converted from natural language text to a logical format. The concepts in each requirement are matched to the source ontology. The matched concepts are tagged in the source ontology. Next, the multidimensional elements such as fact and dimensions are automatically derived using reasoning. The proposed framework is represented in Figure 2.

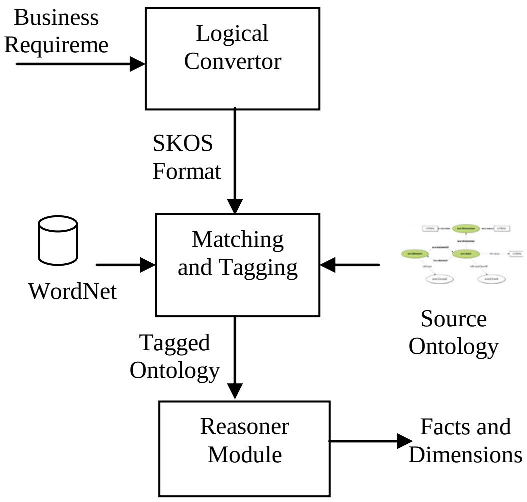

Figure 2 Proposed Framework

Our approach differs from the existing ontology based multidimensional schema design approaches in the following ways: i) The requirements of the data warehouse can be represented in natural-language text format. ii) The Concepts which are of interest to the user for analysis is identified at the early stage of design iii) Deriving multidimensional elements such as facts and dimensions are done automatically using reasoner. The different phases followed in developing the above framework is explained below.

3.1 Representing data source

Today web has become one of the largest sources which may consist of structured (e.g. Database), semi-structured (e.g. XML) and unstructured (e.g. Text) data. Our approach represents each source in ontology (e.g. OWL) format. The need for representing each source in ontology format is that concepts in the domain are well captured using ontology. Another advantage is that any syntactic, structural or semantic conflicts can be easily resolved during integration of the sources. A relational source can be converted to ontology using RDBtoOnto [21] tool. Similarly xml and text sources can be converted to ontology representation using JXML2OWL [22] and OntoLT [23] tools respectively. Using PROMPT [24] an ontology mapping and merging plug-in for Protege tool the local ontologies can be integrated to global ontology. Here we assume that such global ontology representing different sources exists, and our framework takes this global ontology as input which needs to be verified with the requirements.

3.2 Representation of Requirements

The next driving force for data warehouse conceptual design is the end-user requirements. The requirement analysis phase of a data warehouse is different from that of a conventional operating system. In the data warehouse scenario the information requirements can be stated easily by the end-users as it consists of data that is required in the decision making process [11]. Here, we assume that a formal requirement analysis has been carried out earlier and the end-user requirements are stated as information requirements in normal text. For example “Analyze balance and turnover by customer location”, “Analyze balance, turnover and interest by branch” etc., would be the end-user requirements in a banking domain. In order to map the requirements with the global ontology it has to be represented in logical format. Our framework uses logical convertor which takes the requirement as input. This input text is parsed using Stanford Parser and stored in SKOS (*Simple Knowledge Organization System)* format. A Knowledge Organization System (KOS) is a set of elements, often structured and controlled, which can be used for describing (indexing) objects, browsing collections, etc., [24]

3.3 Matching Requirements with Ontology

This phase derives the concepts from the global ontology which are of interest for analysis. This can be achieved by matching the requirements with the concepts in the global ontology. We use string-based, sense-based and gloss-based algorithms [26], [27] for matching the concepts. The string-based matchers used in our framework are: Prefix matcher which checks whether one input string starts with the other one and returns the equivalence relation in this case, Suffix matcher which checks whether one input string ends with the other one and returns the equivalence relation in this case and Edit distance matcher which calculates the edit distance measure between two strings. The calculation includes counting the number of the simple editing operations, such as delete, insert and replace needed to convert one string into another one and dividing the obtained number of operations with max (length (string 1), length (string 2)). If the resultant value exceeds a given threshold the equivalence relation is returned. For further matching we use sense-based matcher that uses the structural properties of the WordNet hierarchies and gloss-based matcher that compares two textual descriptions (glosses) of WordNet senses to produce the relatedness. Since each requirement is represented in SKOS format it can be easily matched with the global ontology in OWL format using the following steps:

1. Each concept in the requirement is matched with the concepts (classes, subclasses and properties) in the global ontology using ontology matching algorithms.
2. For each matching algorithm the similarity between two concepts is calculated using similarity measures such as Levenshteinv, Resnik etc., [27].
3. The similarities between the concepts from requirements and global ontology are represented using SIM (Similarity Assessment Matrix) between i and j elements of the matrix. $SIM := (S_{i,j})n\times m$, $1 < i < n$ and $1 < j < m$. Where, S is the degree of similarity that has been determined by a particular matching algorithm. Table 1 represents a sample of the matrix.
4. Concepts with high similarity values above defined threshold are tagged in the global ontology.

Table 1. Similarity Assessment Matrix

Ontology 1	Ontology 2	S	Matcher
Sales	Salesman	0.68	Sub-String
Article	Publication	1.0	Gloss-Based

3.4 Deriving Facts and Dimensions using Reasoning

The main aim of our approach is to automate the task of identifying facts and dimensions. From the above tagged ontology the multidimensional elements are extracted automatically. We identify a concept as fact if it contains ratio of numerical attributes or number of instances greater than the threshold specified by the designer. The numerical attributes become the measures of the fact. For each fact identified we derive the dimensions by making use of class subsumption and multidimensionality principle that n elements of fact are related to atleast and atmost one element of a dimension through an object property. Finally, we check for levels of each dimension by traversing it recursively. Since a reasoner can compute subsumption (i.e., class A is subsume of class B), identify class taxonomies (i.e., given a class find all its subclasses and superclasses) and property taxonomies (reasoning over properties), we make use of ontology reasoning to compute several steps of our algorithm. The algorithm to compute facts and dimensions is shown below.

```
Algorithm : Compute Facts and Dimensions
Input : Tagged ontology O
Output : Multidimensional elements
Compute_fact (O)
    for each taggedconcept c in O do
        for each dataproperty of c do
            if isnumeric(dataproperty.range) then
                num_list= dataproperty;
                na++;
            else
                nonnum_list= dataproperty;
            end if
        ta++;
        end for
        rna = na / ta;
        ins=count( reasoner.getInstances(c);
        if (rna >  trna or ins > tins) then
            fact_list = c;
            c.measure_list=num_list;
            c.level_list= nonnum_list;
            Print(c,c.measure_list,c.level_list);
            Compute_dimension(c);
        end if
    end for
Compute_dimension(c)
    for each taggedconcept c' in O do
        if (c.subclassOf(c')) then
            if (c. objectproperty allValueFrom c' &&
                maxCardinality = = 1) then
                c.dimension_list =
                c.dimension_list+c';
                end if
            end if
    end for
    Print(c.dimension_list);
    for each concept d in dimension_list do
        Compute_level(dimension_list);
    end for
    //Compute_level
```

```
35      Compute_level(dimension_list)
36          level_list=reasoner.directconcepts(d);
37              if level_list ! = null then
38                  Print(level_list);
39                  Compute_level(level_list);
40              end if
```

Here, O denotes the tagged ontology, c, c' denote concepts available in the ontology O. Step 1–21 of the algorithm computes facts and measures of the given tagged ontology. For each data property of the concept c we compute, i) Ratio of numerical properties (rna) = na / ta, where na is the number of numerical properties for a concept and ta is the total number of data properties for a concept. ii) The total number of instances (ins) for each concept is obtained using reasoner. Concepts with rna > trna or ins > tins are marked as facts. Where, trna and tins are the threshold values for numerical properties and total number of instances respectively. Threshold values for rna and ins can be set by the designer. The numerical properties of the fact are identified as measures (num_list) and non numerical properties (nonnum_list) are identified as a level. From step 22-30, the algorithm computes the dimensions for each fact identified in the previous step. Using reasoner we find concepts involved in a subsumption relationship with fact (i.e., fact c subsume of c'). Here the concepts c' with a many-to-one relationship with fact are identified as dimensions (dimension_list). From step 32-40, we make use of the reasoner to compute the directly related concepts which are identified as levels (level_list). Each dimension is recursively traversed to identify the dimension levels. Our framework uses Jena API [28] and Pellet reasoner API [29] to implement the above algorithm. Generating the multidimensional schema is a straightforward task once facts, measures, dimensions and dimension hierarchies are identified.

3.4 Validating the Multidimensional Elements

The logical and physical schema for the data warehouse can be generated from the above results. The following criteria are used to validate the multidimensional schema generated [15]:

a) Disjointness : Any two dimension concepts belonging to a fact must be disjoint. And levels belonging to the same dimension must also be disjoint. This helps to perform a roll-up or drill down during OLAP operations.
b) Orthogonality : The facts and dimensions are arranged to represent a multidimensional view i.e., each fact instance is related to atleast and atmost one dimension.
c) Summarizability : This implies the functionality of roll-up properties which can be verified by performing some OLAP operations.

4. RESULTS AND DISCUSSION

In this section we illustrate our framework in the car rental domain. EU-Rent is a (fictitious) car rental company, used as a case study in our framework. The business requirements for EU-Rent include the following (A detailed specification of this case study is available at [30]):

a. EU-Rent operates in several countries; in each country it has local areas containing branches
b. EU-Rent rents cars to customers from branches; one-way rentals are allowed
c. Rentals may be booked in advance or "walk-in"
d. Cars are owned by local areas and stored at branches
e. Each car is of a given model; car models are grouped into car groups; all cars in a car group have the same rental tariff
f. Cars are serviced at 5.000 mile intervals

In order to monitor the business the car rental company may need the some performance indicators. For example, each branch must set targets for performance -- numbers of rentals, utilization of cars, turnover, profit, customer satisfaction, etc., If performance targets are not met, control action must be taken. Control action may include: changing the resources at branches (e.g. Number of cars, quotas of cars within each group, number of staff).

To construct a data warehouse for the above car rental company, we assume that the data source is represented as ontology format and a formal requirement analysis has been carried out before our multidimensional schema design. The information requirements are stated in natural language text. For example "Analyze basic price by branch location ", "Analyse basic price and best price by customer location, branch, branch type, rental duration and time" etc., may be the requirements for our domain. Each requirement is parsed as shown in Figure 3 and saved to skos format.

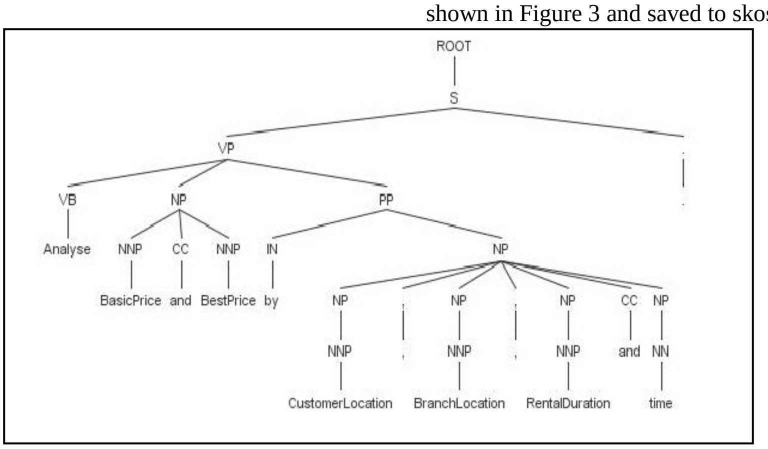

Figure 3 Results after parsing requirement

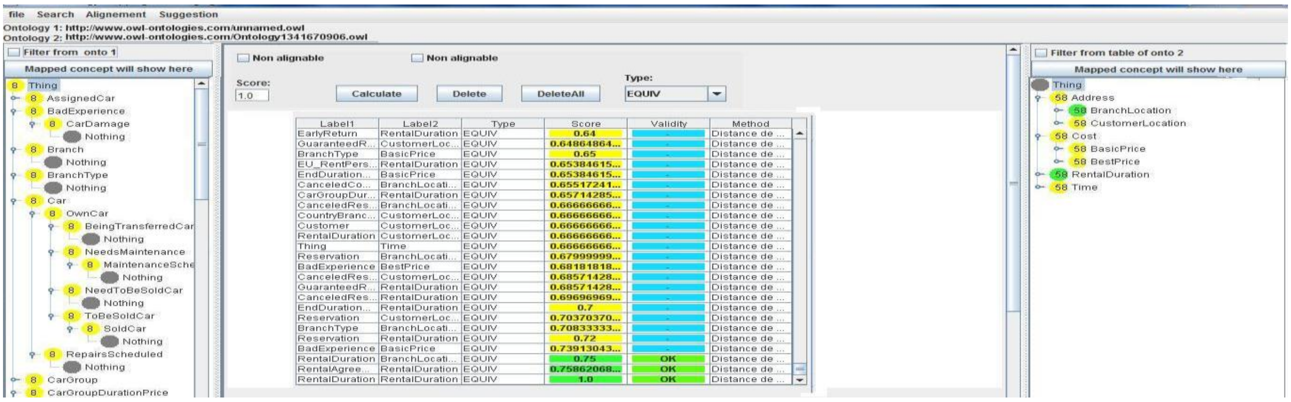

Figure 4 Gloss-based matcher results

In the second phase of our framework we match the requirements with the global ontology. Here we load the EUCarRental ontology and requirements represented in skos format to our mapping and tagging component. Figure 4 shows the results of gloss-based matcher used in our framework. Label 1 represents the concepts from EUCarRental ontology and Label 2 represents the concepts from requirements. The equivalence score after matching is displayed, where we find that RentalDuration from ontology and RentalDuration from requirements have an exact match with a score value as 1 whereas, EarlyReturn and RentalDuration has a score value of 0.64. The concepts with higher score value (e.g >= 0.75) are tagged in the EUCarRental ontology. The concepts Branch, Customer, RentalDuration, Discount, Country, Rental Agreement in EUCarRental ontology are tagged for the above requirement.

Our reasoner module takes the tagged ontology as input and automatically identifies the facts, dimensions, measures etc., Here, the Rental agreement concept is identified as fact as it has rna value greater than the threshold value. The numerical attributes of the rental agreement are basic price and best price which are identified as measures. The Rental agreement is related to concepts such as branch, customer, rental duration and time through a many-to-one relationship which are marked as dimensions. Finally, the designer could use the GUI to add or modify any entities of his choice. Figure 5 represents the multidimensional elements identified for the given requirement. Here we find that as we include new requirements for our domain the number of multidimensional concepts identified increases. Once we found all the interesting concepts for analysis we derive no new concepts, since the information requirements stated may have overlapping concepts. This is illustrated in Figure 6.

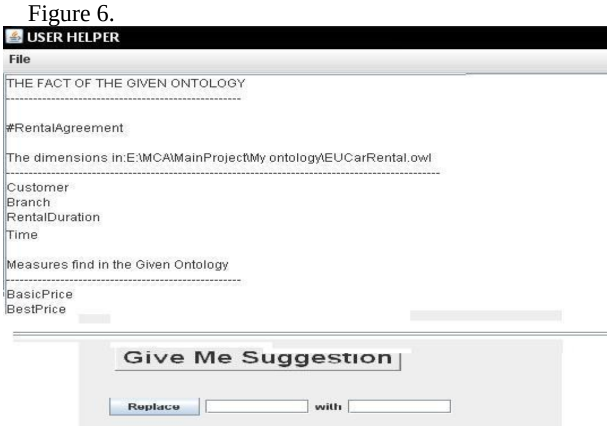

Figure 5 Multidimensional Elements

To implement the above framework we used Java in Net Beans 6.9.1 as a development environment. For working over ontologies and reasoning we used Jena API and pellet reasoner API respectively. Using the results obtained a logical schema can be constructed and it can be validated using the criteria's mention in section 3.5. In our example as the dimensions Customer, Branch, Rental Duration and Time are disjoint sets the schema satisfies disjointness. Analyzing the relationship between fact and dimensions we found that the concepts are orthogonal. Summarizability can be validated after the physical schema is constructed from the above logical schema with OLAP queries containing roll-up or drill down operations.

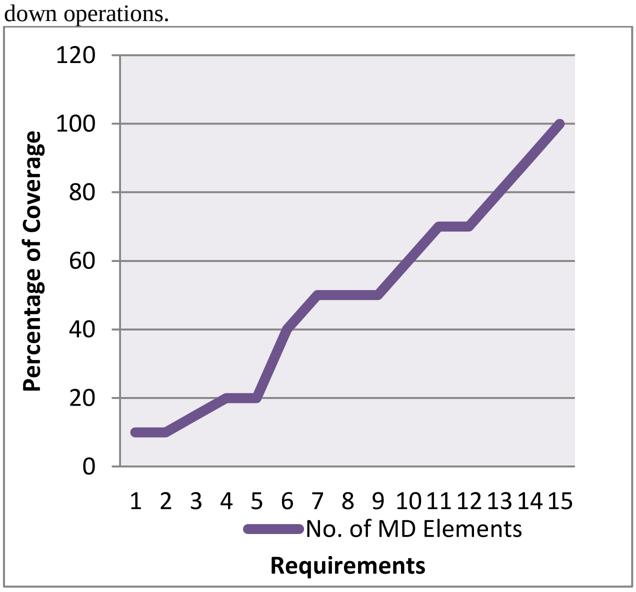

Figure 6 Multidimensional Coverage

Next we compare our proposed approach with existing approaches to multidimensional schema design for the data warehouse. In MIO [15], S^{2}WRC [16] and GEM [19] approach the multidimensional elements such as facts, measures, dimensions etc., is mentioned explicitly in the requirements hence we do not include them in our comparison. We use certain general parameters and parameters specific to multidimensional element identification for comparison. Table 2 represents the comparison of the proposed approach with AMDO [17] and AMDMM [18]. The AMDMM approach follows a hybrid methodology similar to the proposed one by analyzing the source and requirements before the design task. But the steps to identify multidimensional elements are not clearly stated. Similar to the proposed approach the AMDO approach makes use of the reasoner to automate the task of deriving multidimensional elements. But the results are exhaustive as they use a supply driven methodology to fully analyze the source ontology. They need to filter the results manually by means of requirements.

Table 2. Comparison of the Ontology based approaches

Features	**AMDO**	**AMDMM**	**PROPOSED APPROACH**
Hybrid	No	Yes	Yes
Fully Automatic method	No (Semi)	No (Semi)	Yes
Tool	Yes	No	Yes
Fact Identification			
→Numerical Value	Yes	No	Yes
→Connectivity	Yes	No	Yes
→Cardinality	Yes	Yes	Yes
Measure Identification	Yes	No	Yes
Dimension Identification			
→Fact Centered	No	No	Yes
→Functional dependencies	Yes	No	Yes
Level Identification	Yes	No	Yes

5. CONCLUSION

As a multidimensional model plays an important role in the data warehouse design, there is a need to automate the modeling task. The application of ontology for the conceptual design of data warehouses has relieved the burden of the designer to automate the task in a more meaningful way. We propose a more comprehensive framework which involves expressing business requirements in the natural language format; reconcile requirements and global ontology (representing source) to derive interesting concepts and use of efficient reasoning algorithm to extract multidimensional elements from the ontology. As a future work we plan to work over large sized dynamic ontologies in a distributed environment.

6. REFERENCES

[1] Golfarelli M., Maio D., Rizzi S., “The dimensional fact model: a conceptual model for data warehouses”, Int. J. Coop. Inf. Syst. 7 (2–3), 1998, 215–247.

[2] M.R. Jensen, T. Holmgren, T.B. Pedersen, Discovering multidimensional structure in relational data, 6th Int. Conf. on Data Warehousing and Knowledge Discovery, LNCS, vol. 3181, Springer, 2004, pp. 138–148.

[3] D. Moody, M. Kortink, From enterprise models to dimensional models: a methodology for data warehouse and data mart design, Proc. of 2nd Int. Workshop on Design and Management of Data Warehouses, CEUR-WS.org, 2000.

[4] Hüsemann B., Lechtenbörger J., Vossen G., “Conceptual data warehouse modeling, Proc. of 2nd Int. Workshop on Design and Management of Data Warehouses”,CEUR-WS.org, 2000, p.6.

[5] R. Winter, B. Strauch, A method for demand-driven information requirements analysis in DW projects, Proc. of 36th Annual Hawaii Int. Conf. on System Sciences, IEEE, 2003, pp. 231–239.

[6] P. Giorgini, S. Rizzi, M. Garzetti, Goal-oriented requirement analysis for data warehouse design, Proc. of 8th Int. Workshop on Data Warehousing and OLAP, ACM Press, 2005, pp. 47–56.

[7] N. Prat, J. Akoka, I. Comyn-Wattiau, A UML-based data warehouse design method, Decision Support Systems 42 (3) (2006) 1449–1473.

[8] Song, I., Khare, R., & Dai, B. (2007). SAMSTAR: A Semi-Automated Lexical Method for Generating STAR Schemas from an ER Diagram In I. Song, T. B. Pedersen (Eds.), Proceedings of ACM 10th International Workshop on Data Warehousing and OLAP; pp 9-16, Lisbon, Portugal: ACM Press.

[9] Phipps C., & Davis K. C., “Automating Data Warehouse Conceptual Schema Design and Evaluation”. In L. V. S. Lakshmanan (Ed.), Proceedings of 4th International Workshop on Design and Management of Data Warehouses, 2002, pp 23-32, Toronto, Canada: CEUR-WS.org.

[10] J.-N. Mazon, J. Trujillo, J. Lechtenborger, Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms, Data & Knowledge Engineering 23 (3) (2007) 725–751.

[11] Romero, A. Abelló, Automatic Validation of Requirements to Support Multidimensional Design, Data & Knowledge Engineering 69 (2010) 917–942.

[12] M. Gagnon, Ontology-based Integration of Data Sources, 10th International Conference on Information Fusion, Quebec, Canada, 2007.

[13] Jesús Pardillo, Jose-Norberto Mazón, “Using ontologies for the Design of data warehouses”, International Journal of Database Management Systems (IJDMS), May 2011.

[14] Yannis Kalfoglou, and Marco Schorlemmer, Ontology mapping: the state of the art, The Knowledge Engineering Review Journal, Vol 18, No. 1, pp.1-31, 2003.

[15] V. Nebot, R. Berlanga, J. M. Perez, M. J. Aramburu, and T. B. Pedersen. Multidimensional Integrated Ontologies: A Framework for Designing Semantic Data Warehouses. JoDS XIII, 5530:1{35, 2009.

[16] Selma Khouri, Bellatreche Ladje, “A Methodology and Tool for Conceptual Designing a Data Warehouse from Ontology-based Sources”, Ecole nationale Supérieure d’Informatique Algiers, Algeria,2010.

[17] O. Romero, Alberto Abelló: A framework for multidimensional design of data warehouses from ontologies. Data Knowl. Eng. 69(11): 1138-1157 (2010)

[18] Lihong Jiang1, Junliang Xu1, Boyi Xu2, Hongming Cai1. An Automatic Method of Data Warehouses Multidimension Modeling for Distributed Information Systems. In Proceedings of 15th International Conference on Computer Supported Cooperative Work in Design (2011) IEEE, 9-16.

[19] Oscar Romero, Alkis Simitsis, and Alberto Abelló, GEM: requirement-driven generation of ETL and multidimensional conceptual designs. In Proceedings of the 13th international conference on Data warehousing and knowledge discovery (DaWaK'11), Alfredo Cuzzocrea and Umeshwar Dayal (Eds.). Springer-Verlag, Berlin, Heidelberg, (2011) 80-95.

[20] M. Thenmozhi, K. Vivekanandan, A Framework to Derive Multidimensional Schema for Data Warehouse Using Ontology, Proceedings of National Conference on Internet and WebSevice Computing, NCIWSC (2012).

[21] http://sourceforge.net/projects/rdbtoonto/

[22] Toni Rodrigues, Pedro Rosa, Jorge Cardoso. Moving from syntactic to semantic organizations using JXML2OWL. Computers in Industry, 59(8):808-819, 2008.

[23] Paul Buitelaar, Daniel Olejnik, Michael Sintek (2003) OntoLT: A Protege Plug-In for Ontology Extraction from Text Demo Session at the International Semantic Web Conference, Sanibel Island, Florida, USA

[24] Natalya Fridman Noy, Mark A. Musen: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. AAAI/IAAI 2000: 450-455

[25] http://www.w3.org/2004/02/skos/

[26] Giunchiglia, Fausto and Shvaiko, Pavel and Yatskevich, Mikalai. Semantic Matching: Algorithms and Implementation. Technical Report DIT-07-001, Department of Information Engineering and Computer Science, University of Trento. In Journal of Data Semantics (JoDS), IX, 2007.

[27] Jérôme Euzenat, Pavel Shvaiko: Ontology matching. Springer-Verlag, Berlin Heidelberg (DE), 2007: 1-333

[28] http://jena.apache.org/

[29] http://org.mindswap.pellet

[30] L. Fr´ıas, A. Queralt, and A. Oliv´e. EU-Rent Car Rentals Specification. Technical report, ”Dept. de Llenguatges i Sistemes Inform`atics”, 2003. www.lsi. upc.edu/dept/techreps/llistat_detallat.php?id=690.