ONTOLOGICAL WAREHOUSING ON SEMANTICALLY INDEXED DATA

Reusing Semantic Search Engine Ontologies to Develop Multidimensional Schemas

Filippo Sciarrone and Paolo Starace
Business Intelligence Division, Open Informatica srl, Via dei Castelli Romani 12/A, Pomezia, Italy

Keywords: Business intelligence, Data Warehousing, OLAP.

Abstract: In this article we present a first experimentation of a Business Intelligence solution to dynamically develop multidimensional OLAP schemas through a reuse of ontologies, stored in concept and relations dictionaries and used by semantic indexing engines. The particular aspect of the proposed solution consists in the integration of semantic indexing techniques of non-structured documents, based on ontologies, with dynamic management techniques of unbalanced hierarchies in a Data Warehouse. As a case study, we embedded our solution into a real system, built for the analysis and management of experts' curricula in an e-government environment. We show how it is possible to automatically build OLAP dimensions, inheriting the hierarchic structure of ontologies, with the goal of using the semantically indexed data to carry out multidimensional OLAP analyses. The first experimental results are encouraging.

1 MOTIVATIONS AND GOALS

In this article we present a solution for the dynamic management of ontologies in a BI environment. The main features of our proposal concern the development of a process to dynamically extract concepts from a semantically indexed database. Our approach allows us to expand the traditional $On-Line-Analytical-Processing$ (*OLAP*) analysis, in order to design and build dimensions over ontologies. The basic idea of our work is a new methodology, aimed at reusing predefined ontologies in a concept-based dictionary to develop multidimensional *OLAP* schemas.

Dimensions are obtained from the structure of the ontologies in a dynamical way, namely, by defining only the root level of the very ontology and allowing the system to build the cube dimension automatically. Other studies carried out in this field focused on different aspects of this problem. Some aim at extracting schemas without involving the human being. In this context, sometimes ontologies are used to describe the application domain (Simitsis et al., 2008; Skoutas and Simitsis, 2006), to generate mediators (Critchlow et al., 1998) and to semantically describe data sources (Toivonen and Niemi, 2004) to support and automate the definition of $Extract-Transform-Load$ (ETL) processes. In all such cases, the use of ontologies occurs at a lower level in the application architecture with respect to our.

The paper breaks down as follows: Section 2 presents a description of the *Ontology Processing* process, that is, the characteristics of the integrated processes on which this proposal is based, ranging from the working hypotheses, to the treatment of *Bridge Tables*. Section 3 gives a description of the case study through which our proposal was tested. Finally, Section 4 deals with the conclusions and sets the work to be done in future.

2 ONTOLOGY PROCESSING

In this section we present the paramount characteristics of integrated processes supporting the treatment and management of ontologies.

In order to try out our experiment in a real case, we adapted our solution to a pre-existing system, implemented to execute a two-step indexing process of non structured documents in the three layers illustrated in Figure 1. During the first step, from the *unstructured docs* layer to the *terms set* layer, the engine, based on the API of the Open Source search engine *Lucene* [1], indexed each document, thus obtaining a set of index-terms. In the second step, from the *terms set* layer

[1] http://lucene.apache.org

Sciarrone F. and Starace P. (2009).
ONTOLOGICAL WAREHOUSING ON SEMANTICALLY INDEXED DATA - Reusing Semantic Search Engine Ontologies to Develop Multidimensional Schemas.
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, pages 315-318
DOI: 10.5220/0002307103150318

to the *ontologies* layer, these terms were contextualized and associated with the concepts of predefined ontologies.

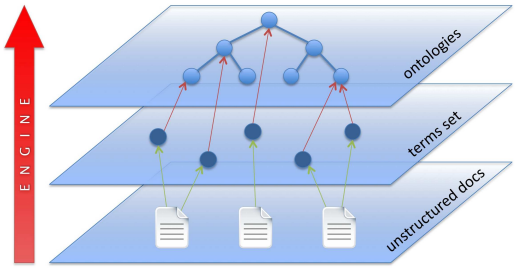

Figure 1: The Two-Steps Indexing Process.

In order to integrate our module with the aforesaid system, the following assumptions were imposed:

- The concepts included in the dictionary were exclusively linked by hypernymy and hyponymy relations;
- Each ontology was based on a hierarchic structure.

The aforesaid restrictions obviously entailed an experimentation that was to be limited to the context, albeit it could also go for other real cases.

In the case in point, an ontology is therefore to be considered as the number of non uniform levels, whose number may change and, above all, is not known beforehand. In the development of our solution it was deemed crucial to make sure that the ontological tree be extracted dynamically, so as to make the use of concepts as dimensions flexible and easily adaptable. Representing an arbitrary and irregular hierarchy is an intrinsically hard task in a relational environment. The adopted solution envisages the inclusion of a bridge table between the concept dimension and the facts table. In literature, Kimball suggests this method to manage the dimensions that recursively refer to records on the same table (Kimball et al., 1998; Kimball and Ross, 2002). The goal of the bridge table was to help the *OLAP* engine aggregate data more quickly.

Dynamic dimensions are generated starting from the tree's root concept. This is a central idea in the project we are presenting. The goal is to uncouple the user from the manual definition of hierarchies, placing him/her on a higher level in the application architecture. By doing so, the user may actually not know the logical structure of the hierarchy because the latter is defined automatically.

Figure 2 shows the entire process performed by the overall system. In particular we developed a custom *ETL* module in order to integrate semantically in-

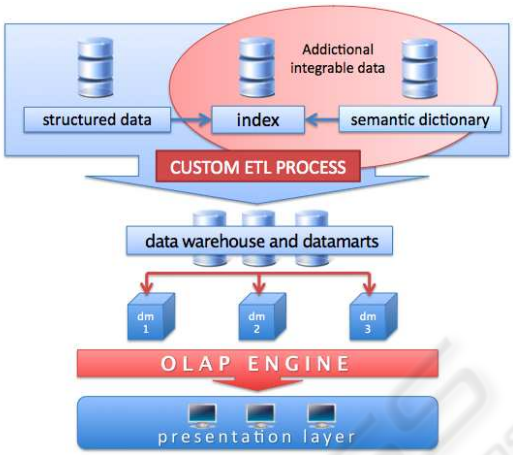

Figure 2: The Overall Process.

dexed data with operational ones. Particular attention is to be given to the management of the ontologies' structure. In order to ensure a hierarchic navigation it is necessary to bring it back to a tree structure. The presence of navigable cycles on the structures is ruled out - even theoretically - from the typology of relation existing between the concepts, i.e., the *part of* relation. One must therefore consider the management of *Directed Acyclic Graphs* (*DAG*).

3 CASE STUDY

In this section we illustrate the Case Study, namely, the integration of our module with the pre-existing system and the *OLAP* engine. The proposed so-

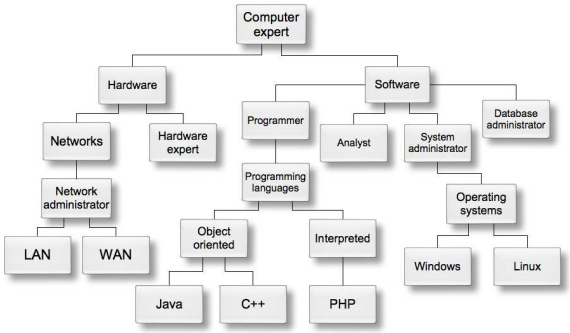

Figure 3: The Computer Expert Ontology.

lution was implemented successfully in an experimentation within a corporate context for the analysis and semantic search of curricula of experts, through data generated by a web application used in the e-governance field. The web application indexed the non-structured textual information contained in the

curricula uploaded by candidates submitting their application for jobs offered by an Italian government body. The application's function was to retrieve the information stored in these text documents, in order to create a complete profile for the worker and make it easier to be found. In this context we chose to define a star-schema to show the application of the suggested study. The experimentation parameters are the ones shown in Table 1. We must point out that the system works off-line with respect to the user: in this way all the parameters shown in the Table do not affect the on-line performance.

Table 1: The Experimental Parameters.

Variable	Value
# curricula	150,000
# ontologies	10
# concepts per candidate	>10
ontology extraction avg. time	10"
index cleaning avg. time	5'

3.1 An Ontology Star-schema Example

The ontology to be integrated in the schema was one partially defined in a manual way (likewise, it could be possible to choose a predefined one in the semantic dictionary), whose contents provide a description of the *IT expert* concept. The operational database provides contents regarding the candidates' personal, working and academic data. We therefore thought it would be interesting to show the relations existing between such data and a concept regarding a working environment such as the chosen one. The structure of the *IT expert* concept ontological tree is shown in Figure 3. Consistently with the limitations mentioned in the previous paragraphs, the ontological graph results in an oriented n-ary tree. To highlight the use of the ontology, the star schema we chose to implement involves this dimension only. The resulting schema is shown in Figure 4. The measure on which the aggregation is to be made is the number of candidates referring to the single concept. The resulting table will therefore be a factless fact table (Kimball and Caserta, 2004). In this situation it is evident that managing *many-to-many* relations is a crucial aspect, because a superficial management of the issue inevitably leads to an inconsistent result. Going further into detail, the aforesaid problem is illustrated by referring to two concepts stemming from the same parent concept. If two stemming concepts were to simultaneously refer to the same candidate, aggregating them to the parent concept level would lead to wrong value. This is the typical problem we come across in data warehousing (Song et al., 2001; Golfarelli et al., 1998). Our proposal doesn't drift away from the issue. The problem is neither worsened nor dampened, all its aspects are inherited. It does occur more frequently however, because the tackled issues can be easily affected in this regard. In fact, when addressing concepts such as hierarchic dimensions, the use of many-to-many relations is frequent. The answer to the problem may be searched, for example, in the weighted management of the graph nodes (Song et al., 2001). The problem does not exist if the concepts are on different levels in the same branch, since it is possible to remove the records that refer to the upper levels, thus maintaining the result consistent with the interrogation level. The dimensions that may be aggregated, both standard and dynamic ones, are not number-limited, and this makes the solution adaptable to any type of problem.

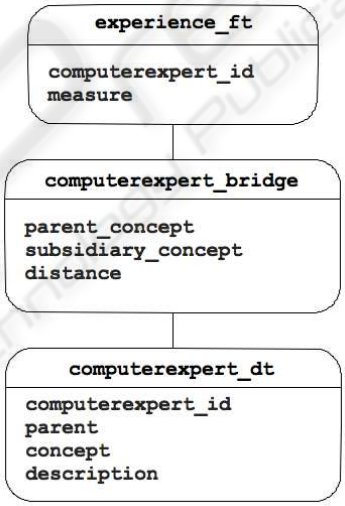

Figure 4: The Star-Schema with Bridge Table.

3.2 The *OLAP* module

The execution of the *ETL* process is ended by the generation of a table of facts with a dimension stemming from the *IT expert* concept. For its interrogation we chose to use the OLAP MONDRIAN engine of PENTAHO Business Intelligence Suite[2]. It was therefore necessary to create the required structures for interaction with the specific interrogation tool which, aside from the specific choice made in this example, can be chosen without any particular restrictions. Regardless of the choice, it is best to manage the creation of structures through a specifically-made solution, because necessities drift from the standard use of the tools. The PENTAHO suite presents the data

[2]Pentaho Business Intelligence Suite - http://www.pentaho.org

	Measures
EspertoInformatico	Esperto
+All EspertoInformatico	2.412
−hardware	429
tecnico hardware	33
− rete rete_di_computer reti reti_di_computer	349
cyberspazio internet	31
rete web www	51
+Network Administrator (Amministratore di rete)	121
LAN	39
WAN	45
−software	1.942
+Amministratore di Sistema	368
−Database administrator	205
+linguaggi di programmazione (database)	100
DBMS	31
Database	33
+ esperto_informatico programmatore programmatore_di_computer	1.252
Analista software	36
Project manager	40

Figure 5: The Resulting Pivot Table.

processed by the *OLAP* MONDRIAN engine through jsp libraries that generate pivot tables for the navigation of multidimensional cubes, making roll-up and drill-down operations. The result obtained from the navigation of the schema illustrated in the previous paragraph is shown in Figure 5. In the image it is possible to notice that the quality of the ontologies contained in the dictionary is the basic element for a good performance of the presented information. Therefore, the names in the attribute field have deliberately not been refined, so as to highlight this aspect.

4 CONCLUSIONS AND FUTURE WORK

In this article we have proposed a solution for the integration of indexing data generated by semantic search engines and the re-use of ontologies defined in their dictionaries as *OLAP* dimensions. The goal is that of dynamically developing multidimensional schemas for *BI* applications regarding ontologies. We use this technology to simplify the management of ontology-based information and reduce, without bringing to zero, human involvement. We could consider the idea of implementing the studies mentioned in Section 1 to enhance the process generating ontology-based dimensions. The defined process is currently stable and yields positive results in a company environment. Finally, the fact-definition process can be improved, extending the logic to the ”join” base of the data. In order to provide a complete *BI* service, the system must be able to make several types of aggregations, not just the basic ones. As future work we plan the enhancement of indexed data management, with the introduction of a *Cache-Based* engine, and on the resolution of problems related to the management of many-to-many relations.

REFERENCES

Critchlow, T., Ganesh, M., and Musick, R. (1998). Automatic generation of warehouse mediators using an ontology engine. In Borgida, A., Chaudhri, V. K., and Staudt, M., editors, *KRDB*, volume 10 of *CEUR Workshop Proceedings*, pages 8.1–8.8. CEUR-WS.org.

Golfarelli, M., Maio, D., and Rizzi, S. (1998). Conceptual design of data warehouses from e/r schema. In *HICSS ’98: Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences-Volume 7*, page 334, Washington, DC, USA. IEEE Computer Society.

Kimball, R. and Caserta, J. (2004). *The Datawarehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming and Delivering Dasta*. Wiley.

Kimball, R., Reeves, L., Thornthwaite, W., Ross, M., and Thornwaite, W. (1998). *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses with CD Rom*. John Wiley & Sons, Inc., New York, NY, USA.

Kimball, R. and Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition)*. Wiley.

Simitsis, A., Skoutas, D., and Castellanos, M. (2008). Natural language reporting for etl processes. In *DOLAP ’08: Proceeding of the ACM 11th international workshop on Data warehousing and OLAP*, pages 65–72, New York, NY, USA. ACM.

Skoutas, D. and Simitsis, A. (2006). Designing etl processes using semantic web technologies. In *DOLAP ’06: Proceedings of the 9th ACM international workshop on Data warehousing and OLAP*, pages 67–74, New York, NY, USA. ACM.

Song, I.-Y., yeol Song, I., Medsker, C., Ewen, E., and Rowen, W. (2001). An analysis of many-to-many relationships between fact and dimension tables in dimensional modeling. In *Proc. of the Intl Workshop on Design and Management of Data Warehouses*, pages 6–1.

Toivonen, S. and Niemi, T. (2004). Describing Data Sources Semantically for Facilitating Efficient Creation of OLAP Cubes. In *Poster Proceedings of the Third Interntional Semantic Web Conference*.