PROCEEDINGS

INTERNATIONAL WORKSHOP ON

Ontologies-based techniques for DataBases and Information Systems

Workshop Co-Chairs:
Martine Collard and Jean-Louis Cavarero

CO-LOCATED WITH THE 31ST INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES

Table of contents

Invited Talk

Conceptual Knowledge Processing

Gerd Stumme, University of Kassel

Abstract:
Knowledge processing mainly deals with the organizational and technical support of knowledge processes. Important activities of knowledge processing are discovering and structuring knowledge, derivation of new knowledge, and communication of the knowledge. Since human thinking is conceptually organized, a major issue in knowledge processing is a semantics-based support of these tasks. In the presentation, the exploitation and generation of conceptual knowledge in the related research areas Knowledge Discovery, Information Retrieval, and Peer to Peer is discussed by three examples: (i) conceptual clustering with background knowledge, (ii) conceptual email management, (iii) semantic-based routing in peer to peer systems.

Gerd Stumme is Full Professor of Computer Science, and is leading the research unit on Knowledge and Data Engineering at the University of Kassel. He earned his PhD in 1997 at Darmstadt University of Technology, and his Habilitation at the Institute AIFB of the University of Karlsruhe in 2002. In 1999/2000 he was Visiting Professor at the University of Clermont-Ferrand, France, and Substitute Professor for Machine Learning and Knowledge Discovery at the University of Magdeburg in 2003.

Gerd Stumme published over 80 articles at national and international conferences and in journals, and chaired several workshops and conferences. He is member in the Editorial Boards of the Intl. Journal on Data Warehousing and Mining and of the International Conference on Conceptual Structures, and was also member of several conference and workshop Program Committees. Gerd Stumme led several national and European projects, eg, the current national project "Personalized Access to Distributed Learning Resources (PADLR)". Additionally he is co-chairing the Web Mining Forum of the European Network of Excellence KDNet. More information on Gerd Stumme can be found at <http://www.kde.cs.uni-kassel.de/index_en.html>.

Using WordNet Ontology to automatically enrich dimension hierarchies in a data warehouse

Jose-Norberto Mazón, Juan Trujillo

Dept. of Software and Computing Systems
University of Alicante, Spain
Apto. Correos 99. E-03080
{jnmazon,jtrujillo}@dlsi.ua.es

Manuel Serrano, Mario Piattini

Alarcos Research Group
University of Castilla-La Mancha
Paseo Universidad, 4; 13071 Ciudad Real, Spain
{Manuel.Serrano,Mario.Piattini}@uclm.es

Abstract

OLAP (On-Line Analytical Processing) operations, such as roll-up or drill-down, depend on data warehouse dimension hierarchies in order to aggregate information at different levels and support the decision-making process required by final users. However, operational data could not be enough for supplying information to construct adequate hierarchies, which have enough aggregation levels. In this paper, we apply knowledge given by relationships among concepts from WordNet Ontology to overcome this problem. Therefore, more complete dimension hierarchies will be specified in the data warehouse, and OLAP tools will be able to show proper information to improve decision-making process. Finally, we will show the benefits of our approach by providing a case study in which a poor hierarchy is enriched with new levels of aggregation.

1. Introduction

According to Inmon's definition [8], a data warehouse (DW) is "a subject oriented, integrated, non-volatile, and time variant collection of data in support of management's decision". In order to support the decision-making process, OLAP (On-Line Analytical Processing) tools are commonly used. These tools allow users to query DW by analyzing its large amount of data. In this analysis, operations such as roll-up or drill-down are used to aggregate or disaggregate data, depending on levels of aggregation which must be explicitly specified by organizing the members of a given dimension into hierarchies [2,9,12,15,21,22,26]. Thus, hierarchies must be properly defined for analyzing data stored in DW according to user requirements in order to improve the decision-making process. In fact, the richer a hierarchy is defined, the more meaningful users' queries will be answered and the better decisions will be made.

Based on our experience on designing DWs [13,14], we consider that the right way of defining hierarchies is as follows: from user requirements we use conceptual modeling approach to build a conceptual schema. Then we use operational data sources to complete this first version of the conceptual schema (of course, from now on we would proceed with the following design stages such as logical and physical design). Nevertheless, even though we use operational data sources to complete hierarchies, we found that these hierarchies could not be specified as many terms and data are missed in the operational data sources. Consequently, operational data could be not enough for constructing adequate hierarchies, and some kind of guidance would be appropriate to provide mechanisms to enrich them. A dimension hierarchy is enriched by adding levels of aggregation to accomplish with information analysis requirements and improve decision-making process.

In this paper we present an approach (see figure 1) to automatically complete hierarchies using relationships among concepts provided by an ontology. The reason is that dimension hierarchies are derived from abstraction processes that represent semantic relations between values, like generalization ("is-a-kind-of" or hypernym) or aggregation ("is-a-part-of" or meronym) [1,2,12,15,21, 22,23]. In our approach, WordNet is used like an ontology (using its hierarchy of concepts) because (i) it provides concepts from many domains, (ii) it can be easily extended to other languages, apart from English, by means of EuroWordNet [28], and (iii) it presents relations between concepts which are easy to understand and use.

The benefit of our proposal is clear: using knowledge provided by an ontology to improve quality of dimension hierarchies by means of adding new hierarchy aggregation levels, which allow DW users to achieve their analysis information needs and, in this way, to better support the decision-making process.

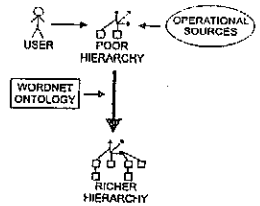

Figure 1. Applying WordNet to obtain a richer hierarchy.

The remain of this paper is structured as follows. Section 2 presents an overview of works about WordNet and ontologies. Section 3 describes our approach for modeling DWs and their dimension hierarchies based in UML (Unified Modeling Language). Section 4 overviews WordNet Ontology. Section 5 defines our approach for enriching dimension hierarchies using WordNet Ontology. In section 6, a case study is presented. Finally, in Section 7 we point out our conclusions and sketch some future works.

2. Related work

Traditionally, WordNet has been used to improve natural language processing systems. It has supported several kinds of tasks, such as information retrieval and extraction, document structuring and categorization, etc. A comprehensive review of applications related to WordNet can be found in [19].

On the other hand, within multidimensional environments, linguistic knowledge provided by ontologies has been specially used for data integration. Kedad and Métais [10] use linguistic knowledge during the process of data cleaning in multisource information systems to solve terminological conflicts between data instances. Toivonen and Niemi [27] present an ontology-based method to find suitable data from different sources and to semantically integrate them into one OLAP cube. A review of the use of ontologies for data integration can be found in [29]. For a more general review, we refer reader to [3].

Furthermore, several works [2,9,15] have paid attention to the importance of dimension hierarchies in multidimensional modeling. However, to the best of our knowledge, our contribution is the first work about employing ontologies for improving the design of dimension hierarchies in DWs.

3. Data warehouses, dimensions, and their hierarchies

Multidimensional databases, OLAP applications, and DWs provide companies with many years of historical information for the decision-making process. It is widely accepted that these systems are based on multidimensional modeling which structures information into facts and dimensions. A fact contains interesting measures (*fact attributes*) of a business process (sales, deliveries, etc.), whereas a dimension represents the context for analyzing a fact (product, customer, time, etc.) by means of dimension attributes hierarchically organized. A set of fact measures is based on a set of dimensions that determine the granularity adopted for representing facts.

In this paper we follow our object oriented (OO) conceptual model from [12,26]. This approach has been specified by means of a UML profile that contains the necessary stereotypes in order to carry out conceptual modeling successfully [12]. The structural properties of multidimensional modeling are represented by means of a UML class diagram [20] in which the information is clearly organized into facts and dimensions represented by means of fact classes and dimension classes, respectively.

Fact classes are defined as composite classes in shared aggregation relationships of n dimension classes. The minimum cardinality in the role of the dimension classes is 1 to indicate that all the facts must always be related to all the dimensions. The relations "many-to-many" between a fact and a specific dimension are specified by means of the cardinality 1...* in the role of the corresponding dimension class. In our example in figure 2, we can see how the *Sales* fact class has a many-to-many relationship with the *Product* dimension.

A fact is composed of measures or fact attributes. By default, all measures in the fact class are considered to be additive. For non-additive measures, additive rules are defined as constrains and are included in the fact class. Furthermore, derived measures can also be explicitly represented (indicated by /) and their derivation rules are placed between braces near the fact class.

Our approach also allows the definition of identifying attributes in the fact class (stereotype OID). In this way, degenerated dimensions can be considered [11], thereby representing other fact features in addition to the measures for analysis. For example, we could store the ticket number (*ticket_number*) as degenerated dimension, as reflected in figure 2.

Regarding dimensions, there are two kinds of hierarchies: classification hierarchies, represented by association relationships, and categorization hierarchies, represented by means of generalization relationships.

Classification hierarchies defined on certain dimension attributes are crucial because the subsequent data analysis will be addressed by these hierarchies. A dimension attribute may also be aggregated (related) to more than one hierarchy, and therefore multiple classification hierarchies and alternative path hierarchies are also relevant. For this reason, a common way of representing and considering dimensions with their

classification hierarchies is using Directed Acyclic Graphs (DAG). Nevertheless, classification hierarchies are not so simple in most of the cases. The concepts of "strictness" and "completeness" are important, not only for conceptual purposes, but also for further steps of multidimensional modeling. "Strictness" means that an object of a lower level of hierarchy belongs to only one of a higher level, e.g. a *city* is only related to one *state*. "Completeness" means that all members belong to one higher class object and that object consists only of those members. For example, suppose we say that the classification hierarchy between the *state* and the *city* levels is "complete". In this case, a *state* is formed by all *cities* recorded and all the *cities* that form the *state* are recorded. In our DW conceptual model, each level of a classification hierarchy is specified by a base class (see figure 2). An association of base classes specifies the relationship between two levels of a classification hierarchy. The only prerequisite is that these classes must define a DAG rooted in the dimension class. Due to the flexibility of UML, non-strict hierarchies and complete hierarchies can be also considered by means of the cardinality of the roles of the associations. See figure 2 for an example of all kinds of classification hierarchies.

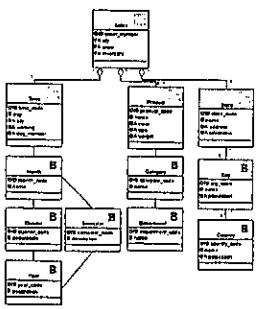

Figure 2. Example of an Object Oriented DW conceptual model using UML.

Lastly, categorization hierarchies are useful when OLAP scenarios become very large as the number of dimensions increases significantly. This fact may lead to extremely sparse dimensions and data cubes. In this way, there are attributes that are normally valid for all elements within a dimension while others are only valid for a subset of elements. For example, attributes *number of passengers* and *number of airbags* would only be valid for *cars* and will be *"null"* for *vans*. In our DW conceptual model, categorization hierarchies are considered by means of the generalization/specialization relationships of UML.

Once the structure of multidimensional model has been defined, final users require fulfilling a set of initial analysis requirements as a starting point for the subsequent analysis phase. From these initial requirements, users can apply a set of operations (OLAP operations) to the multidimensional view of data for further analysis. OLAP operations related to dimension hierarchies are usually as following: roll-up (increasing the level of aggregation) and drill-down (decreasing the level of aggregation) along one or more classification hierarchies.

4. Hierarchies in WordNet

WordNet [16] is a linguistic resource that provides lexical information about words and their senses. Furthermore, WordNet also provides a variety of semantic relations which are defined between concepts [17], so it can be used like ontology. Syntactic category of each word determines its potential semantic relationships. In this paper, we focus on noun semantic relations (since dimension attributes are usually nouns) namely:

- Synonymy: it is a symmetric relation between word forms. It is a similar relationship: synonymy indicates that two concepts have a similar meaning. For example: *pipe* and *tube* are synonyms.
- Antonymy: it is also a symmetric relation between word forms. It is an opposite relationship: antonymy indicates that two concepts have an opposite meaning. For example: *hell* and *heaven* are antonyms.
- Hyponymy/Hypernymy: they represent transitive relations between concepts. It is a subtype/supertype relationship. Giving two concepts X and Y, it is expressed as X is-a-kind-of Y, where X is a more specific concept (hyponym) and Y is a more generic concept (hypernym). An example: *cake* is-a-kind-of *baked goods*. In figure 3, an example of a more comprehensive hypernym hierarchy is given: *chocolate cake* is-a-kind-of *cake*, which is-a-kind-of *baked goods*, which is-a-kind-of *food*.
- Meronymy/Holonymy: they are complex semantic relations, such as components parts, substantive parts, and member parts. They are whole-part relationships. Giving two concepts X and Y, it is expressed as X is-a-part-of Y, where X is a concept that represents a part (meronym) of whole concept Y (holonym). For example: *wheel* is-a-part-of *car*.

chocolate cake -- (cake containing chocolate)
=> cake -- (made from or based on a mixture of flour and sugar and eggs)
=> baked goods -- (foods (like breads and cakes and pastries) that are cooked in an oven)
=> food -- (any solid substance (as opposed to liquid) that is used as a source of nourishment; "food and drink")

Figure 3. Hypernym hierarchy for *chocolate cake*.

These semantic relations allow us to organize concepts into hierarchical structures (an example of a hypernym hierarchy is shown in figure 3). In particular we are interested in hypernymy ("is-a-kind-of" or generalization) and meronymy ("is-a-part-of" or aggregation) relations between nouns; since, they are the most useful relationships in a dimension hierarchy [1,2,12,15,22].

5. Enriching dimension hierarchies

Dimension hierarchies in DWs show the relationships between domains of values from different dimension attributes (set in levels of aggregation). As above-described, WordNet also presents hierarchy relationships between concepts, such as hypernymy/hyponymy and meronymy/holonymy. Thereby, we will use this hierarchical organization of WordNet to automatically complete dimension hierarchies.

We focus on the dimension hierarchy definition provided by [12], described in section 3. Since UML is used for designing a DW, hierarchies are modeled by using UML relationships. Particularly for classification hierarchies we use associations (including aggregations) between levels and generalizations for categorization hierarchies. For generalization we will use hypernymy/hyponymy relationship provided by WordNet. Association relationship from UML is more general, since it only specifies that two elements are connected. Thus, we will use hypernymy/hyponymy or meronymy/holonymy relationships from WordNet depending on the domain of dimension attributes: if an association is considered as an aggregation then we use meronymy/holonymy, else we use hypernymy/hyponymy. For example, in the case of the hierarchy *city-state-country*, we will use meronymy/holonymy relationship due to the fact that *city* is a part of *state* and *state* is a part of *country* (e.g. *Boston* is a part of *Massachusetts* and *Massachusetts* is a part of *USA*). However, if hierarchy is *product-family-class*, hypernymy/hyponymy relationships will be used, because of every *product* is a kind of *family* and every *family* is a kind of *class* (e.g. *cake* is a kind of *baked good* and *baked good* is a kind of *food*).

For the sake of clarity when explaining our proposal, from now on, we assume that only strict hierarchies are taken into account. So, non-strict hierarchies (multiple and alternate path hierarchies) are not considered. It can be assumed because of WordNet restrictions regarding relationships, since there is usually only one hypernym for each word sense [4,16].

Our approach consists of grouping word senses whose hypernyms/meronyms are equal, into a new set of word senses. This new set corresponds to a level of a dimension hierarchy. Each set of senses is described by its common hypernym/meronym. In order to create another level in a hierarchy, grouping again into hypernym/meronym senses (by its common upper concept) is required until the needed level of aggregation is achieved. Before starting, word senses must be disambiguated to obtain the right sense for each one. For disambiguation we have used the specification marks WSD (Word Sense Disambiguation) algorithm from [18], since it offers good results when every word for disambiguating belongs to the same domain. We assume that every possible value of a certain dimension attribute belongs to the same domain. For example, all possible values of the attribute *city* will be *names of cities*.

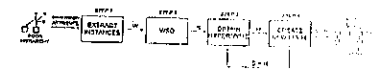

Figure 4. Overview of our approach

Following, we explain the main steps of our approach (an overview is shown in figure 4):

Prerequisite 1. A dimension attribute is chosen. The dimension hierarchy will be specified starting from the attribute.

Prerequisite 2. A level of aggregation (called n) is specified. This is the number of aggregation levels required to properly analyze data from the DW.

Prerequisite 3. Create a variable t. This variable increments its value when a new level of aggregation is created. It must be initialized: $t=0$.

Step 1. Extract all values (without repeating any value) from chosen dimension attribute. These values are nouns and they constitute the input (or context) for specification marks WSD algorithm:

$W=\{w_1, w_2, ..., w_m\}$; *where w_i denotes every value of the selected dimension attribute*

Step 2. For each word in W, we have to obtain its correct senses from WordNet using specification mark WSD. Here s_i represents the correct sense for context value w_i.

$S=\{s_1, s_2, ..., s_m\}$; *s_i is the sense of w_i*

Step 3. For each sense in S, we obtain one hypernym/meronym (only the lowest one) as h_i.

$H_t=\{h_i\}$ $\forall s_i \in S$, *h_i is the lowest hypernym/meronym of s_i*

The set of every hypernym/meronym senses obtained from every H_t without repeating is also formed:

$H=\{h_1, h_2, ..., h_n\}$

Step 4. A new hierarchy level is created and every hypernym/meronym sense from H_t is added as instance.

Step 5. Take new input values as all hypernym/meronym senses: $S=H$.

Stop condition. $t=t+1$. If the required level of aggregation is reached ($t=n$) or S has only one element (all input attributes already have a common hypernym/meronym), then maximum level of aggregation has been reached for these input values. Otherwise, go to step 3.

In figure 4, every step of our approach is illustrated. From a dimension in multidimensional model which not

accomplish with users requirements because its hierarchy does not have enough levels of aggregation, a dimension attribute is chosen and all its values form the context for specification marks WSD in order to obtain right senses. for each value of dimension attributes. Afterwards, iterations start to obtain hypernyms/meronyms of values, a new level of the dimension hierarchy is created, and values are mapped into this new level of hierarchy. Iterations are repeated until a richer hierarchy, with every level of aggregation required, is obtained.

6. Case study

In this section, we will show the benefits of our approach by providing a little case study in which a poor hierarchy is enriched with new levels of aggregation. Our case study consists of a retail sales business composed of several grocery stores spread over several regions. In each store several products are sold. This business process deals with analyzing what quantity of products are selling in which stores on what date. The store manager needs to further study these sales, analyzing them by means of several levels of aggregation (e.g. user needs to analyze the sales aggregating by classes of product). However, only name of the product is available in the operational sources (see table 1), so the original hierarchy provided by these sources, can only consist of one level: *product*. According to the DW conceptual model overview in section 3, a multidimensional class diagram is built from user requirements according to available operational sources (see figure 5). Dimension hierarchies are shown in this class diagram. We can see that product dimension has not enough levels to accomplish with user requirements.

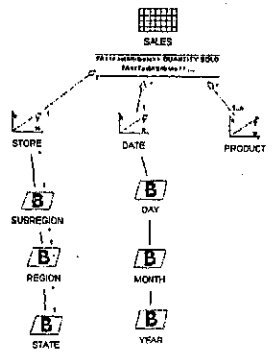

Figure 5. Case study's initial multidimensional class diagram.

Since user requirements are not achieved by this hierarchy, we apply the approach above described to introduce new levels in the dimension hierarchy and enrich it. Original hierarchy consists of an aggregation level, named *product* (see figure 5). However, user needs to aggregate data in three more levels: a lower level called *subtype*, an intermediate level called *type* and a higher level called *class* (see figure 6). Three new levels have to be added to the original hierarchy in order to enrich it. We consider that the user knows the semantic of each level, so levels will be denoted as *level 1*, *level 2*, and *level 3*.

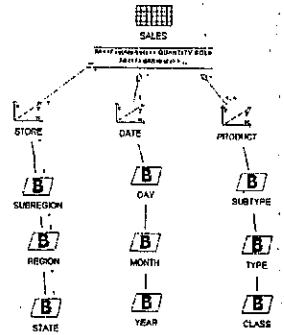

Figure 6. Multidimensional class diagram which accomplish with user requirements.

Now, every step to obtain the final hierarchy from the original one is described:

Prerequisite 1. Dimension attribute *product* is chosen (see table 1).

Prerequisite 2. Three new levels of aggregation are required, so $a=3$.

Prerequisite 3. $t=0$.

Quantity	*Product*	Date
2	*Bourbon*	17/01/2002
3	*Merlot*	01/02/2002
2	*Chardonnay*	03/02/2002
2	*Cabernet*	10/01/2002
1	*Scotch*	09/02/2002

Table 1. Some of data stored in operational source.

Step 1. Input values are the following:
W={bourbon, merlot, chardonnay, cabernet, scotch}

Step 2. For each word in the input, its correct senses using specification marks WSD are the following:
S={s1, s2, s3, s4, s5}={bourbon#2, merlot#2, chardonnay#2, cabernet#1, scotch#1}

Step 3. Hypernyms for each sense of *S* are obtained from WordNet (only the lowest hypernym for each sense).
$H_{bourbon\#2}$={whisky#1}, $H_{merlot\#2}$={red wine#1}, $H_{chardonnay\#2}$={white wine#1}, $H_{cabernet\#1}$={red wine#1}, $H_{scotch\#1}$={whisky#1}
H={whisky#1, red wine#1, white wine#1}

Step 4. Level 1 is added (see table 2).

Product	*Level 1*
Bourbon	*Whisky*
Merlot	*Red wine*
Chardonnay	*White wine*
Cabernet	*Red wine*
Scotch	*Whisky*

Table 2. First hierarchy level created and its mapped values.

Step 5. Definition of new values for S:
S=H={whisky#1, red wine#1, white wine#1}
Stop condition. *t=1. t<a*, then go to *step 3*.

Step 3. Hypernyms for each sense of *S* are obtained:
$H_{whisky\#1}$={liquor#1}, $H_{red\ wine\#1}$={wine#1}, $H_{white\ wine\#1}$={wine#1}
H= {liquor#1, wine#1}

Step 4. Level 2 is added (see table 3).

Step 5. S=H={liquor#1, wine#1}.

Stop condition. *t=2. t<a* then go to *step 3*.

Step 3. Hypernyms for each sense of *S* are obtained
$H_{liquor\#1}$={alcohol#1}, $H_{wine\#1}$={alcohol#1}
H={alcohol#1}

Step 4. Level 3 is added (see table 4).

Step 5. S=H={alcohol#1}.

Stop condition. *t=3=a* then *stop*.

Product	*Level 1*	*Level 2*
Bourbon	*Whisky*	*Liquor*
Merlot	*Red wine*	*Wine*
Chardonnay	*White wine*	*Wine*
Cabernet	*Red wine*	*Wine*
Scotch	*Whisky*	*Liquor*

Table 3. First and second levels created and its values.

Product	*Level 1*	*Level 2*	*Level 3*
Bourbon	*Whisky*	*Liquor*	*Alcohol*
Merlot	*Red wine*	*Wine*	*Alcohol*
Chardonnay	*White wine*	*Wine*	*Alcohol*
Cabernet	*Red wine*	*Wine*	*Alcohol*
Scotch	*Whisky*	*Liquor*	*Alcohol*

Table 4. Hierarchy levels created by our approach and its values.

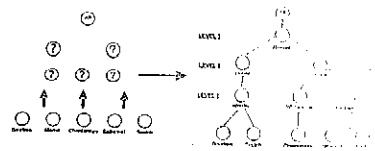

Figure 7. Instances before and after applying our approach.

After applying this method, an enriched hierarchy is obtained (see figure 6 and table 4) which accomplish with user requirements: *analyzing quantity of product sold aggregating by several levels of aggregation (subtype, type, and class of product)*, despite only one level of aggregation (*product*) was available. Then, our approach was applied starting with a poor hierarchy (see figure 5 and table 1) only with one level of aggregation (*product*) and an enriched hierarchy have been obtained which helps users to fulfill their requirements (aggregating by *subtype*, *type*, and *class of product*). In figure 6, the enriched hierarchy is shown. Instances of this hierarchy can be both observed in table 4 and figure 7.

7. Conclusion and future work

Using richer hierarchies when querying DW with OLAP tools is crucial to improve decision-making process. In this paper we present an approach to automatically enrich dimension hierarchies in DWs. In our proposal we start with a conceptual multidimensional model based on UML [12,26] and user requirements regarding dimension hierarchies. Then, we apply hypernym or meronym relationships from WordNet to obtain a richer hierarchy. This enriched dimension hierarchy allows users to accomplish with their information analysis needs and improve their decisions.

In spite of using WordNet, we have to point out that it presents several ontological problems [4] which must be overcome in a next future. For instance, WordNet does not have enough relations, such as attribution ("is an attribute-of") [24], which could be used to enrich every level of the hierarchy by adding several possible attributes (i.e. for *city*, attributes like *population* or *area* could be added). Some kind of formal specification of WordNet (like OntoWordNet [5]) could be used to solve these ontological problems.

In the line of [25] we can study a methodology for creating and managing domain ontologies to properly apply them in our approach.

Finally, we can use WordNet within DWs systems to overcome dimension update problems [7] or to resolve integration problems [10] and inaccurate problems regarding summarizability [6].

8. Acknowledgements

This work has been partially supported by the METASIGN project (TIN2004-00779) from the Spanish Ministry of Education and Science and by the MESSENGER project (PCC-03-003-1) from the Regional Science and Technology Ministry of Castilla-La Mancha (Spain).

References

[1] Abelló A., Samos J., Saltor F. Understanding Analysis Dimensions in a Multidimensional Object-Oriented Model. Int. Workshop on Design and Management of Data Warehouses (DMDW), 2001.

[2] Akoka J., Comyn-Wattiau I., Prat N. Dimension Hierarchies Design from UML Generalizations and Aggregations. ER 2001. LNCS 2224, pp. 442-455, Springer-Verlag, 2001.

[3] Chandrasekaran B, Josephson J.R., Benjamins V.R. Ontologies: What are they? why do we need them? IEEE Intelligent Systems and Their Applications, 14(1):20–26, 1999. Special Issue on Ontologies.

[4] Gangemi A., Guarino N., Masolo C., Oltramari A. Sweetening WORDNET with DOLCE. AI Magazine 24(3): 13-24 (2003).

[5] Gangemi A., Navigli R., Velardi P. The OntoWordNet Project: Extension and Axiomatization of Conceptual Relations in WordNet. CoopIS/DOA/ODBASE 2003. LNCS 2888, pp. 820-838, Springer-Verlag, 2003.

[6] Horner J., Song I-Y., Chen P. An analysis of additivity in OLAP systems. 7th ACM Int. Workshop on Data Warehousing and OLAP (DOLAP), pp. 83-91, 2004.

[7] Hurtado C.A., Mendelzon A. O., Vaisman A. A. Maintaining Data Cubes under Dimension Updates. 15th International Conference on Data Engineering (ICDE) 1999: 346-355. IEEE Computer Society Press.

[8] Inmon W., Building the Data warehouse, John Wiley & Sons, 1996.

[9] Jagadish H. V., Lakshmanan L. V. S., Srivastava D. What can Hierarchies do for Data Warehouses?. 25th VLDB Conference, 1999.

[10] Kedad Z., Métais E. Ontology-Based Data Cleaning. NLDB 2002. LNCS 2553, pp. 137-149, Springer, 2002.

[11] Kimball R. The Data Warehouse Toolkit: Practical Techniques For Building Dimensional Data Warehouse. John Wiley & Sons, 1996.

[12] Luján-Mora, S., Trujillo, J., Song I-Y. Extending UML for Multidimensional Modeling. 5th International Conference on the Unified Modeling Language (UML 2002), pp. 290-304: LNCS 2460, 2002.

[13] Luján-Mora S., Trujillo J. A Comprehensive Method for Data Warehouse Design. In Proceedings of the 5th International Workshop on Design and Management of Data Warehouses (DMDW'03), pages 1.1–1.14, Berlin, Germany, September 2003.

[14] Luján-Mora S., Trujillo J. A Data Warehouse Engineering Process. 3rd Biennial International Conference on Advances in Information Systems (ADVIS'04), Izmir, Turkey. LNCS 3261, pp. 14-23. 2004.

[15] Malinowski E., Zimányi E. OLAP Hierarchies: A Conceptual Perspective. Advanced Information Systems Engineering CAiSE 2004, LNCS 3084, pp. 477-491.

[16] Miller G.A., Beckwith R., Fellbaum C., Gross D., Miller K.J. WordNet: An on-line lexical database, International Journal of Lexicography, 3(4), 1990.

[17] Miller G.A., Fellbaum C. Semantic networks of English. Lexical And Conceptual Semantics. Blackwell Cambridge and Oxford, England, 1992. pp. 197-229.

[18] Montoyo A., Palomar M. WSD Algorithm Applied to a NLP System. NLDB 2000. LNCS 1959, pp. 54-65, Springer-Verlag, 2001.

[19] Morato J., Marzal M.A., Lloréns J., Moreiro J. WordNet Applications. Proc. of the 2nd International WordNet Conference (GWC) 2004, pp. 270-278.

[20] Object Management Group (OMG). Unified Modeling Language Specification 1.5. http://www.omg.org/cgi-bin/doc?formal/03-03-01. 2004.

[21] Pourabbas E., Rafanelli M. Characterization of Hierarchies and Some Operators in OLAP Environment. In Proc. of the 2nd ACM Int. Workshop on Data Warehousing and OLAP (DOLAP), pp. 54-59, 1999.

[22] Schneider, M. Well-formed Data Warehouses Structures. 5th International Workshop Design and Management of Data Warehouses, 2003.

[23] Smith J.M., Smith D.C.P. Database Abstractions: Aggregations and Generalizations. ACM TODS, 2(2), 1977.

[24] Storey V. Understanding Semantic Relationships. VLDB Journal,2, 455-488 (1993).

[25] Sugumaran V., Storey V. Ontologies for conceptual modeling: their creation, use, and management, Data & Knowledge Engineering, Vol. 42(3), 2002, pp 251-271.

[26] Trujillo J., Palomar M., Gómez J., Song I.Y. Designing Data Warehouses with OO Conceptual Models. IEEE Computer, 34(12):66-75, 2001.

[27] Toivonen S., Niemi T. Describing data sources semantically for facilitating efficient creation of OLAP cubes. 3rd International Semantic Web Conference (ISWC2004). Hiroshima, Japan, November, 2004.

[28] Vossen P. EuroWordNet: building a multilingual database with wordnets for European languages. Published in: The ELRA Newsletter, February 1998, Vol. 3 n.1, ISSN: 1026-8300. Paris. p. 7-10.

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

Automatically Generating Aggregations for Ontologies from Database Schema: some alternatives to type hierarchies

Csaba Veres

Norwegian University of Science and Technology
Sem Saelands vei. 7-9
Trondheim
Norway
Csaba.Veres@idi.ntnu.no

Abstract

Type hierarchies (a particular kind of taxonomy) are a fundamental part of ontologies, but are less generally evident in databases. It is therefore important to discover reliable and preferably automated methods for identifying taxonomic relationships in existing database instances. But we also argue that the role of type hierarchies is overstated, and show several aggregations that can replace generalization in some circumstances. In light of this we propose that simplistic methods for discovering potential taxonomies in databases can be useful if they are complemented by sophisticated pruning mechanisms which ensure that the correct taxonomies and aggregations are adopted. We show that the most appropriate type of relationship can be discovered on the basis of the grammatical properties of the terms as used in texts of natural language, which can therefore form a basis for the pruning mechanism.

1 Introduction

The basic organising relation for ontologies is a taxonomy of types [1]. In a survey of relatively well developed ontologies of significant size, [2] find that "taxonomy is the center part of most ontologies". Taxonomies have had a long history in knowledge representation, appearing in the guise of semantic networks, which have their origins in ancient Greek philosophy [3]. More recently [5] argue that taxonomies " ... have become an important conceptual tool for database schemas, knowledge-based systems, and semantic lexicons." Taxonomy is defined in [5] as " ... a partial-ordering relation commonly known as is-a, class inclusion, or subsumption ...". To support the primacy of taxonomies, a fundamental axiom for asserting facts about concepts in OWL (the W3 endorsed Web Ontology Language) is *rdfs:subClassOf*, which is used in asserting the taxonomic sub type relation between concepts.

The role of concept type hierarchies in databases is somewhat less than straightforward, and can depend on the particular modeling tool, the stage of database design, the choice of database architecture, and so on. For example the ER conceptual modeling notation does not have the facility for concept subtyping, but it is a feature of EER and even more expressive semantic data models [8]. Moreover, full support for abstract datatypes in object databases makes class hierarchies into first class citizens. Identifying taxonomic relationships in existing database descriptions is therefore going to depend on the format of those descriptions. In the absence of an expressive data model where such relationships are expressly defined, they must be inferred from other aspects of the database relations. That is, flattening the concepts into tables can lead to a loss of information about the concepts and their roles, especially regarding hierarchies. For example different classes of PRODUCT might simply be distinguished in a database by an *product_type* attribute, in which case the subtype relationship is not directly-indicated. Yet PRODUCT might serve as an excellent candidate as a superclass for the different products offered by a retailer. Another possibility is that instances of the subclass relationship might be inferred through processing the data dictionary. There are many existing techniques for automatically generating taxonomies from glossary definitions and unstructured text, which can

Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005