International Journal for Modern Trends in Science and Technology, 6(12): 28-32, 2020

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST061206
Available online at: http://www.ijmtst.com/vol6issue12.html

An Ontology Approach to Data Integration using Mapping Method

Dr.A.Mekala

Department of BCA, Sacred Heart College, Tirupattur, Tamil Nadu

To Cite this Article
Dr.A.Mekala , "An Ontology Approach to Data Integration using Mapping Method", *International Journal for Modern Trends in Science and Technology*, 6(12): 28-32, 2020.

Article Info
Received on 08-November-2020, Revised on 16-November-2020, Accepted on 25-November-2020, Published on 30-November-2020.

ABSTRACT

Text mining is a technique to discover meaningful patterns from the available text documents. The pattern sighting from the text and document association of document is a well-known problem in data mining. Analysis of text content and categorization of the documents is a composite task of data mining. Some of them are supervised and some of them unsupervised manner of document compilation. The term "Federated Databases" refers to the in sequence integration of distributed, autonomous and heterogeneous databases. Nevertheless, a federation can also include information systems, not only databases. At integrating data, more than a few issues must be addressed. Here, we focus on the trouble of heterogeneity, more specifically on semantic heterogeneity – that is, problems correlated to semantically equivalent concepts or semantically related/unrelated concepts. In categorize to address this problem; we apply the idea of ontologies as a tool for data integration. In this paper, we make clear this concept and we briefly explain a technique for constructing ontology by using a hybrid ontology approach.

KEYWORDS: *Federated Databases, Ontology, Semantic Heterogeneity.*

I. Introduction

Data integration means merging data from several assorted sources. While performing the data incorporation you have to agreement with several issues such as data redundancy, inconsistency, duplicity and many more. Data integration merges information from a number of heterogeneous sources to attain meaningful data. The source involves several databases, multiple files or facts cubes. The integrated information must excused inconsistencies, discrepancies, redundancies and disproportion. Data integration is significant as it provides a unified view of the scattered information not only this it also maintains the accurateness of data. This helps the data-mining program in mining the useful in sequence which in turn helps the executive and managers in taking the strategic decisions for the betterment of the project.

II. FEDERATED DATABASE STRATEGY

Autonomy: The users and the applications can access to the information through a federated system or by your own limited system. The autonomy can be classified in three types [7,20]: *plan autonomy*, *statement autonomy* and *execution autonomy*.

Distribution: Nowadays most computers are connected to some type of network, especially the Internet, and it is natural to think of combining application and data sources that are physically located on different hosts, but that can communicate through the network.

Heterogeneity: it can be classified into four categories, *structure, syntax, system, and semantic*. The *structure* heterogeneity involves different data models; the *syntax* heterogeneity involves different languages and data representations and the *system* heterogeneity involves hardware and operating systems. The semantic heterogeneity can be classified as: semantically equivalent concepts semantically unrelated concepts.

III. DATA INTEGRATION TECHNIQUES

Manual Integration

This method avoids the use of automation during data integration. The data analyst himself collects the information, cleans it and integrates it to provide useful in order. This method can be implemented for a little association with a small data set. But it would be tedious for the great, complex and recurring integration. Because it is a time taking process as the entire procedure has to be completed manually.

Middleware Integration

The middleware software is employed to collect the in sequence from different sources, normalize the data and stored into the consequential data set. This technique is adopted when the enterprise wants to integrate data from the legacy systems to modern systems. Middleware software acts as an exponent between the legacy systems and advanced systems. You can take an example of the adapter which helps in between two systems with different interfaces. It can be applied to some system only.

Application-Based Integration

This technique makes use of software application to extract, transform and load the data from the heterogeneous sources. This method also makes the data from disparate source compatible in order to ease the transfer of the data from one system to another.

This technique saves time and effort but is little complicated as designing such an application requires technical knowledge.

Uniform Access Integration

This technique integrates data from a more discrepant source. But, here the position of the data is not changed; the information stays in its original location.

This method only creates a unified view which represents the integrated information. No separate storage space is required to store the integrated data as only the integrated view is created for the end-user.

IV. DATA INTEGRATION USING ONTOLOGIES

There are a group of advantages in the employ of ontologies for data integration. the ontology provides a wealthy, predefined vocabulary that serves as a stable conceptual boundary to the databases and is independent of the database schemas; the facts represented by the ontology is sufficiently comprehensive to support translation of all the relevant information sources; the ontology supports dependable management and the acknowledgment of inconsistent data; etc.

As we can see, the method has three main stages: building the shared expressions, building local ontologies and defining mappings. Each step embodies a set of tasks that must be achieved.

I: Building the shared vocabulary:

As Figure 1 shows, this chapter contains three major steps: *analysis of information sources*, *search for terms (or primitives)* and *defining the global ontology*. The first step implies a total analysis of the in order sources, e.g., what information is stored, how it is stored, the meaning of this in order (the semantic), etc.

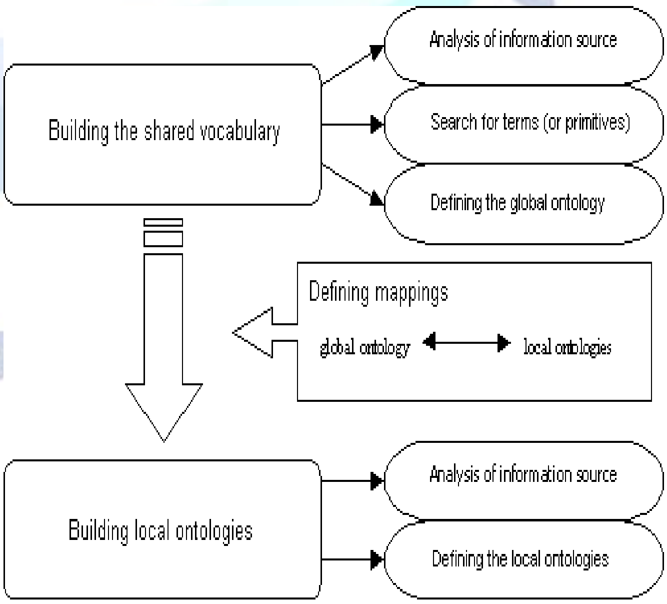


Figure 1: Ontology Construction Method

The primary difficulty is reflected by the *amount* attribute in both systems since there are two classes with the same meaning but with dissimilar representations: liter and gallon.

System 1 has the *milk* and *truck* classes with the *amount* quality to represent the amount of transported milk. The instances of the *amount* attribute are in *liters*. On the extra hand, System 2 has the same classes but the *amount* attribute is represented in *gallons*. The second problem, *different classification*, is reflected by the *truck* class in System 2 and, in System 1 by the hierarchy of trucks: *truck_with_refrigeration* and *truck_without_refrigeration*. Both systems use different classifications to indicate the same things. The *truck* class in System 2 includes the two sub classifications of System 1.

The second step, *search for terms (or primitives)*, implies the choice of the list of terms or concepts in conformity with the shared vocabulary. In our example, the list of terms can be: *milk, gallon, amount, truck, truck_with_refrigeration* and *truck_without_refrigeration*. We comprise the terms of the hierarchy because this classification is more descriptive. As we contain previously mentioned, the

The third and last step, *defining the global ontology*, uses the terms selected in the last step to create the universal ontology.

We could have represented the ontology without the *gallon* class. Thus, the *transporting* relation would be:

```
;;; Transporting
(Define-Relation Transporting (?Truck ?Milk
?Number) "the quantity of milk transported by
a truck" :Def (And (Truck ?Truck) (Milk ?Milk)
(Number ?Number)))
```

The primitive type *Number* is replacing the *gallon* class. This representation, although acceptable for Ontolingua, is not clear enough and does not supply the complete semantic information available in the system. In fact, adding up of classes describing attribute types in the ontology is the finest choice for providing the semantic in order required.

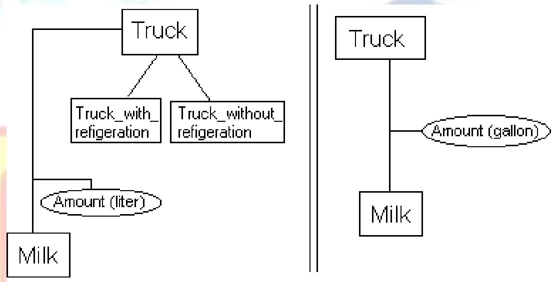
II: Building local ontologies:, This stage contains two major steps: *analysis of information source* and *defining the local ontologies*.

III: Defining Mappings: In this phase we define the mappings stuck stuck between the concepts defined in the universal ontology and in the local ontologies. This phase must solve the *truck* class in System includes both trucks with refrigeration and without refrigeration.

The first step, is similar to the first stage previously explained. A complete analysis of the information sources must be made.

This analysis is performed independently, that is, lacking taking into account the other information sources. With this analysis, the second step can be performed.

Every Ontology defines it's have classes and relationships. Ontology 1 has *milk, liter* and *truck* (with the subclasses) classes and the *the amount* relative to represent the domain. Ontology 2 has the similar classes and relations except for *liter class* that is replaced by the *gallon* class indicating dissimilar milk measures.



System 1 **System 2**

semantic heterogeneity problems creation connections between the two stages.

In our example, the global ontology has the *gallon* class to stand for the metric measure of the milk. The *liter* group of Ontology 1 is not represented in the global ontology and we must include an axiom to relate these classes. A liter equals 0.22 gallon.

(<=>(Liter ?x) (Gallon ?x * (0.22)))

This mapping is performed because users could query the integrated system by asking for information about the amount of milk in liter measure. And the global ontology only has the *gallon* class. Therefore, when users make queries, the global ontology and the mapping are used to retrieve the information needed.

No mapping is needed for the *truck* classes in together systems since they encompass the same name and they denote the same things.

added

and relations (of the new source) that are not in the global ontology must be added.

Ontology 1	Ontology 2
;;; ------------------ Classes -------------- ;;;Milk (Define-Class Milk (?X) "the set of types of milks" :Def (And (Thing ?X))) ;;;Liter (Define-Class Liter (?X) "the liters" :Def (And (Thing ?X))) ;;;Truck (Define-Class Truck(?X) "the set of trucks" :Def (And (Thing ?X))) ;;;Truck_With_Refrigeration (Define-Class Truck_With_Refrigeration (?X) "the set of trucks with refrigeration" :Def (And (Truck ?X))) ;;;Truck_Without_Refrigeration (Define-Class Truck_Without_Refrigeration (?X) "the set of trucks without refrigeration" :Def (And (Truck ?X))) ;;; ------------------ Relations -------------- ;;; The_Amount (Define-Relation The_Amount (?Frame ?Value) "the amount expressed in liters" :Def (And (Liter ?Frame) (Number ?Value))) ;;; Transporting (Define-Relation Transporting (?Truck ?Milk ?Liter) "the amount of milk transported by a truck" :Def (And (Truck ?Truck) (Milk ?Milk) (Liter ?Liter)))	;;; ------------------ Classes -------------- ;;;Milk (Define-Class Milk (?X) "the set of types of milks" :Def (And (Thing ?X))) ;;;Gallon (ine-Class Gallon (?X) "the set of gallons" :Def (And (Thing ?X))) ;;;Truck (Define-Class Truck(?X) "the set of trucks" :Def (And (Thing ?X))) ;;; ------------------ Relations -------------- ;;; The_Amount fine-Relation The_Amount (?Frame ?Value) the?Frame) (Number ?Value))) ;;; Transporting (Define-Relation Transporting (?Truck ?Milk ?Gallon) "the amount of milk transported by a truck" :Def (And (Truck ?Truck) (Milk ?Milk) (Gallon llon)))

Our technique serves as a practical guide to analyze integrated information taking benefit of the use of ontologies. The ontologies allow the capturing of all the semantic in sequence provided by the system. more than a few semantic problems can be localized and solved when our method is followed. Two of them, property- *type* mismatch and *different classification,* have been described in the last section.

As our suggestion is based on an hybrid ontology move toward, it has two main advantages:

(1) new information sources can be

Also, the limited ontology and the mappings among the new extra terms must be defined.

Also, we are extending our method to include the idea of *contexts* assuming that each term is true or false according to the context it is in, that is, the context determines the truth or falsity of a statement as well as its meaning.

One main advantage of contexts is the avoidance of the homonym problem. If two systems use the same word (term) each denoting different meanings they will be in different contexts and they need not be compared.

Each ontology might be related to several contexts indicating the different roles of one database. For example, the *use cases* of a UML specification] might be the source to obtain some of the contexts. Each context contains a series of terms included in the ontology. Then, we will define relationships among the contexts of different ontologies. Thus, simply the terms included in the related contexts will be compared.

Accordingly, these two new extensions to our approach help finding and solving more heterogeneity problems. The inclusion of the similarity functions gives a more precise comparison among the terms of different ontologies. Also, the combination of the use of ontologies and contexts provides a higher degree of semantic information needed for consistent data integration.

VI. CONCLUSIONS

Our in progress researches on data incorporation uses the "ontology" concept to help solving the semantic heterogeneity problems. We create a useful and realistic method for the construction of a hybrid ontology approach. The method has three main stages: ***building the shared vocabulary***, ***building confined ontologies*** and d***efining mappings***. Each period embodies a number of steps that must be followed. Every stride serves like a guide to make out all the cases of semantic heterogeneity and the ways to resolve them.

Our look at is ongoing and there are a numeral of aspects being analyzed. We aim at including the "context" concept to solve, for example, the homonym problem, different representations, etc. Also, we are working on including similarity functions to find similarity terms within the different local ontologies.

Finally, the method and its extensions need be validated by using more complex examples and real cases for revise.

REFERENCES

[1] Q. Mei and C.X. Zhai, "Discovering Evolutionary Theme Patterns from Text—An Exploration of Temporal Text Mining," Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery in Data Mining, pp. 198-207, 2005.

[2] J. Allan, editor. Topic Detection and Tracking: eventbased information organization. Kluwer Academic Publishers, 2002.

[3] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Soren Auer, Christian Becker, Richard Cyganiak, Sebastian Hellmann. DBpedia - A Crystallization Point for the Web of Data.

[4] Leonhard Hennig, Winfried Umbrath , Robert Wetzker. An Ontology-based Approach to Text Summarization, 2008.

[5] Xing Jiang, Ah-Hwee Tan. Learning and inferencing in user ontology for personalized semantic web search, Information Sciences, 2009, 2794-2808.

[6] Marek Obitko, Vaclav Snasel, Jan Smid. Ontology design with formal concept analysis, Edited by Vaclav snasel, Radim Belohlavek. In: Proc of the CLA 2004 Intl. workshop on Concept Lattices and their Applications Ostrava, Czech Republic, Sept. 2004, 111-119.

[7] Hele-Mai Haav. A semi-automatic method to ontology design by using FCA, Edited by Vaclav Snasel, Radim Belohlavek, In: Proc. of the CLA 2004 lnfl. Workshop on Concept Lattices and their Applications Ostrava, Czech Republic, Sept. 2004, 13-24.

[8] Lixin Han, Guihai Chen. A fuzzy clustering method of construction of ontology-based user profiles,Advances in Engineering Software, 2009, 535-540.

[9] Deryle Lonsdale, David W. Embley, Yihong Ding, Li Xu, Martin Hepp. Reusing ontologies and language components for ontology generation, Data & Knowledge Engineering, 2010, 318-330.

[10] Rung-Ching Chen, Cho-Tscan Bau, Chun-Ju Yeh. Merging domain ontologies based on the Word-Net system and Fuzzy Formal Concept Analysis techniques, Applied Soft Computing, 2011, 1908-1923.

[11] T. R. Gruber, "Toward Principles for the design of Ontologies used for Knowledge Sharing", in Proc. of International Workshop on Formal Ontology, March 1993.

[12] Y. Labrou and T. Finin, "Yahoo! as Ontology: Using Yahoo! Categories to Describe Documents," in Proc. of The Eighth International Conference on Information Knowledge Management, pp. 180-187, Nov 1999, Kansas City, MO.

[13] N. Guarino, C. Masolo, and G. Vetere, "OntoSeek: Content-based Access to the Web," IEEE Intelligent Systems, Volume 14, no. 3, pp. 70-80, 1999.