Measuring Database Objects Relatedness in Peer-to-Peer Databases

M. Basel Almourad, *Member, IACSIT*

Abstract—**Peer-to-Peer database management systems have become an important topic in the last few years. They rise up p2p technology to exploit the power of available distributed database management system technologies. Identifying relationship between different peer schema objects is one of the main activities so semantically relevant peer database management systems can be acquainted and become close in the overlay. In this paper we present our approach that measures the similarity of peer schema objects and ultimately depicts database management systems closeness in the overlay.**

Index Terms—**Meta-data, P2P databases, schema similarity, semantic relatedness, semantic similarity.**

I. Introduction

Peer-to-Peer (P2P) systems become an active research because of the opportunities for real –time communication, ad-hoc collaboration and information sharing in a large scale environment. The P2P systems are initially created for sharing digital media files (e.g. music files, video, and photos). A typical P2P system like Napster and Gnutella uses file properties (e.g. file name, author name, etc) and provides keyword-based exact matching lookup services [1].

Whereas unstructured data sharing has been very successful, P2P systems are important and useful for more than just sharing of flat files. In the last few years, various researches have been conducted on P2Pdatabase management systems [2]-[5]; the aim is to rise up P2P technology to exploit the power of available distributed database management technologies. A P2P database management system (PDMS) is envisaged as a distributed data integration system that provides transparent access to collection of heterogeneous and autonomous databases without the need for centralized logical schema [6].

In a PDMS, each peer database (i.e. pDBMS) is an autonomous source that has a local schema. Sources store and manage their data locally, disclosing part of their schemas to the rest of the peers [7].

Due to the absence of the global schemas, peers convey and answer queries based on their local schemas. Peers also perform local coordination with their neighbors in the overlay (i.e. acquaintees). During the acquaintance procedure, the two peers exchange information about their local schemas and created mediation mapping semi-automatically [8]. Peers with new schemas simply need to provide a mapping between their schema and any other schema already used in the system to be part of the network [9]. A main question is how semantically relevant peer DBMS are acquainted or become close in the overlay [10].

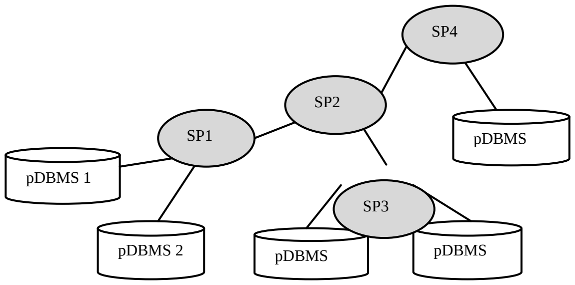

Fig. 1. Illustration of pDBMS relatedness in PDMS.

For example how pDBMS1 and pDBMS2 (seeFig. 1) will be clustered together and belong to the same super peer (SP).

In this paper we present our technique that measures peers schema relatedness so semantically related peers can be acquainted and become semantically close in the overlay

II. Database Design from the Semantic Point of View

The designers of DB have to deal with different ‘worlds’: The Real world, the Conceptual world and the Represented world [11]. The real world is made up of real world objects out of which a selected set (universe of discourse, UoD) and specific view(s) on these selected objects is being considered in particular DB design. The conceptual world comprises conceptualization. A conceptualization is an abstraction that simplified view of a UoD for some purpose [12] and is made up of concepts and abstract relationships [11] hence a conceptualization can be viewed as a model of the UoD. What real world objects comprise the UoD and what characteristics of these objects should be included in a conceptualization are basically guided by the requirements of the intended applications and their users. A concept in conceptualization typically represents a collection of real world objects satisfying the notation of the concept. Thus concepts group individual real world objects logically into collections [13]. The representation of the meaning of a concept, in a designer’s conceptualization, lies in the set of relationships the concept has with other concepts [11]. This is an abstraction of the designer’s conceptualization due to his requirements and the limitations of the representation system. The represented world comprises representations of conceptualization in the DB schemas therefore a DB schema can be considered as a representation of conceptualization that a group(s) of people share about a universe of discourse.

When building schemas the DB designers invent names to label schema elements. Since DBs are usually designed to model the real world, schema element names are normally natural language nouns chosen to provide a bridge between

Manuscript received October 9, 2012; revised January 15, 2013.

M. B. Almourad is with the Zayed University, P. O. Box 19282, Dubai, UAE (e-mail: basel.almourad@zu.ac.ae).

DOI: 10.7763/IJCCE.2013.V2.190

them and their corresponding conceptualization [14].

III. Meta-Data and Its Use in Discovering Peer Schema Object Similarities

Building an ideal concept space for set existing DBs is not an easy task as it requires the elicitation (involvement) of the conceptualization of those who designed the DBs. However, approximate concept space may be built to represent the real world concepts modeled by the DBs by analyzing their meta-data and eliciting relevant knowledge from the DBAs.

Meta-data is defined as data about data. Meta-data is considered to be valuable resource for managing information sources [15], [16].

In a typical DB integration exercise, a DB schema integrator needs to locate DB schema objects that are relevant to the user information requirement. An appropriate sub-set of schema objects could be selected. In this exercise various types of met-data are required to facilitate the above tasks. In current typical DB integration research a common approach is to use conceptual knowledge about information content of DBs in the federation for the purpose of locating relevant DB schema objects. Such knowledge is used to link schema objects with relevant concepts in some conceptual structure(s) such as ontologies. Two types of ontology have been used general global ontology and domain specific ontology. The size of the general global ontology makes its management difficult in PDMS and the scope of concepts covered by the ontology may not be sufficient to describe conceptualization in some domains. On the other hand using domain-specific ontologies is inappropriate since the semantic of peer schema in PDMS is not known and it is not permanent certain domain

In our approach we use met-data of peer schema objects as heuristics to determine peer schema object similarities and use these similarities to measure the closeness of peers in PDMS. The definition of attribute equivalence varies in the literature. For example,the authors in [17] define two schema objects as equivalent if a mapping function could be defined between the values of their domain, whereas other authors [18], [19] define two schemas to be equivalent if one can be obtained from the other by a pre-defined set of transformations. Reference [14] defines two schema objects as equivalent when they represent the same real world concept. Reference [20] introduces the concept of semantic proximity in order to formally specify various degrees of semantic similarity among related objects in different application domains. Our technique reasons about the meaning and resemblance of peer objects in terms of their meta-data representation in order to identify those that could be semantically related. The technique uses kind of heuristics to determine the similarity of objects based on the occurrences of related attributes in the objects and the percentage of related attributes. The basis of heuristic depends on attribute equivalence to determine whether objects are semantically equivalent. Wordnet thesaurus is used in [19] to help automate the identification of semantically similar properties.

IV. The Role of Wordnet Thesaurus

A thesaurus comprises a collection of terms which are formally organized so that relationships between the terms are made explicit [21]. In database integration research, thesauri and thesauri-like concept structures have been used for various tasks, e.g. to enable users to formulate queries in thesaurus terms [21]; to help automate the identification of semantically similar entities despite their local representational differences; and to construct semantic dictionaries. In addition, techniques developed in IR systems have been used to determine object type similarities to assist database integration [22] and to index databases to facilitate resource discovery.

WordNet is a machine readable, on-line lexical database of English words (nouns, verbs and adjectives) [19]. The words are grouped into synsets: sets of synonyms (lists of synonym word forms that are interchangeable in some context). The words in a synset are selected so that they represent a single lexical concept. One of the main assumptions underlying WordNet is that the different meanings or senses of a given word can be conceived unambiguously by considering the other words in the corresponding synsets to which they belong due to their semantic relationships.

Semantic relations are represented by a pair of synset_ids, in which the first synset_id is generally the source of the relation and the second is the target. If the pair synset_id, w_num is present, the operator represents a lexical relation between word forms. In our technique we consult synset predicates to detect synonyms that may be used when identifying similar objects. The synset predicate has the following syntax:

s (*synset_id, w_num, 'word', ss_type, sense_number, tag_state*)

where an *s* operator is present for every word sense in WordNet and w_num specifies the word number for this word in the synset.For example, to find whetherscholar and student are synonyms or not we use the following predicate:

s(X,_, scholar, _, _, _),s(X, _,student, _, _, _).

The result is either true or false depending on whether scholar and student belong to the same synonym set or not.

V. Degree of Relatedness Computation

Semantic closeness between relations of peers in PDMS is usually of interest only when the relations have some sort of resemblance so that they can be integrated in a way that satisfies their context and fits the user requirements.

Various types of semantic relationship are possible between different pDBMS relations in PDMS (e.g. equivalent, overlap, inclusion, disjoint). There are a number of classifications reported in the literature. For example, they can be classified according to the real world objects they represent [13], or they can be classified with respect to a concept space constructed for a federation [23].

We presume that two relations are Semantically Related when they have corresponding intended Real World Semantics (RWS) for some universe of discourse and Semantically Incompatible when they are not semantically related. The real-world semantics of a relation C, RWS(C), is defined as the set of objects in the real world defined by C's pDBMS schema definition. As we cannot depend on the extension of relations in reality, we use relation properties as the basis for relation comparison, assuming that the properties represent the intended meaning of the relations. In a real life application, complete semantic or syntactic equivalence between the related components of databases being integrated should not be expected to occur very often. Therefore, we adopt the notion of similarity rather than equivalence between database properties as the basis of our research. To detect whether two relations are similar, we designed a heuristic procedure known as Relation Similarity Detector (RSD). The RSD quantifies the measure of similarity between two relations in disparate peers according to a hierarchical aggregation of similar properties. If the measure exceeds or equals a certain threshold (which can be altered by the user), then we consider the two relations to be similar and ultimately close.

The heuristic used in RSD is based on a hierarchical aggregation of the results of applying several similarity functions [24], [5]. The advantage of this approach is that our similarity functions use meta-data that is acquired from the properties of the relations. The following equation is used to determine whether two relations *C*1, *C*2 from two pDBMSs are similar/related or not:

$$\sum_{i=1}^{n} F_i(C_1, C_2) \times W_i \geq Threshold \qquad (1)$$

where

1) C_1, C_2 are the two relations being compared.
2) F_i are the set of similarity functions (e.g. matching relation names, attribute names, attribute types). The result of applying each function is a value in [0, 1].
3) $W_i$are the function similarity weights given by the user to the particular function, and

$$\sum_{i=1}^{n} W_i = 1$$

4) The threshold is a value in [0,1]. There is a default value for the threshold and this value can be altered by the user to suit requirements. A high value forces the heuristic algorithm to detect only the relations that have highly similar properties and ultimately very close relations and this could ignore some potentially related relations. However a low value could mean the heuristic algorithm has to reject many relations as non-close which might be related.

Each function's similarity weight can be altered and a higher weight can be assigned for a function that is believed to be more efficient or important than other functions, or can be given a zero value to a weight, if we want to eliminate the corresponding functionality.

In what follows, we will give a definition of the similarity functions and show how the value of each function is calculated. We call each function a factor:

1) Relation Name Similarity Factor (RNSF) - This is determined by the following values:

$$RNSF = \begin{cases} 1, & iftheyarethesamename \\ [0.3, 0.5], & iftheyarespeltdifferentlyorare \\ & synonym(basedonWordnet) \\ 0, & iftheyaredifferentnames \end{cases}$$

2) Relation Property Similarity Factor (RPSF): property here refers to both attribute and its data type. To detect whether two properties are similar or not we consider the name and type of each property. The RPSF is given by the following equation:

$$RPSF = \frac{Numberofequivelantproperties}{Theavergaenumberofpropertiesacrossbothrelations}$$

Two properties are considered as equivalent if the *Property Similarity (PS)* factor exceeds or equals a threshold value, where PS is calculated by the following equation:

$$PS = \begin{cases} 0, & ifPNS \times W_{PNS} = 0 \\ PNS \times W_{PNS} + PTS \times W_{PTS}, & Otherwise \end{cases}$$

where *PNS* is the *Property Name Similarity* factor and PTS is the *Property Type Similarity*factor, and W_{PNS} and $W_{PTS}$are the weights for *PNS* and *PTS*, respectively.The following values are given to similar names (*PNS*) and similar types (*PTS*):

$$PNS = \begin{cases} 1, & iftheyarethesamename \\ [0.3, 0.5], & ifthearespeltdifferentlyorare \\ & synonym\ (basedon(WordNet)) \\ 0, & iftheyaredifferentnames \end{cases}$$

$$PTS = \begin{cases} 1, & iftheyarethesametype \\ 0.5, & iftheyarecompatibletype \\ 0, & iftheyaredifferenttypes \end{cases}$$

Two data types are considered compatible if they are both members of a certain type set. For example, varchar, varchar 2, char, char (n) are compatible because they are members of a character data type set. If the types are non-primitive objects (i.e. user defined types), RSD is re-consulted to detect the similarity of these types. Note that a recursive process can occur.

The current values of threshold, weights and factors have been chosen through our experience of running the algorithm on a number of component database schemas. It should be stressed here that all these values can be altered by users. Some of the above factors could be augmented or enhanced in the future to make the function more complex.

VI. Example

Let us take the following two relations: Employee (see Table I) and Worker (see Table II)in pDBMS1 and pDBMS2 respectively:

TABLE I: Employee Relation

Attribute	Data Type
F-Name	Varchar2(25)
Last_name	Varchar2(20)
Home_address	Varchar2(50)
Date_of_birth	Date
Salary	Number
Position	Varchar2(20)

TABLE II: Worker Relation

Attribute	Data Type
First_name	Varchar2(25)
Surename	Varchar2(25)
Address	char(40)
Birthday	Date
Wage	Number
Sex	Char

We will show how the heuristic module Relation Similarity Detector measures whether the Employee and Worker relations in pDBMS_1 and pDBMS2 respectively are close or not. We will show how each similarity function (factor) has its value calculated and then how the aggregation of all similarity function values is applied to evaluate whether two relations are similar and altimetry close in the overlay.

A. Relation Name Similarity Factor (RNSF)

As the relations Employee and Worker have different names, the WordNet thesaurus is consulted and it is found that Employee and Worker are synonym; therefore RNSF is assigned the value of 0.5.

B. Relation Property Similarity Factor (RPSF)

This factor reflects the number of similar properties. Identifying two similar properties requires that these two properties must have at least similar names and compatible data types and their combination must exceed or equal a threshold value. Let us assume that WPNS (the weight of *PNS*) is 0.8, WPTS (the Weight of *PTS*) is 0.2 and the threshold value is 0.4.

Let us take F_name form Employee relation and First_name from Worker relation. Since both attributes are spelt differently therefore *PNS* is assigned the value of 0.3.

Similarly the data type is equivalent therefore PTS is assigned the value of 1.

The value of PS is 0.3 × 0.8 + 1 × 0.2 = 0.44 and we consider F_name and First_name are similar properties.

Similarly let us take Salary from Employee relation and Wage from Worker relation. Since attributes are spelt differently, the WordNet thesaurus is consulted and it is found that Salary and Wage are synonym therefore *PNS* is assigned the value of 0.5. Similarly the data type is equivalent therefore PTS is assigned the value of 1. The value of PS is 0.5 × 0.8 + 1 × 0.2 = 0.6 and we consider Salary and Wage are similar properties. The property similarities in both relations and how PS factor is calculated are shown in (Table III).The average number of properties across both relations is 12/2= 6. Therefore RPSF = 4/6 = 0.66

After calculating all factors, we can measure the closeness of the Employee and Worker relations. If we assume that W_{RNSF} =0.4 and W_{RPSF} = 0.6 and the threshold value is 0.6. The closeness is calculated using (1).

TABLE III: PROPERTY SIMILARITIES OF EMPLOYEE & WORKER

Employee	**Worker**	**PS**	**>=0.4**
F-Name	First_name	0.3*0.8+1*0.2 = 0.44	Yes
Last_name	Surename	0.5*0.8+0.5*0.2 = 0.5	Yes
Home_address	Address	0.3*0.8+0.5*0.2= 0.34	No
Date_of_birth	Birthday	0.5*0.8+0.5*0.2 = 0.5	Yes
Salary	Wage	0.5*0.8+1*0.2 = 0.6	Yes
Position	-	= 0	No
-	Sex	= 0	No

From previous equation and 0.72 >= 0.6 we measure that the relations Employee and Worker are close and can belong to the same acquaintees.

The above example demonstrates our methodology for two relations in two different pDBMSs. In PDMS each peer is an autonomous source and normally discloses part of its schema to the rest of the peers. The above methodology can be performed against all disclosed relations of particular peers. To measure the relatedness of completely two peers we can apply the methodology against all relations in the peers to find out the number of equivalent relations and then apply the following equation:

$$\frac{Numberofequivelantrelations}{Theavergaenumberofrelationsacrossboth\ \text{peers}}$$

When the above ratio exceeds certain threshold the two compared peers can be considered related and therefore close enough to belong to the same acquaintees. The value of the threshold is determined by how much we want to be strict. The bigger threshold value determines more relatedness of two peers and vice versa.

VII. CONCLUSION

In this paper we have explained the importance of measuring the semantic relationship between different pDBMS schema objects in PDMS. We have presented our approach which depends on using PDMS schema properties to measure the closeness of different pDBMS schema objects in PDMS. We described RSD, the heuristic we use to quantify the measure of similarity between two relations according to a hierarchical aggregation of similarity factors. An example to clarify the methodology was given.

REFERENCES

[1] Gnutella website. [Online]. Available: http://www.gnutella.wego.com

[2] A. Bonifat, P. K. Chrysanthis, A. M. Ouksel, and K. Sattler, “Distributed databases and peer-to-peer databases: past and present,” *SIGMOD Rec*., vol. 37, no. 1, pp. 5-11, 2008.

[3] K. Norvag, E. Eide, and O. H. Standal, “Query planning in P2P database systems,” in *Proc. the 2nd International Conference on Digital Information Management*, 2007, pp. 376-381.

[4] V. Kantere and T. Sellis, “Data Exchange Issues in Peer-to-Peer Database Systems,” *Enterprise Information Systems*, Heidelberg: Springer Berlin, 2008, pp. 29-37.

[5] R. Mohamed, M. B. A. Mourad, and Y. M. A. Khalifa, “A Vision to Construct Multiple Data Views in Peer Data,” *Lecture Notes on InformticsLNIi*, vol. 84, 2006.

[6] A. L öser, W. Siberski, M. Wolpers, and W. Nejdl, “Information Integration in Schema-Based Peer-To-Peer Networks,” *Advanced Information Systems Engineering*, Heidelberg: Springer Berlin, 2003.

[7] G. D. Giacomo, D. Lembo, M. Lenzerini, and R. R. D. Calvanese, “Inconsistency tolerance in P2P data integration: An epistemic logic approach,” *Information Systems*, vol. 33, no. 4-5, pp. 360-384, 2008.

[8] A. Y. Halevy, Z. G. Ives, D. Suciu, and I.Tatarinov, “Schema Mediation in Peer Data Management Systems,” in *Proc. the 19th International Conference on Data Engineering*, 2003, pp. 505.

[9] Z. Majkić, “Intensional Semantics for P2P Data Integration,”*Journal on Data Semantics VI,* Heidelberg: Springer Berlin, vol. 4090, pp. 47-66, 2006.

[10] V. Kantere, D. Tsoumakos, and T.Sellis, “A Framework for semantic grouping in P2P databases,” *Information Systems*, vol. 33, pp. 611-636, 2008.

[11] F. Slator and M. G. Solaco, “Diversity with cooperation in database schemeta: Semantic relativisim,” in *Proc. the 14th International Conferences on Information Systems*, New York, 1993, pp. 247-254.

[12] T. R. Gruber, “A translation approach to portable ontology specifications,” *Knowledge Acquisition*, vol. 5, no. 2, pp. 199-220, 1993.

[13] J. M. Blanco, A. Illarramendi, and A. Goni, “Building a federated relational database system: An Approach using a knowledge-based system,” *International Journal on Intelligent and Cooperative Information systems*, vol. 3, no. 4, pp. 415-455, December 1994.

[14] K. Kashyap and A. Sheth, “Semantic and schematic similarities between database objects: a context-based approach,” *VLDB Journal,* vol. 5, pp. 276-304, 1996.

[15] T. M. Connolly and C. E. Begg, *Database Systems: A Practical Approach to Design, Implementation and Management*, 5th ed.: Addison Wesley, 2009.

[16] V. Kantere, D. Tsoumakos, T. Sellis, and R. Nick, "GrouPeer: Dynamic clustering of P2P databases,"*Information Systems*, vol. 34, no. 1, pp. 62-86, March 2009.

[17] J. A. Larson, S. B. Navathe, and R. Elmasri, "A theory of Attribute equivalence in databases with application to schema integration," *IEEE Transactions on Software Engineering*, vol. 15, no. 4, April 1989.

[18] S. B. Navathe and S. G. Gadgil, "A methodology for view integration in logical database design," in *Proc. the 8th International Conference on Very Large Data Bases*, Mexico City, 1982, pp. 142-164.

[19] S. Milliner, A. Bouguettaya, and M. Papazoglou, "A scalable architecture for autonomous heterogenous database interactions," in *Proc. the 21st International Conference on Very Large Databases*, Zurich, Switzerland, 1995, pp. 515-526.

[20] A. P. Sheth, S. K. Gale, and S. B. Navathe, "On Automatic reasoning for schema integration," *International Journal of Intelligent and Cooperative Information Systems*, vol. 2, no. 1, pp. 23-50, 1993.

[21] A. Amba, N. Narasimhamurthi, K. C. O'Kane, and P. M. Turner, "Automatic linking of thesauri," in *Proc. the 19th Annual International ACM SIGIR Conference on Research and Development in Information Rerieval*, Zurich, Switzerland, 1996, pp. 181-186.

[22] B. Eaglestone and N. Masood, "Schema interpretation: An aid to schema analysis in federated database design," in *Proc. the International CAiSE97 Workshop on Engineering Federated Database Systems*, University of Magdeberg, Germany, 1997.

[23] M. Jarke, R. Gallersdorfer, M. A. Jeusfeld, and M. Staudt, "Concept-Base - a deductive object base for meta data management," *Journal of Intelligent Information Systems*, vol. 3, pp. 167-192, 1994.

[24] R. Mohammed, M. B. A. Mourad, and Y. M.A. Khalifa, "Super-peer P2P Systems Utilizing Mediated Knowledge-based and User-defined Views ," *Journal of Computer Networks and Internet Research*, vol. 1, no. 1, July 2007.

M. B. Almouradhas received his BSc in Informatics Engineering from Aleppo University, Syria, and his PhD degree in Computer Science from Cardiff University of Wales, United Kingdom**.** He has served in various International Universities. He is currently an Assistant Professor in Zayed University, Dubai, UAE.Dr. Al-Mourad is active in several research areas where he published numerous articles including: Heterogeneous Data Management, Semantic Web, Web Accessibility, Community of Practice and Distributed and Agent Based Systems.