A Proposal to Maintain the Semantic Balance in Cluster-based Data Integration Systems

Edemberg Rocha Silva[1], Bernadette Farias Lóscio[2] and Ana Carolina Salgado[2]
[1]*Federal Institute of Education, Science and Tecnhology of Paraíba, João Pessoa, Brazil*
[2]*Federal University of Pernambuco, Recife, Brazil*

Keywords: Semantic Balance, Semantic Clusters, Dynamic Data Integration Systems, Schema Evolution, Clustering Measure.

Abstract: With the large volume of data sources on the Web, we need a system that integrates them, so that the user can query them transparently. For efficiency in queries, integration systems can group these sources in clusters according to the semantic similarity of their schemas. However, the sources have autonomy to evolve their schema, and to join or to leave the integration system at any time. This autonomy may cause a problem which we define as semantic unbalance of clusters. The semantic unbalance can compromise the formation of clusters and hence the efficiency of the submitted queries. In this paper, we propose a solution to the semantic balance of clusters in dynamic data integration systems based on self-organization. We also introduce a measure to evaluate how much the clusters are semantically unbalanced.

1 INTRODUCTION

The increasing number of distributed, autonomous and heterogeneous data sources, (ie: XML documents, relational database and HTML pages among others), has motivated the need for data integration systems, allowing users to query those sources transparently (Roth and Skritek., 2013; Pires et al., 2012). In this sense, dynamic data integration systems, as Peer Data Management Systems (PDMS) (Halevy et al., 2006), Grid (Zamboulis et al., 2010) or systems based on a pay-as-you-go strategy (Halevy et al., 2008), have been used to improve the data sharing of data sources distributed on the Web or on the cloud (Wall and Angryk, 2011). Some of these systems have a dynamic behaviour, i.e., their data sources have autonomy to join/leave the system and to change their own schemas. By convention, we will call these dynamic data integration systems as data integration systems and their data sources as peers.

Aiming to reduce the search space for queries, their response time and to diminish the message traffic on the network, some data integration systems organize their networks in clusters (peer grouping) (Raftopoulou and Petrakis, 2008; Montanelli et al., 2011; Kantere et al., 2008). According to Raftopoulou and Petrakis (2008), the query processing in a data integration environment can be improved if peers are grouped. In order to group peers, semantic similarity measures (Montanelli et al., 2011; Pires et al., 2012) between peer schemas may be used. Once these peers are grouped, a query can be sent to the cluster that may offer the best answer. Data integration systems like ESTEEM (Montanelli et al., 2011), SPEED (Pires et al., 2012) and OntoZilla (Joung and Chuang, 2009) create their clusters according to the semantic similarity of their schemas. In these systems, peers' schemas are represented by a conceptual ontology, called *peer ontology*. In addition, some systems have a conceptual ontology that represents the cluster schema, which is called *cluster ontology*.

Some of the clusters-based systems have two levels of connections between clusters: the intra-cluster and the inter-cluster level (Ayyasamy and Sivanandam, 2010). Each of these levels has an overlay network whose connections are established through the semantic similarity of the respective data source schemas. The intra-cluster level comprises the connections between peers that belong to the same cluster. In this level, a connection between peers is established only if there is a minimum semantic similarity between the peer ontology and the cluster ontology. On the other hand, the connection at the inter-cluster level comprises the

Rocha Silva E., Farias Lóscio B. and Carolina Salgado A..
A Proposal to Maintain the Semantic Balance in Cluster-based Data Integration Systems.
DOI: 10.5220/0004897000900098
In *Proceedings of the 16th International Conference on Enterprise Information Systems* (ICEIS-2014), pages 90-98
ISBN: 978-989-758-027-7

connections between different clusters. In this case, a minimum semantic similarity between clusters' ontology is required. Then, clusters that are connected to other clusters are called semantic neighbors.

The dynamic behaviour of peers in this type of data integration systems may cause evolutions of clusters' ontologies. These evolutions can result in a situation in which a cluster has peers and/or neighbors with low semantic similarity. When this situation occurs, we say that the cluster is semantically unbalanced. This unbalance can occur both at the intra and at the inter-cluster level.

A semantic unbalance may cause an undesirable behaviour during query processing on cluster-based data integration systems. At the intra-cluster level, with the existence of peers with low semantic similarity with the cluster, these peers will probably not contribute to respond expressively the queries that arrive to the cluster. These peers could contribute more to other clusters with which they have more semantic similarity.

As the inter-cluster connections are established by semantic similarity between clusters, when a query is submitted it is routed through inter-cluster connections. When semantically unbalanced clusters exist at the inter-cluster level, a query can be routed to clusters which do not offer meaningful results for this query. Furthermore, this situation will contribute to the degradation of query response time.

We propose, in this paper, a solution to the semantic unbalance problem, which is based on a self-organization approach, i.e., without human intervention. This paper examines the types of dynamic behaviours (peer joining/leaving or peer schema evolution) that may cause the semantic unbalance of clusters, and proposes a solution that automatically produces semantically balanced clusters. To evaluate the network organization before and after actions of semantic balance, we use the measure called clustering efficiency (Raftopoulou and Petrakis, 2008). Furthermore, we introduce a measure that quantifies the level of semantic balance of all the network of peers, called semantic balance coefficient.

The remainder of this paper is organized as follows. Section 2 discusses the environment that is used to illustrate the semantic unbalance problem. The schema evolution in data integration systems will be discussed in Section 3. Section 4 discusses how this evolution may cause semantic unbalanced clusters. Section 5, describes our solution based on self-organization (Conforti et al., 2004; Pires et al., 2012) (without human intervention) for the semantic unbalance problem. Section 6 presents measures that evaluate the network organization. Our experiments and results are described in Section 7. Related works and concluding remarks are in Sections 8 and 9, respectively.

2 THE SPEED SYSTEM

In this section, we present a semantic cluster-based dynamic data integration system called SPEED (Semantic PEEr Data Management System) (Pires et al., 2012). In SPEED, the peers represent their schemas through ontologies. In addition, each cluster has its own cluster ontology which describes the schemas of their peers. The cluster is managed by a peer called super-peer. The super-peer is a peer belonging to the cluster, which has more processing capacity and good network connectivity. It is responsible for managing query processing and data integration.

The cluster ontology is stored in the super-peer. In SPEED, the semantic similarity between peers is computed using a tool called SemMatcher (Pires et al., 2012). This tool performs a semantic-based ontology matching process that considers, besides the traditional terminological and structural matching techniques, a semantic-based one. The matching process produces a set of semantic correspondences (alignments) and a Global Similarity Measure (GSM) between the two matched ontologies. The GSM is calculated considering the semantic similarity measure of the generated correspondences.

For a peer to join a cluster, the GSM between the peer ontology and the cluster ontology should be greater than or equal to the threshold of minimum semantic similarity of the cluster, called cluster threshold. However, if this similarity is a value below the cluster threshold, the peer will find another semantic similar cluster or form a new one. This new cluster will be a neighbor of another cluster if the semantic similarity between their cluster ontologies is greater than or equal to the minimum neighborhood threshold, called neighbor threshold.

When a peer joins a cluster, a merge between its peer ontology and the cluster ontology is performed to generate a new cluster ontology. The SPEED uses the OntMerger tool (Pires et al., 2012) to perform the merge between ontologies. The OntMerger tool takes as arguments two ontologies (i.e, the cluster ontology and the peer ontology) and the set of correspondences between them (generated by

SemMatcher). As a result, the tool produces a new version of the cluster ontology containing the elements of both input ontologies as well as the semantic correspondences between the new cluster ontology and the peer ontology.

3 SCHEMA EVOLUTION IN DYNAMIC DATA INTEGRATION SYSTEMS

As stated by (Curino et al., 2013), the schema evolution is a change likely to happen over the life cycle of the systems, but it needs to be considered so as not to affect the queries. Actions should be taken so that the queries are not harmed, by having empty results or different results from what was expected (Genevès et al., 2011).

Once the data integration systems are composed by data sources, the schema evolution of these sources may compromise some of the functionalities of the integration system (such as queries, composition of semantic clusters, among others). When there is a schema that represents a cluster (cluster ontology), based on the peers' schemas, the concern is even worst because the cluster schema should describe the real schemas of the peers within the cluster. Furthermore, all intra and inter-cluster connections are established based on clusters' schemas. The cluster ontology can evolve in three situations, as described below:

i) when a peer joins a cluster: in this case, a merge between the current cluster ontology and the peer ontology is performed, evolving the cluster ontology.

ii) when a peer leaves a cluster: a peer may leave the cluster for some reasons: problems in the physical network, the peer wants to leave the integration system or it no longer has a minimum semantic similarity with the cluster. When a peer leaves a cluster, the schema elements that belong exclusively to this peer should be removed from the cluster ontology causing the evolution of the cluster schema.

iii) when a peer schema evolves (evolution of the peer ontology): the evolutions are related to changes in the logical definition of the peer data source (add, delete, reset, or split columns, change the cardinality of the relationship from 1: N to M: N, among others) (Sockut and Iyer, 2011). These evolutions should be also reflected in the cluster ontology.

In all described situations, the cluster schema evolution may cause problems in the peer clustering, which will be described in next section.

4 SEMANTIC UNBALANCE

The evolution that occurs in the clusters' ontologies can let the clusters on a state in which some intra-cluster and inter-cluster semantic connections become below the established thresholds. When this occurs, we say that there is a semantic unbalance at the intra-cluster and/or at the inter-cluster levels. We present the definitions related to the unbalance problem as follows.

4.1 Intra-cluster Semantic Unbalance

Let C be a cluster and $Sp = \{p_1, p_2, p_3, p_4, ..., p_n\}$ the set of peers belonging to C. C is semantically unbalanced at the intra-cluster level if there is, at least, one peer $p_i \in Sp$; in such a way that the GSM between $O(p_i)$ and $O(C)$ is below the cluster threshold, denoted by θ. Thus, we can define the intra-cluster semantic unbalance in the following way:

$\forall p_i \in Sp, \exists p_i \mid \text{GSM}(O(p_i), O(C)) < \theta$, C is semantically unbalanced at the intra-cluster level.

There are two actions that can cause an intra-cluster semantic unbalance: either by a peer joining the cluster or by the evolution of a peer ontology. Although a peer leaving the cluster provokes an evolution of the cluster ontology, it does not result in an intra-cluster unbalance. When a peer leaves the cluster the exclusive elements of its peer ontology will be removed from cluster ontology. Therefore, the semantic similarity between others peers' ontologies and the cluster ontology can remain the same or it can be increased.

We will present an example in order to illustrate how the evolution of the cluster ontology may cause a semantic unbalance. Consider that the system wants to integrate five peers p_1, p_2, p_3, p_4 and p_5. The peers' schemas are represented by ontologies and are part of the same domain (for example, *Education* domain). Suppose that these peers will form a single cluster C, i.e., each peer ontology has a semantic similarity with C cluster ontology greater than or equal to the cluster threshold (θ). We assume that value of θ is equal to 0.7. The C ontology is denoted by $O(C)$ and the p_i ontology by $O(p_i)$. The p_1 will be the first peer to enter C and $O(C)$ will be equal to $O(p_1)$. In the sequence, the others peers will join C as summarized in Table 1. The table columns mean:

- Peer: the peer which is joining the cluster C.

- Schema Evolution: the way as the *C* ontology evolved. The OntMerger tool was used to performer the merge between cluster ontology and peer ontology.
- GSM: the GSM between $O(C)$ and $O(p_i)$, when p_i is joining the cluster. The GSM values were computed by the SemMatcher tool.

When *p2* joins *C*, the merge between $O(C)$ and $O(p_2)$ is performed, evolving the $O(C)$. After, the similarity measures are recomputed. We can observe the GSM between $O(C)$ and $O(p_1)$ decreased from 1 to 0.84, after p_2 joins the cluster. However, all the GSM remain above θ. Similar to p_2 join, when p_3, p_4 and p_5 join the cluster *C*, the $O(C)$ also evolves. According to Table 1, after p_5 has joined the cluster,

Table 1: Example of semantic unbalance due evolution of cluster ontology.

Peer	Schema Evolution	GSM
p_1	$O(C)=O(p_1)$	$p_1 = 1.0$
p_2	merge($O(p_2)$,$O(C)$)	p_1=0.84 p_2=0.92
p_3	merge($O(p_3)$,$O(C)$)	p_1=0.75 p_2=0.85 p_3=0.85
p_4	merge($O(p_4)$,$O(C)$)	p_1=0.72 p_2=0.81 p_3=0.81 p_4=0.79
p_5	merge($O(p_5)$,$O(C)$)	p_1=0.60 p_2=0.71 p_3=0.70 p_4=0.69 p_5=0.85

there were two intra-cluster semantic unbalances within *C*. We can observe that GSM($O(p_1)$,$O(C)$) and GSM($O(p_4)$,$O(C)$) decreased from 0.72 and 0.81 to 0.60 and 0.69, respectively (below θ). These two peers are provoking the semantic unbalance of *C*.

4.2 Inter-cluster Semantic Unbalance

Let *C* be a cluster and $Nc = \{v_1, v_2, v_3, v_4,..., v_k\}$ the set of neighbor clusters of *C*. *C* is semantically unbalanced at the inter-cluster level if there is at least a neighbor cluster $v_j \in Nc$, such a way that the GSM between $O(v_j)$ and $O(C)$ is below the neighbor threshold, denoted by φ. In this way, we can define the inter-cluster semantic unbalance in the following way:

$\forall v_i \in Nc, \exists\ v_i \mid \text{GSM}(O(v_i), O(C)) < \varphi$, *C* is semantically unbalanced at the inter-cluster level.

The evolutions occurred in the cluster ontologies can cause a semantic unbalance at the inter-cluster level. As the inter-cluster level is established according to the GSM computed between cluster ontologies, if at least one of these ontologies changes, the semantic similarity between them can decrease for a value below φ.

To understand how changes in the cluster schema may cause inter-cluster semantic unbalance, let's consider *C* formation according to the Table 1. After p_5, the peer p_6 joined the system and the GSM between $O(C)$ and $O(p_6)$ is equal to 0.4. This value is below to θ, thus p_6 will form its own cluster (*C'*). Considering φ equals 0.4, *C* and *C'* will become semantic neighbors. Suppose that in other time, the peer p_7 joined the system and the values of GSM($O(C)$,$O(p_7)$) and GSM($O(C')$,$O(p_7)$) are 0.53 and 0.83, respectively. Thus, p_7 will join *C'*, evolving $O(C')$. After p_7 joined, the value of GSM($O(C)$,$O(C')$) decrease from 0.4 to 0.37, value above the φ. Therefore, the join of p_7 caused an inter-cluster semantic unbalance.

5 ACTIONS TO SEMANTIC BALANCE

In this paper, we propose a self-organization based solution for the previously described semantic unbalance problem. In our proposal, each cluster is able to self-organize in order to keep the semantic balance at the intra-cluster level as well as at the inter-cluster level, with no need of human intervention.

In a general way, the solution for the semantic unbalance problem consists of periodically checking if there is a semantic unbalance both at the inter-cluster and at the intra-cluster level. This can be done recalculating the semantic similarity measures between the cluster ontology and the peers' ontologies, as well as between clusters' ontologies.

The cluster could check if there is a semantic unbalance when the actions that can provoke the evolution of cluster ontology are detected. When this occurs, all the GSM must be recomputed and the cluster checks whether they are below the thresholds. This action may cause an overload in the cluster due to the dynamic nature of this type of data integration system. To avoid this overload, we will perform, periodically, the detection of the semantic unbalance situation. This period can be set in the system. Figure 1 illustrates the semantic balance algorithm for the intra-cluster and inter-cluster levels.

Let C_i be a cluster and $Sp = \{p_1, p_2, p_3, p_4,..., p_n\}$ the set of peers belonging to C_i (lines 9 and 10). C_i has the super-peer s_i (line 11) and the set of semantic neighbors $S_V = \{v_1,..., v_m\}$ (line 12). Initially, the semantic balance solution tries to repair the problems at the intra-cluster level, given that the evolution of the cluster ontology, due to semantic balancing actions in this level, can cause a semantic unbalance at the inter-cluster level. However, the balance actions at the inter-cluster level do not impact the intra-cluster level because the cluster ontology remains unchanged with the input or output of new neighbors.

At the intra-cluster level, each unbalanced peer p_i will be inserted into a list L_P in ascendant order, according to the GSM between $O(p_i)$ and $O(C_i)$ (lines 15 to 20). If the super-peer is an unbalanced peer (line 22), a new super-peer among the other peers that are not unbalanced (lines 24 and 25). This new super-peer will store $O(C_i)$ (line 26). While L_p is not empty (line 30), for each peer p_k removed from L_P (lines 32 and 33), s_i sends a message, to each neighbor v_j, to search for a new cluster to receive p_k (lines 36 and 37). The GSM of the remaining peers of L_P are recalculated (line 34) because with the output of p_i, it is likely that some peers in this list have become balanced, as discussed in Section 4.1. The message contains the $O(p_i)$. The search will end when a particular TTL (Time-to - Live), is achieved (line 39). Only the clusters that have greater semantic similarity and capacity to absorb the load generated by p_k will send back to s_i a message of interest by the peer (lines 41 and 42). Each interested cluster is added into a list S_C (line 43).

```
1.  Algorithm Semantic Balance()
2.  {
3.     θ, φ:Threshold;
4.     v,Ci : Cluster;
5.     SC, SV: Set of Clusters;
6.     pi,pk,si : Peer;
7.     SP,LP: Set of Peers;
8.
9.     Ci ← retrieve current cluster;
10    SP ← retrieve the peers of Ci;
11    si ←retrieve current super-
    peer;
12    SV ← retrieve neighbors of Ci;
13
14    LP ← ∅;
15    FOR EACH pi IN SP DO
16      IF (GSM(O(pi),O(Ci))< θ) THEN
17      {
18        add pi into LP in ascendant
19          order by GSM(O(pi),O(Ci));
20      }
21
22    IF si ∈ LP THEN
23    {
24       si ← select new super-peer
25             from {SP – LP};
26      store O(Ci) into si;
27    }
28
29    SC ← ∅;
30    WHILE (LP ≠ ∅) DO
31    {
32       pk ← remove first element
33             from LP;
34       IF (GSM(O(pk), O(Ci))< θ)
    THEN
35       {
36          FOR EACH vj IN SV DO
37            send O(pk)to vj;
38       }
39       WHILE (TTL ≠ 0)
40       {
41          Cz ← receive interested
42                cluster;
43          add Cz into SC;
44       }
45       IF (SC ≠ ∅) THEN
46       {
47          Cz ← select cluster from
48                SC with greater
49                GSM(O(pk), O(Ci);
50          unmerge(O(pk),O(Ci));
51          disconnect pk from Ci;
52          connect pk to Cz;
53       }
54       ELSE
55         create new cluster with
    pk;
56    }
57
58    FOR EACH vj IN SV DO
59    {
60       IF (GSM(O(vj),O(Ci))< φ) THEN
61          disconnect vj from Ci;
62    }
63 }
```

Figure 1.

After receiving all messages of interest, s_i will choose from S_C, the cluster C_z which has the highest semantic similarity to receive p_k (lines 47 to 49). s_i will run the "*unmerge*" between the $O(C_i)$ and $O(p_k)$ (line 50) and disconnect p_k from itself (line 51). The "*unmerge*" is inverse to the merge operation and it

removes the schema elements that exclusively belong to $O(p_k)$ from $O(C_i)$. Afterwards, p_i will connect to C_z (line 52). If no cluster shows interest by p_i, this peer will form its own cluster, according to the process of cluster formation established by the system (lines 54 and 55).

Once the actions of intra-cluster balance are finished, it is time to check whether there is an inter-cluster semantic unbalance and then balance them. During the inter-cluster balancing, the GSM for each v_j in S_V is recomputed (lines 58 to 60). Each unbalanced v_j is disconnected from C_i (line 61). However, C_i periodically sends a message in the network to search for new semantic neighbors. The clusters that have minimum semantic similarity with $O(C_i)$ will be connected to C_i.

6 MEASURING CLUSTERING QUALITY

We introduce the semantic balance coefficient $\bar{\lambda}$ as a measure that quantifies the level of semantic balance of the whole network. To evaluate the network organization, i.e., the set of clusters that composed the system, before and after the semantic balance actions, we present a measure called clustering efficiency $\bar{k}$ (Raftopoulou and Petrakis, 2008) that quantifies network organization by exploiting the underlying network structure.

6.1 Clustering Efficiency

To measure the efficiency of the network organization, we use the clustering efficiency measure $\bar{k}$ (Raftopoulou and Petrakis 2008), which is defined based on the set of clustering efficiency measures for the clusters that compose the network. Formally, the clustering efficiency k_i for a cluster C_i is defined as the number of peers p_j semantically similar to C_i ($\text{sim}(O(C_i),O(p_j)) \geq \theta$) that can be reached from C_i within a TTL, divided by the total number of peers p_k in the network similar to C_i. The $dg()$ function is the distance (measured in number of hops) of peers in the network.

$$ki = \frac{\sum_{j=1}^{N} pj : \{dg(Ci, pj) \leq TTL, sim(O(Ci), O(pj) \geq \theta\}}{\sum_{k=1}^{N} pk : \{sim(O(Ci), O(pk))\}} \quad (1)$$

The clustering efficiency for the network as a whole $\bar{k}$ is defined as the clustering efficiency average (over all clusters in the network) . We use the clustering efficiency to evaluate the peer clustering after and before the semantic balance actions.

$$\bar{k} = \frac{1}{N}\sum_{i=1}^{N} ki \quad (2)$$

The clustering efficiency measure gives information about the underlying overlay network, and looks at how the network is structured at a larger scale. Clustering efficiency measure will have values in the interval [0;1]. According to (Raftopoulou and Petrakis, 2008), the highest the value of clustering efficiency is, the better the underlying network organization is.

6.2 Semantic Balance Coefficient

To evaluate how much the network is semantically balanced, we introduce the semantic balance coefficient measure based on recall measure (Rijsbergen, 1979). We propose this measure because the clustering efficiency measure doesn't consider the intra-cluster and inter-cluster connections that are semantically unbalanced. For each cluster we considered the number of intra and inter-cluster connections that are semantically balanced over the total number of cluster connections.

First, we defined the intra-cluster semantic balance coefficient α for a cluster C_i as the number of peers p_i belonging to C_i and whose $\text{GSM}(O(C_i),O(p_i))$ is greater than or equals θ, divided by the total of intra-cluster connections n in C_i, i.e., the number of peers within C_i:

$$\alpha = \frac{1}{n}\sum_{i=1}^{n} pi : \{pi \in Ci, GSM(O(Ci), O(pi)) \geq \theta\} \quad (3)$$

The α measure is equal to one if the whole set of peers p_i of C_i is semantic balanced. Conversely, the α measure is equal to zero if all peers of C_i are semantic unbalanced.

Similar to the α, the inter-cluster semantic balance coefficient β for a cluster C_i is defined as the number of C_i neighbors (v_i)whose GSM ($O(C_i)$, $O(v_i)$) is greater than or equals φ, divided by the total of inter-cluster connections m in C_i, i.e., the number of neighbors of C_i.

$$\beta = \frac{1}{m}\sum_{j=1}^{m} vi : \{vi \in Ci, GSM(O(Ci), O(vi)) \geq \phi\} \quad (4)$$

The β measure is equal to one if all neighbors of C_i are semantically balanced at the inter-cluster level. Conversely, the β is equal to zero if all

neighbors of C_i are semantically unbalanced at the inter-cluster level. So, the semantic balance coefficient λi for a cluster C_i is the ratio between its α and β coefficients and its number of intra and inter-cluster connections:

$$\lambda i = \frac{\alpha + \beta}{n + m} \quad (5)$$

The semantic balance coefficient for all the network $\overline{\lambda}$ is the ratio between the sum of all λi and the number of clusters N in the network.

$$\overline{\lambda} = \frac{1}{N} \sum_{k=1}^{N} \lambda i \quad (6)$$

The semantic balance coefficient is a measure that gives information about how the network is semantically balanced at a large scale. This measure is used to evaluate our solution for the semantic unbalance problem, as we discuss in the next section.

7 EVALUATION

In this section, we present the results of some tests performed to evaluate our proposal. The tests were performed using the SPEED prototype. The main goal of the proposed experiment is initially to create some clusters and in the next step to evaluate the network organization when one or more clusters become unbalanced. To evaluate our proposal, we consider two different scenarios: in the first one, our proposal was used to solve the semantic unbalancing problem and in the second scenario our proposal was not considered.

Both the prototype and our solution to the semantic balance were implemented in Java language. Our experiments were done on a single CPU (Core 2 Duo E8400, 3Ghz), simulating forty five (45) peers. Issues about performance were evaluated and discussed in (Silva et al., 2013).

7.1 Experimental Set-up

Our experiments have been performed in a set of peer ontologies of the *Education* domain. Each peer ontology is represented in OWL (OWL, 2013) and has about six concepts on average. The cluster threshold was set to θ=0.7 and the neighbor threshold to φ=0.4.

Five different scenarios were generated with different orders of new peer joins, forming twenty one clusters on average. In each scenario some peers were randomly chosen to evolve their schemas. At each time interval of 200K milliseconds, the clusters check whether there are semantic unbalances. To guarantee that all clusters were visited in the semantic balance solution, we set the TTL to the number of peers, i.e., equals 45.

7.2 Experimental Validation

Figures 2 and 3 illustrate an overview of the network, considering the measures $\overline{\lambda}$ and $\overline{k}$. The measures were computed two times for the situations: (i) without, and (ii) with semantic balance actions. Therefore, each point on the graphics is the average value of five scenarios to both situations.

Figure 2 illustrates the variations of clustering efficiency $\overline{k}$ for each peer join operation, in the two situations. The graphics pointed out that when the semantic balance actions are performed, $\overline{k}$ has better results. In situations (i) and (ii) (Figure 2) the values of $\overline{k}$ decreased with the join of some peers. These decrease are justified by the dynamic behavior of system that allowed new clusters, semantically balanced, to be formed with the join of new peers in the system.

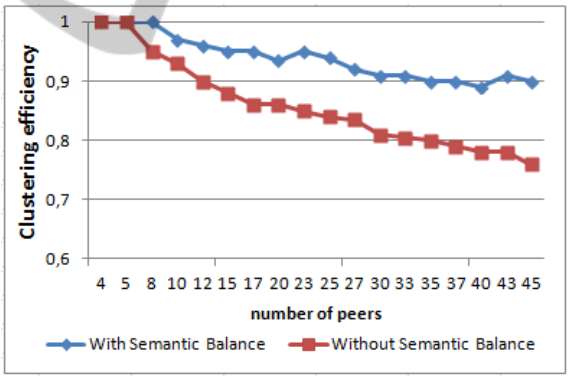

Figure 2: Clustering efficiency as a function of peers joining the network with or without semantic balance actions.

Figure 3 illustrates how semantic balance coefficient varies over time, in the two situations. In situation (i) the coefficient has worst value. After 9x200K milliseconds the coefficient was equal to 0.5, i.e., fifty percent of the connections in the network were semantically unbalanced. In situation (i) the values of $\overline{\lambda}$ increased in the time intervals 2x200K<t<3x200K and 5x200K<t<8x200K. These increase are justified by the dynamic behavior of the network that allows new clusters, semantically balanced, to be formed with new peers joining the

system. However, the other clusters remained unbalanced.

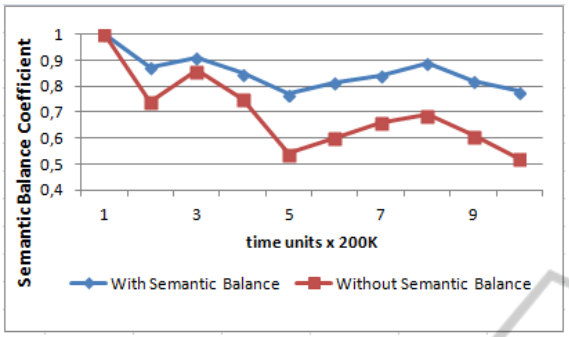

Figure 3: Semantic balance coefficient as a function of time with or without semantic balance actions.

Also in Figure 3, we observe that in situation (ii) the values of $\overline{\lambda}$ remain above the values found in (i). Over time, the semantic balance actions self-organize the network as close as possible to a balanced condition, i.e., $\overline{\lambda}$ close to value 1. In situation (ii) (Figure 3) the values of $\overline{\lambda}$ decreased in the time intervals 1x200K<t<2x200K, 3x200K<t<5x200K, and 8x200K<t<10x200K. This situation is also justified by the system dynamic behaviour that allows new peers to join/leave the system or peers to evolve their schema, provoking new semantic unbalances.

The experiments above showed that the use of our approach for semantic balancing achieves better values of clustering efficiency and semantic balance coefficient. Even with the system dynamic behaviour, our solution presents better results. The semantic balance actions provoked a network reorganization, but the clustering efficiency measure indicated better peer clustering after this actions. However, the issues related to cost of cluster reorganization (like complexities in time and message traffic) will be precisely analyzed in the future works.

8 RELATED WORK

In this section, we make a brief summary of some works similar to ours, which intend to maintain the semantic balance of their clusters.

In (Montanelli et al., 2011), a peer forms a cluster (founder peer) based on its schema. The schema that represents the cluster is the one related to the peer that created it. In this work, only the intra-cluster level is established. The peer joining/leaving or the schema evolution of the other peers do not make the cluster schema to evolve. Only the schema of the founder peer will evolve and it represents the cluster. New peers can be found and added to the cluster by means of probe message that is submitted to the network. If cluster finds new peers more similar than the others already belonging to the cluster, it will replace the less similar peer by the more similar new one.

Kantere *et al.* (2008) developed an algorithm for cluster peers considering a minimum value of the similarity threshold. There is a schema that represents the cluster and it is obtained through merging operations of the schemas of the sources. The algorithm is able to detect the peers of the clusters that had its semantic similarity decremented due to the evolution of the cluster schema. As only connections at the intra-cluster levels are established, Kantere *et al.* (2008) solve the semantic unbalance only at this level.

In (Raftopoulou and Petrakis, 2008), the cluster discards the less similar peers by the more similar ones found in the network. Each cluster starts a process of reconnection of the more similar peers every time there is an intra-cluster semantic unbalance. The reconnection procedure finds the less similar peers and substitutes them by the more similar ones according to an established cluster threshold. There is no inter-cluster level of connections.

Conforti *et al.* (2004) describe a set of formation of clusters following a super-peer topology. Each super-peer owns a list containing the peers' schemas connected to it and it integrates them, i.e., it creates a super-peer schema (representing the general view of the cluster schema). By means of a self-organization process, the peers are grouped in clusters according to the semantic similarity between the peers' schemas. Inter-cluster connections are established between semantically similar clusters, considering the super-peers' schemas. Nevertheless, there is no procedure that identifies the semantic unbalance and that fulfils the necessary semantic balance at the intra-cluster and inter-cluster levels.

Compared to the works discussed above, our algorithm carries out the semantic balance both at the intra-cluster and at the inter-cluster levels. Furthermore, it can be used in a super-peer topology or not. Any schema evolution that causes the semantic unbalance is also considered.

9 CONCLUSIONS AND FUTURE WORKS

In this paper, we propose a solution to solve the problem of semantic unbalance of clusters at the intra and at the inter-cluster levels in dynamic data integration environments. Our proposal ensures that the clusters will always have semantically similar peers according to the established threshold for both intra-cluster and inter-cluster connections. The experimental results demonstrated that our solution is able to detect a semantic unbalance problem and to perform the corresponding balance actions in order to keep the semantic balance of the systems clusters. We are currently improving the solution to reduce the overload, over the cluster, when verifying the semantic unbalance.

REFERENCES

Ayyasamy, S., and Sivanandam, S., 2010. A Cluster Based Replication Architecture for Load Balancing in Peer-to-Peer Content Distribution. *International Journal of Computer Networks & Communications (IJCNC)*, vol.2, pp. 158-172.

Conforti, G., Ghelli, G., Manghi, P., and Sartiani, C., 2004. A Self-organizing XML P2P Database System. *Proceedings of the 2004 international conference on Current Trends in Database Technology*, pp. 456-465.

Curino, C., Moon, H. J., D., Alin, and Zaniolo, C., 2013. Automating the database schema evolution process. *Published in The VLDB Journal – The International Journal on Very Large Data Bases*, vol. 22, pp. 73-98.

Genevès, P., Layaïda, N., and Quint, V., 2011. Impact of XML Schema Evolution. *Published in Journal ACM Transactions on Internet Technology (TOIT)*, vol. 11, article 4.

Halevy, A., Rajarama, A., and Ordille, J., 2006. Data Integration: The Teenage Years. Proceedings of the 32nd International Conference on Very large data bases, pp 9-16. Seoul, Korea.

Halevy, A., Sarma, A. D., and Dong, X., 2008. Bootstrapping pay-as-you-go data integration systems. *Proceeding of the 2008 ACM SIGMOD International Conference of Data*, pp. 861-874. Vancouver, Canada.

Joung, Y., and Chuang, F., 2009. OntoZilla: An ontology-based, semi-structured, and evolutionary peer-to-peer network for information systems and services. *Journal of Future Generation Computer Systems*, vol. 25, n° 1, pp. 53-63.

Kantere, V., Tsoumakos, D., and Sellis, T. ,2008. A framework for semantic grouping in P2P databases. *Published in Journal Information Systems*, vol. 33, pp. 611-636.

Montanelli, S., Bianchini, D., Aiello, C., Baldoni, R., Bolchini, C., Bonomi, S., Castano, S., Catarci, T., Antonellis, V., Ferrara, A., Melchiori, M., Quintarelli, E., Scannapieco, M., Schreiber, A., and Tanca, L.,2011. The ESTEEM platform: enabling P2P semantic collaboration through emerging collective knowledge. *Published in Journal of Intelligent Information Systems*, vol. 36, n° 2.

Pires, C. E., Santiago, R., Kedad, Z., Bouzehoub, M. and Salgado, A. C., 2012. Ontology-based Clustering in a Peer Data Management System. *Published in International Journal of Distributed Systems and Technologies (IJDST)*, vol. 3, Issue 2, pp. 1-21.

Raftopoulou, P., and Petrakis, E. G. M., 2008. A Measure for Cluster Cohesion in Semantic Overlay Semantic. *Proceedings of the 2008 ACM workshop on Large-Scale distributed systems for information retrieval*. Napa Valley, USA.

Rijsbergen, C. J., 1979. Information Retrieval, 2nd Edition, MA: Butterworths.

Roth, A., and Skritek, S., 2013. Peer Data Management. *In Data Exchange, Information and Streams,* vol. 5, pp. 185-215.

Silva, E. R., Salgado, A. C., 2013. Load Balance for Semantic Cluster-based Data Integration Systems. *Proceeding of the 17th International Database Engineering & Applications Symposium (IDEAS'13)*. Barcelona, Spain.

Sockut, G. H., and Iyer, B. R, 2011. Online Reorganization of Databases. *Published in Journal ACM Computing Surveys (CSUR)*, vol. 41, article 14.

Terwilliger, J. F., Bernstein, P. A., and Unnitha, A., 2010. "Worry-Free Database Upgrades: Automated Model-Driven Evolution of Schemas and Complex Mappings". Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 1191-1194. Indianapolis, USA.

Tian, Y., Song, B., and Huh, E. N.. "Dynamic content-based cloud data integration system with privacy and cost concern". Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, pp. 193-199. Redmond, USA. 2011.

Wall, B., and Angryk, R., 2011. Minimal Data Sets vs. Synchronized Data Copies in a Schema and Data Versioning System. *Proceedings of the 4th workshop on Workshop for Ph.D. students in information & knowledge Management*, pp. 67-74. Glasgow, United Kingdom.

W3C. "OWL – Web Ontology Language", 2013. Available in http://www.w3.org/TR/owl-features. Accessed on October 1st..

Zamboulis, L., Martin, N., and Poulovassillis, A., 2010. Query performance evaluation of an architecture for fine-grained integration of heterogeneous grid data sources. *Published in Journal Future Generation Computer Science Systems*, vol. 26, pp. 1073-1091.