Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 60 (2015) 197 – 205

Procedia Computer Science

19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

Adaptive global schema generation from heterogeneous metadata schemas

Rim Zghal Rebaï[a,] *, Fatma Mnif[a], Corinne Amel Zayani[a] and Ikram Amous[a]

[a]*MIRACL-ISIMS, Sfax University, Tunis Road Km 10, 3021 Sfax, Tunisia*

Abstract

The access to heterogeneous data through their metadata needs a matching process of the metadata schemas. This process identifies the correspondence relations called "Mappings" between the schemas to identify a global schema. This latter allows a uniform access to heterogeneous data. In this context, several works are proposed. However, the obtained mappings and the global schema are identified regardless of the user's profile. Thus, the queries results are the same for any user despite the various profiles. In this paper, we present a matching process that (i) deals with the metadata schema heterogeneities and (ii) considers the users' profile.

Peer-review under responsibility of KES International

Keywords: Heterogeneous metadata schema; matching process; mappings; global schema; adapted global schema.

* *E-mail address:* rim.zghal@gmail.com / rim_zghal@yahoo.fr

Peer-review under responsibility of KES International
doi:10.1016/j.procs.2015.08.119

1. Introduction

Metadata is data that describe, identify and improve the filtering and retrieval of other data. It can be also heterogeneous at the level of schemas and languages. Therefore, a uniform access to the latter becomes more and more difficult. For this reason, the heterogeneous metadata integration becomes crucial. In this context, several techniques have been proposed among which we cite: (i) the standardization of schema definition language (e.g., XML Schema, RDF Schema, OWL, etc.) which addresses the schema heterogeneities and (ii) the standardization of metadata schema (e.g., MPEG-7, MPEG-21, Dublin Core, etc.) which can eliminate or reduce the heterogeneities of the description languages. However, for many reasons (conceptual, strategical, etc.), it is impossible to use one of these techniques (one single schema definition language standard or metadata schema standard). Moreover, other solutions have been proposed. An analysis of these solutions was performed in[7] and showed that metadata mapping is the appropriate technique. That is why we are focusing on it in this paper.

Mappings are the semantic correspondence relations between two different schemas[11]. They are identified after a schema matching process[14]. Hence, the latter allows creating a schema which contains all the information on the metadata to be integrated. It is called “Global schema” and acts as a common interface used for querying all the heterogeneous metadata. Once the global schema is identified, a manual mapping is performed in order to extract the existing mappings between the metadata. Nevertheless, this manual mapping, which is costly and time-consuming, was not a success. As a consequence, several semi-automatic and automatic mapping methods are proposed to provide the user with a uniform access to heterogeneous metadata. However, while creating the global schema, these methods do not take into account the user's profile (interests, preferences, etc.). So, the obtained result of a query based on this schema is usually the same for all the users.

Motivated by the identification of an adapted global schema specific for each user, we propose, in this paper, an automatic process for heterogeneous metadata integration which takes into account the user’s preferences. Therefore, the obtained adapted global schema provides a uniform access to heterogeneous metadata and adapted result content.

This paper is organized as follows: Section 2 presents a state of the art of some works dealing with the metadata schema matching and discusses their limitations. In section 3, we describe our proposed integration process. An evaluation study is presented in section 4. Finally, we present the conclusion and future work.

2. State of the art

To ensure the uniform access to heterogeneous metadata and subsequently to their related data, a schema matching is necessary. It is defined as the generation process of the mappings between two different schemas[13]. For the implementation of this process, several techniques have been proposed. Among the most used techniques we cite: the linguistic and structural techniques. The linguistic matching techniques are mainly based on syntactic and semantic comparison between the element names. The structural matching techniques consider that the similarity between two elements (e1, e2) of two schemas (S1, S2) depends on the relation between the connected elements to (e1, e2). The latter can be the adjacent elements[9], the ascendant or the descendant ones[5]. In the literature several systems based on these techniques are proposed.

MUSE[1] and Clip[12] are based on Clio[10]. The former identifies the existing mappings between two schemas (relational or XML) by using examples of data. This can help the user understand, design and refine the identified mappings. As a result, two elements are provided to the user; MUSE-G to design the mappings and MUSE-D to interpret and refine the ambiguous mappings. As for Clip, it manipulates only XML schemas. It provides the user with a graphical interface through which he indicates the similar elements in two different schemas given as input, and based on this manual task, it generates all the existing mappings.

CUPID[8] is a system that uses the example of relational and XML data to identify the mappings. It is based on three steps: (i) a matching linguistic step which resolves the element name conflicts by using an external dictionary, (ii) a transformation step of schemas in trees and (iii) a correct mapping selection step.

MuMIE is[2] a matching system that takes into account the heterogeneities at the level of the schema and schema description languages (XML, RDF and OWL). It allows generating the existing mappings between two schemas via several steps. The first step transforms the schemas to directed labeled graphs. The identified semantic and structural

information are saved to be used in the second step. The latter consists in the matching process which normalizes the element names, calculates the linguistic, structural similarities and selects the correct mappings.

All these systems can detect the existing mappings between different schemas through a matching process. This process varies from one system to another according to the information taken into account. We note that most of the structural and semantic information excepting in MuMIE[2], is not used. Moreover based on these systems, we can obtain a global schema for querying heterogeneous data but the final result is usually the same for users with different profiles.

The user's profile contains metadata that describe the user's characteristics. According to Brusilovsky[3]: "a user's profile is composed of a set of categories: personal data, user's knowledge, interests, history, and preferences". It is a basic component in adaptive systems. These systems provide each user with the adapted result (documents, links, etc.) according to his profile. Hence, they can adapt the navigation, the presentation and the content[15].

Thus, to deal with the heterogeneity problem and mainly to obtain an adapted result for each user, without using a content adaptation process, we define a process that allows getting a global schema that contains the user's preferences called "Adapted Global Schema". This schema is obtained based on the mappings identified via two matching steps. The first step is related to the heterogeneous schemas and the second one is related to the schema and the user's profile. Moreover, our process takes into account the metadata schema and the schema definition language heterogeneities, the semantic and structural information.

3. The proposed matching process

In order to interrogate heterogeneous data by taking into account the user's preferences, we propose a process that builds, from multiple heterogeneous schemas (XSD, RDFS), an Adapted Global Schema "A_GS" for each user. Therefore, the process can treat both the metadata integration and the content adaptation.

Based on the state of the art, we notice that the matching process is generally performed between two different schemas in order to identify the existing mappings. In our work, we propose to perform two different types of matching. The first called "S_Matching" is performed between the different schemas. It allows constructing a global schema "GS" and identifying the existing mappings called "S_Mappings". The second matching, which is called "P_Matching", is performed between the obtained GS and the user's profile. It allows identifying the existing mappings called P_Mappings.

3.1. The proposed user's profile Matching of schemas (S_Matching)

Before starting the S_Matching process, we extract, from each schema, the linguistic information (names of nodes), the structural information (hierarchy of nodes) and the semantic information (properties of nodes). Then, we select one of the local schemas as a temporal global schema "T_GS" and consider it as a **target schema**. This selection is based on the number of simple nodes in the first level. This means that the schema with the highest number of nodes in the first level will be selected as T_GS. In case of equality, one schema is arbitrarily selected. After that, the S_Matching process starts. It takes as input a set of heterogeneous local schemas and matches them one by one with the T_GS based on the algorithm illustrated in table 1

For each local schema graph, the algorithm S_Matching performs a matching with the selected T_GS based on the recursive function "Match" (lines 4-6) which usually starts by nodes at the first level (line 5). It verifies, for each node of the local schema graph, its existence or a similarity relation between the nodes at the same level of the T_GS based on the function called "Similarity" (line 11). If it is the case, the algorithm adds the identified reations in the mappings file (line 12) and, if the node has children nodes, function "Match" restarts the same process by taking into account the next level (lines 13-15). Thus, the hierarchy of the nodes is respected. Moreover, the node and its children nodes, if they exist, are added to the T_GS and the mappings file is updated (lines 16-18). At the end of this algorithm, we obtain the GS graph that includes all the local schemas given as input and all the existing mappings stored in an XML file called "S_Mapping". To detect the existing synonymies between the names of nodes, we should have a "Similarity" function based on the WordNet dictionary[5]. Fig. 1 illustrates a GS generated from two schemas XSD and RDFS and the obtained mappings file.

Table 1.Algorithm S_Matching

S_Matching

```
1. Input:  graphs_schema_source : List, SGtemp : Graph
2. Output:  SG: Graph
3. Begin
4. For each (graph_source in graphs_schema_source) do
5.   level=1;
6.   SGtemp:=match(N, graph_source, SGtemp);
7. end for
8. match (level, graph_source, SGtemp): Graph
9. begin
10. for each (graph_source.[level].[node]) do
11.  if(∃(SGtemp.[level].[node.name])==(
graph_source.[node.name]))or(similarite
(graph_source.[node.name],SGtemp.level]. [node.name])==true)then
12.    S_Mapping:= add_mapping (S_Mapping);
13.     if(graph_source.[node].[child]!= null) then
14.      level ++;
15.      SGtemp:=match(level,graph_source, SGtemp);
16.     end if
17.    end if
18.    else
19.    add(graph_source.[node])to(SGtemp.[level]);
20.    S_Mapping:= add_mapping (S_Mapping);
 21.   end else
22.   end for
23.     return SG:= SGtemp;
24. end match
25. End S_Matching.
```

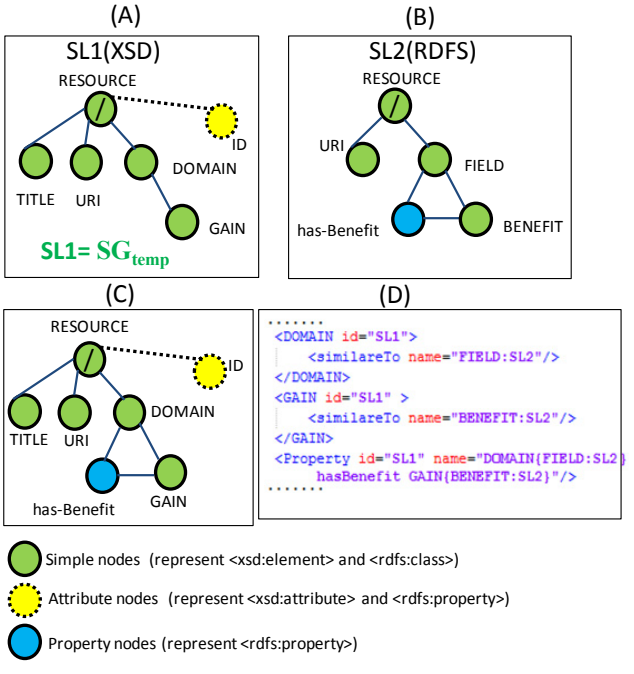

Fig. 1. Example of schema mappings: (A) XSD local schema, (B) RDFS local schema, (C) The GS result and (D) Extract of the obtained mappings

As we can see in Fig.1.D, the obtained mapping file called “S_Mapping” contains all the information about the two graphs (figure 1.D and figure 1.B) and the existing relations between them. The obtained GS is illustrated in figure 1.C. For example, for the node “GAIN”, we save: (i) its original schema “id=SL1” and (ii) its synonyms “SimilarTo name=“BENEFIT:SL2” ”. As for all the existing properties, we save the original schema and the related nodes. In Fig. 1 the property “has-Benefit” is described as follows: “Property id=“SL1” name=“DOMAIN {FIELD:SL2} has-Benefit GAIN {BENEFIT:SL2}”.

3.2. Matching of the GS and the user's profile schema (P_Matching)

The P_Matching is the new matching type that we propose to perform between the obtained GS (result of the S_Matching) and the user's profile (Specifically the user's preferences). This process has two main advantages. On the one hand, it allows enriching queries related to each data source by the user’s preferences at the same time of their rewriting. On the other hand, after each proposal of a new query in the same session, it avoids consulting the user's profile to provide preferences and enrich queries. In other words, in each new query, the system is based only on the A_GS to rewrite and enrich queries instead of visiting the user's profile each time. Therefore, the contribution of this matching essentially consists mainly in a reduction of the system response time. An overview of the general principle of the P_Matching algorithm is illustrated in table 2.

Table 2.Algorithm P_Matching

P_Matching

```
1. Input: GS, profile : graph
2. Output:  P_Mapping : XML document
3. Begin
4.preferences[]:=extract_Pref(profile);
5.  j:=0;
6.    N:=preferences[].size();
7.for (i:=1 to N) do
8.   if((preferences[i].name== SG.[node.name])or
(similarite(preferences[i].name,SG.[node.name]))   then
9.    P_mapping:= add_mapping(P_mapping);
10.   added_elts[j]:= preferences[i];
11.   j++;
12.  end if
13. end for
14. for each (k :=1 to added_elts.size()) do
15.  preference_value:= Search_Preference_Value (profile, added_elts[k]);
16.  P_mapping:= add_Value(P_mapping);
17. end for
18. return P_mapping;
19. End P_Matching.
```

The input of the P_Matching is the GS result of the S_Matching and the user's profile. It begins by extracting the preferences of the user (line 4). For each user’s preference, if it exists in the GS or similar to any node in the GS, it is added to the profile mappings file “P_Mapping” (lines 8-12). After that, for each preference found in the GS, the algorithm searches its value in the profile and adds it to the “P_Mapping”, (lines 14-17). Fig. 2 illustrates an example of GS/User's profile matching and the obtained mappings.

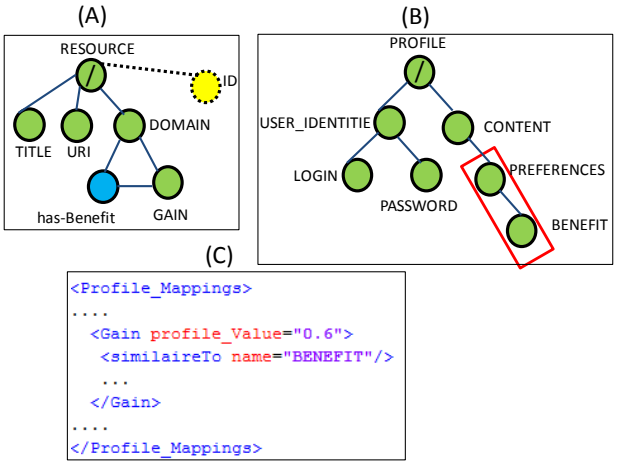

Fig. 2. Example of GS/User's profile matching: (A) The GS, (B) The user's profile graph and (C) The mapping result

As illustrated in Fig. 2, the P_Matching is performed only on the user's preferences which are detected and updated after each navigation session in the different document collections. The XML result mapping file "P_Mappings" contains all the user's preferences found in the GS. For each preference, we found : (i) its value taken from the profile, in our example "Gain profile_Value="0.6" ", and (ii) its synonyms in the profile "similarTo name="BENEFIT" ".

At the end of the two matching processes, we obtain two XML mapping files "S_Mapping" and "P_Mapping" which contain all the relations between local schemas and the user's preferences. They are very important to interrogate the heterogeneous distributed collections.

3.3. The adding of mappings

In an attempt to reduce the time of the queries rewriting process and to avoid the access to both GS and all the mapping files, we propose to add the content of "S_Mapping" and "P_Mapping" into the GS. Fig. 3 illustrates an example.

4. Evaluation

To evaluate the usefulness and effectiveness of our process, we use several document collections which are localized on different machines and have heterogeneous metadata (in terms of schema and schema definition language (XSD and RDFs)). The document collections consist of the INEX 2007 collection (110000 French version documents) divided into five collections. The first four collections are described by XML metadata (XSD) and collection 5 is described by RDF metadata (RDFs). For this purpose, we carried out a series of experiments performed by 20 users who propose 15 queries. At first, we evaluated the users' satisfactions concerning the obtained result documents. Second, we compare the obtained response time by using an A_GS and a classical GS.

4.1 Evaluation of the users' satisfactions

In order to study the impact of the A_GS on the result, we provided the users with two result versions (with A_GS and with classical GS) who indicated the pertinence of each document. A document is not-pertinent when it is removed from the adapted result (obtained based on the A_GS), while the user wants to visit it. The obtained users' satisfactions are illustrated in figure 4 which clearly shows that the users are satisfied. In fact, the average satisfaction rates obtained by using the A_GS vary between ≈ 0.94 and ≈ 0.98. Furthermore, the 20 users' average

satisfaction rate is ≈ 0.96. However, by using a classical GS, we found that these values vary between ≈ 0.51 and ≈ 0.64. As result, these values can affirm the users' satisfactions and the usefulness of the A_GS in the content adaptation result as well as the dealing with the metadata heterogeneities.

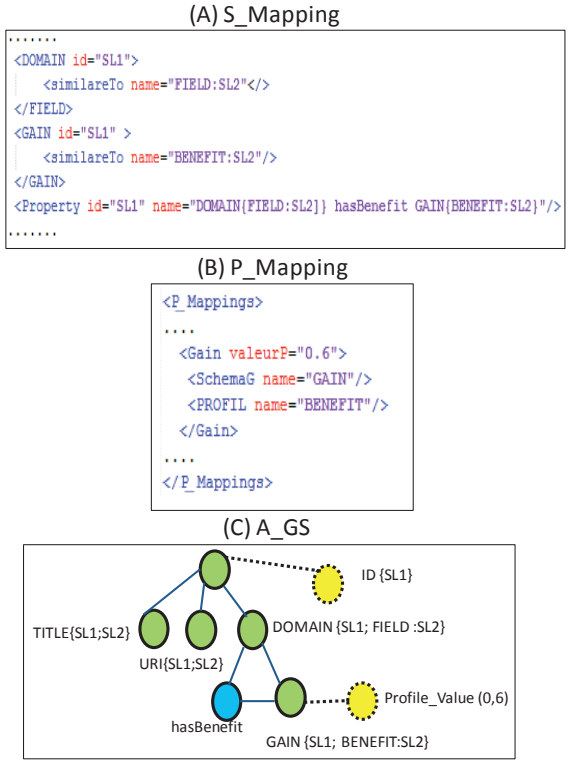

Fig. 3. Example of GS/User's profile matching: (A) The GS, (B) The user's profile graph and (C) The mapping result

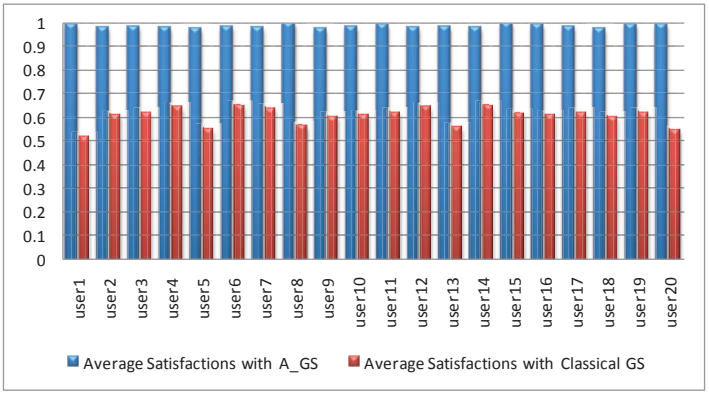

Fig. 4. The users' average satisfactions related to the result documents

4.2 Evaluation of the response time

After having studied the impact of the A_GS on the result, we studied its impact on the system time response to the users' queries. For this reason, we first measured the system response time (per second) based on the A_GS during one session. Secondly, we measured the response time of the system based on the classical GS by taking into account the users' preferences. In this case, the obtained result is also adapted and the response time includes the global schema identification time, the profile consultation time and the rewriting, enrichment and execution of the queries time. The obtained results are illustrated in Fig. 5.

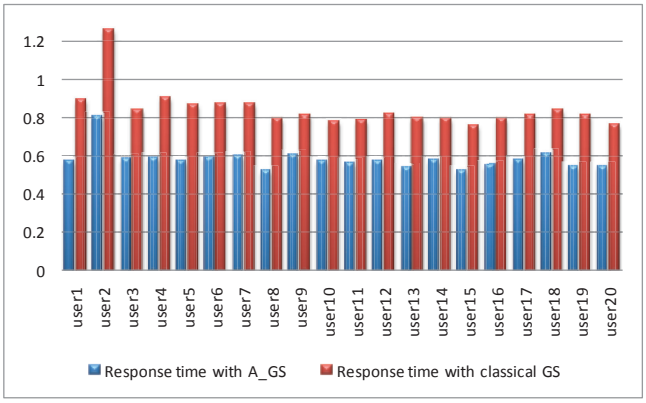

Fig. 5. Response time with A_GS and classical GS

Based on Fig. 5, we can notice that the use of the A_GS enables to obtain an adapted result in a shorter time compared to the required time to provide an adapted result based on the classical GS. The percentage of the gain varies between ≈25.9% and ≈36.4% for the users. Thus, the average gain is ≈30.8%. For example, for a session of 60 minutes, the user can save ≈18 minutes. We can explain this gain as follows. When the system uses the A_GS, it identifies the GS, consults the profile and matches it with the GS to obtain an A_GS only once. Besides, during the session, it executes all the user's proposed queries based on the A_GS. However, when the system is based on a classical GS, it identifies this schema once. On the other hand, it consults the user's profile on each new proposed query to enrich it with the user's preferences.

5. Conclusion

In this paper, we have presented a process that help us deal with (i) the metadata schema heterogeneity in order to access heterogeneous documents and (ii) the content adaptation based on an adapted global schema "A_GS". This schema is the result of the proposed process which is based on two matching steps. In the first step called "S_Matching" a classical matching is realized between the different metadata schemas. This matching is based on linguistic and structural techniques. The second step is a new matching realized between the GS result obtained by the S_Matching and the user's profile schema. This matching helps us obtain an adapted global schema related to any user's profile and adapts the content without using an adaptation process.

As it is shown in the evaluation process, the use of the A_GS can reduce the number of the result documents and the response time to generate an adapted result. The users' feedback evaluation can prove their satisfactions (average satisfaction rate is ≈ 0.96). As for the response time, we obtained ≈30.8% as an average time gain.

In the continuation of our work, we aim to propose a process that deals with the user's profile heterogeneities. Moreover, we are going to propose and implement a learning method to the heterogeneous user's profile which automatically removes the irrelevant content after several updating operations.

References

1. B. Alexe, L. Chiticariu, R. Miller, D. Pepper and W. Chiew Tan, " Muse: a system for understanding and designing mappings", In Proceedings of the International Conference on Management of Data (ACM SIGMOD), 2008, pp. 1281–1281. Doi:10.1145/1376616.1376755.
2. S. Amir, I. Bilasco Marius, Ch. Djeraba, "MuMIe: Multi-level Metadata Mapping System. Journal of Multimedia", 6, 3, 2011, pp. 225–235, Doi:10.4304/jmm.6.3.225–235.
3. P. Brusilovsky, "Adaptive hypermedia", In User Modeling and User Adapted Interaction, Vol.11, 2001, pp. 87-110
4. J. Feki, I. Ben Messaoud, G.Zurfluh, "Building an XML document warehouse", In Journal of Decision Systems, 2013, pp.122–148.
5. Ch. Fellbaum, Editor, "WordNet: An Electronic Lexical Database", MIT Press, Cambridge, MA, 1998.
6. F. Giunchiglia, P. Shvaiko and M. Yatskevich. "S-match: an algorithm and an implementation of semantic matching", In ESWS The Semantic Web: Research and Applications Lecture Notes in Computer Science, Vol. 3053, 2004, pp. 61–75.
7. B. Haslhofer and W. Klas, "A survey of techniques for achieving metadata interoperability", ACM Computing Surveys, 42, 2, 2010, Doi: 10.1145/1667062.1667064.
8. J. Madhavan, Ph. A. Berstein and E. Rahm, "Generic schema matching with cupid", In Proceedings of the International Conference on Very large Data Bases (VLDB), 2001, pp. 49–58.
9. S. Melnik, H. Garcia-Molina and E. Rahm, "Similarity flooding: A versatile graph matching algorithm and its application to schema matching" In ICDE Proceedings of the 18th International conference on Data Engineering, 2002, pp. 117–128.
10. R. J. Miller, M. A. Hernandez, L. M. Haas, L. Yan, C. T. H. Ho, R. Fagin and L. Popa, "The Clio Project: Managing Heterogeneity", In Proceedings of the International Conference on Management of Data (ACM SIGMOD). 30, 1, 2001, pp.78–83. Doi: 10.1145/373626.373713.
11. M. St. Pierre and W. P. LaPlant, Issues in cross walking content metadata standards, July, 1998, http://www.niso.org/press/whitepapers/crosswalk.html.
12. A. Raffio, D. Braga, S. Ceri, P. Papotti, M. A. Hernández, "Clip: a tool for mapping hierarchical schemas", In Proceedings of the International Conference on Management of Data (ACM SIGMOD), 2008, pp. 1271–1274, Doi: 10.1145/1376616.1376753.
13. C. R. Rivero, I. Hernández, D. Ruiz and R. Corchuelo, 2011. "A Reference Architecture for Building Semantic-Web Mediators", Advanced Information Systems Engineering Workshops, Lecture Notes in Business Information Processing. Vol. 83, 2011, 330–341.
14. P. Shvaiko and J. Euzenat, "Ten challenges for ontology matching", In Confederated International Conferences (OTM), 2008, pp. 1164–1182. Doi:10.1007/978-3-540-88873-4_18.
15. J. Conklin and M. L. Begeman, "gibis : A hypertext tool for team design deliberation", In Hypertext, 1987, pp. 247–251.