Context-based Query Answering for Semantic Interoperable Information Systems

Benharzallah Saber, Kazar Okba
Department of Computer science
University of Mohamed Khider Biskra, 07000
Algeria
{sbharz, kazarokba @yahoo.fr}

ABSTRACT:*The importance of semantic interoperability has grown in recent years with the emergence of increasingly interrelated information systems. We present in this paper a context-based query answering for semantic interoperable information systems. The algorithm is self-adapted to the changes of the environment, offers a wide aptitude and solves the various data conflicts in a dynamic way, it reformulates the query using the schema mediation method for the discovered systems and the context mediation for the other systems. Another advantage of our approach consists in the exploitation of intelligent agents for query reformulation and the use of a new technology for the semantic representation.*

Keywords: Query answering, Semantic mediation, Multi-agent systems and OWL DL

Received: 1 September 2010, Revised 4 October 2010, Accepted 14 October 2010

1. Introduction

Recently, research on information systems has increasingly focused on how to effectively manage and share data in heterogeneous distributed environments. The cooperation of systems is confronted with many problems of heterogeneities and must take account of the open and dynamic aspect of modern environments. Querying the distributed ontologies is one major task in semantic interoperable information systems.

Various types of heterogeneity can be encountered cited as follow: technical, syntactic, structural and semantic heterogeneity. The resolution of semantic heterogeneity is becoming more important than before. Its types appear as: naming conflicts (taxonomic and linguistic problems) and values conflicts [28][03].

The high number of the information sources implies the increase and the diversification of the conflicts number, as well as an increase in the time of ocalization of relevant information. It increases also the time of transmission of the queries towards all these information sources and the time response of the information sources. Therefore, the solutions of semantic interoperability should have an intelligent processor for query answering that allows the adaptation of the environment's changes and solves the various data conflicts in a dynamic way. Each solution provides some advantages to the detriments of others. Each one of them treats just one part of the data conflicts.

We propose a context-based query answering approach for semantic interoperable information systems. Our algorithm is self-adapted to the changes of the environment, offers a wide aptitude and solves the various data conflicts in a dynamic way. It reformulates the query using the schema mediation method for the discovered systems and the context mediation for the other systems.

In the following, section 2 presents a synthesis of the various existing approaches. Section 3 and 4 describe the architecture of the mediation and the basic concepts of our architecture. The section 5 describes the query answering and the section 6 presents the technical aspects and prototype implementation.

2. Related works

As the query answering problem in distributed systems has been discussed in traditional databases and Semantic Web, two possible orientations have been proposed: the integration guided by the sources (schema mediation), and the integration guided by the queries (context mediation) [21][5] [6][8][4][10][11].

The schema mediation is a direct extension of the federate approach. Data conflicts are statically solved. In the schema mediation; the mediator should be associated with a knowledge set (mapping rules) for locating the data sources. The query answering follows an execution plan established by rules which determine the relevant data in order to treat a query (static resolution of queries). It requires a preknowledge on the systems participating in the cooperation. The mediator's role is to divide (according to the global schema) the user query in several sub-queries supported by the sources and gathers the results. The global schema is generally specified by object, logic, XML or OWL interfaces [24][17][3][5][22]. In all these works, the objective is to build a global schema which integrates all the local schemas. When one operates in an evolutionary world where sources can evolve all the time, the elaboration of a global schema is a difficult task. It would be necessary to be able to reconstruct the integrated schema each time a new source is considered or each time an actual source makes a number of changes [4]. Generally, the time response of the queries of this approach is better than the context mediation which requires much time (it uses the semantic reconciliation). In this approach; the transparency (is to give the illusion to the users whom they interact with a local system, central and homogeneous) is assured. The degree of automation of the resolution of the data conflicts is weak, and the scalability (the system effectiveness should be not degraded and the query answering remains independent of the addition or the suppression of systems in a given architecture) and evolutionarity (to control the update, the remove and the addition of information systems) are less respected compared to the context mediation.

Many works are dedicated to the proposition of automatic approaches for schemas/ontologies integration [30][31]. The schemas mapping notion have been particularly investigated in many studies, therefore it leads to the elaboration of several systems such as DIKE [7], COMA [13], CUPID [14]. It is possible to find analyses and comparisons of such systems in [18]. Several ontologies based approaches for integration of information were suggested. In [20] and [4] survey of this subject is presented. Among the many drawbacks of these works is that they do not describe the integration process in a complete way; they always use assumptions like pre-existence mappings [23][33] from a part, and from another part, they provide methods to calculate mappings between general or specific ontologies [30] and they do not indicate how to really exploit it for automatic integration or for the query reformulation [22][33].

In [21][3] the authors have proposed an extended schema mediation named DILEMMA based on the static resolution of queries. The mediation is ensured by a couple mediator/wrapper and a knowledge base associated with each system that takes part in the cooperation. The mediator comprises a queries processor and a facilitator. This approach provide a better transparency and makes it possible to solve the semantic values conflicts, but in a priori manner. The automation degree of the resolution of the data conflicts is enhanced compared to the schema mediation. This later involves always the recourse of an expert of the domain. It has a low capacity to treat evolutionarity and the scalability.

The role of the mediator in the context mediation approach is to identify, locate, transform and integrate the relevant data according to semantics associated with a query [21][3]. The resolution of data conflicts is dynamic and does not require the definition of a mediation schema. The user's queries are generally formulated in terms of ontologies. The data are integrated dynamically according to the semantic information contained in the description of the contexts. This approach provides a best evolutionarity of the local sources and the automation degree of the resolution of the data conflicts is better compared to schema mediation. Two categories of context mediation are defined: - the single domain approach SIMS [9], COIN [10] working on a single domain where all the contexts are defined by using a universal of consensual speech. The scalability and evolutionarity are respected but remains limited by the unicity of the domain. - Multi-domains approaches Infosleuth [11], Observer [12] they use various means to represent and connect heterogeneous semantic domain:ontologies, hierarchy of ontologies and method of statistical analysis.

In the context mediation approach the data conflicts are dynamically solved during the execution of the queries (dynamic query resolution), allowing the best evolution of the local sources and the automation degree is enhanced compared to the schema mediation, this to the detriment of time response of the queries (it uses the semantic reconciliation). Concerning the semantic conflicts, the majority of the projects solve only the taxonomic conflicts (Coin [10]). The resolution of the values conflicts is either guided by the user (Infosleuth [11]), or unsolved in the majority of cases (Observer [12] [28]).

The agent paradigm gives a new insight for the systems nature development such as: complex, heterogeneous, distributed and/or autonomous [15][34][35][38][40]. Several works of semantic interoperability use the agent paradigm [16][11][29] [32].

Infosleuth project [11] is used to implement a set of cooperative agents which discover, integrate and present information according to the user or application needs for which they produce a simple and coherent interface. The Infosleuth's architecture project consists of a set of collaborative agents, communicating with each other using the agent communication language KQML (Knowledge Query and Manipulation Language). Users express their queries on a specific ontology using KIF (Knowledge Interchange Format) and SQL. The queries are dispatched to the specialized agents (agent broker, ontological, planner...) to retrieve data on distributed sources. The resolution of many semantic conflicts remains guided by the user [3]. They use specialized agents seen as threads which are widely different from the usual definition of the cognitive agent given in the distributed artificial intelligence.

In [28], the authors propose a multi-agent system to achieve semantic interoperability and to resolve semantic conflicts related to evolutive ontologies domain. In this approach, the query answering and the validation of the mappings are completely related to the users. In [29] propose an agent based intelligent meta-search and recommendation system for products through consideration of multiple attributes by using ontology mapping and Web services.

This framework is intended for an electronic commerce domain. All the approaches cited above use a approach fixed in advance (schema or context mediation), it can cause problems scalablilité and adaptation to the changing environment. Our main contribution is what does not fixed in advance for query answering. Our system dynamically adapts to change of the environment and processes queries according to available information on suppliers (or environment).

3. Generic Architecture based Agent for Context and Schema mediation (GAACSM)

Our architecture is divided into two levels (Figure 1): physical entities level and the agent level. The first level includes especially existing information systems (ISs), including legacy systems. These systems are developed using conventional technologies such as databases management systems, they can be relational, object, XML, ... Originally, these systems are designed to meet local needs and not necessarily work together.

The second agent level is designed on the top of existing physical systems. There are two types of agents: intelligent agents (IAs) and routing agents (RAs). An IA is an intermediary between an information system and the semantic mediation environment. An information system can play the role of information supplier and/or consumer. So an IA can play the role of IA supplier (IAS) and/or consumer (IAC). In other words, if the AI asks the query, so in this case is called AIC, if an another agent asked the IA to run a query in this case it is called AIS.

The routing agents play an important role in the GAACSM architecture. They assume context mediation, and organize intelligent agents in near domains semantically, it is ensured using a semantic proximity which allows to separate the semantic interoperable information systems in segments (or groups, or set of domains). The agents belonging to the same group are considered as near agents semantically, we use this segmentation to avoid the useless communications between agents, and to improve our query answering approach.

The cooperation suggested in our solution is based on: A preliminary construction of information before its integration in the architecture system and we use the static and dynamics query resolution.

The integration phase of a new information system (IS) in our proposed mediation system begins with the creation of an IA and continues with the fastening of this last to a routing agent (RA) which is *nearest semantically*.

The new IA integrated into the system of mediation applies the Contract Net protocol and sends an invitation describing its domain. The RAs receiving the call and provide their ability (*semantic proximity rate*). As soon as, the IA receives answers from all RAs, then it evaluates these rates, and makes its choice on a RA which is the nearest semantically. The chosen RA adds the previous IA to its net contacts.

Our approach does not use a global schema or some predefined mappings. Users interrogate the consuming system (the queries formulated in term of the consuming schema). At the beginning, the intelligent agent consuming (IAC) applies the dynamic query resolution protocol (context mediation) because it does not have information on the suppliers systems. This protocol is applied via the RA which is the nearest semantically with the IAC. During the dynamic evaluation of the query, the intelligent agent suppliers (IASs) update their histories and add information (mapping between terms of query ontology and their ontologies) to facilitate their dynamic integration with the IAC.

Each IAS replies with results, the RA updates its KB and reorders the list of IASs that are the most important to previous IAC (in other words; the IASs which contain results are at the head of the list). If no IAS replies, the RA sends the query to other RAs. If there are replies, the RA adds the IASs of other RAs to its KB (auto reorganization).

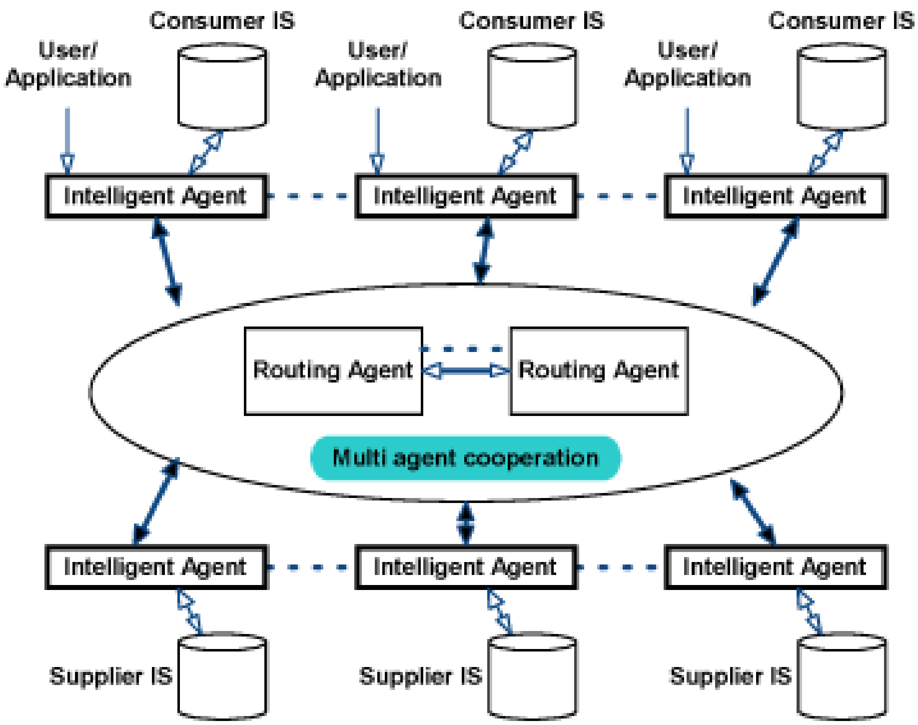


Figure 1. General architecture of the proposed approach

During the operation of the mediation system, the IAC applies the protocol to discover suppliers which are the nearest semantically to its domain, and to integrate them dynamically in order to use them in the schema mediation. For this aim, it cooperates with the RA. Indeed; the RA updates its KB during its communication with the other agents. Particularly, its KB contains for each IA an ordered list of its IASs which are not discovered yet. These IASs should be near semantically to it. The first IAS in the list is the one which has largest number of responses of IA. After that the first IAS becomes the next supplier solicited to the following dynamic integration done by IA.

After the dynamic integration, the IAC updates its knowledge base by mapping rules.

During the operation of the system, the IAC discovers some suppliers and adapts itself with the environment. So, to treat a query two protocols should be applied: the static query resolution protocol is adopted for the discovered systems and the dynamic query resolution for other systems (algorithm3).

3.1 The principal interne structure of the agents of GACSM
In this section we present the principal interne structure of an intelligent agent (IA) and a routing agent (RA).

3.2.1 The principal interne structure of an intelligent agent
The figure 2 presents the principal interne structure of an IA.

The components of an intelligent agent are:

- The interface of communication: allows an agent to communicate with theenvironment. Using the interface of communication, an IA sends queries to the local IS associated with it, and receives responses, it responds to user queries. It also cooperates with other agents in the system.

- Messages parser: it analyzes the received messages and determines their types.

- Conversations manager: it identifies the cooperation protocol used for the dialogue between the sender and the receiver, and controls the progress of the conversation.

- The wrapper plays three main roles:

1- The translation of written requests by using the language adopted to the proprietary language to the local source,

2 - the translation of the local source schema to a schema written using the common data model (OWL DL),

3 - the translation of the results obtained from the local source to the common XML format.

-Knowledge base: contains the agent's knowledge on the environment, it is always updated by the agent during its communication with other agents. It contains mainly the context of the local information system, the name of the domain, the ontology describing the domain name, ontology of semantic conflicts of values. It also contains the mapping rules between the schema of the agent and the schemas of other agents that are discovered (by this agent).

- The schema mediator: realize the schema mediation

- The context mediator: realize the context mediation

- Interpreter: It makes the management synchronization between different threads. So depending on the type of the received message, it activates and calls the component necessary for queries processing.

- Executor: The role of the executor is manifold; according to the mission for which it is created. For example: an Executor is created by the mediator schema to reconstruction the results of a query

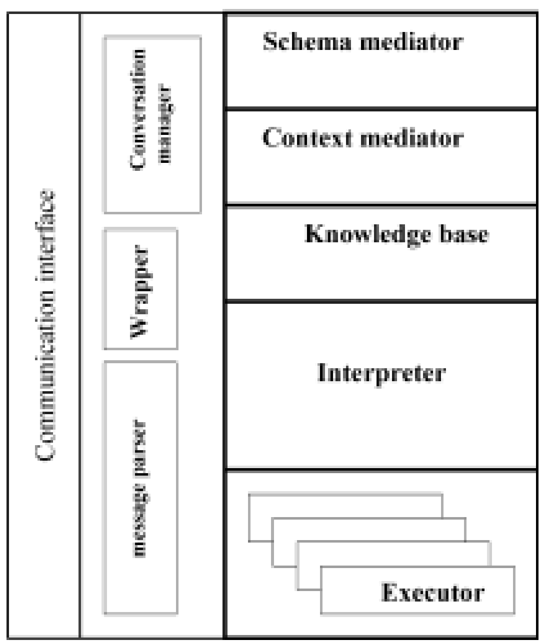


Figure 2. The interne structure of an IA

3.2.2 The principal interne structure of a routing agent

The routing agents play an important role in the GAACSM architecture. They assume context mediation, and organize intelligent agents in near domains semantically, the figure 3 present the principal interne structure of a RA.

The components of a RA are:

- The communication interface, messages parser and conversations manager, have the same rules as in an IA.

-Knowledge base: contains the agent's knowledge on the environment, it is always updated by the agent during its communication with other agents. In particular, the Knowledge base contains for each IA, an ordered list of its IAS that are not yet discovered by it and which are near semantically to it (Figure 4). If an agent IA_i has obtained the largest number of responses to its requests from the IAS_j, then the agent AR place IAS_j in the header of the ordered list of the IA_i. Thus IAS_j is the sought supplier for the next dynamic integration conducted by IA_i.

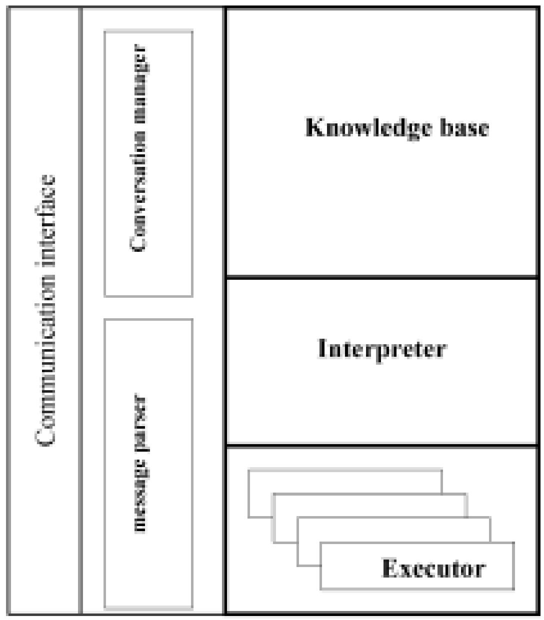


Figure 3. The interne structure of a RI

-Interpeter: It makes the management synchronization between different threads. So depending on the type of the received message, it activates and calls the component necessary for queries processing.

- Executor: The role of the executor is manifold; according to the mission for which it is created. For example: an Executor is created by the interpreter to reconstruction the results of a query in the case of context mediation.

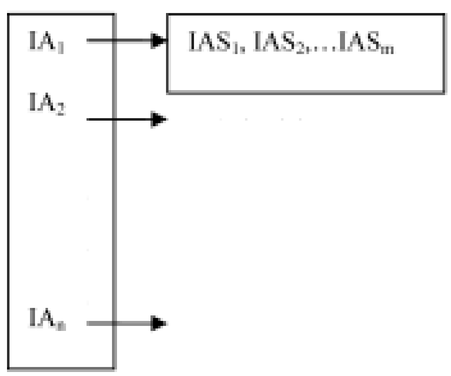


Figure 4. The list of IAs near semantically to a RA

4. Basic concepts of the GAACSM architecture

In what follows, we present a cooperation scenario which will be used throughout this paper.

4.1 Cooperation scenario

In this section we describe an interoperability example between heterogeneous systems. A given company wishes to provide an information service regarding the concerts of various artists (extension of the example cited in [26]) from the world. We chose this example for reasons of simplification.

The schema of the consuming system is as follows: Class (CS) FunctionalProperty (nbC domain (CS) range (xsd:integer)) DatatypeProperty (artistN domain (CS) range (xsd:string)) DatatypeProperty (dateC domain (CS) range (xsd:date)) DatatypeProperty (Pfree domain (CS) range (xsd:integer)) DatatypeProperty (Psold domain (CS) range (xsd:integer)) DatatypeProperty (Pprice domain (CS) range (xsd:float)) nbC (integer): number identifying a concert, artistN (string): Name of artist, dateC (date): Date of concert, Pfree (integer): number of a free places, Psold (integer): number of sold places, Pprice (float): price of a place (Euro)	The schema of the supplier system 1 is given below: Class (SS1) Class (Place) FunctionalProperty (id domain (SS1) range (xsd:integer)) DatatypeProperty (nam domain (SS1) range (xsd:string)) DatatypeProperty (seance domain (SS1) range (xsd:date)) ObjectProperty (EidPlc domain (SS1) range (Place)) DatatypeProperty (ticket domain (SS1) range (xsd:float)) FunctionalProperty (idplc domain (Place) range (xsd:integer)) DatatypeProperty (nbP domain (Place) range (xsd:integer)) DatatypeProperty (totP domain (SS1) range (xsd:integer)) FunctionalProperty (id domain (SS1) range (xsd:integer)) id (integer) : number identifying a concert, nam (string) :name of artist, seance (date) : date of a seance, Eidplc (integer) : an identifier reference to the relation Placeidplc, ticket (float) : price of a place (Dinars), idplc (integer) : identifier identifies nbP and totP, nbP (integer) : number of a free places, totP (integer) : number of total places
The schema of the supplier system 2 is the following: Class (SS2) FunctionalProperty (nomCons domain (SS2) range (xsd:integer)) DatatypeProperty (NamArtist domain (SS2) range (xsd:string)) DatatypeProperty (ConsDate domain (SS2) range (xsd:date)) DatatypeProperty (soldP domain (SS2) range (xsd:integer)) DatatypeProperty (totalP domain (SS2) range (xsd:integer)) DatatypeProperty (Tprice domain (SS2) range (xsd:float)) numCons : number identifying a concert, NamArtist : Name of artist, ConsDate: Date of seance, soldP: number of a sold places, totalP : Number of total places, Tprice : Price of a place (Dollars)	

Our approach uses the OWL DL [19] as common data model. The OWL DL enriches the RDF Schemas model by defining a rich vocabulary to the description of complex ontologies. So, it is more expressive than RDF and RDFS which have some insufficiency of expressivity because of their dependence only on the definition of the relations between objects by assertions. OWL DL brings also a better integration, evolution, division and easier inference of ontologies [19].

To build an ontology from a schema; we propose the following steps: a) We use the schema to extract the concepts and the relations between them, in other words; Find the semantic organization of the various concepts (used in the schema) and the relation between them (initial construction). b) We add the synonyms and the antonyms of each name of class in *'label'*, c) We add comments on the name of classes by using *'comment'*, d) We add for each name of a class its sub concepts, its super concepts and its class's sisters.

The construction of this ontology is closely related to the context of the application domain of the information system.

Example 1

The following example indicates the schema ontologies of the consuming, supplier

1 and 2 systems built by using the preceding steps (it is a concise representation, fig 5, 6 and 7)

4.2 Definitions

Definition 1: Schema-ontology mapping. Given a schema *S* and its ontology *O*. a schema-ontology mapping is expressed by the function:

$$\begin{aligned} MSO: S &\rightarrow O \\ x &\rightarrow e \end{aligned}$$

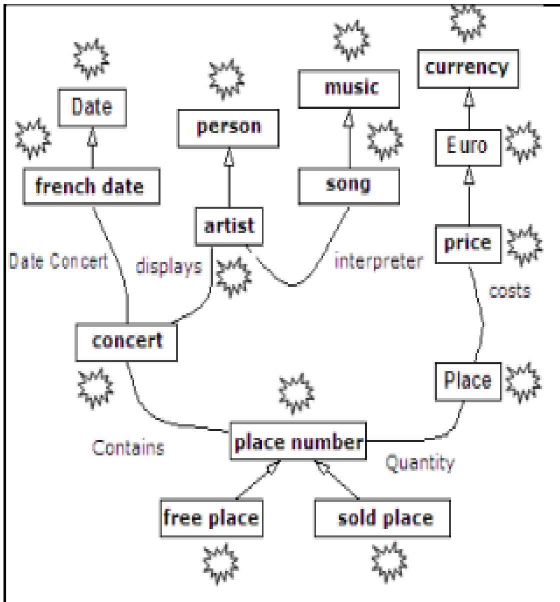


Figure 5. Ontology of consuming system
The b,c, d information are represented by:

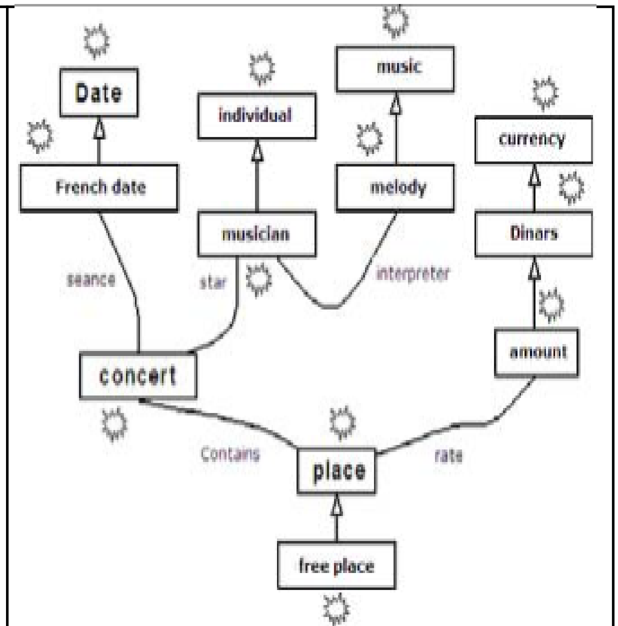


Figure 6. Ontology of supplier 1 system

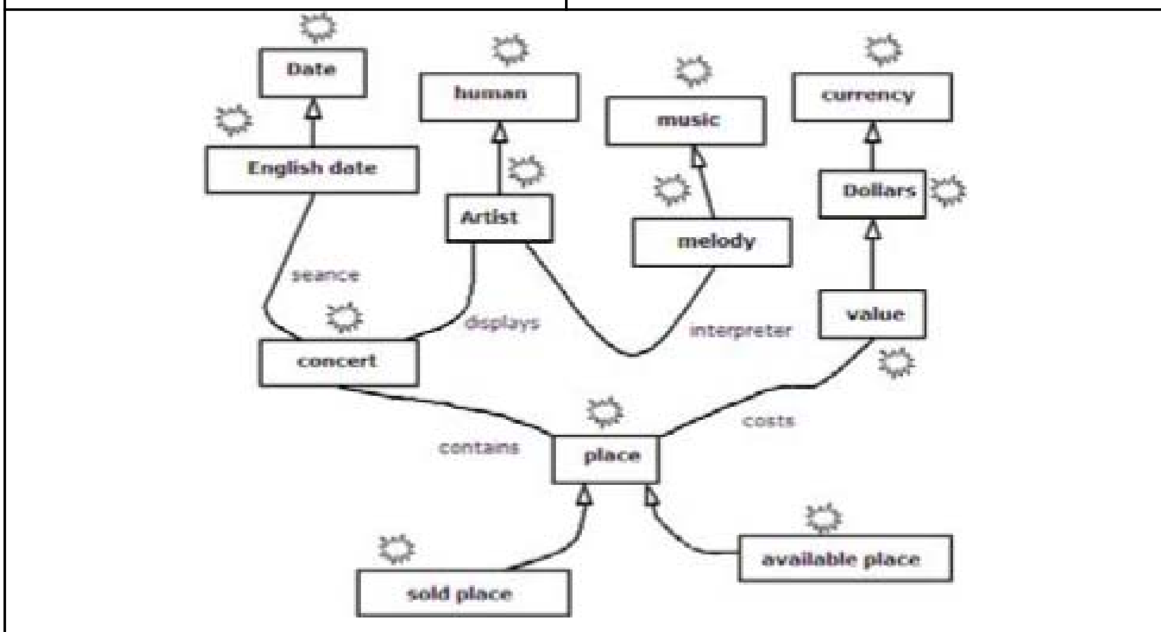


Figure 7. Ontology of supplier 2 system

Example 2

Mappings *MSO* between the consuming schema and its ontology are the following:

<Concert rdf:ID=”nbC”> </Concert> <Concert rdf:ID=”CS”> </Concert> <Artist rdf:ID=”artistN”> </Artist> <Date rdf:ID=”dateC”> </Date>	<freeplace rdf:ID=”Pfree”> </ freeplace> <soldplace rdf:ID=”Psold”> </ soldplace> <Price rdf:ID=”Pprice”> </Price>

Definition 2: Context. It describes the assumptions, the explicit information of definition and use of a data. In our approach, the context is defined by (*S*, *SCV*, *O*, *MSO*) such as: *S* is a schema, *SCV* is a semantic conflicts of values, *O* defines an ontology and *MSO* is a schema-ontology mapping.

Definition 3: Query language. We adapted the language defined in [2] as a query language in our architecture. Given L the set of individuals and values belonging to OWL DL data types. Given V the set of variables disjoint from those of L. A query Q_i in ontology O_i is of the form ,$Q_C^i \wedge Q_P^i$ where

- Q_C^i is a conjunction of $C^i(x)$ where $C^i \in C$ and $x \in L \cup V$
- Q_P^i is a conjunction of $P^i(x, y)$ where $P^i \in P$ and $x,y \in L \cup V$

Example 3

This query is formulated in terms of the consuming schema.
$Q = CS(x) \wedge$ artistN(x, "artist1") $\wedge$ dateC(x, y) = x . Which means the knowledge of the date or the dates of the concerts of the artist «artist1».

Definition 4: Semantic similarity. The calculation of the semantic similarity between two concepts is calculated from the elementary calculations of similarity which take into account the various elements of the environment of a concept in its domain. The various adopted measures are: the terminology of the concept and environment in which the concept is located. These measurements are selected from a deep study of the various similarities measures [1] [36] [34] and from the definition of an ontology of schema in GAACSM architecture. Our algorithm which calculates the semantic similarity between two elements e1, e2 is as follows (figure 8):

Algorithm 1 Sim(e1,e2)

Require: ontology $\boldsymbol{O_1}$ and $\boldsymbol{O_2}$,e1 $\in O_1$,e2 $\in O_2$

1: Calculation SimN of e1,e2,

2: Calculation SimC of e1,e2,

3: Calculation SimV of e1,e2,

4: Calculation SimR of e1,e2,

5: $SimTer\ (e1,e2) = \alpha_1 \times SimN + \alpha_2 \times SimC$

6: $SimStruc\ (e1,e2) \leftarrow \beta_1 \times SimV + \alpha_2 \times SimR$

7: $Sim(e1,e2) \leftarrow \alpha \times SimTer + \beta \times SimStruc$

End

Figure 8. Semantic similarity algorithm

où:$\alpha \in [0,1], \beta \in [0,1], \alpha_1 \in [0,1], \alpha_2 \in [0,1], \beta_1 \in [0,1], \beta_2 \in [0,1]$. SimTer: terminological similarity. SimStruc: structural similarity. SimN : Similarity of names using their synonyms and antonyms. SimC: Comments similarity of the two concepts. SimV: Structural similarity vicinity (Our approach is based on the assumption that if the neighbors of two classes are similar, these two classes are also considered as similar). SimR: Roles similarity (The roles are the links between two OWL DL classes)

Definition 5: Comparison of two ontologies. The comparison of two ontologies, belonging to different IAs, The comparison is defined by the Comp function as follows: Comp: O $\rightarrow$ O' such as Comp(e1) = e'1 if Sim(e1, e'1) > tr where O and O' are two ontologies to be compared, tr indicates a minimal level of similarity belonging to the interval [0,1], e1 $\in$ O and e'1 $\in$ O'.

Definition 6: Sub schema Adaptation of an IA. Given two intelligent agents A, B.

- Given the schema S_a , the ontology O_a of A, and the ontology O_b of the agent B.
- Given the function Comp: $O_a \rightarrow O_b$ the comparison between two ontologies O_a and O_b of A and B respectively.
- Given CO_{ab} the set of the elements e $\in O_a$, such that Comp(e) = e' and Sim(e, e') > tr with e' $\in O_b$.
- Given a sub-schema Ss_a the set of elements x $\in S_a$ such as MSO(x) =e with e $\in CO_{ab}$.

The adaptation of the sub schema Ss_a of S_a (of the agent A) on the ontology O_b of the agent B is the function:

Adapt: $Ss_a \rightarrow O_b$

$X \rightarrow e'$

with: X $\in Ss_a$, e' $\in O_b$ where there exist e $\in CO_{ab}$ such that Comp(e) = e' and Sim(e,e') > tr, MSO(x) =e.

Definition 7: Semantic enrichment of a query. Given the context C represented by (S, O, MSO) and $Q = Q_C^i \wedge Q_P^i$ a query formulated in term of the schema S. The semantic enrichment of this query, by using the ontology O, is defined by the following rules:

1) Find using the function MSO, the equivalent classes of C^i (x) and , P^i (x,y) of the query Q_C^i and Q_P^i respectively in the ontology O. They are noted by OC^i (x) and OP^i(x ,y) respectively.

2) Find by using the subsumption relation, the ancestors classes of each class of OC^i (x) and OP^i(x ,y). They are noted by pOC^i(x) and pOP^i(x ,y) respectively..

3) Find by using the subsumption relation, the sub classes of each class of OC^i (x) and OP^i(x ,y). They are noted by cOC^i(x) and cOP^i(x ,y) respectively.

4) Find by using the equivalent relation, the equivalent classes of each class of OC^i(x) and OP^i(x ,y). They are noted by eOC^i (x) and eOP^i(x ,y) respectively.

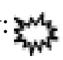
5) We clarify, by using the schema S, the semantic conflicts of values which exist in the query Q. This information is noted by *csvQ*.

A query Q enriched semantically is composed of : $Q_C^i \wedge Q_P^i$, OC^i(x),{eOC^i(x)},{cOC^i(x)},{pOC^i(x)},OP^i(x ,y),{eOP^i(x ,y)},{pOP^i(x ,y)},{cOP^i(x ,y)},*csvQ*).
This enrichment is called *query ontology*.

Example 4
Given the following query formulated in terms of the consuming schema
$Q = CS(x) \wedge$ artistN(x,"artist1") $\wedge$ date$C$(x,y). The semantic enrichment of the query is as follows: We have $Q = Q_C^i \wedge Q_P^i$

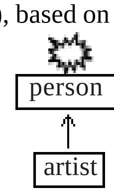
- The Correspondent of the concept *CS* (*x*)

- based on the function MSO, is the concept OC^i(x) =*concert*.

- Concepts {eOC^i (x)},{cOC^i(x)},{pOC^i(x)} are represented by: 

- The Correspondent of the concept OP^{i1}(x ,y) = artistN(x,"artist1"), based on the function MSO, is the concept *Artist*.

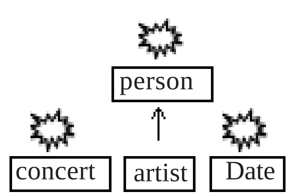
- Concepts {eOC^1 (x)},{cOC^1(x)},{pOC^1(x)} are represented by :



- The Correspondent of the concept OP^2(x ,y) = dateC(x,y), based on the function MSO, is the concept *Date*.

- Concepts {eOC^2(x)},{cOC^2(x)},{pOC^2(x)} are represented by : 

The semantic enrichment of the query (query ontology) $SC(x) \wedge$ artistN(x,"artist1") $\wedge$ date$C$(x,y) is the ontology:



And *csvQ*= {dateC : < OSCV: Date > French date < OSCV: Date />}

Definition 8: Semantic evaluation of a query ontology. The semantic evaluation of a query enriched semantically (query ontology) $O_Q = Q_C \wedge Q_P$,OC^i(x),{eOC^i(x)},{cOC^i(x)},{pOC^i(x)},OP^i(x ,y),{eOP^i(x ,y)},{pOP^i(x ,y)},{cOP^i(x ,y)},*csvQ*) is defined by the algorithm 2 (figure 9):

Example 5
Given the previous query $CS(x) \wedge$ artistN(x,"artist1") $\wedge$ date$C$(x,y)

Algorithm 2
Require: query ontology O_Q *of IAC and ontology* O_{IAS} of IAS
1: Calculation of the function *Comp:* $O_Q \rightarrow O_{IAS}$
2: Calculation of the set CO_{QIAS}
3: Calculation of the sub-schema Ss_Q
4 : Calculation of the function *Adapt* : $Ss_Q \rightarrow O_{IAS}$
5: /*Evaluate of the semantic query *Adapt* (Ss_Q) */
6: For each $X \in Ss_Q$ and $e \in$ *Adapt* (X) do
7: Search Y $\in S_{IAS}$ where: MSO(Y)=e
8: If Y exists then to replace X by Y in $Q_C^i \wedge Q_P^i$ ($Q_C^i \wedge Q_P^i \in O_Q$)
9: Endfor
10: the quey which will be to evaluate in IAS is $Q_C^i \wedge Q_P^i$
11: End

Figure 9. Algorithm 2. Semantic evaluation of a query ontology

The semantic evaluation of its query ontology, on the source of supplier 1 is done by the application of the algorithm 2. The steps are as follows:

- Calculation of the similarities between the query ontology and the ontology of supplier 1.

- Calculation of the set CO_{QIAS} = {Concert, Artist, Person, Date,…}

- Calculation of the sub schema Ss_Q ={*CS*(x) $\wedge$ artistN(x,"artist1") $\wedge$ date*C*(x,y)}

- Calculation of the function Adapt :$Ss_Q \rightarrow O_{IAS}$. Its values are :{AdaptartistN(x,"artist1")=Musician, Adapt(date*C*(x,y) =Date}

- Semantic evaluation of *Adapt*(Ss_Q), which requires the calculation of the function MSO reverse. Hence: {MSO^{-1} (Concert)=SS1, MSO^{-1} (Musician)=nam, MSO^{-1} (Date)=seance}.

- Concerning the semantic conflicts of values, the attribute *seance* uses the same format like *dateC,* else it is necessary to take into account the change of the results and to convert the format using the OSCV ontology (a transformation function).

Finally, the query which will be carried out on the level of the source of supplier 1 is as follows: Q = SS1(x) $\wedge$ *nam* (x,"artist1") $\wedge$ *seance*(x,y).

Definition 9: Mapping rules. A schema mapping is a triplet (S1, S2, M)[2], where: S1 is the source schema; S2 is the target schema; M the mapping between S1 and S2, i.e. a set of assertions $q_s \mapsto q_T$, with q_s and q_T are conjunctive queries over S1 and S2, respectively, having the same set of distinguished variables x, and $\mapsto \in \{\subseteq, \supseteq, \equiv\}$.

Definition 10: Query Reformulation. Let Q_i be a query in schema S_i and Q_j be a query in schema j S described by classes and properties in the mapping Mij .

- Q_j is an equivalent reformulation of Q_i if $Q_j \subseteq Q_i$ and $Q_i \subseteq Q_j$ which is noted by $Q_j \equiv Q_i$.
- Q_j is a minimally-containing reformulation of Q_i if $Q_i \subseteq Q_j$ and there is no other query Q'_j such that $Q'_i \subseteq Q_j$ and $Q'_j \subseteq Q_j$.
- Q_j is a maximally-contained reformulation of Q_i if $Q_j \subseteq Q_i$ and there is no other query Q'_j such that $Q_j \subseteq Q'_j$ and $Q_j \subseteq Q_i$.

To find the approximate query reformulation we use the mapping rules M (definition 15), we substitute the terms of i Q by their correspondents [02].

5. Queries answering

The query answering is divided into several steps, and during this process, the multi-agents system uses a set of protocols. The principal steps are (figure 10):

5.1 Static query resolution

The static resolution is applied to the systems have been already discovered.

Step 1: query validation the IAC checks the validity of the query.

Step2: query reformulation: the query is divided into a recombining query of the results and sub queries intended for the IAS which contain data necessary to the execution of the query. The decomposition of the query is done by the use of the mapping rules.

```
Algorithm 3. Query answering
Given L the list of discovered agents and their mappings
If QueryValidation() then
1:if L <> empty then
     - QueryReformulation()
   - StaticRecombiningResults()
2: Dynamic query resolution
  - SemanticEnrichmentQuery()
  - TransmissionSemanticallyEnrichedQuery()
  - SemanticEvaluation() /*algorithm 2*/
  - DynamicRecombiningResults()
```

Figure 10. Algorithm 3. Query answering

Step3: recombining of the results: the IAC executes the recombining query for the results.

5.2 Dynamic query resolution

The dynamic resolution makes it possible to take into account the appearance of new IASs. The principal steps are:

Step 1: Semantic enrichment of a query. The IAC enriches the query semantically by using the ontology and the links schema-ontology which are in its own knowledge base (definition 7).

Step 2: Transmission of the semantically enriched query. The IAC applies the cooperation protocol of dynamic query resolution. So it transmits the semantically enriched query to the routing agent which is nearest semantically. This latter sends it to all IASs of its net contacts.

Step 3: Semantic evaluation of the semantically enriched query (algorithm 2). Each IAS answers according to its capacity to treat the query:

1) To compare elements of the query with its ontology. The elements of the query and its ontology are compared by using a semantic distance. The identified elements as equivalent are retained.

2) The query is rewritten in terms of the equivalent elements of its ontology (then interpreted on its schema) to take into account the semantic conflicts of values (each intelligent agent has library of functions for the conversion of the types).

3) The answer is sent latter to the routing agent, indicating the manner of treating the query, so that this letter can build recombining queries of the results.

If no IAS answers, the routing agent sends the query to the other routings agents of other domains and if there are answers the routing agent updates its net contacts.

Stage 4: Results recombining: the routing agent recomposes the results obtained by IASs. Then it sends the final result to the IAC, this latter recomposes the results of static and dynamic query resolution.

6. Technical aspects and prototype implementation

Our implementation is based on three class libraries: OntoSim [39], Alignment API [25] and Jade [37]. OntoSim provides many similarities measurements between character strings. Alignment API allows to integrate new methods of similarities measurement (between two OWL ontologies) by implementing a Java interface. Jade (Java Agent Development Framework) [37] is used for the construction of the multi agents systems and the realization of applications in conformity with FIPA specifications. The cooperation protocols are implemented using the Jade platform. Concerning the local information systems, the local database of the consuming system and the database of the supplier system 1 are established under the Access DBMS and the Windows XP operating system. The database of the supplier system 2 is implemented in XML files and the same operating system. The scalability and the performances of the transport system of Jade message were treated in [27][28]. The obtained results confirm the fact that Jade deals well with the scalability according to several scenarios intra or inter framework. The figure 11 presents an example of comparison between two ontologies of the consuming system and the supplier system 2. Figure 12 presents the graphical interface, an example of query and the obtained results. In this example, the IAS1 is discovered by agent IAC. This last applies the schema mediation in order to reformulate the query. The IAC applies the context mediation for other agents, which are not yet discovered (IAS2). It communicates with the agent RA.

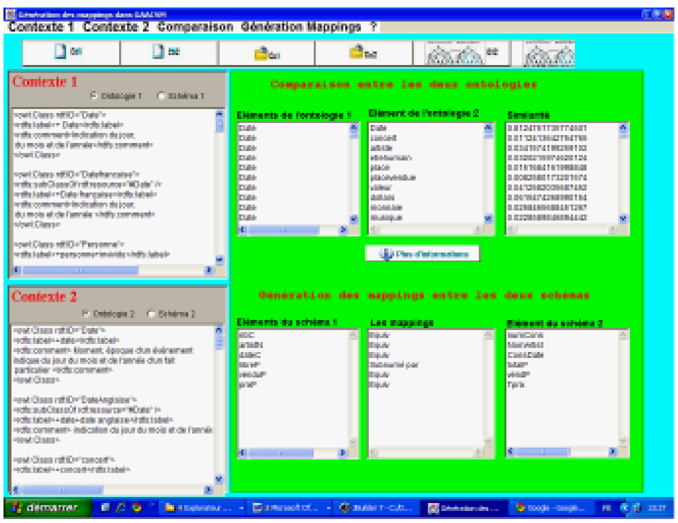


Figure 11. Automatic Mapping Generation

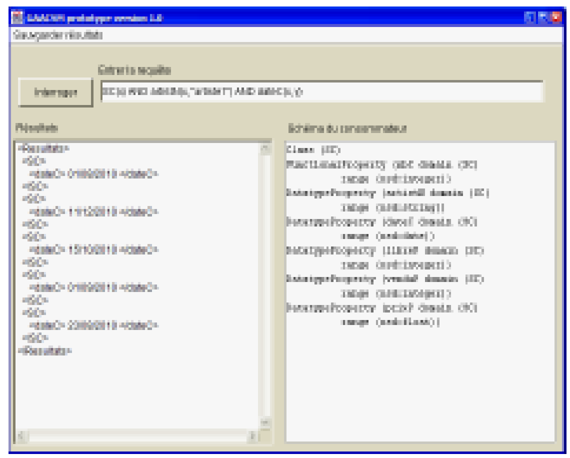


Figure 12. A simple exemple

7. Conclusion and future research

In this paper we presented a context-based query answering approach for semantic interoperable information systems. The main advantage of our query answering algorithm consists in its robustness with regard to the evolution of systems, adaptation to the changes of environment and the resolution of the most various data conflicts in a dynamic way. The developed prototype shows the functionality of suggested approach. Our future research consists to employing the intelligent methods (for segmented comparison of large ontologies of arbitrary size) in order to reduce the time of ontologies comparison so we avoid the influence on the scalability of the suggested architecture.

References

[1] Rahm E. et Bernstein P. A.(2001). A survey of approaches to automatic schema matching, VLDB Journal The International Journal on Very Large Data Bases, 10 (4) 334 – 350.

[2] Akahani, Jun-ichi., Hiramatsu, Kaoru., Satoh, Tetsuji (2003). Approximate Query Reformulation for Ontology Integration, Semantic Integration Workshop (SI-2003) Second International Semantic Web Conference October 20, 2003 Sanibel Island, Florida, USA.

[3] Fabrice Jouanot, (2001). DILEMMA : vers une coopération de systèmes d'informations basée sur la médiation sémantique et la fusion d'objets », université de bourgougne, novembre 2001.

[4] Schneider, Michel., Bejaoui, Lotfi., Bertin, Lotfi (2009). A Semantic Matching Approach for Mediating heterogeneous Sources, metadata and semantic, Springer US, p 537-548.

[5] Busse, Susanne., Kutsche, Ralf-Detlef., Leser, Ulf., Weber., Herbert (1999). Federated Information Systems: Concepts, Terminology and Architectures, Technical Report Nr.99-9 Berlin University.

[6] Hakimpour., Farshad (2003) Using Ontologies to Resolve Semantic Heterogeneity for Integrating Spatial Database Schemata, thesis:, Universität Zürich Zürich.

[7] Palopoli, Luigi., Terracina, Giorgio., Ursino., Demenico (2003). DIKE: a system supporting the semi-automatic construction of cooperative information systems from heterogeneous databases. Softw., Pract. Exper. 33 (9) 847-884.

[8] Leclercq, Eric., Benslimane,Djamal., Yétongnon, Kokou(1999). "ISIS: A Semantic Mediation Model and an Agent Based Architecture for GIS Interoperability, *In*: Proceedings of the International Database Engineering and Applications Symposium (IDEAS'99), Montreal, CANADA, August 2-4, 1999. Published by IEEE Computer Society Press, p. 81—92

[9] Arens,Y., Chung-Nan Hsu, Hsu, C.,(1996). Query processing in the SIMS Information Mediator. *In* TATE A. Ed. Advanced planning technology, Menlo park, CA, AAAI Press.

[10] Bressan, Stephane., Goh, Cheng., Levina, Natalia., Madnick, Stuart., Shah,Ahmed., Siegel, Michael (2000). Context Knowledge Representation and Reasoning in the Context Interchange System, Applied Intelligence, 13 (2) p. 165-180.

[11] Bayardo, R. J. Jr.et al,(1997). InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments, Microelectronics and Computer Technology Corporation.

[12] Mena, E., Kashyap,V., Sheth, A., Illarramendi, A. (2000). OBSERVER : An approach for Query Processing in Global Information Systems based on Interoperation across Preexisting Ontologies, *International Journal on Distributed And Parallel Databases DAPD,* 8 (2) 223-272.

[13] Do,Hong-hai., Rahm, Erhard (2002). COMA - A System for Flexible Combination of Schema Matching Approaches. VLDB 2002, p. 610-621.

[14] Madhavan, Jayant., Bernstein, Philip A., Rahm, Erhard (2001). Generic Schema Matching with Cupid. VLDB 2001, p. 49-58.

[15] Girardi, R.(2001). An analysis of the contributions of the agent paradigm for the development of complex systems", In (SCI 2001) and (ISAS 2001), Orlando, Florida.

[16] Purvis, M., Cranefield, S., Bush,G., Carter, D., McKinlay, M., Nowostawski, M., Ward,R.(2000), The NZDIS Project: an Agent-Based Distributed Information Systems Architecture, Proceedings of the Hawai'i International Conference On System Sciences, January 4- 7, 2000, Maui, Hawaii.

[17] Suwanmanee, Seksun., Benslimane, Djamal., Thiran, Philippe(2005). OWL Based Approach for Semantic Interoperability, Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA 05)

[18] Mohsenzadeh, M. , Shams, F., Teshnehlab, M (2005). Comparison of Schema Matching Systems. WEC (2) 141-147.

[19] Deborah, L., McGuinness and Frank van Harmelen.(2004). Owl web ontology language. Technical report.

[20] Wache , H., Vögele, T., Visser , U., Stuckenschmidt, H., Schuster , G., Neumann , H., Hübner, S (2001).Ontology-based integration of information - a survey of existing approaches. In Stuckenschmidt, H., ed., IJCAI-01 Workshop: Ontologies and Information Sharing, p 108-117.

[21] Jouanot, Fabrice., Cullot, Nadine., Yetongnon, Kokou (2003) Context Comparison for Object Fusion CAiSE 2003, LNCS 2681, p. 536"551.

[22] Rubén Tous, Thesis: Data Integration with XML and Semantic web Technologies. Universitat Pompeu Fabra, Barcelona, June 2008.

[23] Goasdoué, François., Rousset.,Marie-Christine(2003). Querying Distributed Data through Distributed Ontologies: ASimple but Scalable Approach, *IEEE INTELLIGENT SYSTEMS*,60-65.

[24] Nang Tom Hawm Saw (2000). OWL-Based Approach for Semantic teroperating and Accessing Heterogeneous XML Sources, IEEE.

[25] http://alignapi.gforge.inria.fr/lib.html

[26] Fabrice Jouanot.(2000). Un modele sémantique pour l'interoperabilité de systemes d'information, laboratoire LE2I, INFORSID.

[27] Cortese, E., Quarta, F., Vitaglione,G., (2002). Scalability and Performance of Jade Message Transport System. Presented at AAMAS Workshop on AgentCities, Bologna, 16 th July.

[28] Magali Séguran., (2003). Résolution des conflits sémantiques dans les systèmes d'information coopératifs : proposition d'un modèle d'interaction entre agents, Université Jean Moulin, Lyon3.

[29] Wooju Kim., Dae Woo Choi and Sangun Park., (2008) Agent based intelligent search framework for product information using ontology mapping, J Intell Inf Syst (2008) 30:227–247. Springer Science + Business Media, LLC.

[30] Fleur Mougin, Anita Burgun, Olivier Bodenreider, Julie Chabalier., Olivier Loréal4., Pierre Le Beux1, (2008). Automatic Methods for Integrating Biomedical Data Sources in a Mediator-Based System, DILS 2008, LNBI 5109, p. 61–76, Springer-Verlag Berlin Heidelberg.

[31] Hansen, H. L., Ingvaldsen, K. (2005). Dynamic Schema Integration The ACRoS Prototype THESIS, The Faculty of Social Sciences, Department of Information Science and Media Studies.

[32] Gal, Avigdor., Segev., Aviv (2005). Agent Oriented Data Integration, J. Akoka et al. (Eds.): ER Workshops 2005, LNCS 3770, p. 98–108, 2005.Springer-Verlag Berlin Heidelberg.

[33] Peter (2007). Haase,dissertation, semantic technologies for distributed information systems, universitatsverlag karlsruhe.

[34] KnowledgeWeb.(2005).Deliverables of KWEB Project: D2.2.3: State of the art on ontology alignment. EU-IST-2004-507482, [On Line] http://knowledgeweb.semanticweb.org/.

[35] Carbonell, J,G., Siekmann, J.(2006). Lecture Notes in Artificial Intelligence, Agent-Oriented Information Systems III 7th International Bi-ConferenceWorkshop, AOIS 2005 Utrecht, Springer Berlin Heidelberg NewYork.

[36] Euzenat J., Shvaiko P.,(2007) Ontology Matching. Springer.

[37] Bellifemine, Fabio., Caire, Giovanni., Greenwood, Dominic(2007). Developing Multi-Agent Systems with JADE, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, Copyright.

[38] Ai-Ping, Li., Yan , Jia., WU Quan-Yuan., (2007). A Study on Organizational Knowledge in Multi-Agent System, *JCIT: Journal of Convergence Information Technology*, 2 (1) 61-65.

[39] https://gforge.inria.fr/projects/ontosim/

[40] Trajkovski, G.(2011). Developments in Intelligent Agent Technologies and Multi-Agent Systems: Concepts and Applications.