Domain Ontologies Evolutions to Solve Semantic Conflicts

Guilaine Talens[1], Danielle Boulanger[1], and Magali Séguran[2]

[1] MODEME, Université Jean Moulin
6 cours Albert Thomas – BP 8242
69355 Lyon cedex 08, France
{talens,db}@univ-lyon3.fr
[2] SAP Labs France, SAP Research
805, avenue du Dr. Maurice Donat
06254 Mougins Cedex
magali.seguran@sap.com

Abstract. The growth and variety of distributed information sources imply a need to exchange and/or to share information extracted from various and heterogeneous databases. The cooperation of heterogeneous information systems requires advanced architectures able to solve conflicts coming from data heterogeneity (structural and semantic heterogeneity). To resolve semantic conflicts relatively to evolutive domain ontologies following databases evolution according to the dialogue between agents, taking care of scalability issues, we propose a multi-agent system. These interaction protocols allowing ontologies evolution are currently implemented by using Java and the JADE (Java Agent DEvelopment framework) platform.

Keywords: Cooperation of heterogeneous information systems, Ontology elicitation from databases, Ontology evolution.

1 Introduction

The growth and diversity of automated information systems in organizations make the cooperation of information from heterogeneous databases [1], [2] and/or knowledge bases necessary. Every cooperative architecture has to face heterogeneity problems: technical heterogeneity (refers to various operating systems and platforms), syntactic (concerns the diversity of choices regarding data models and query languages) and application heterogeneities. This heterogeneity refers to schema, structural (like generalization/specialization conflict) and semantic heterogeneities.

In order to achieve semantic interoperability the meaning of the exchanged information must be understood across the different systems. Semantic heterogeneity is information sources dependent: semantic conflicts arise when two contexts do not use the same interpretation of the information. Semantic conflicts are classified as follows:

M. Collard (Ed.): ODBIS 2005/2006, LNCS 4623, pp. 51–67, 2007.

- Synonymy conflicts (two entities semantically similar could have two different names)
- Homonymy conflicts (two entities semantically different could have similar names)
- Confounding conflicts (two attributes could be represented by different values or precisions)
- Scaling conflicts (two attributes semantically similar could be represented by different units).

Since a few years, the use of ontologies to extract implicit knowledge is a research-intensive approach to overcome semantic heterogeneity difficulties in the context of cooperation of heterogeneous information sources.

From an other point of view an agent-based solution seems well-adapted to solve semantic heterogeneity problems. In a multi-agent system, it is natural to deal with heterogeneities and conflicts: agents communicate by interaction and negotiation protocols to treat these conflicts.

In our proposal, cooperation is achieved by means of an abstract descriptive layer supporting advanced reconciliation processes and a multi-agent system. The metadata involves descriptive data objects and links constituting a knowledge base (ontology) rich enough to describe: various data models, and constraints, syntactic expressions of local available data, semantic links between local data depending on various application contexts. The knowledge base is integrated in a global project based on a multi-agent approach for heterogeneous information sources cooperation.

Because metadata are distributed in the cooperating agents we have several ontologies but as they share a common description, our approach could be qualified 'hybrid ontology approach' [3].

In first, we have briefly presented the context and the second point reminds some related works. In a third point, agent ontologies are defined. The fourth point focuses on the evolution of the agent ontologies with the interaction protocols during the semantic conflicts resolution. Finally, we conclude and expose some perspectives.

2 Related Works

Numerous projects, based on information brokering have partially dealt with the semantic conflicts solving. These systems use advanced technologies such as information mediation, agent technology or semantic representation based on ontologies, metadata or contexts [4]. For instance, whereas recent works emphasize the need for adaptive ontologies following data source evolutions [5], [6], projects often utilize global [7], [8], [9] and non scalable ontologies. The SIMS [7] model of the application domain offers a hierarchical terminological knowledge base. Each information source is related to one global ontology. INFOMASTER [8] also use single ontology approach. InfoSleuth [10] captures developments such as agent technology, domain ontologies and brokerage to support interoperation of data and services in a dynamic and open environment. InfoSleuth emphasizes on ontologies and brokers. Ontologies give a uniform and declarative description of semantic information and an ontology agent provides an overall view of ontologies. Specialized broker agents semantically match information needs in order to route requests to the relevant resources. The InfoSleuth architecture consists of a set of collaborating agents communicating by the query language KQML. Users express queries over specified ontologies via applet-based user interfaces. KIF (Knowledge Interchange Format) and SQL are used to represent queries over ontologies. Queries are routed by mediation and brokerage to specialized agents for data retrieval from distributed sources and for integration [10]. But the exact description of ontologies integration is not proposed.

In OBSERVER [11] the semantics of one information source is described by one separate ontology. It is not mentioned that the different ontologies share a common vocabulary. To compare the different ontologies, mapping rules are needed. In practice, to define inter-ontology mappings is not trivial.

SCROL [12] proposes a common ontology which specifies consensual vocabulary. The authors argue that a common ontology and the use of a semantic data model provide a complete agreement within the application domains.

COIN project [13], [14] uses a lightweight ontology coupled with powerfull algorithms to realise context mediation.

The approach chosen in PICSEL project [15] is to define an information server as a knowledge-based mediator (called domain ontology) in which the language CARIN is used as the core logical formalism to describe both the domain of applications and the contents of the relevant information sources.

Most recent projects propose an architecture of multi-agent system based on evolutive ontology in a context of e-commerce as [16]. The DASMAS project [17] presents a dialogue framework-based for resolving semantic interoperability in multi-agent systems. The approach is characterized by: several multi-agent systems with real world heterogeneous ontologies, the resolution of semantic differences at run-time through an adapted protocol and the use of WordNet lexicon in the resolution process. An ontology is associated to one multi-agent system and WordNet permits to find semantically similar concepts in the heterogeneous ontologies.

To address the problem of ontology evolution, research projects propose to build different versions of an ontology.

The problems of versioning and evolution in ontologies is significantly different with those in the relational databases [18], [19], [20]. The authors [20] define ontology versioning and evolution as '*the ability to manage ontology changes and their effects by creating and maintaining different variants of the ontology*'.

In ontology evolution and versioning, two techniques exist : the first keeps track of changes in a new version or compares ontologies and computes differences or mappings between them. The second proposes automatic techniques based on heuristics comparisons to find similarities and differences between the different versions.

The OntoView system [21] helps a user to manage changes in ontologies and keeps the ontology versions. It compares the versions of ontologies and highlights the differences. It also allows the users to specify the conceptual relations between the different versions of concepts.

In [22], the researchers propose a general framework for ontology evolution that allows tools supporting different evolution tasks to share change information and leverage change information obtained by other external tools. A structural comparison of ontology versions is also proposed.

SHOE [23] does not keep track of changes from one version to another. SHOE maintains each version of the ontology as a separate web page. The ontology designer copies the original ontology file, assigns it a new version number, and adds or removes elements as needed.

In [24] through the notion of evolution strategy, the users guide the ontology evolution. They can control and customize the evolution process. [25] keeps track of different versions of an ontology and offers the possibility to allow branching and merging operations. Protegé [26] keeps track of, and records, ontology changes within the ontology itself. It also compares versions of the same ontology.

On the market, Software AG [27] emerges and has developed an XML integration solution allowing the integration of data sources as databases, XML-files and Web Services. More recently, the ‘Information Integrator’ [28] proposes a single and coherent view of disparate information sources by using a common ontology. This domain ontology so-called ‘business ontology’ reinterprets the data described in the local data-source ontologies. This reinterpretation is a way to represent complex knowledge interrelating these data. This reengineering process of the data source contents cannot be done automatically.

Scalability (the complexity of creation and maintaining the interoperation services should not increase exponentially with the number of participating local information sources) and extensibility (the ability to incorporate local information system changes without having adverse effects on other parts of the larger system) are not really treated in the case of multi-domain approaches.

Therefore, cooperative architectures with a multi-domain approach have difficulties to deal with scalability and extensibility. Thus, they do not deal with adaptative ontologies.

So, we present a proposition for semantic conflict resolution that integrates domain ontologies evolution.

3 Agent Ontologies

This work is involved in the ACSIS (Agents for the Cooperation of Secure Information System) project [29], [30], [31].

In our proposal actual information sources cooperation is based on agents interactions. Each local source is represented by one or several agents and the set of agents constitutes a multi-agent system. The scope of distributed artificial intelligence brings techniques to implement multi-agent architectures able to dynamically face the various emerging problems of information systems cooperation. The reasons for modelling a system using multiple cognitive agents are various, they range from agent cognitive capabilities to multi-agent dynamic features [32], [33]:

- Agents are autonomous, thus they can define their own internal goals and plans,
- they are able to deal with high level interactions through domain independent communication messages,
- a multi-agent architecture can dynamically evolve according to the problem to solve and even during the problem resolution,

- agents can detect changes in their environment, modify their behaviour and update their internal knowledge base describing the environment,
- they are able to cooperatively solve problems (in particular knowledge-intensive ones like semantic conflict resolution) through interactions and negotiation protocols,
- agents allow the construction of open and scalable architectures (easy addition or removal of data sources).

ACSIS architecture aims at resolving technical, syntactic, application (structural and semantic) heterogeneities that appear during the cooperative processes. Our architecture [29] comprises several levels to treat these different types of heterogeneities:

- The technical heterogeneity between information sources is performed by using a CORBA (Common Object Request Broker Architecture) middleware.
- The syntactic heterogeneity is resolved by Data Descriptive Objects (cf. paragraph 3.2) ensuring the homogenization of local data or knowledge bases.
- The structural and semantic heterogeneity is resolved during query processing by using multi-agent system and interaction protocols.

Scalable domain ontologies are used to represent the agents' knowledge corpus. Each agent owns its ontology. The agents and their ontologies are described as follows.

3.1 The Agent Model

An agent comprises several units (ontology unit constituted by Data Descriptive Objects and links between these objects), acquaintances (list of closed known agents), reasoning, communication, behaviour.

The defined multi-agent system is composed of different types of agents (see Fig. 1).

The *Wrapper Agent (WA)* ensures the participation of local data to the cooperative processes. Each WA is linked to a domain from a local database and DDOs (Data Descriptive Objects) and intra-base links form its ontology.

The *Information Agent (IA)* structures the exchange between WAs during the processing of global queries and semantic conflict resolution. Its ontology is composed of the semantics links at the global level (inter-bases links). Each IA groups WAs according to semantic characteristics. An IA accesses to at least one, and potentially many information sources, and is able to collate and manipulate information extracted from these sources in order to answer the users and other IAs.

Each IA is a multi-domain agent. Its ontology is formed by the inter-bases links.

The *Interface Agent* insures intermediation between the user (expert or user role) and the other agents:

- The User Agent manages the query, validates the results and asks the re-execution of the query if the results are not correct.
- the domain Expert Agent defines some intra-base links, chooses the database type (relational/object) and gives a representative name of the domain.

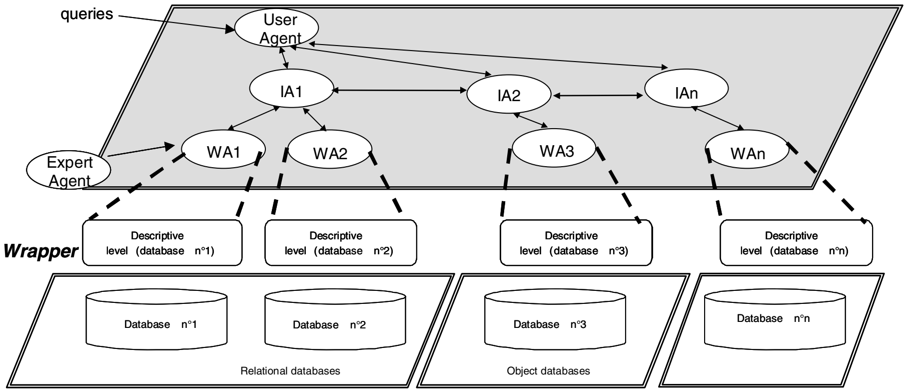

Fig. 1. The different agents

Agents exchange information by interaction protocols to solve semantic conflicts and to manage the evolution of domain ontologies.

3.2 Scalable Domain Ontologies

Models describing ontologies come from Distributed Artificial Intelligence, Knowledge representation or Databases [34]. Two different directions are envisaged:

- the first is Distributed Artificial Intelligence oriented and proposes descriptive logic with inference tools
- the second is Database-oriented and presents extended conceptual models so as to represent all the informations.

We will adopt the ontology definition in a database/knowledge sharing approach. Nevertheless, we integrate some inference rules. Ontology is an explicit, partial specification of a conceptualization [35]. A conceptualization could be a set of concepts, relations, objects and constraints defining the domain semantic model. An ontology can be defined as a specific vocabulary and relationships used to described certain aspects of reality and a set of explicit assumptions regarding the intended meaning of the words vocabulary [36].

Recently, other definitions are used in the context of oriented mediation-cooperation projects. Mena gave the following precise definition [37]:
'ontology is a description of the concepts and relationships that can exist for an agent or a community of agents. This definition is consistent with the usage of ontology as set-of-concept-definitions, but more general. And it is certainly a different sense of word than its use in philosophy. Ontology is a set of terms of interest in a particular information domain and the relationships among them'.

In our approach, the ontology of each agent contains Data Descriptive Objects (DDO) and links between these objects [29]. The DDOs contain the description of data from local information sources as well as the access primitives to this data. Local information entities (relation, relation attribute, primary key, object type, object attribute…) are described so that each information source involving the cooperation process is represented by a set of DDOs. The *relation/object* DDOs describe a class or a relation. There is no difference between relationship or entity in our modeling process, each concept is a *relation* DDO. The *attribute* DDOs include object attributes (it could be *object attribute* or *reference object attribute* that stores a pointer on an object) and *relation attributes* (it could be *primary key*, *foreign key* or *relation attribute*).

The ***links*** connect DDOs, according to schematic, structural or semantic characteristics.

Schematic Links between these DDOs are automatically extracted. The figure 2 presents the relations: *firm* (id firm, name), *office worker* (id, firstname, wage, id firm).

The *dependence links* allow connecting the attribute DDOs to a relation/object DDO.

The *reference links* allow to connect a reference DDO and a refereed DDO.

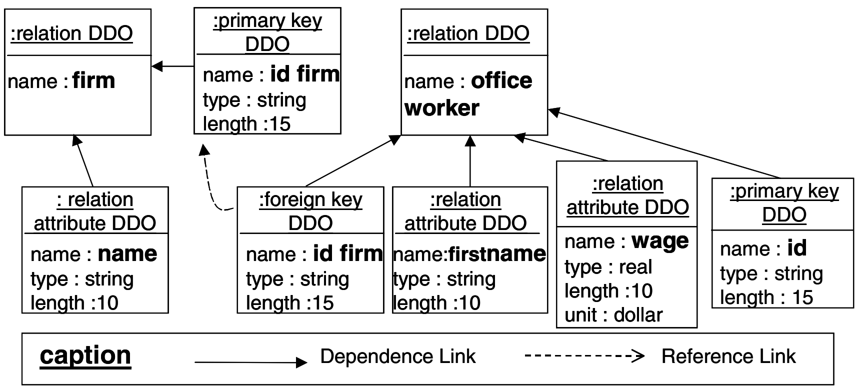

Fig. 2. Reference and dependence links between DDOs

Structural Links (generalization, specialization) are automatically extracted in the case of object approach or defined by a domain expert in the case of relational approach.

Semantic Links connect two DDOs, according to their semantic characteristics. The links are defined either by a domain expert or are automatically created during the query processing.

Synonymy Links describe a similar sense between two DDOs with different name (for example between *employee* DDO and *office worker* DDO).

Non Synonymy Links describe a different sense between two DDOs with different names.

Similarity Links describe a similar sense between two DDOs with same name.

Homonymy Links describe a different sense between two DDOs with same name (*name* DDO and *name* DDO if *name* is the attribute of *project* and *name* is the attribute of *employee* (see Fig. 3)).

Scale Links describe a same scale between DDOs which have a same unit (employee *wage* DDO and director *wage* DDO with *Dollar* unit).

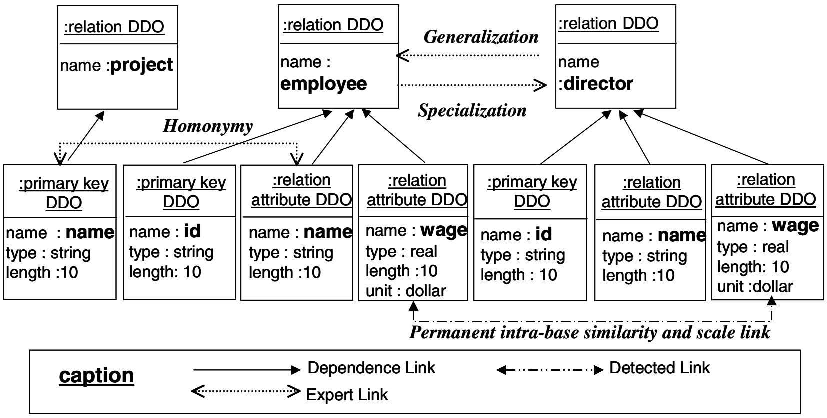

Fig. 3. Ontology example

Different Scale Links specify a unit existence between two DDOs with same name (employee and director *wage* DDOs with *Dollar* unit because there are US and Canadian dollar).

Conflict semantic resolution is performed by the use of links and DDOs and by the new links detection during the query processing. The user must validate these detected links. Therefore, these synonymy, similarity and different scale links could be **temporary** links (detected by the system and have to be validated), **permanent** links (created by an expert, or validated) or **user** links (link inserted by the user). Non synonymy, homonymy and scale links are permanent links. An **intra-base** link connects two DDOs extracted from the same database; an **inter-bases** link connects two DDOs extracted from two different databases.

DDOs hierarchy along with these local semantic links forms an ontology.

4 Interaction Protocols and Ontologies Evolutions

The interactions between agents are managed by a set of rules that forms interaction protocols dedicated to conflict resolution.

In ACSIS project, interactions reuse the FIPA protocols [38]. The conflict resolution is performed in a dynamic way during the insertion of a new information source and the global query processing. The conflicts are solved relatively to the link exploitation. The interaction protocols allow the domain ontologies to evolve with the automatic detection of new links. When the detected links are validated, a semantic inference process also allows the ontologies evolution.

In [39], an ontology is used to describe interaction protocols. Thanks to the global ontology, the agents can dynamically adapt their behavior. In ACSIS project, the interaction protocols cannot be changed because only the ontologies encapsulated in the agents evolve, not the agents. These latter transfer the informations according to predefined interaction protocols.

4.1 Insertion of a New Information Source

The registration step begins with the creation of a WA (a Wrapper Agent is created for one database) and continues with the attachment to an IA (Information Agent).

Creation Step of a WA

The DDO hierarchy with the extracted links is encapsulated in a WA. Once created, each WA can automatically detect temporary intra-base similarity links relatively equivalence based on the DDO's name (for example, *director.wage* (dollar) and *employee.wage* (dollar)). If a unit is specified in the *attribute DDOs,* some scale difference links are created. For the scale, we cannot detect equivalence with the name of the unit because wage (dollar) and wage (dollar) could not be a scale link (for example, it could be US dollar or Canadian dollar). *Temporary similarity and different scale* links are created if there are not existing homonymy and scale links.

Some *similarity* and *scale* links between *attribute DDOs* could be created in a *permanent* mode if specialization/ generalization links exist between the respective relation DDOs.

The following example (see Fig. 3) presents the relations: *project* (name), *employee* (id, name, wage), *director* (id, name, wage). The *director* relation DDO specializes *employee* relation DDO. So, there are a permanent similarity and scale link between the *wage* attribute DDO (depending of *director* relation DDO*)* and the *wage* attribute DDO (depending of *employee* relation DDO*).* The expert creates only some intra-base links which cannot be automatically created, for example the homonymy links. He also specifies the database domains.

Registration of a WA to an IA

The Registration protocol allows the registration of a WA (when a new source integrates the system) and therefore increases the WA network attached to an Information Agent. Each WA dynamically queries to be joined to the IAs that are previously created. The IA, whose domain is semantically the closest, integrates this WA into its acquaintance network (the Contract Net Protocol allows to choose the WA). The IA establishes a comparison between the network's WA so as to create temporary inter-bases similarity and different scale links (the *Fipa Query Protocol* is used).

When a new WA is recorded into an IA, the different DDOs are sent to the other WAs in order to discover new temporary inter-bases links. In our example, five similarity links are created:

– firm name DDO respectively with employee name, with director name, with project name,
– office worker wage respectively with employee wage DDO, with director wage DDO.

Two different scale links are added: office worker wage respectively with employee wage DDO, with director wage DDO.

4.2 The Global Query Processing Protocol

The global query processing protocol organizes the negotiation phases between the IAs and the WAs to resolve semantic conflicts. There are different phases in the global query processing protocol (principally based on the *Fipa Query Protocol*):

Transmission of the Query from the User Agent to IAs and WAs
Each Information Agent looks for its inter-bases semantic links and broadcasts the query to the WAs of its acquaintance network all the while taking into account its inter-bases links.

Semantic evaluation
Each WA accepts or refuses the query request relatively on knowledge of query elements by using synonymy intra-base links and homonymy links.

Links of others IAs
When the WAs of the acquaintance network don't have sufficient knowledge to answer the query, the IA asks other IAs inter-bases synonymy links to modify the concepts of the query. The modified query is send again to its connected WAs.

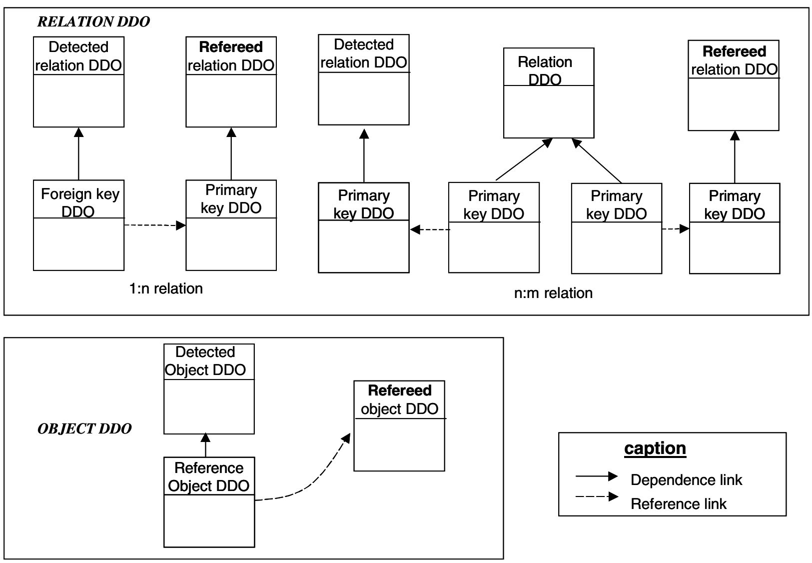

Fig. 4. Selection of the refereed DDO with reference and dependence links

Creation of New Temporary Intra-base Synonymy Links
During this semantic evaluation, each WA can create temporary intra-base synonymy through some schematic links (like the *reference links* and *dependence* links) according to the following method:

- Item 1. For each query element, if none DDO corresponds, there is a selection of the refereed DDO relatively dependence and reference links, from detected DDO (see Fig. 4).
- Item 2. If there is no attribute specified in the query, or if the specified attribute is equivalent to the attribute DDO depending of the object/relation DDO, a temporary intra-base synonymy link is created between the reference DDO and a virtual object/relation DDO (a virtual DDO is a DDO only created for the representation of this temporary link). If the attribute specified in the query matches with the attribute DDO depending of the reference DDO, a temporary intra-base similarity link is created between this attribute DDO and the attribute element of the query.

In the following example (see Fig. 5), the database contains the relations: *project* (name), *work* (id, name) and *employee* (id).

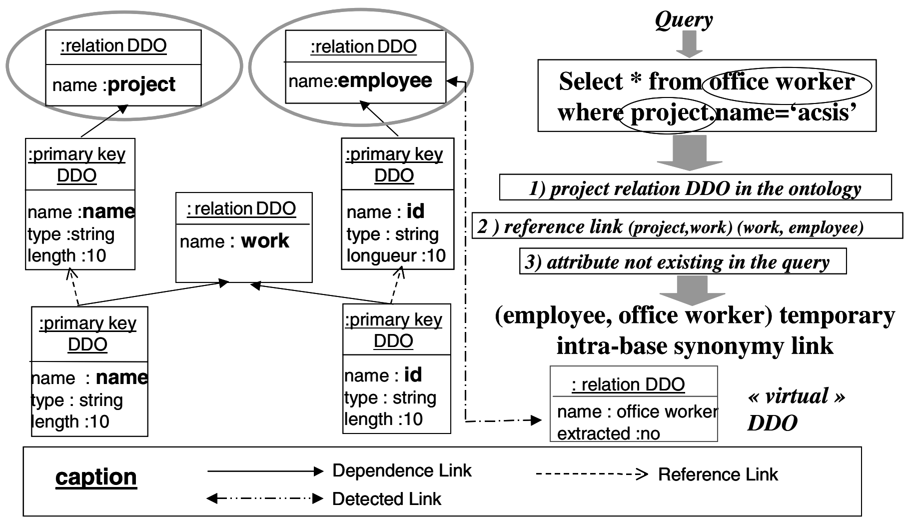

Fig. 5. Creation of a temporary intra-base synonymy link

The query is "select * from office worker where project.name='acsis'". The *office worker* DDO does not exist in the WA's ontology. Relatively to the *reference link* between *project* (existing element in the ontology) and *work* and, *work* and *employee*, the refereed *employee* DDO is selected. There is no attribute relevant *to office worker* indicated in the query (respect of item 2). Therefore, a temporary intra-base synonymy link is created between the *employee DDO* and an *office worker* virtual DDO. The WA replies with these temporary intra-base synonymy links and with the DDO names equivalent to the query elements.

Creation of New Temporary Inter-bases Synonymy Links

When the WAs send some temporary intra-base synonymy links to their IA, the IA could create some temporary inter-bases synonymy links if there is the same term in an other WA of its acquaintance network. For example, the *office worker* DDO exists in another WA. A temporary inter-bases synonymy link is created between the *office* *worker* DDO and the *employee* DDO. When there is a creation of synonymy and similarity link, a corresponding link to validate is instantiated, and passed from the IA or WA to the User Agent. It allows simplifying the queries execution on the local databases. A sub-query comprises only the global query parts on which the IA has the relevant knowledge and replaces term by using temporary inter-bases synonymy links. The IAs then contact the WAs which contain knowledge in order to perform the sub-query.

Retrieving the Results

Each WA can accept or refuse (agree/refuse performative) to process the query. If it agrees, it queries its local database using temporary and permanent semantic intra-base links and structural links, retrieves data coming from local sources via DDOs and sends them to its IA (inform or failure performative).

The global query processing is presented (see Fig. 6) by using Agent UML [40].

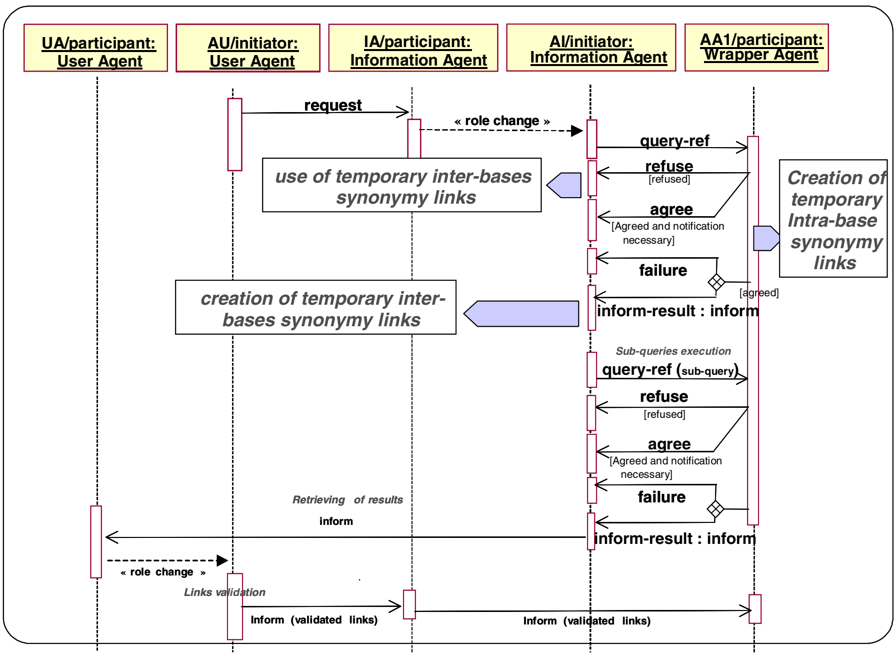

Fig. 6. Protocol for global query processing (in Agent UML)

For example, if the query is "select * from employee where employee.name='smith'", with the specialization/generalization link, the following query "select * from director where director.name='smith'" is also executed. When they retrieves the results, each WA (at the local level) and each IA (at the global level) uses intra-base or inter-bases different scale links to translate data in the expected format (if conversion functions are detected in the DDOs).

Validation of Results

Each IA restructures the responses obtained from its WAs and sends them to the user agent. At the end of the protocol, the ontologies are updated when new semantic links are discovered and after the validation of these links by the User Agent. If synonymy, similarity and different scale links are validated, they become permanent links : if they are not validated, they become non synonymy, homonymy and scale links. The User Agent can insert synonymy links if the results are not correct. Some creation of synonymy links could be performed by the WA or the IA:

- An *inter-base synonymy link (user type)* is created if the two involved terms are situated in the two different WA.
- An *intra-base synonymy link (user type)* is created if the two involved terms are situated in the same WA. An intra-base synonymy link is created using a virtual DDO as soon as a term is in a WA.

Sub-queries (WA or IA level) are re-executed in several cases (see Fig. 7):

- when intra-base or inter-bases links are not validated, the sub-queries are re-executed at the WA level or the IA level,
- when synonymy links are inserted by the user, only the modified parts of the query are re-executed towards the WA, the results being preserved at the level of each IA,
- when conversion functions are inserted by the user (they are encapsulated within the validated links towards IAs which transmit to the WAs), the corresponding sub-query must be again re-executed.

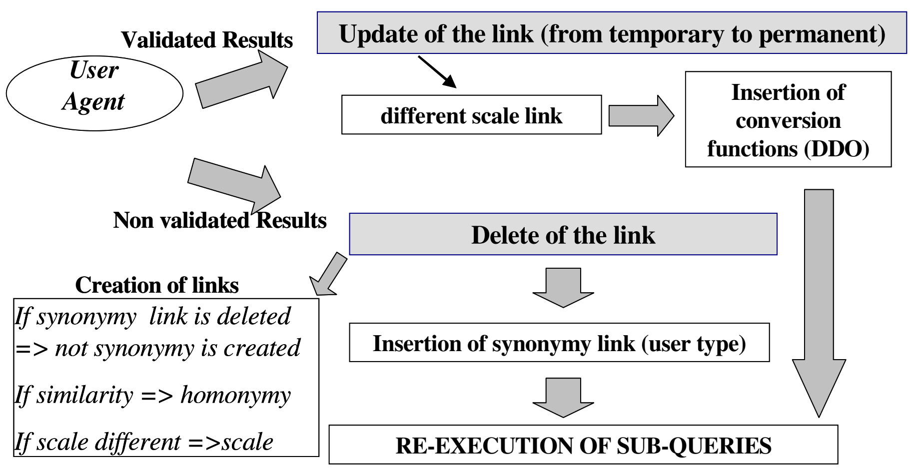

Fig. 7. Links validation process

4.3 Database Update

Different problems arise with the ontology modifications: Incompatibility of instances and incompatibility of the related applications. In ACSIS, they are managed by the database administrator in respect of the local sources autonomy.

Further, we think to use ontology versioning to capture the evolution proposed by the user as long as it is not a sound and validated evolution.

Different modifications can be performed on a database. Incompatibilities of linked ontologies are managed as follows.

	Addition	Deletion	Modification
Relation/Class	Adding of the corresponding DDO	Deletion of the DDO and updating of the concerned links	- name - attribute adding - attribute deletion
Attribute	Adding of the corresponding DDO	Deletion of the DDO and updating of the concerned links	- name - type (modification of the DDO) - unit

- Addition :
  The creation of similarity and different scale intra-base links is processed as for the insertion of a new database as previously explained. The new DDOs are compared with the other DDOs in order to create new links.
- Deletion :
  The intra-base links are deleted but not the inter-bases links because they constitute global knowledge.
- Modification of attribute, relation or class name:
  Modification of the concerned DDO, creation of a virtual DDO in order to store the old name and creation of a synonymy link between the two DDOs.
- Modification of attribute unit :
  Modification of the DDO and updating of the scale and different scale intra-base links.

The different modifications are sent to the IA and the latter sends them to its WAs.

The creation of new links is therefore performed in a dynamic way during the insertion of a new information source and the global query processing. It also performed after the User Agent validation when links become permanent. When a link becomes permanent, each WA or IA could automatically create new semantic links in respect of semantic inferences rules [31]. Semantic inferences also contribute to perform the ontology evolution.

5 Conclusion

Semantic conflict resolution is processed by using ontology during two steps: the insertion of a new information sources and the global queries resolution. The real dialogue between agents managed by a protocol, enables agents' ontologies to evolve. The scalability of the system comes from the new link detection (scalable domain ontology) and the ability of following the evolution of local databases (relatively to the DDOs level). The cooperative architecture with the interaction protocols is implemented by using Java and the JADE platform (Java Agent DEvelopment framework) [41][42]. JADE is a software framework to develop agent-based applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems and provides a library of FIPA interaction protocols ready to be used. JADE provides a support for content language and ontologies allowing the developers manipulating information within their agents as Java objects without the need of any extra work. The JADE support performs the conversion between the information represented as a string or a sequence of bytes at ACL Message level and Information represented as Java objects (easy to manipulate) at agent level. In our platform, this support has been useful to pass links and queries objects. Currently the prototype runs. We have implemented the main protocols and processed some queries. We have proposed a solution based on an extended conceptual model integrating some principles coming from Distributed Artificial Intelligence like interaction protocols between agents and inference rules on the detected links. It is a mixed approach combining advantage from descriptive logics and an extended conceptual model. In our future works, we wish to continue to integrate tools of reasoning in the model of evolutive ontologies proposed in this article.

The concept of version must be developed on the ontology to capture the modifications performed by the user. The ontology must be not directly modified as long as the modifications have not been completely validated. The user's modification becomes public after its validation by a super user or an administrator. Our ontology versions will keep track of the different modifications (add, update, delete) in order to better follow the evolution and to perform the impacts on the other ontologies after the version validation.

References

1. Bouguettaya, A., Benatallah, B., Elmagarmid, A.K.: Multidatabase Systems: Past and Present. In: Elmagarmid, A.K., Rusinkiewicz, M., Sheth, A. (eds.) Distributed and Heterogeneous Database Systems. Morgan Kauffmann, San Francisco (1998)
2. Fankhauser, P., Gardarin, G., Lopez, M., Muñoz, J., Tomasic, A.: Experiences in Federated Data-bases From IRO-DB to MIRO-Web. In: 24th VLDB'98, New-York, USA, pp. 655–658 (1998)
3. Wache, H., Vogele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hubner, S.: Ontology-Based Integration - a survey of Existing Approaches. In: IJCAI'01 Workshop: Ontologies and Information Sharing, pp. 108–117 (2001)
4. Sheth, A.: Changing Focus on Interoperability in Information Systems: from System, Syntax, Structure to Semantics. In: Interoperating Geographic Information Systems. Kluwer Academic Publishers, Boston, MA (1999)
5. Kahng, J., Mac Leod, D.: Dynamic Classificational Ontologies. In: Arbid, M.A., Grethe, J. (eds.) Computing the Brain : A Guide to Neuroinformatics. Academic Press, San Diego (2000)
6. Gal, A.: Semantic Interoperability in Information Services. Experiencing with CoopWARE. ACM Sigmod record 28(1) (1999)
7. Arens, Y., Chee, C.Y., Hsu, C., Knoblock, C.A.: Retrieving and Integrating Data from Multiple Information Sources. IJCIS 2(2), 127–157 (1993)
8. Genesereth, M.R., Keller, A.M., Duschka, O.M.: Infomaster: an Information Integration. System. In: Proceedings of the ACM SIGMOD Conference (1997)
9. Aparicio, A.S., Farias, O.L.M., dos Santos, N.: Applying Ontologies in the Integration of Heterogeneous Relational Databases. In: Australian Ontology Workshop, AOW'2005, Sydney, Australia (2005)

10. Nodine, M., ali.: Active Information Gathering in InfoSleuth. IJCIS 9(1-2), 3–28 (2000)
11. Mena, E., Illarramendi, A., Kashyap, V., Shet, A.: Observer: an approach for query processing in global information systems based on interoperation across pre-existing ontologies. Int'l Journal Distributed and Parallel Databases 8(2), 223–271 (2000)
12. Ram, S., Park, J.: Semantic Conflict Resolution Ontology (SCROL): An Ontology for detecting and resolving data- and schema-level semantic conflicts. IEEE Transactions on knowledge and data engineering 16(2), 189–202 (2004)
13. Goh, C.H.: Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources. PhD. MIT, Cambridge, MA (1997)
14. Zhu, H., Madnick, S.: Structured contexts with lightweight ontology. In: VLDB Workshop on Ontologies-based techniques for DataBases and Information Systems, Korea (2006)
15. Reynaud, C., Safar, B.: Representation of ontoloies for Information Integration. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, Springer, Heidelberg (2002)
16. Rosaci, D.: A Model of Agent Ontologies for B2C E-Commerce. In: Proceedings of the International Conference on Enterprise Information Systems (ICEIS 2004), Porto, Portugal, pp. 3–9 (2004)
17. Orgun, B., Dras, M., Nayak, A.: DASMAS – Dialogue based Automation of Semantic Interoperability in Multi Agent Systems. In: Australian Ontology Workshop, AOW'2005, Sydney, Australia (2005)
18. Bounif, H.: Predictive Approach for Database Schema Evolution. In: Ma, Z.M. (ed.) Intelligent Databases. Idea Group Publishing, USA (2006)
19. Bounif, H., Pottinger, R.: Schema Repository for Database Schema Evolution. In: Bressan, S., Küng, J., Wagner, R. (eds.) DEXA 2006. LNCS, vol. 4080, pp. 647–651. Springer, Heidelberg (2006)
20. Noy, N.F., Klein, M.: Ontology Evolution: Not the same as Schema Evolution. Knowledge and Information Systems 5 (2003)
21. Klein, M., Kiryakov, A., Ognyanov, D., Fensel, D.: Ontology versioning and change detection on the web. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473. Springer, Heidelberg (2002)
22. Klein, M., Noy, N.F.: A component based framework for ontology evolution. In: Workshop on Ontologies and Distributed Systems, IJCAI'03, Mexico (2003)
23. Heflin, J., Hendler, J.: Dynamic Ontologies on the Web. In: Proceedings of the 17th national conference on Artificial Intelligence, AAAI 2000, pp. 443–449. AAAI/MIT Press, Menlo Park, CA (2000)
24. Stojanovic, L., et al.: User-driven Ontology Evolution Management. In: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, Ontologies and the semantic web (2002)
25. Auer, S., Herre, H.: A Versioning and Evolution Framework for RDF Knowledge Bases. In: Proceedings of Ershov Memorial Conference (2006)
26. Liang, Y., Alani, H., Shadbolt, N.R.: Ontology change management in Protégé. In: Proceedings of Advanced Knowledge Technologies Doctoral Colloquim, United Kingdom (2005)
27. Software AG: http://www.softwareag.com/fr/
28. Angele, J., Gesmann, M.: Semantic Information Integration with Software AGs Information Integrator. In: Second International Conference on Rules and Rule markup languages for the Semantic Web, Athens, Georgia, USA (2006)
29. Boulanger, D., Dubois, G.: An Object Approach for Information System Cooperation. Information Systems 23(6), 383–399 (1998)

30. Couturier, V., Séguran, M.: Patterns and Components to Capitalise and Reuse a Cooperative Information System Architecture. In: Int'l Conf on Enterprise Information System Architecture ICEIS 2003, Angers, April 23-26, pp. 225–231 (2003)
31. Séguran, M.: Résolution des conflits sémantiques dans les systèmes d'information coopératifs. PhD thesis, Université Jean Moulin, Lyon, France (2003)
32. Papazoglou, M., Laufmann, S., Sellis, T.: An Organizational Framework for Cooperating Intelligent Information Systems. IJCIS 1(1) (1992)
33. Klusch, M.: Intelligent Agent Technology for the Internet: A Survey. Journal on Data and Knowledge Engineering. In: Fensel, D. (ed.) Special Issue on Intelligent Information Integration, vol. 36(3), Elsevier Science (2001)
34. Cullot, N., ali.: Ontologies: A contribution to the DL/DB debate. In: Proceedings of the 1rst Int'l Workshop on Semantic Web and Database (SWDB'2003) co-located VLDB'2003, Germany, pp. 109–130 (2003)
35. Gruber, T.: The Role of a Common Ontology in Achieving Sharable, Reusable Knowledge Bases. In: Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning, pp. 601–602. Cambridge University Press, Cambridge (1991)
36. Guarino, N.: Formal Ontology and Information Systems. In: Proceedings of the 1st International Conference on Formal Ontology in Information Systems (FOIS'98), Torino, pp. 3–15 (1998)
37. Mena, E., Illarramendi, A.: Ontology-based Query Processing for Global Information Systems. Kluwer Academic Publishers, Boston, MA (2001)
38. Fipa: http://www.fipa.org
39. Toivonen, S., Helin, H.: Representing Interaction Protocols in DAML. In: van Elst, L., Dignum, V., Abecker, A. (eds.) AMKM 2003. LNCS (LNAI), vol. 2926, pp. 310–321. Springer, Heidelberg (2004)
40. Odell, J., Parunak, H., Bauer, B.: Extending UML for Agents. In: Wagner, G., Lesperance, Y., Yu, E. (eds.) Proceedings of Agent-Oriented Information Systems Workshop at the 17 th National conference on Artificial Intelligence. Icue Publishing, Austin, Texas (2000)
41. Bellifemine, F.: Jade and beyonds. Presentation at AgentCities Information Day 1, Lausanne (2002)
42. Jade: http://sharon.cselt.it/projects/jade