# DOMAIN ONTOLOGY EVOLUTION BY VERSIONING

Guilaine Talens and Danielle Boulanger
*MODEME, Université Jean Moulin 6 cours Albert Thomas – BP 8242 69355 Lyon cedex 08, France*

Keywords: Ontology creation, Ontology evolution, Ontology versioning, Semantic interoperability, Heterogeneous information sources.

Abstract: To solve the semantic conflicts in the cooperation of heterogenous databases, we propose a multi-agent system which contains domain ontologies. When a local source is added by an expert, ontology is dynamically created by the system. The expert can also completed it. It evolves with the update of the local base but also when an user performs a query. A process examines the query and creates temporary semantic links. These must be validated or not by the user. The result validation implies the version creation from temporary links. The modification and deletion of database elements by the expert perform the ontology evolution. Our research work treats evolution by the version concept.

## 1 INTRODUCTION

The cooperation of heterogeneous information systems requires advanced architectures able to solve conflicts coming from data heterogeneity (Sheth, 1999). We propose a Multi-Agent System (MAS) to resolve semantic conflicts relatively to evolutive domain ontologies following databases evolution according to the dialogue between agents, taking care of scalability issues. These interaction protocols allowing ontologies evolution.

In our research, we focus on ontology versioning to treat ontology evolution. Ontology versioning is defined as 'the ability to manage ontology changes and their effects by creating and maintaining different variants of the ontology' in (Klein & al., 2002). This functionality is important in scenarios where users access an ontology in a distributed manner.

In different previous research projects, ontologies often evolve and the old versions are lost because only the latest exists. Sometimes old and new versions are archived but no mechanisms are exhibited to expose the differences between versions. Currently most of the works in the ontology evolution and versioning captured their ideas from schema versioning and schema evolution in databases. But the content and the management of ontologies are more difficult than database schemas (Noy and Klein, 2003): ontologies incorporate much more semantics and thus help to solve integration problems between several information sources. 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. So, the characteristics of ontologies do that the concept of evolution and versioning in database schemas cannot be applied directly in ontologies.

Ontology versioning has to focus on several aspects :

- providing frameworks for ontology versioning and evolution in the multi-ontology management context,
- modelling and representing the details of changes between ontology versions,
- specifying ontology changes operations and analysing the impacts of these operations in different contexts,
- developing algorithms and tools to compare the different versions of a same ontology.

The current research works only treat few or none of these points.

In first, we have briefly presented the problem. In a second point, the domain ontology concept is described. The third point focuses on ontology evolution during the multi-database query execution. The fourth point exhibits the evolution database through version concept. Finally, we conclude by highlighting the specific features of our contribution.



Talens G. and Boulanger D. (2009).
DOMAIN ONTOLOGY EVOLUTION BY VERSIONING.
In *Proceedings of the International Conference on Agents and Artificial Intelligence*, pages 185-190
DOI: 10.5220/0001557001850190

## 2 DOMAIN ONTOLOGIES

The ACSIS (Architecture Cooperative of Secure Information System) architecture (Boulanger and Dubois, 1998) allows the structural and semantic conflicts resolution throws a multi agent system.

In our approach, the ontology of each agent contains Data Descriptive Objects (DDO) and links between these objects. An ontology can be defined as a specific vocabulary and relationships used to described some aspects of reality and a set of explicit assumptions regarding the intended meaning of the words vocabulary (Gruber, 1993) (Guarino, 1997). Recently, the Observer project (Mena and Illarramendi, 2001) has described an ontology as the concepts and the links which exist for an agent or for several agents.

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 field, primary key, object type, object attribute…) are described so that each information source participating in the cooperation process is represented by a set of DDOs.

The DDOs describe a class (in the object oriented databases) or a relation (in the relational databases).

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.

Three agent types interact:

- 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. The DDOs 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 obtained from these sources in order to answer the users and other IAs (Klusch, 2001). Each IA is a multi-domain agent. Its ontology is formed by 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 satisfying.

* the domain *Expert Agent* defines some intra-base links, chooses the database type (relational/object) and gives a representative name of the domain.

Agents exchange information (see figure 1) by the mean of interaction protocols to solve semantic conflicts and to manage the evolution of domain ontologies.

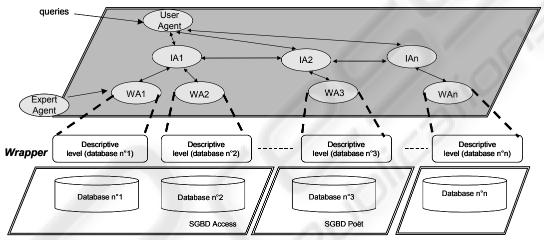


Figure 1: The different agents.

We explain our ontology through an example created from a relational database (see figure 2). A WA is created through the Expert Agent request to add a new database. This new WA is named “business” by the expert. A second WA named “recreation” from an object oriented database (see figure 3). These two WAs are attached to the same IA.

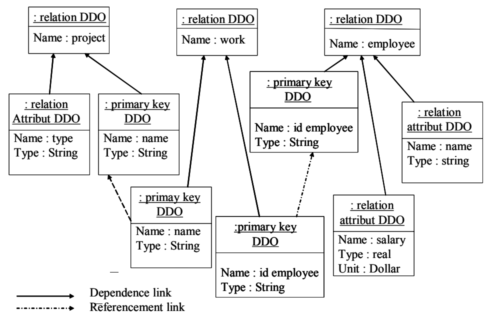


Figure 2: Ontology created from relational database.

The DDOs and the schematic links are automatically extracted from the database. The dependence links allow the attribute DDOs connecting to a relation DDO. The reference links allow connecting the key DDOs. For example in the figure 2, the *id employee key* of the *work* relation references the *id employee key* of the *employee* relation.

After the automatic extraction, the expert can add semantic links:

*Synonymy Links* describe a similar sense between two DDOs with different names (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 the same name.

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

*Scale Links* describe a same scale between DDO with same name (*wage* DDO with *Dollar* unit and *wage* DDO with *Dollar* unit).

*Different Scale Links* specify a unit existence between two DDOs with same names (*wage* DDO with *Dollar* unit and *wage* DDO with *Dollar* unit when dollar is US or Canadian).

A link can have two states: permanent and temporary. The state is temporary if the system creates it; it is permanent, if it is created by a database expert or if five users have validated the system link creation.

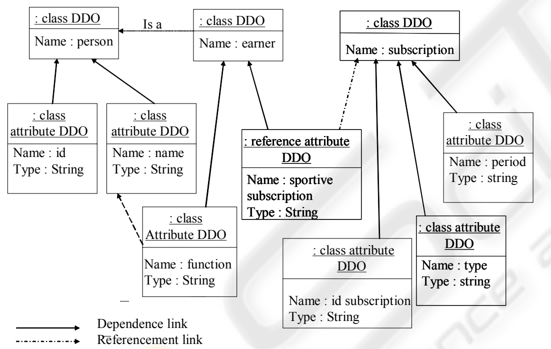


Figure 3: Ontology created from object oriented database.

All these DDOs and links are created when a new database is added: creation of a WA and the WA attachment to an existing IA or creation of a new IA (Talens, Boulanger, Séguran 2007).

These links can be also automatically created during the query processing. A query example is detailed to explain the ontology evolution.

# 3 ONTOLOGY EVOLUTION

## 3.1 Multi-base Queries

When a query is executed through the User agent, different agents are contacted. Negotiation protocols are defined for the communication between the different agents: User Agent, Wrapper Agents and interface Agents (Talens, Boulanger and Séguran, 2007). We only focus on the domain ontology.

The user writes through a User Agent the following query: "SELECT * From employee WHERE employee. salary=1500 euros AND subscription . type="year"". This query is sent to Information Agents. The IAs exploit their ontology in order to find interesting inter-bases semantic links. After, each IA broadcasts to the network's Wrapper Agents the modified query (in the case of relevant links have been identified, on the contrary the initial query is sent).

Each WA extracts the different elements (attribute, relation) of the query and compares them to the DDOs. The intra-base synonymy and homonymy links are also examined.

The business WA has the employee relation DDO and the salary attribute DDO. A different scale link is created (see figure 4) because the unit is dollar in the attribute and euros in the query. A virtual attribute is created to represent the different scale link. A virtual DDO represents a query term; it is only created for the representation of a temporary link.

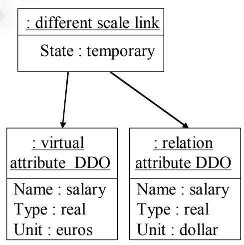


Figure 4: Different scale link between dollar and euros.

The subscription relation is not found. When one or several elements are not found, a process is applied for each of them to select a potential relevant DDO. A temporary intra-base synonymy link is created without the User Agent intervention thanks to some schematic links (like the *reference links* and *dependence* links) between the selected DDO and the query element. This latter does not exist, so a virtual DDO is created; it is a DDO only created for the representation of this temporary link.

For each query element, if none DDO corresponds in the ontology, there is a selection of the *refereed DDO* thanks to *dependence and reference* links, from detected DDOs. Moreover, if there is no attribute specified in the query, or if the specified attribute is equivalent to the *attribute DDO* depending of the *class/relation DDO*, a *temporary intra-base synonymy* link is created between the *reference DDO* and a *virtual class/relation DDO.* If the specified attribute 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 our example, thanks to the dependence link from type attribute, the project relation DDO is selected. A temporary synonymy link is created between the project DDO and a virtual DDO named subscription (query element). Another temporary link is created; it is a similarity link between the two type attributes. These two links and the different scale link and the employee, salary, project and type DDOs are returned to the IA.

For the second database (see figure 2), the recreation WA has found the subscription class DDO and the type attribute DDO. The employee and salary DDO are not selected because from the reference link the person and earner classes are been potential relevant DDOs but they don't contain the type attribute. Only, the subscription and type DDOs are sending to the IA.

When the WAs send some temporary intra-base synonymy links to their IA, the IA could create some temporary inter-bases synonymy links if the same term exists in another WA of its acquaintance network.

In our example, the recreation WA has the subscription DDO and the business WA has the project DDO (intra-base synonymy). A temporary inter-bases synonymy link is created between these two DDOs of these two WAs.

The IAs ask to the WAs to execute the query or a part of it (selected DDOs of the previous step). For the achievement, each WA (at the local level) and each IA (at the global level) use intra-base or inter-bases different scale links to translate data in the expected format (if conversion functions are detected in the DDOs).

In our example, the business WA performs the following query: "SELECT * FROM employee WHERE employee. salary=1500 AND project . type="year"". The recreation WA has only the query concerning the subscription: SELECT * FROM subscription WHERE subscription. type="year"".

# 4 RESULTS

The User Agent organises the different answers of each IA. The results are provided and also the temporary links because they must be validated or not by the user. The inter-bases and intra-base synonymy links between project and subscription is invalidated. The intra-base similarity link is also invalidated. The different scale link is validated; the user must insert a conversion function to convert euros in dollar. Informations are sent to the concerned IA and after to the WA. The modified query is performed "SELECT * FROM employee WHERE employee. salary=1500 euros" to the business WA. For the other WA, the execution is finished because the system does not require a validation.

In the IA, a version of the synonymy link is created. It contains the invalidated state, the time and the identification of the user who has invalided the synonymy link.

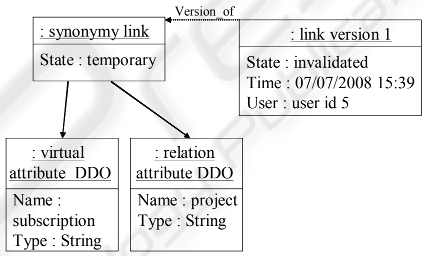


Figure 5: Synonymy link version.

In the WA, a similar version (see figure 5) is created from intra-base synonymy link. A version is also created from the similarity link between subscription type and project type. A version is also created from the different scale link. This version contains the validated state, time, user and the conversion function.

The system creates versions from the different links because the new information is inserted in the ontology only when it has been validated by several users. In this way, the ontology consistency is maintained.

For the synonymy link when it has been invalidated by five users, it is transformed in a permanent non synonymy link. In fact a non synonymy link is created. If the temporary synonymy link is validated five times, a new version is created with the permanent state. When a link version became permanent the WA domain ontology evolves.

The advantage to store version is in a first time the user validation or invalidation is not directly inserted in the ontology. The second is when the same query term is searched, the process to select a DDO is not performed. The synonymy link is directly proposed.

# 5 LOCAL DATABASE EVOLUTION

Different modifications can be performed on a database. They are made by the local expert. He manages the instance consistency because data cannot be acceded. Only the database schema is known by the system. The different modifications relatively to a database are applied to the ontology of the concerned WA and IA. The WA sends the modifications from its IA, this one sends them at all these WAs.

These modifications concern:

- Addition relation/class or attribute: adding a new relation or class DDO and/or new attributes DDO. The creation of similarity and different scale intra-base links is processed as for the insertion of a new database. The new DDOs are compared with the other DDOs in order to create new links.

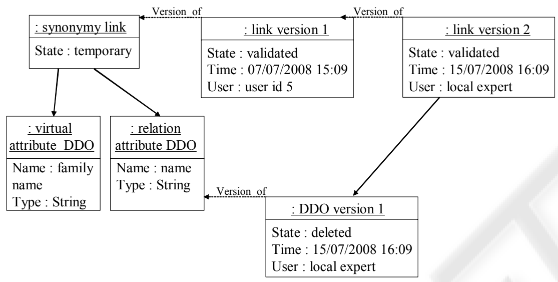


Figure 6: Versions from deleted DDO.

- Deletion relation/class or attribute: a version is created from the concerned DDO with the "deleted" state. A version is created from all the links referencing this DDO. The state is validated or invalidated, with mention of the time and by the local expert. The state is the same as the previous, the attribute deletion has no consequence on the link state. The inter-bases links are not concerned because they constitute global knowledge. To know if a DDO is a synonym or an homonym of an other DDO is very important for the IA because the query is directly translated, none research is necessary.

For example, a synonymy link exists between the name attribute and the family name virtual attribute (query term). When the local expert deletes the name attribute, a version is created from the DDO with the deleted state (see figure 6). This version creation brings about the version creation of the entire links which referenced this DDO. In our example, a creation version is made from the synonymy link. The version creation is performed for all the leaf versions of link. A version is only created from the link version 1, its state is validated because the previous version has the validated state.

- Modification:
  - Attribute addition or deletion from a relation or class (see the previous paragraphs),
  - Class or attribute name: a version is automatically created with the modified name state, the new name, the time and the expert. A synonymy link is also created between the DDO and the version. For example, the local expert modifies the name of the attribute salary. This modification brings about the version creation from this attribute and the creation of a synonymy link between the attribute DDO and the attribute version. The state is permanent because the modification has been performed by the database local expert.
- Modification of attribute unit: Version creation from the DDO with unit modification and version creation from the scale and different scale intra-base links. A DDO version is created from the attribute to modify the attribute unit. The system automatically creates successive versions from the different scale links. The expert must give the conversion function to be applied between the previous and new units.
- Modification of attribute type: a version is created with the "type modified" state, the new type, time and the database expert. A conversion function is asked to the expert if scale or different scale links exist for this attribute and therefore version creation is performed to store the conversion.

# 6 RELATED WORKS

Several projects use ontologies and MASs. In (Toinonen and Helin, 2003) an ontology describes interactions protocols. Agents protocols dynamically evolve with the ontologies. In our MAS system, interactions protocols do not evolve but the ontologies evolve.

The DASMAS project (Orgun et al., 2005) provides a dialogue framework to treat semantic interoperability in MASs. DASMAS runs with several MASs. At each MAS corresponds a domain ontology. The resolution conflict process uses a negotiation protocol and WordNet Lexicon detects semantical similar concepts in the heterogeneous ontologies.

Some systems capture ontology evolution: Inside the ontology or with ontology versioning. The systems are also different in the manner to maintain or not the changes between two successive versions. SHOE (Heflin et Hendler, 2000) cannot keep track of changes from a version to an other. SHOE maintains each version of the ontology as a separate web page. The ontology designer copies the original ontology file, assigns it a version number and adds or removes elements as needed. On the contrary, Protégé (Liang et al., 2005) keeps track and records ontology changes within the ontology itself for ongoing comparisons. OntoView (Klein et al., 2002) system helps the user to manage changes in ontologies and keeps ontology versions; it also allows the users to specify conceptual relations between different versions of concepts. The system of Auer and Here (Auer and Here, 2006) keeps track of different versions of an ontology and thus offers merging operations or sub-hierarchies design.

# 7 CONCLUSIONS

Semantic conflict resolution with agents consists of exploiting ontology links and DDOs. The set of protocols permits the domain ontologies to evolve by detecting new semantic links. The system dynamicity comes from the detection of new links during the query running and also from its ability to follow local evolution databases. A prototype is currently running on Jade platform (Bellifemine, 2002) and it permits to validate our proposition.

We apply the version concept in order to not directly insert the modifications within the ontology. The different versions constituting the historical evolution are archived. The ontology versions keep track of different modifications to better follow their evolutions and to propagate modifications to interconnected ontologies. The user's modifications become public after several validations from different other users. The local databases evolution is also managed by the version concept to store the modifications and to propagate them along the all existing semantic links.

# REFERENCES

Auer S. and Herre H., "A Versioning and Evolution Framework for RDF Knowledge Bases", In *Proceedings of Ershov Memorial Conference*, 2006.

Bellifemine, F.: Jade and beyonds. Presentation at AgentCities Information Day 1, Lausanne, (2002)

Boulanger, D., Dubois, G., 1998. An Object Approach for Information System Cooperation. *Information Systems*, Vol. 23, n°6 (1998) 383-399.

Gruber T.R., 1993. A translation approach to portable ontologies. *Knowledge acquisition,* vol. 5, n°2, , p. 199-220, 1993.

Guarino N., 1997. Understanding, Building, And Using Ontologies. *International Journal of Human-Computer Studies,* vol. 469, n°2-3, Feb./March, 1997.

Heflin J. and Hendler J., 2000. Dynamic Ontologies on the Web. In *Proceedings of the seventeenth national conference on Artificial Intelligence*, AAAI 2000, Austin, 2000.

Klein, M, Noy, N., 2003. A Component-Based Framework for Ontology Evolution. Technical Report, IR-504, CS Dept, Vrije Universiteit, Amsterdam, 2003.

Klein M., Kiryakov A., Ognyanov D. and Fensel D., 2002. Ontology versioning and change detection on the web. In *13th International Conference on Knowledge Engineering and Knowledge Management*, EKAW02, Spain, 2002.

Klusch, M., 2001. Intelligent Agent Technology for the Internet : A Survey. Journal on Data and Knowledge Engineering. Special Issue on Intelligent Information Integration, D. Fensel (Editor), Vol. 36, n°3, Elsevier Science (2001).

Liang Y. Alani H. and Shadbolt N.R., 2005. Ontology change management in Protégé. *Proceeding of Advanced Knowledge Technologies Doctoral Colloquim,* United Kingdom, 2005.

Mena E., Illarramendi A., 2001. Ontology-Based Query Processing for global Information Systems. Kluwer Academic Publishers, June, (2001).

Noy N.F. and Klein M., 2003. Ontology evolution: Not the same as Schema Evolution. Knowledge and *Information Systems*, 5, 2003.

Orgun B., Dras M., Nayak A.,2005. DASMAS – Dialogue based Automation of Semantic Interoperability in Multi Agent Systems. *Australian Ontology Workshop*, AOW'2005, Sydney, Australia, 2005.

Sheth A., 1999. Changing Focus on Interoperability in Information Systems: from System, Syntax, Structure to Semantics", In *Interoperating Geographic Information Systems*. M F. Goodchild, M.J. Egenhofer, R. Fegeas, C.A. Koffman (editors), Kluwer Academic Publishers, 1999.

Talens G., Boulanger D., Séguran M., 2007. Domain Ontologies Evolutions to Solve Semantic Conflicts. In *Ontologies-Based Databases and Information Systems* , volume 4623/2007, LNCS Springer Berlin /Heidelberg, ISBN 978-3-540-75474-9_4, September (2007), pp 51-67.

Toivonen S. and Helin H., 2003. Representing Interaction Protocols in DAML, In van Elst, L., Diagnum, V., and Abecker, A. (Eds.): Agent-Mediated Knowledge Management: Selected Papers from AAAI 2003 Spring Symposium, volume 2926 of Lecture Notes in Artificial Intelligence, Berlin, Germany, 2003.