Ontology Evolution in the Corporate Semantic Web

Fatma Chamekh, Guilaine Talens and Danielle Boulanger
Magellan Center- Information System Team, University of Jean Moulin, 6 cours Albert Thomas 69008, Lyon, France

Keywords: Semantic Web, Ontology Evolution, Multi-Agent System, Knowledge Management, Ontology Version.

Abstract: We present in this paper an approach for ontology evolution in the corporate semantic web. We particularly focus on the ontology and resources evolution which are two important components of the corporate semantic web. Ontology evolves with added and updated resources. The concerned ontological entities by the resource modifications are linked to the documents. To manage dynamically changes we use the MAS paradigm. The evolution process is distributed in the different agents of the system. Each of them has a particular role.

1 INTRODUCTION

Nowadays, to better capitalize and share their knowledge, the companies set up knowledge management systems. They enrich their in-house knowledge by capturing knowledge extracted from external information sources. The new generation of the web technologies as the semantic Web- authorizes to describe sources by increasing meaning of documents via metadata.

To represent knowledge of a specific domain, the ontology concept is a possible approach (Grüber, 1993; Studer et al., 1998).

The integration of semantic Web technologies in knowledge management systems provides new perspectives. The coupling between the company's communication tools (intranet, intraweb) as well as semantic web technologies leads to create company's memory as a Corporate semantic Web (Gandon, 2002; Luong et al., 2006). It consists of resources (documents), ontologies and, semantic metadata/annotations.

The environment of organisations is widely heterogeneous, distributed and evolutive. The main challenge consists of the capture of this changing environment to satisfy the dynamic experts'needs and requirements. Indeed, the application of changes to the one of components (document, and ontology) involves the evolution of the Corporate Semantic Web. In the context of intranet, the employees share the documents and reuse them to write others.

The knowledge extracted from documents allows the ontology evolution. This evolution is performed by a MAS (Multi-Agent System). Each agent manages a step of evolution process: Change identification, change analysis, change propagation and version management.

This paper is structured as follows: Section 2 lists related works about ontology evolution and multi-agent systems. Section 3 describes our approach and system architecture. In section 4, we present a case study. Finally, section 5 gives a conclusion and some perspectives.

2 RELATED WORKS

In this section, we describe related works concerning ontology evolution and multi-agent systems.

2.1 Ontology Evolution

The management of the ontology evolution is defined by (Stojanovic et al., 2003) as the "timely adaptation of ontology to the arisen changes and the consistent propagation of these changes to dependent artefacts".

(Klein and Noy, 2003) define ontology versioning as "the ability to manage ontology changes and their effects by creating and maintaining different variants of the ontologies".

(Stojanovic, 2004) proposed an approach for the management of evolution and the maintaining of consistency for KAON ontologies.

(Klein, 2004) suggested a process of ontology's version management.

Chamekh F., Talens G. and Boulanger D..
Ontology Evolution in the Corporate Semantic Web.
DOI: 10.5220/0004438401820189
In *Proceedings of the 15th International Conference on Enterprise Information Systems* (ICEIS-2013), pages 182-189
ISBN: 978-989-8565-60-0

Based upon the previous works, several approaches have been developed:

Onto-Evoal (Ontology Evolution-Evaluation) (Djedidi et al., 2010) is a system to manage ontology evolution and evaluation. It is made up of three levels: Evolution level process, pattern level and historical level.

CoSWEM (Corporate Semantic Web Evolution Management) (Luong et al., 2006) designed to manage Corporate Semantic Web Evolution especially on the two components ontology and semantic annotation. This approach focuses on the propagation of ontology changes to the semantic annotations.

Rogozan (Rogozan and Paquette, 2005) developed an approach for ontology evolution relatively to educational semantic web. The author presented a process of evolution in two phases: a phase of ontology evolution and a phase of version management.

Consistology (Jaziri et al., 2010) is a tool to manage ontology evolution and versioning. The authors proposed an ontology evolution approach based on change kits which control the inconsistencies generated by each type of change. The consistency of ontology is anticipated by suggesting all the possible resolutions and their effects on the ontology according to a set of rules defined by the system. A new version of ontology will be created. Each version of ontology is stored in a log.

Evolva (Zablith et al., 2008) is an ontology management framework, aiming to reduce user intervention. It explores background knowledge sources (Wikipedia, WordNet, online ontologies….). The system is composed of five components: information discovery, data validation, ontological changes, evolution validation and evolution management.

(Freddo and Tacla, 2009) proposed an integration approach of social web with semantic web. This approach is composed of two phases:

- Ontology learning from folksonomies: it consists of populating the tag ontology (SCOT), identifying the relation between each pair of tags (tag1, tag2) (tag1, tag3), and interacts with user in order to create concepts and instantiate concepts and relations.
- Ontology evolution from folksonomies: in this phase the authors propose to update the base ontology after each change of folksonomies. After identifying the changes, the system studies the relation between the extracted tag and the entities in the “base ontology” using an ontology alignment method.

Most of the existing systems for the ontology evolution uses patterns, plugins, and software modules. Thus, the task of automation system remains difficult.

The multi-agent system can be a tool to solve the problem related to the evolution of ontology, among others resolution of inconsistencies.

2.2 Multi-Agent System (MAS)

A software agent is considered as an entity with goals, is able to actions endowed with domain knowledge and situated in an environment (Stone and Veloso, 2000). MAS is suitable for the domains that involve interactions between different people or organizations with different (possibly conflicting) goals and proprietary information (Jennings, 1995). An agent evolves in an environment and is able to perceive what surrounds it, to communicate with the other agents. It has an autonomous behaviour in the aim of satisfying its objectives. Moreover, each agent has knowledge of its environment.

Dynamo (Sellami et al., 2009) is a MAS for dynamic ontology construction from domain specific text documents. Architecture is composed of two modules:

- DYNAMO Corpus analyzer is a twofold module of textual corpus processing: An extractor of terms fills up the database with candidate terms, and an extractor of lexical relations provides triplets constituted of two candidate terms and the syntactic relation between them.
- DYNAMO MAS is composed of term agent and concept agent. It uses as an entry the triplet provided by the first module. The built ontology is proposed to the user. He can accept, refuse or modify it. DYNAMO generates ontology in OWL file relatively RTO model (Resources Termino-Ontological).

(Deen et al., 2006) proposed an approach for dynamic ontology integration and mapping between global and local ontologies in a multi-agent environment. Each agent captures knowledge about its own schema (local knowledge) and also knowledge about the schemas of its acquaintances (partial global knowledge). The authors employed a specific thesaurus to resolve semantic conflicts. The agent can add new knowledge (concept, property or instance) but it is not really an ontology evolution.

In ACSIS (Boulanger et al., 2000), a MAS allows to dynamically solve semantic conflicts for the cooperation of information systems (Talens and Boulanger, 2009). The global knowledge base –

ontology– is distributed inside informational and wrapper agents involved in negotiation protocols to solve conflicts and insuring the completeness of the answer to a global multi-base query.

These meta-data encapsulated inside agents build several ontologies sharing a common description. Different agents interact to answer to a user query. In order to give the better results the agent ontologies dynamically evolve to deduce information between the database schemas and the query.

2.3 Discussion

From the analysis of the related works (see table1), two types of works exist:

- The user can add or modify directly the ontology. The evolution of ontology is based on the process described in (Stojanovic, 2004). Moreover, (Klein, 2004; Rogozan and Paquette, 2005) and (Luong et al., 2006) propose approaches based on the correction of inconsistencies after their production. In (Jaziri et al., 2010) an anticipatory approach is proposed to prevent and anticipate inconsistencies. (Rogozan and Paquette, 2005) and (Luong et al., 2006) describe Semantic web evolution approach that allows changing two components: ontology/semantic annotation or ontology/resources.
- (Zablith et al., 2008) and (Freddo and Tacla, 2009) focuse on identifying new information added to the ontology. The second type performs either through the analysis of the Folksonomy created by open and uncontrolled systems (social tagging system), or by using of the background knowledge sources.

Other approaches with MAS only propose design or integration of ontologies with evolution. However, Dynamo allows ontology modification but there is no process of ontology evolution.

In ACSIS (Talens and Boulanger, 2010), the ontologies inside agents dynamically evolve relatively the negotiation protocols. The ontology evolution is performed by versioning. The MAS role is to build the semantic interoperability between databases. In our knowledge, none system performs the ontology evolution through a MAS. We propose an approach of ontology evolution based on a MAS. Each agent realises a part of the evolution process designed by (Stojanovic, 2004). The ontology dynamically evolves by adding of new documents. Indeed concepts, properties and instances extracted from documents imply ontology evolution. The ontological entities may index the corresponding documents.

3 PROPOSED APPROACH

3.1 Context

The organisations share knowledge among their members, create and collect new knowledge. Sharing the activity in a group also implies to share the knowledge involved in the activity. Inside an enterprise, different services exist. Each of them produces documents; they can be consulted by the users. The goal of our proposition is to classify the documents thanks to ontologies. Furthermore, ontology servers help an organization to keep track of all concepts and notions used in its documents and to clarify the importance of transactions and business processes. The ontology is defined in (Grüber, 1993) as "an explicit specification of a conceptualization ". The adding and the updating of documents perform the updating of ontologies. Therefore, the documents evolve; the ontologies must follow the evolution. But, the modification of the ontology can involve a new classification of the documents.

Table 1: Comparative analysis.

	Stojanovic	***Klein and al***	***Onto-Evoal***	***Coswem***	***Rogozan***	***Consistology***	***Evolva***	***ACSIS***	***Freddo and al***
Ontology evolution process	*Yes*	*No*	*Yes*	*Yes*	*Yes*	*Yes*	Yes	*No*	*No*
System	*Semi-automatic*	*Semi-automatic*	*Semi-automatic*	*Automatic*	*Automatic*	*Semi-automatic*	*Automatic*	*Automatic*	*Automatic*
Ressources	*Internal*	*Internal*	*Internal*	*Internal*	*Internal*	*Internal*	*External*	*Internal*	*Floksonomies*
Ontology Consistency	*Yes*	*No*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*No*	*No*
User intervention	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*No*	*No*	*No*

At the beginning, experts of each domain must build the domain ontology to describe the vocabulary related to a specific domain. For example, marketing, human resources, financial ontologies are designed. Our framework allows the user to evolve the domain ontology when he drops off a new document on the intranet. After, when a user searches a document, key words are asked to find it. The system proposes all the documents containing these terms in all the ontologies. To realize these changes we must follow a multi-step process presented in the next section.

3.2 Evolution Process

Our approach follows an evolution process based on Stojanovic and Klein's process. It includes four phases:

- Identification of changes enables to represent the transformations at the request of the user (or expert) and to clarify them formally. We distinguish the changes that can make by the expert when he:

-Adds a new document includes add concept, modify concept, add property, modify property, add instance, modify instance.
-Updates a new document includes add concept, modify concept, add property, modify property, add instance, modify instance.
-Removes a document includes delete concept, delete property, delete instance.

An expert can directly add, update or delete a concept, property and instance if he considers necessary to handle ontology. He can consult the changes history stored in the log of changes.

- The analysis of the changes consists in studying the effect of the changes on the consistence of ontology and the consistency between ontology and documents. To resolve this problem, we defined consistence rules. Examples of consistence rules:

-Ontology should not have isolated concepts
-Ontology should not contain two occurrences for a same concept
-Each ontological entity should be connected to concept, property or instance.
-Each ontological entity should be linked to document

The rules must manage the effects of each modification in the ontology. We present the authorized changes in table 2.

To treat inconsistencies when they occur, we defined additional operations. To identify the adequate corrective operations related to each type of change, it is necessary to determine the types of changes and inconsistencies. If different possibilities exist, i.e., different additional operations can be applied with different effects, the users have to choose the appropriate additional changes to implement. The various operations and their impact to the consistence of ontology are displayed to the users in order to assist them.

- The propagation of the changes consists in checking the consistence of the dependent artefacts (ontology, document) after each change. When ontology is modified, the different mappings must be checked between the ontologies.
- The management of the versions (Vn and Vn+1) consists of validation of the changes enables to create the Vn+1 version and keeping record of the ontology library and annotating the whole of the changes. The annotations are recorded in log of changes. The classified documents can be differently referenced. It concerns changes in ontology, in document but also the classification of documents. Finally, the document is referenced in different concepts, attributes and instances. The URI is stored. The linked data are used to better manage and make easier information retrieval. The key concept of

Table 2: Effects of Changes

Changes	***Ontological entity position***	***Consequences***
Add Concept	Leaf concept	None
	Non leaf concept	Consequences on the sub concepts and instances
Delete Concept	Leaf concept	Consequences instances
	Non leaf concept	Consequences on the sub concepts, upper concepts, and instances
Rename Concept	Leaf concept	Consequence on the concepts and upper concepts Creation semantic link
	Non leaf concept	Consequence on the concepts and upper concepts Creation semantic link
Add property	Leaf concept	Consequence on the upper concepts
	Non leaf concept	Consequence on the upper concepts and sub concepts
Delete property	Leaf concept	None
	Non leaf concept	Consequence on the sub concepts and instances
Rename property	Leaf concept	Creation semantic link Consequence on the concepts and upper concepts
	Non leaf concept	Creation semantic link Consequence on the upper concepts and sub concepts

Linked Data (D'Aquin, 2010; Health and Bizier, 2011; Health et al., 2012) is based on the idea that the mechanisms used to share and interlink documents on the Web can be applied to share and interlink data and metadata about these documents, as well as the concepts they relate to.

3.3 Architecture of the System

The architecture of our system (see figure 1) is founded on a multi-agent system. These agents interact together to meet the change needs for the user while respecting the structural and semantic constraints. The user starts the evolution process by carrying out a request for change.
The agents interact to run different processing steps. In order to provide the consistence of ontology, our system guides the user by suggesting many choices of change operations. The system architecture (see Figure1) is made up of four agents. Firstly we assigned to each agent a process step.

- User agent detects the change to be realized by analyzing the user request. If the user adds/modifies a new document the user agent throws a term extractor module. This one backs the entities, the agent informs the ontology management agent of a type of changes and concerned entities. If the user deletes a document, the agent gives the document URI to the ontology management agent.
- Ontology management agent analyses a change. It searches similar ontological entity on the domain ontology. We use existing similarity measures like Jaccard (Jaccard, 1901), Levenstein (Levenshtein, 1966) and n-grammes (Damerau et al., 1971). It informs the inconsistency management agent about the changes that can be held.
- Inconsistency management agent checks the effects of changes on the ontology consistence. After each request for change, it receives the consistence type from the ontology management agent. So, it must propose additional changes in order to guide the user to work out the operation of change. It is a BDI agent (Beliefs, Desires, and Intentions). It is composed of three layers: The beliefs (knowledge) are the whole of the consistence rules as formal concepts. The desires are the whole of the changes rules. The intentions are the additional changes corresponding to the user's changes.
- Version management agent: it generates a new version of ontology after validation of the changes. The Vn+1 version are stored in an ontology library. However, the Vn management is stored in the log of ontology versions to memorize change traces.

The architecture of system contains: Log changes for storing all annotations of changes; Log of versions contains all versions of ontology. Ontology Library is contains all ontologies of domain. A learning module is implemented to extract news terms from the document. We use existing tools text2onto (Maedche and Staab, 2000), Gate (Cunningham, 2001), RapidMiner (Hunyadi, 2010)…).

4 CASE STUDY

We explain the ontology evolution process through an example (see Figure1).

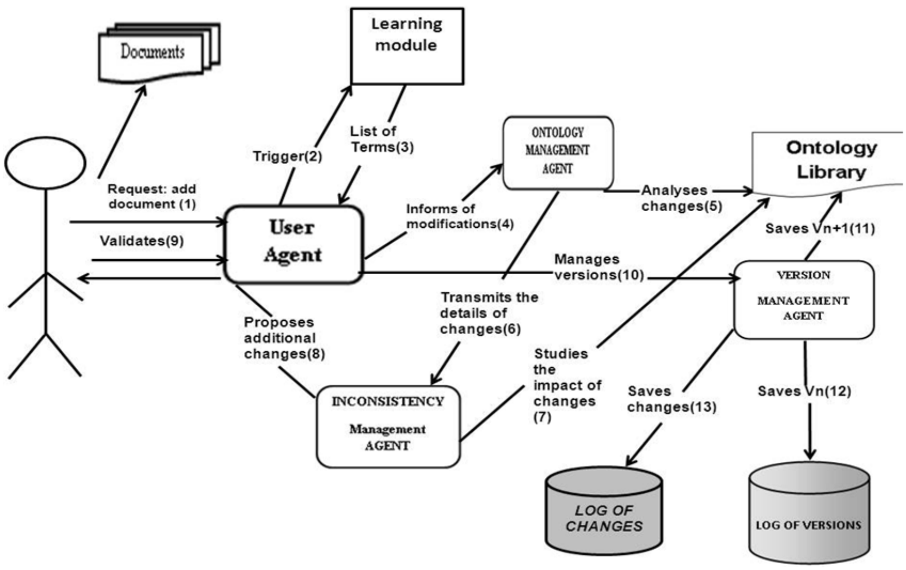

Figure 1: Architecture of the system.

4.1 Add New Document

Our system offers a guide to an expert in the process of the corporate semantic web evolution. We present it through an example. We suppose the expert want to add a new document, he sends a request through the research interface (1). The user agent treats the request and activates the learning module (2). This one extracts terms from the document. Different tools are currently in study (text2onto, GATE…). The user agent passes on the list of generated terms in the XML file (see Figure 2) (3). The extracted XML file contains two concepts “mobile commerce” and “electronic transfer”. The user agent informs the ontology management agent about the changes which are Add document and transmits the list of extracted terms (4). This one searches in the domain ontology (see Figure 3) similar concepts (5). In this step, different existing similar measures (Jaccard, Levenstein …) are used to find concepts, properties and instances. The similar concepts are “commerce”, “electronic commerce” and “E-commerce” to “mobile commerce” and “electronic commerce” and “Electronic data exchange” to “electronic transfer”. The ontology management agent suggests that:

- “mobile commerce” is a subclass of “commerce”
- “mobile commerce” is an upper class of “commerce”
- “mobile commerce” is a sub class of “electronic commerce”
- “mobile commerce” is an upper class of “electronic commerce”
- “mobile commerce” is a sub class of “E-commerce”
- “mobile commerce” is an upper class of “E-commerce”
- “electronic transfer” is sub class of “electronic commerce”
- “electronic transfer” is an upper class of “electronic commerce”
- “electronic transfer” is sub class of “electronic data exchange”
- “electronic transfer” is sub class of “electronic data exchange”

```
<?xml version='1.0'encoding='ISO-8859-1?>
<entity>
<concept1>mobile commerce</concept1>
<concept2>electronic transfer</concept2>
</entity>
```

Figure 2: Extracted XML file.

The ontology management agent transmits the details of changes to the inconsistency management agent (6). This one studies the impact of changes on the domain ontology (7). For this, it verifies the consistence rules:

- “mobile commerce” and “electronic transfer” are concepts; they should be connected to a concept, property or instance.
- “mobile commerce” and “electronic transfer” does not exist in the domain ontology. They are not redundant concepts.
- “electronic data exchange” is a leaf concept. If we add “electronic transfer” there is none consequence
- “commerce”, “electronic commerce” and “e-commerce” are non leaf concepts. The ontology management agent studies the impact of changes to the properties of “commerce” “electronic commerce”, “e-commerce” and “electronic data exchange”. The inconsistency management agent proposes the additional changes (8). When the expert chooses one of these: We suppose that he validates the “mobile commerce” subclass of “commerce” and “electronic transfer” is a sub class of “electronic commerce”. The user agent transmits this information to the version management agent. This later saves the Vn version in the log of version and Vn+1 version in the ontology library. It saves the modifications in the log of changes. The document URI is stored in each ontological entity.

4.2 Delete Document

We suppose that the expert wants to delete a document, he sends request through the research interface. The user agent treats the request. It informs the ontology management agent about the changes and transmits the URI of the document. The ontology management agent researches the instances, properties and concepts related to this document. This one transmits the details of changes.

The inconsistency management agent studies the impact of the changes and proposes additional changes. When the expert chooses one of these, the user agent communicates the information to the version management agent. This one saves the Vn version in the log of version and Vn+1 version in the ontology library. It saves the modifications in the log of changes.

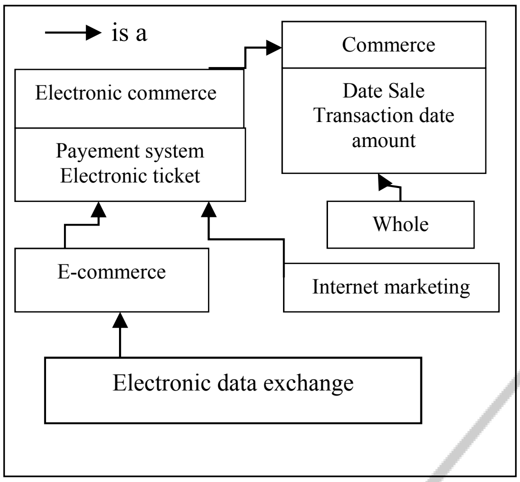

Figure 3: Commerce ontology.

5 CONCLUSIONS

Our research work tries to bring up two crucial points about corporate semantic web evolution and ontology evolution. The first point is how to change each component and the second point is how to apply this change to the ontology. We proposed an approach to manage the corporate semantic web evolution founded on multi-agent system. The agents interact between them to manage evolution process. The ontology evolves dynamically when a new document is added or modified by the expert. The evolution history is stored to keep track of changes. Currently we experiment and compare different tools to extract terms or concepts. Then our perspective is to link all the documents of a domain by the linked data technologies.

REFERENCES

Boulanger, D., Disson, E., Dubois, G., 2000. Object Oriented Metadata for Secured Cooperation of Legacy Information Systems. *In International Workshop on Model Engineering IWME'00, 14th European Conference on Object Oriented Programming*,Sophia-Antipolis and Cannes, France, p 75-82.

Cunningham, H., Maynard, D., and Bontcheva, K., 2001. Text processing with Gate.

Damerau, F., Hirano, M., Oka, K., and Tagawa, Y., 1971 Markov Models & Linguistic Theory, The Hague Mouton.

D'Aquin, M., Putting linked data to use in a large higher-education organisation, 2010. In Workshop on Interacting with Linked Data (ILD) at the Extended Semantic Web Conference (ESWC).

Djedidi, R., Aufaure, M. A., Onto-Evo an Ontology Evolution Approach Guided by Pattern Modelling and Quality Evaluation, 2010. In the 6th International Symposium on Foundations of Information and Knowledge Systems, LNCS: Vol. 5956, G. Berlin, Germany: Springer. Sofia, Bulgaria, p. 268-305.

Deen, S. M., Ponnamperuma, K., Dynamic ontology integration in a multi-agent environment, 2006. In 20th International Conference on Advanced Information Networking and Application AINA'06.

Freddo, A. R., Tacla, C. A., Integrating social web with semantic web-Ontology learning and ontology evolution from folksonomies, 2009. In International Conference on Knowledge Engineering and Ontology Development, Portugal, p. 247-253.

Gandon, F., 2002, Distributed Artificial Intelligence and Knowledge Management: Ontologies and multi-agent systems for a corporate semantic web, PhD Thesis INRIA.

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

Hass, P., Stojanovic, L., Consistent evolution of OWL ontologies, 2005. *In Proceedings of the 2nd European Semantic WebConference (ESWC)*, LNCS 3532, p. 182-197.

Hunyadi, D., Rapid miner e-commerce, 2010. In Proceedings of the 12th WSEAS international conference on Automatic control, modelling & simulation, p. 316-321.

Health, T., Bizier, C., Linked Data: Evolving the Web into a Global Data Space, 2011. Morgan & Claypool.

Health, T., Singer, R., Sahbir, N., Clark, C., and Leaves-ley, J., Assembling and applying an education graph based on learning resources in universities, 2012. *In Linked Learning (LILE) Workshop at the World Wide (WWW) conference.*

Jaziri, W., Sassi, N., Gargouri, F., Approach and tool to evolve ontology and maintain its coherence, 2010. *Metadata Semantic and Ontologies*, vol. 5, p. 151-166.

Jennings, N., Controlling cooperative problem solving in industrial multi-agent systems using joint intentions, 1995. *Artificial Intelligence*.

Jaccard, P., Étude comparative de la distribution florale dans une portion des Alpes et des Jura, 1901. *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37, p. 547-579.

Klein, M., Noy, M., A component based framework for ontology evolution, 2003. *In Workshop on Ontologies and distributed System, IJCAI'03*, Mexico.

Klein, M., 2004. Change Management for Distributed Ontologies. PhD thesis,University of Amsterdam .

Levenshtein, V., Binary codes capable of correcting deletion, insertions, and reversals, 1966. *Soviet Physics Doklady*, vol. 10, n° 8, p. 707-710.

Luong, P-H., Dieng-Kuntz, R. et Boucher, A. Managing semantic annotations evolution in the CoSWEM system.2006. In Proceedings of the Third National Symposium on Research, Development and Application of Information and Communication

Technology (ICT.rda'06), Hanoi (Vietnam), May 2006, p 215 –223.
Maedche, S., Staab, S., 2000. The text-to-onto ontology learning environment", *In Software Demonstration at ICCS-2000-Eight International Conference on Conceptual Structures.*
Noy, N. F., Musen, M., PROMPTDIFF: A fixed-point algorithm for comparing ontology versions, 2002. *In* Proceedings of the 18th National Conference on Artificial Intelligence and Fourteenth Conference on Innovative *Applications of Artificial Intelligence*, Canada.
Rogozan, D., Paquette, G., 2005. Managing ontology changes in the semantic web, *In the IEEE/WIC/ACM* International Conference on Web Intelligence*,* Compiegne, France*.*
Stojanovic, L., Maedche, M., Stojanovic, N., Studer, R., Ontology evolution as reconfiguration design problem solving, 2003. In Proceedings of the 2nd International Conference on Knowledge *CAPture K-CAP'03*, p. 162-171.
Stojanovic, L., 2004. Methods and Tools for Ontology Evolution, PhD thesis, University Karlsruhe.
Stojanovic, L., Maedche, A., Motik, B., Stojanovic , N., User-driven ontology evolution management,2002. In Proceedings of the European Conference on Knowledge Engineering and Management, p. 285-300.
Stone, P., Veloso, M., Multiagent Systems: Survey from a Machine Learning Perspective, 2000, *In Autonomous Robotics*, vol. 8, p345-383.
Studer, R., Benjamins, R., Fensel, D., Knowledge engineering: Principles and methods, 1998. *Data & Knowledge Engineering*, Volume 25, Issues 1-2, p. 161-19.
Sellami, Z., Aussenac-Gilles, N., Gleizes, M.P., Vers un outil de construction d'ontologies à partir de textes à l'aided'un système multi-agent, 2009. Actes de JFO.
Talens, G., Boulanger, D., Séguran, M., Domain Ontologies Evolutions to Solve Semantic Conflicts, 2007. *In Proccedings of Ontologies-Based Databases and Information Systems*, vol. 4623/2007, LNCS Springer, Berlin /Heidelberg, ISBN 978-3-540-75474-9_4, p. 51-67.
Talens, G., Boulanger, D., Domain evolution ontology by versioning, 2009. In Proceedings of the 5th International Conference on Agents and Artificial Intelligence – *ICAART'09*, Porto, Portugal, p. 185-190.
Talens, G., Boulanger, D., Evolutive ontology by versioning, 2010. *In Proceedings of the 4th International Conference Research Challenges in Information RCIS'10*, Nice, France, p. 157-168.
Zablith, F., Sabou, M., d'Aquin M., and Motta, E., Using background knowledge for Ontology evolution, 2008. *In International Workshop on Ontology Dynamics (IWOD) at the International Semantic Web Conference (ISWC)*, Karlsruhe, German.
Zablith, F., D'Aquin, M., Brown, S., and Green-Hughes, L., Consuming linked data within a large educational organization, 2011. In Second International Workshop on Consuming Linked Data (COLD) at International Semantic Web Conference (ISWC).