

---

# Evolution d'un web sémantique d'entreprise à l'aide d'ontologies dynamiques et de système multi-agents

**Fatma Chamekh**

*Université Jean Moulin  
Equipe MODEME  
6 Cours Albert Thomas BP 8242  
69355 Lyon  
Fatma.chamekh@univ-lyon3.fr*

---

*RÉSUMÉ. Les systèmes de gestion des connaissances s'appuient sur les ontologies afin de décrire les connaissances d'un domaine. Dans un contexte multi-acteurs les concepts changent au cours du temps et devient donc nécessaire d'étudier l'évolution des ontologies. Nous proposons une approche qui traite de l'évolution des ontologies dans le cadre d'un web sémantique d'entreprises en nous appuyant sur des agents logiciels intelligents.*

*ABSTRACT. Knowledge management systems rely upon ontologies in order to describe a domain's knowledge. Concepts evolve in a dynamic multi - actor context. Therefore, it is necessary to study the ontology evolution. In this paper, we present an approach for a dynamic evolution of ontologies in semantic web using the agent paradigm.*

*MOTS-CLÉS : web sémantique, évolution d'ontologie, système multi-agents, gestion des connaissances.*

*KEYWORDS: semantic web, ontology evolution, multi-agent system, Knowledge management.*

---

## 1. Introduction

Face aux changements économiques et technologiques, les organisations se trouvent confrontées à une masse importante d'informations. Ces informations sont stockées sans être traitées, ni réutilisées correctement au sein de l'organisation ; pour cette raison le repérage des connaissances (tacites et explicites) demeure nécessaire. L'intégration d'un système de gestion des connaissances permet à l'entreprise de capitaliser ses connaissances et les partager.

Le web sémantique met à disposition des utilisateurs une dimension sémantique de l'information afin de faciliter la recherche des ressources web. Il s'agit de décrire les ressources formellement par des métadonnées et les associer à une ontologie de domaine. Comme les ressources sur le web évoluent, les organisations vivent dans un environnement dynamique ce qui mène à l'évolution de leur système de gestion des connaissances du web sémantique d'entreprise. Les utilisateurs étant dispersés, les ontologies étant partagées, les agents peuvent apporter une solution pour aider à l'évolution de l'ontologie. Dans ce contexte, l'objectif de notre travail est de proposer un système de gestion de l'évolution de web sémantique d'entreprise basée sur une plateforme agent. Ce système offre une interaction avec l'utilisateur qui peut être lui-même l'ingénieur d'ontologie ou l'expert. Une façon de traiter l'évolution des ressources et des ontologies est le paradigme agent.

L'article est structuré de la façon suivante : la section 2 expose les objectifs de notre travail. La section 3 présente un état de l'art comportant trois aspects : web sémantique, évolution d'ontologie et systèmes multi-agents couplés à des ontologies. La section 4 décrit notre proposition.

## 2. Objectifs

Dans un environnement dynamique collaboratif, hétérogène et distribué, nous nous focalisons sur le problème d'évolution du web sémantique d'entreprise. En effet, appliquer des changements au web sémantique d'entreprise entraîne des changements à une ou plusieurs composantes (documents, métadonnées, ontologies). Cependant, la modification de l'une des composantes présente plusieurs difficultés aussi bien sur le plan pratique que théorique. Dans ce projet, nous nous intéressons aux processus d'évolution du web sémantique, ce qui entraîne l'évolution de l'ontologie.

Gérer l'évolution dans chacune des composantes tout en assurant une cohérence globale (entre les composantes) et une cohérence locale (dans chaque composante) est une tâche complexe. Quels types de changements ? Comment assurer la cohérence des composantes ? Comment analyser les effets des changements et les résoudre ? Lorsqu'il y a plusieurs changements lequel choisir ?

Pour répondre à ces questions, nous avons défini un système semi-automatique qui supporte l'évolution du web sémantique. La solution proposée utilise le paradigme agent. Les agents interagissent entre eux pour conduire le processus d'évolution.

Le travail de recherche passe par plusieurs phases : i) préciser la problématique de recherche par comparaison avec des travaux existants. ii) l'architecture et la modélisation de la solution proposée iii) la validation de la proposition par la réalisation d'un prototype.

### **3. Etat de l'art**

Nous présentons dans les sous-sections suivantes les trois thèmes : le web sémantique, l'évolution d'ontologie et les systèmes multi-agents qui sont le fondement de notre approche.

#### **3.1 Web sémantique**

Berners-Lee et al. (2001) expose une illustration du futur web qui se présente comme un vaste espace d'échange entre humains et machines afin d'exploiter des grands volumes d'informations. Le web sémantique permet d'explicitier le contenu sémantique des ressources dans le web (Luong, 2007). Dans le cadre de notre recherche nous nous intéressons aux trois composants principaux du web sémantique : les ressources, les ontologies et les métadonnées. L'ontologie est définie comme une « *spécification formelle et explicite d'une conceptualisation partagée* » (Stüder et al., 1998). Elle permet de représenter les connaissances d'un domaine formellement. Cette spécification est traduite par les concepts, les relations, les axiomes et les instances. Les annotations et les métadonnées permettent de décrire les ressources. En pratique, il y a une distinction entre les deux. Une métadonnée est une description normalisée, attachée à une ressource identifiée. Une annotation est une description située dans une ressource (Prié et Garlatti, 2004).

Par ailleurs, l'infrastructure web sémantique est distribuée et gérée par des acteurs différents. En effet, le changement du domaine d'application ou le changement des besoins d'utilisateurs engendre l'évolution des principaux composants du web sémantique (ressources, ontologies et métadonnées).

#### **3.2 Evolution d'ontologie**

*La gestion de l'évolution des ontologies est définie par (Stojanovic, 2004) comme étant : « l'adaptation dans le temps aux changements produits et la propagation consistante de ces changements dans les artefacts dépendants ». De même (Klein, 2004) a défini l'évolution d'ontologie par : « La capacité de gérer les changements apportés lors de l'évolution en créant et en maintenant différentes versions d'une ontologie. Cette capacité consiste à identifier et à différencier les versions, à modifier les versions, à spécifier des relations qui rendent explicites les changements effectués entre les versions ».*

Stojanovic (2004) propose un processus à six étapes qui supporte l'évolution d'ontologie : détecter les changements exprimés par les experts, représenter les changements, conserver la cohérence de l'ontologie au cours de son évolution

(sémantique des changements), implémenter les changements, étudier les effets des changements sur les objets dépendants (instance, ontologie, application), implémenter physiquement les changements additionnels. L'ensemble des changements est archivé dans un journal de changements.

Klein et al. (2002), proposent un processus de gestion de versions d'ontologie. Ce processus se base sur deux modules : un module de relation entre deux versions et un module d'identification des versions sur le web qui se base sur l'identification de l'URI (Uniform Resource Identifier) de la nouvelle version de l'ontologie suite au changement de l'entité terminologique.

Djedidi (2009) propose l'approche Onto-Evoal (Ontology-Evolution Evaluation) qui s'appuie sur une modélisation gérée par des patrons. L'architecture Onto-Evoal se compose de trois niveaux : niveau processus, niveau patrons et niveau journal d'évolution.

Une évolution du web sémantique d'entreprise (Luong, 2007) conduit à deux types d'interdépendances entre composants (ontologie et annotation sémantique) : i) une interdépendance ontologie-annotation sémantique qui exige que tout changement de l'ontologie conduise à un changement des annotations sémantiques, ii) une interdépendance ressource-annotation sémantique qui révèle que tout changement de ressource entraîne une perte de référence correspondante dans la base des annotations.

Rogozan (2008) présente un processus d'évolution en deux phases : une phase d'évolution d'ontologie et une phase de gestion des versions  $V_n$  et  $V_{n+1}$ . L'auteur propose deux modules pour présenter le cadre de référencement (phase gestion des versions) : le module 'Semantic Annotation Modifier' qui suit le référencement sémantique de façon dynamique. Il permet à l'utilisateur d'accéder facilement aux ressources. Le module 'Change History Builder' qui permet d'archiver les changements de l'ontologie.

Jaziri et al. (2010) décrivent une approche anticipative qui gère les incohérences générées par chaque type de changement. Cette approche s'appuie sur la vérification a priori des incohérences (Stojanovic, 2004) en proposant à l'utilisateur des changements additionnels. Elle suit un processus d'évolution en trois phases basé sur celui proposé par (Stojanovic, 2004).

Les termes « ontologie » et « système multi-agents » (SMA) représentent deux domaines différents. Néanmoins, dans la littérature ils sont souvent mentionnés ensemble. Nous remarquons à travers certains travaux de recherche qu'ils sont complémentaires.

### **3.3 Systèmes multi-agents**

Un agent est une entité intelligente faisant partie d'un système (Ferber, 1995). L'agent possède des propriétés de perception et de communication avec son

environnement. Un agent cognitif encapsule des connaissances concernant l'environnement et les autres agents.

DYNAMO est un outil basé sur un système multi-agents adaptatif pour la construction d'ontologie à partir de textes (Sellami et al., 2009). L'architecture est composée de deux modules : DYNAMO Corpus Analyser traite des corpus textuels et DYNAMO MAS est formé d'agents termes et agents concepts. Le système propose à l'utilisateur une ontologie (constitué par le système multi-agent lui même). Après validation du SMA, DYNAMO produit une ontologie sous forme d'un fichier OWL qui respecte le modèle RTO (Ressources Termino-Ontologique).

COMMA (Gandon, 2002) est un système de gestion d'une mémoire organisationnelle distribuée basée sur un système multi-agents et une infrastructure web sémantique. L'architecture COMMA est composée de l'ensemble des documents et annotations décrivant la mémoire d'entreprise et des agents (agent ontologie, annotation, utilisateur, interface, connexions). L'ensemble des messages échangés entre agents s'appuie sur un lexique consensuel exposé dans l'ontologie O'CoMMA.

Les approches successivement présentées dans l'état de l'art traitent de l'évolution du web sémantique et de l'évolution d'ontologie en utilisant plusieurs méthodes ( patrons, introductions de modules et de plugin...). A notre connaissance aucune n'utilise un système multi-agents pour gérer l'évolution du web sémantique qui à son tour entraîne l'évolution de l'ontologie. Par contre DYNAMO offre la possibilité de modification de l'ontologie ce qui pourrait être assimilé à une évolution d'ontologie mais il n'y a pas, à proprement parlé de processus d'évolution.

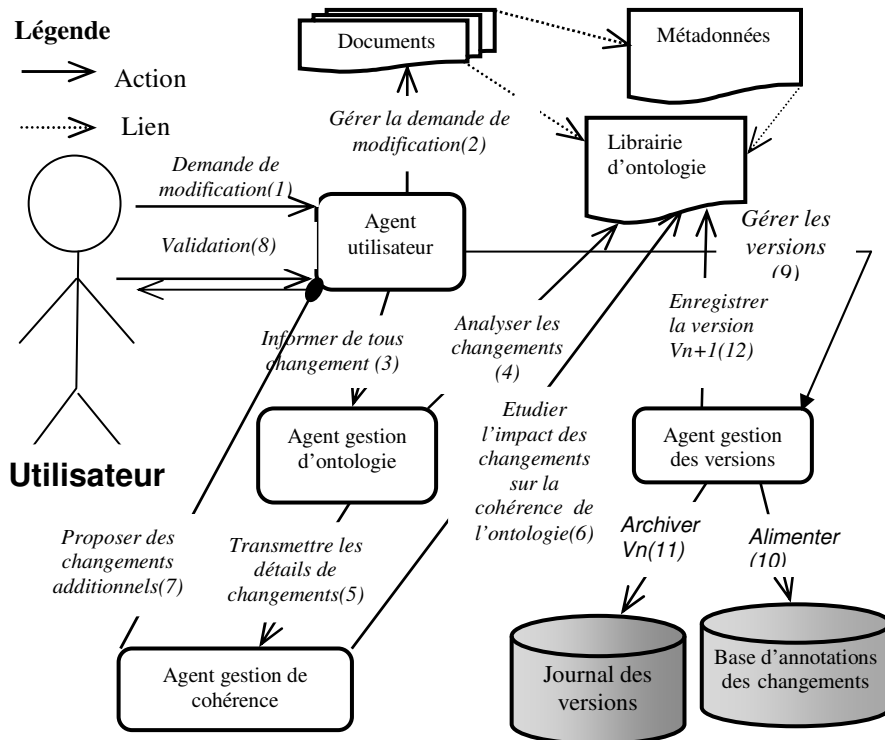
#### **4. Proposition d'une approche d'évolution du web sémantique**

Notre approche suit un processus d'évolution qui s'appuie sur celui de Stojanovic (2004) et Klein (2004). Il comprend cinq phases :

- L'identification des changements permet de représenter les changements à la demande de l'utilisateur (expert) et les expliciter formellement.
- L'analyse des changements consiste à étudier l'effet des changements sur la cohérence de l'ontologie.
- La propagation des changements consiste à vérifier la cohérence des artefacts dépendants après chaque changement.
- Validation des changements sert à créer la version  $V_{n+1}$  et à la sauvegarder dans la librairie d'ontologie.
- Gestion des versions ( $V_n$  et  $V_{n+1}$ ) consiste à comparer les deux versions et annoter l'ensemble des changements. Les annotations sont sauvegardées dans une base d'annotations des changements.

#### 4.1 Architecture et fonctionnement du système

L'architecture du système (cf. figure 1) que nous proposons est fondée sur un système multi-agents. Ces agents interagissent entre eux pour répondre aux besoins de changement de l'utilisateur (ontologies, documents, métadonnées) tout en respectant les contraintes structurelles et sémantiques. L'utilisateur déclenche le processus d'évolution en effectuant une demande de changement. Les agents interagissent entre eux pour exécuter les différentes étapes du processus. Le système guide l'utilisateur en lui proposant des choix d'opérations de changements afin d'assurer la cohérence de l'ontologie. L'architecture du système est composée de quatre agents :



**Figure 1.** Architecture du système

- **Agent utilisateur** : il permet de détecter le changement à réaliser. Ce changement peut être effectué sur un document, les métadonnées ou l'ontologie de domaine. Cet agent est réactif, il réagit en communiquant entre les agents. Il informe l'agent gestion d'ontologie de chaque modification demandée par l'utilisateur.

- **Agent gestion d'ontologie** : il spécifie formellement la demande de l'utilisateur tout en prenant en compte la classification prédéfinie et la taxonomie choisie. Cet agent est un agent cognitif qui se base sur des connaissances qui englobe les concepts formels de changements et une logique de raisonnement monde fermé (ce

qui n'est pas explicitement vrai est faux) qui permet d'inférer sur la base de connaissances.

- Agent de gestion des cohérences : il a pour but de vérifier les effets de changements sur la cohérence de l'ontologie. Il reçoit le type de cohérence et l'axiome concerné à partir de l'agent gestion d'ontologie. Il doit proposer des changements additionnels afin de guider l'utilisateur à élaborer l'opération de changement. C'est un agent BDI (Beliefs, Desires, Intentions). Il est composé de trois couches : les croyances (les connaissances) qui sont l'ensemble des règles de cohérence prédéfinie en tant que concepts formels. Il a comme objectif de vérifier la cohérence de l'ontologie après chaque demande de changement.

- Agent gestion des versions : il permet de générer une nouvelle version d'ontologie après validation des changements. La version  $V_{n+1}$  est enregistrée dans la librairie d'ontologie et la gestion  $V_n$  est enregistrée dans le journal de versions d'ontologie. Il compare les deux versions afin d'alimenter la base d'annotation des changements pour garder toute trace de changement.

Les documents sont liés à leurs métadonnées via un schéma RDF. Nous avons choisi le format Dublin Core (<http://dublincore.org>) pour exprimer les métadonnées. Celles-ci sont stockées sous forme d'un fichier XML. De même les ressources et leurs métadonnées sont liées à une ontologie de domaine archivée dans une librairie d'ontologies. Elle permet d'améliorer la qualité de la recherche d'information dans l'ensemble du document.

#### **4.2 Exemple**

Nous illustrons le fonctionnement du système à l'aide d'un exemple. L'utilisateur demande un ajout de document, le système lui demande d'insérer les descripteurs qui sont eux-mêmes l'ensemble des métadonnées. L'agent utilisateur informe l'agent gestion d'ontologie de l'ensemble des modifications à effectuer au niveau du document et des métadonnées ce qui entraîne une modification de l'ontologie du domaine : si l'utilisateur ajoute la rubrique « domaine de compétence » à la fiche employé. Ce changement entraîne l'ajout d'un nouveau sous descripteur à l'ensemble des métadonnées et l'insertion d'un nouveau attribut à la sous classe « employé » de la classe « personne ». L'agent gestion de l'ontologie traduit les changements en concepts formels en faisant appel à sa base de connaissances. Il précise l'ontologie concernée le type de changement et l'axiome en question. Il transmet l'ensemble des informations à l'agent gestion de cohérence qui examine les effets de changement sur la cohérence de l'ontologie et il propose lui-même des changements additionnels. Une fois que l'utilisateur a validé l'ensemble des changements, l'agent gestion des versions génère la version  $V_{n+1}$ . Il compare la nouvelle version et la version antérieure pour alimenter la base d'annotations des changements pour garder trace des différentes modifications effectuées. L'utilisateur est amené à consulter l'historique des changements avant chaque opération de changement dans un but de cohérence. Il stocke la version  $V_{n+1}$  dans la librairie d'ontologie, la version  $V_n$  dans le journal des versions.

## 5. Conclusion

À partir de l'analyse des différents systèmes étudiés nous avons constaté à ce jour qu'aucune approche d'évolution d'ontologie n'utilise les agents et aucun système multi-agents ne supporte l'évolution d'ontologie. Après la phase d'affinage de la problématique nous compléterons l'état de l'art. Nous approfondirons notre approche en examinant les différentes phases du processus d'évolution, et nous avons à décrire de façon détaillée l'architecture agent. Nous évaluerons notre approche en se basant sur le retour d'expérience des utilisateurs.

## 5. Bibliographie

- Berners-Lee T., Hendler J., Lasila O., « The semantic web, Scientific American », May 2001.
- Djedidi R., Approche d'évolution d'ontologie guidée par des patrons de gestion de changement, thèse de doctorat en informatique, Université Paris-sud, 2009.
- Ferber J., *Les systèmes multi-agents, vers une intelligence collective*, IIA, Paris: InterEditions, 1995.
- Ferber J., Ghallab M., « Problématique des univers multi-agents intelligents », *In Actes des 2èmes Journées du PRC-GRECO IA*, éd. D. Pastre, Toulouse, mars 1988, p. 295-320.
- Gandon F., Distributed Artificial Intelligence and Knowledge Management: Ontologies and multi-agent systems for a corporate semantic web, PhD Thesis INRIA, November 2002.
- Jaziri W., Sassi N., Gargouri F., « Approach and tool to evolve ontology and maintain its coherence », *Metadata, Semantic and Ontologies*, vol. 5, n°2, 2010, p. 151-166.
- Klein M., Kiryakov A., Ognyanov D., Fensel D., « Ontology versioning and change detection on the web », *In 13<sup>th</sup> International Conference on Knowledge Engineering and Knowledge Management, EKAW02*, Spain, 2002.
- Klein M., Change Management for Distributed Ontologies, PhD thesis, U Amsterdam, 2004.
- Luong P., Gestion de l'évolution d'un web sémantique d'entreprise, Thèse de l'école des Mines de Paris, 2007.
- Prié Y., Garlatti S., « Métadonnées et annotations dans le web sémantique », *Revue 13 Information-Interaction-Intelligence*, Vol 4, 2004, p. 45-68.
- Rogozan D., Gestion de l'évolution des ontologies : méthodes et outils pour un référencement sémantique évolutif fondé sur une analyse des changements entre versions d'ontologie, Thèse de doctorat en informatique, UQAM et INSA Rouen, 2008.
- Sellami Z., Aussenac-Gilles N., Gleizes M-P., « Vers un outil de construction d'ontologies à partir de textes à l'aide d'un système multi-agent », *Actes de JFO' 2009*.
- Stojanovic L., Methods and Tools for Ontology Evolution, PhD thesis, U Karlsruhe, 2004.
- Stüder R., Benjamins R., Fensel D., « Knowledge engineering: Principles and methods », *Data & Knowledge Engineering*, Vol 25, March 1998, p.161-190.