

Évolution de l'ontologie utilisée comme référentiel sémantique dans un système de téléapprentissage

Delia Rogozan^{1,2}, Gilbert Paquette¹, Ioan Rosca¹

¹ Centre de recherche LICEF, Télé-université
4750, avenue Henri-Julien, Montréal, H2T 3E4, Canada
<http://www.licef.telug.quebec.ca/eng/index.htm>

² Laboratoire PSI FRE 2645 CNRS, INSA Rouen,
BP 08 Place Emile Blondel, 76131 Mont Saint Aignan, France
<http://psiserver.insa-rouen.fr/psi/>

{drogozan, gpaquett, irosca @licef.telug.quebec.ca}

Auteur responsable de la correspondance : Delia Rogozan
Centre LICEF, 4750 avenue Henri-Julien, Montréal, H2T 3E4, Canada

drogozan@licef.telug.quebec.ca

Tel : 1 (514) 840-2747 (#2447), Fax : 1 (514) 843-2151

Résumé : L'utilisation d'une ontologie comme référentiel sémantique pour les éléments pédagogiques d'un système dynamique de téléapprentissage demande l'évolution de cette ontologie. Dans cet article nous mettons en évidence quelques questionnements concernant l'évolution de l'ontologie et nous proposons une méthodologie pour gérer cette évolution d'une manière consistante. Ensuite, nous montrons que l'évolution de l'ontologie peut causer des incompatibilités qui affectent l'accès aux éléments référencés par des entités ontologiques. Afin de résoudre ce problème, nous proposons une analyse de la compatibilité de l'ontologie évoluée et nous utilisons les résultats de cette analyse pour définir un procédé permettant la préservation du référencement des éléments pédagogique après l'évolution de l'ontologie.

Mots clés : Représentation et gestion de connaissances, Méthodologie, Évolution de l'ontologie, Analyse de la compatibilité, Référencement sémantique évolutif

Abstract: The use of ontology as referencing system for an e-learning environment requires the evolution of this ontology. In this paper, we first discuss some issues regarding the ontology evolution in an e-learning environment. Taking into account these issues, we then propose a generic methodology for managing the ontology evolution in an appropriate manner. Afterward, we show that ontology evolution may cause incompatibilities, which in turn may affect the pedagogical elements that use ontology entities as references. In order to solve this problem, we propose a conceptual investigation of what are the dimensions of the compatibility analysis between ontology versions together with their results. According to these results, we propose a method to classify changes applied to the ontology. Finally, we show how to use this classification of changes to keep all dependant pedagogical elements in a consistent state after the ontology evolution.

Keywords: Formalizing and managing knowledge in organizations, Methodology, Ontology Evolution, Compatibility Analysis, Semantic and evolving classification

1. Introduction

À l'heure actuelle, une initiative importante au niveau international consiste à associer au Web une couche des connaissances formalisées, des ontologies, permettant d'associer aux éléments d'information contenus dans une page Web des référents sémantiques [1] ce qui permettra des traitements d'information au niveau sémantique et non pas simplement syntaxique. Cette nouvelle génération du Web, le Web Sémantique, se présente comme une technologie prometteuse pour la mise en place des *systèmes de téléapprentissage et de gestion des connaissances (eLKMS¹)*. En utilisant des ontologies, le Web sémantique constitue un environnement qui facilite l'interopérabilité et la gestion des ressources provenant des systèmes hétérogènes et réparties ainsi que le partage et la communication des connaissances disponibles dans la communauté. Cet environnement permette alors l'élaboration des contenus et des stratégies d'apprentissage de meilleure qualité pour être proposés aux apprenants via le Web sémantique.

Nos travaux de recherche s'intègrent dans le deuxième thème du projet LORNET² qui vise à élaborer une application permettant la spécification des *eLKMS* concrets qui supportent les apprenants distants à atteindre un certain niveau de compétence. Pour faire cela, les apprenants doivent accomplir un nombre des activités pédagogiques et d'utiliser des ressources d'apprentissage (p.ex. documents, services). Les compétences des acteurs intervenant dans l'acte d'apprentissage, les activités pédagogiques ainsi que les ressources d'apprentissage constituent les *éléments pédagogiques* de base qui composent un *eLKMS* [2]. Tous ces éléments sont caractérisés sémantiquement, en étant référencés par des entités d'une ontologie.

L'utilisation d'une ou plusieurs ontologies comme système commun de référencement pour les ressources, les activités et les compétences des acteurs, permettra la création d'un *eLKMS* mettant en oeuvre des phénomènes éducationnels cohérents et agrégés au niveau sémantique [2]. Dans ce contexte, à mesure que le phénomène éducationnel se concrétise, les connaissances des acteurs et leurs compétences évoluent

¹ La sigle *eLKMS* signifie *e-Learning and Knowledge Management System*.

² LORNET est un projet du réseau de recherche pan-canadien sur les répertoires d'objets d'apprentissage <http://www.lornet.org/>.

ce qui peut demander la modification de l'ontologie si les acteurs considèrent qu'elle ne satisfait plus ses fonctions de référentiel sémantique. De plus, vue comme système commun de référencement, l'ontologie devrait permettre un feedback évolutif au sens que le référencement de nouvelles ressources et activités pourrait demander la modification de leur référentiel commun afin de prendre en compte de nouvelles caractéristiques.

D'un point de vue plus général, plusieurs chercheurs ouvrant dans le domaine du Web sémantique soulignent l'importance des méthodes et des outils supportant la gestion et l'évolution de l'ontologie [3].

Dans le deuxième chapitre nous présentons des questionnements concernant l'évolution de l'ontologie et nous proposons, dans le troisième chapitre, une méthodologie générique pour assurer cette évolution d'une manière consistante. L'évolution de l'ontologie peut causer des incompatibilités qui doivent être analysés afin de pouvoir rendre opérationnelle l'ontologie évoluée. Ainsi, dans le quatrième chapitre nous discutons au sujet de l'analyse de la compatibilité entre les versions d'une même ontologie, cette analyse étant une étape indispensable dans le contexte du Web sémantique. Enfin, en se basant sur les résultats de cette analyse, nous présentons dans le cinquième chapitre un procédé pour assurer la préservation du rôle de l'ontologie évolutive utilisée comme référentiel sémantique pour les éléments d'un eLKMS.

2. Évolution de l'ontologie dans un système de téléapprentissage

Selon [4], l'ontologie est la spécification explicite et formelle d'une conceptualisation partagée. En conséquence, l'ontologie devrait être spécifiée dans un langage ayant une sémantique formelle, comme le OWL³ pour les ontologies du Web sémantique. Une ontologie OWL est une collection d'informations incluant des descriptions de classes, des propriétés et des instances [5]. Une classe décrit d'une façon formelle un *concept*⁴ abstrait du domaine de l'ontologie. Une *instance* est un membre de l'extension d'un concept (classe), cette extension étant définie comme l'ensemble de tous les individus concrets appartenant à ce concept. Une *propriété* est une relation binaire qui énonce des faits

³Web Ontology Language (<http://www.w3.org/TR/2004/REC-owl-features-20040210/>)

⁴ Dans cet article nous utiliserons le terme concept plutôt que le terme classe.

généraux au sujet des concepts ou des faits spécifiques au sujet des instances. Enfin, une ontologie est complètement définie quand les concepts sont structurés par leurs propriétés et ces propriétés sont combinées dans des *axiomes* permettant d'effectuer des inférences sur le contenu de l'ontologie.

Nous définissons *l'évolution de l'ontologie* comme le processus de changement de la version antérieure de l'ontologie en une version nouvelle, tout en préservant les rôles de cette ontologie. Nous nommons *rôle de l'ontologie* le service assuré par l'ontologie, le but de son utilisation. Dans notre contexte de recherche, le but premier de l'utilisation de l'ontologie est celui d'assurer l'agrégation sémantique d'un système de téléapprentissage et de gestion de connaissances (eLKMS). Cette agrégation est réalisée par le référencement des acteurs, activités et ressources pédagogiques, à un référentiel sémantique commun. Ce référentiel est particulièrement une ontologie du domaine d'apprentissage⁵ visé.

Assurer l'agrégation sémantique d'un eLKMS, qui est notamment un système dynamique, multi-acteurs et distribué sur le Web, soulève des questionnements complexes du point de vue de l'évolution de l'ontologie utilisée comme référentiel sémantique commun :

- *Question 1 : Comment faciliter la négociation des acteurs distribués pour leur permettre la décision consensuelle des changements à apporter à l'ontologie ?*

Dans la formation à distance, tel que dispensée par la Téléuniversité [6], il faudrait parler d'un ensemble distribué d'administrateurs de l'ontologie qui sont tous des acteurs dans l'acte du téléapprentissage. Ces acteurs peuvent être les professeurs, les tuteurs ou même les apprenants [7].

- *Question 2 : Comment faciliter la spécification des changements à apporter à l'ontologie dans une manière complète mais conviviale ?*

Assurer l'évolution de l'ontologie dans un environnement dynamique où les acteurs ne sont pas tous des experts dans la construction de l'ontologie, demande la spécification des changements élémentaires, par exemple l'ajout ou

⁵ L'ontologie du domaine d'apprentissage réfère au modèle de connaissance explicitant les connaissances utilisées ou à acquérir par les apprenants.

l'effacement d'une entité ontologique, mais aussi des changements plus complexes, par exemples le déplacement, la fusion ou la séparation des entités ontologiques. Selon [8], un changement complexe et plus facilement utilisable et compréhensible qu'une suite des changements élémentaires ayant le même résultat.

- *Question 3 : Comment préserver le référencement des éléments pédagogiques (i.e. acteurs, activités, ressources) à une ontologie évolutive ?*

Le changement d'une ontologie utilisée comme système de référencement peut produire de pertes des coordonnées sémantiques, c'est-à-dire des références d'éléments établies par rapport à un certain référentiel sémantique. En conséquence, les éléments utilisant ces références ne sont plus accessibles via l'ontologie modifiée [9].

Prenons l'exemple d'un répertoire des ressources pédagogiques dans lequel chaque ressource est référencée par des concepts d'une ontologie du domaine d'apprentissage (p.ex. ontologie du domaine de la physique). Soit la ressource pédagogique « Cahier de Physique – 01 » qui utilise le concept 'loi de Newton' comme référence sémantique. Après l'application d'un changement fusionnant les concepts 'loi de Newton' et 'loi de Pascal' dans un seul concept 'lois de la mécanique', cette ressource ne plus accessible via l'ontologie ainsi modifiée, par exemple, pour une recherche de type « trouver les ressources décrivant les lois de la mécanique (concept dans l'ontologie évoluée)».

3. Processus basique d'évolution de l'ontologie

En tenant compte des questionnements présentés ci-dessus, nous pouvons affirmer qu'assurer l'évolution de l'ontologie constitue un vrai défi que nous tentons de résoudre partiellement par la définition d'une méthodologie. Dans cet article, nous décrivons seulement le modèle conceptuel de cette méthodologie et de outils informatiques qui la supportent.

3.1 Règles d'évolution de l'ontologie

En se basant sur les écrits de [8, 10, 11], nous avons identifié un ensemble des règles à respecter lors de l'évolution de l'ontologie (tableau 1).

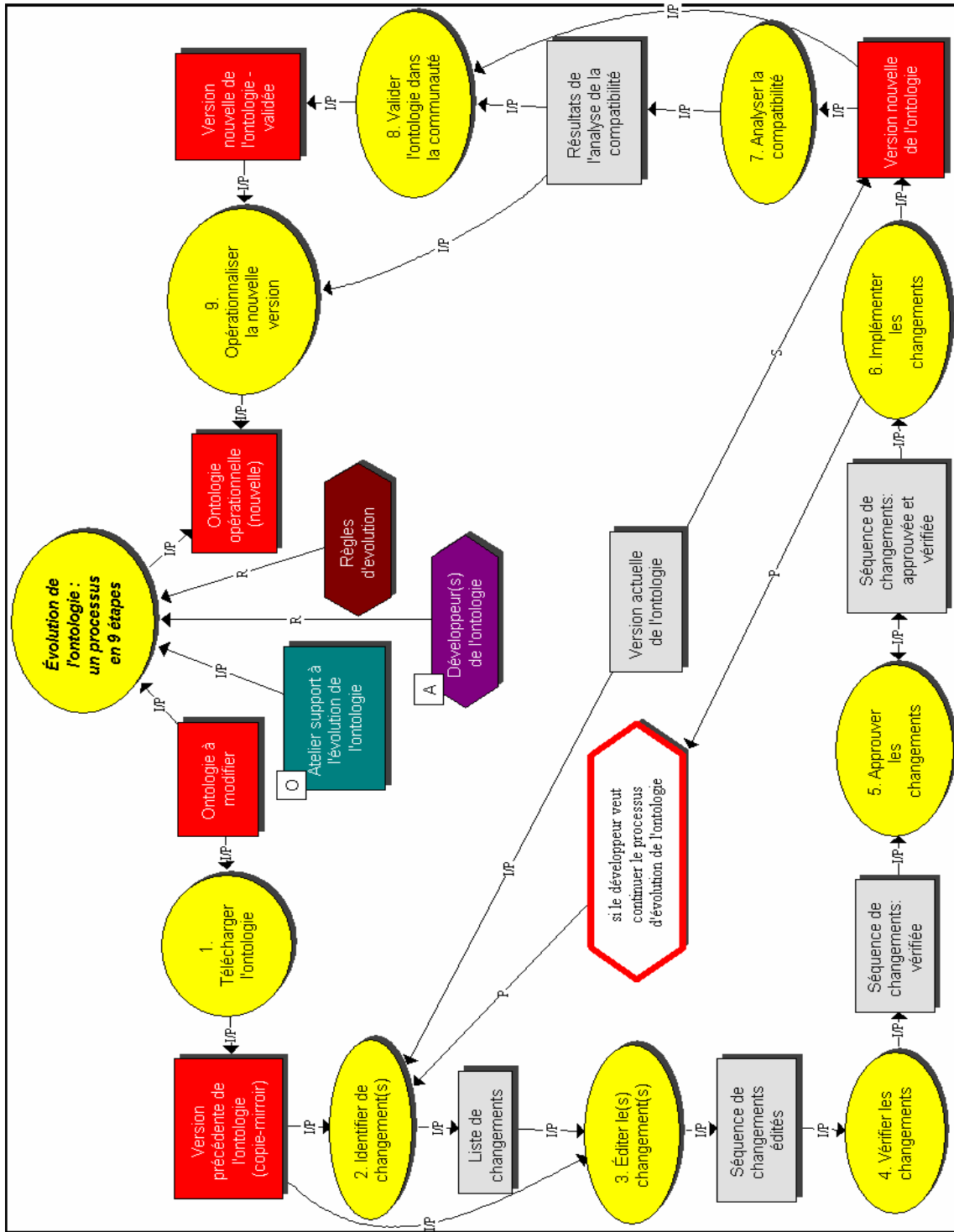
Tableau 1 : Règles d'évolution de l'ontologie

	<i>Règle d'évolution de l'ontologie</i>	<i>Description de la règle d'évolution de l'ontologie</i>
1	Évolution Consistante	L'ontologie doit évoluer d'un état consistant vers un autre état consistant. L'état consistant d'une ontologie est l'état dans lequel toutes les contraintes de l'ontologie sont satisfaites. Ces contraintes concernent la structure ⁶ et les axiomes de l'ontologie.
2	Préservation des rôles de l'ontologie	Les rôles de l'ontologie doivent être préservés après l'évolution de cette ontologie. Dans notre contexte cela signifie la préservation de l'agrégation sémantique d'un eLKMS, réalisée par le référencement de ces éléments constitutifs à une ontologie commune.
3	Acceptation des changements	Les développeurs de l'ontologie doivent approuver les changements apportés à l'ontologie ainsi que la version finale de cette ontologie.

3.2 Méthodologie d'évolution de l'ontologie

Pour concevoir la méthodologie d'évolution de l'ontologie nous nous sommes inspiré de plusieurs recherches [8, 10, 12, 9, 13, 14]. Comme montré dans la figure 1, nous avons développé cette méthodologie comme un processus régit par les règles d'évolution de l'ontologie et accompli par les développeurs de l'ontologie. Nous définissons *le développeur de l'ontologie* comme tout acteur impliqué, à un moment donné, dans le processus d'évolution de cette ontologie et nous précisons qu'il est possible d'avoir un ou plusieurs développeurs qui réalisent certaines tâches lors de ce processus.

⁶ Par exemple, dans une ontologie OWL toute propriété (de type *object property*) doit avoir minimum un concept comme domaine et minimum un concept comme co-domaine.



Légende : Le formalisme (MOT,[15]) s'interprète comme suit : l'ellipse représente une procédure, le rectangle un objet ou concept, le hexagone une règle, le hexagone ayant le signe A représente un acteur qui régit la procédure sous-jacente et le hexagone avec le signe O un outil qui supporte cette procédure. Le lien C est un lien de Composition, I/P de Intrant/Produit, P de Précédence.

Figure 1 : Modèle général de la méthodologie d'évolution de l'ontologie

Les développeurs accomplissent le processus d'évolution de l'ontologie à l'aide d'un atelier composé des plusieurs modules informatiques, chaque module ayant des fonctionnalités spécifiques (figure 2).

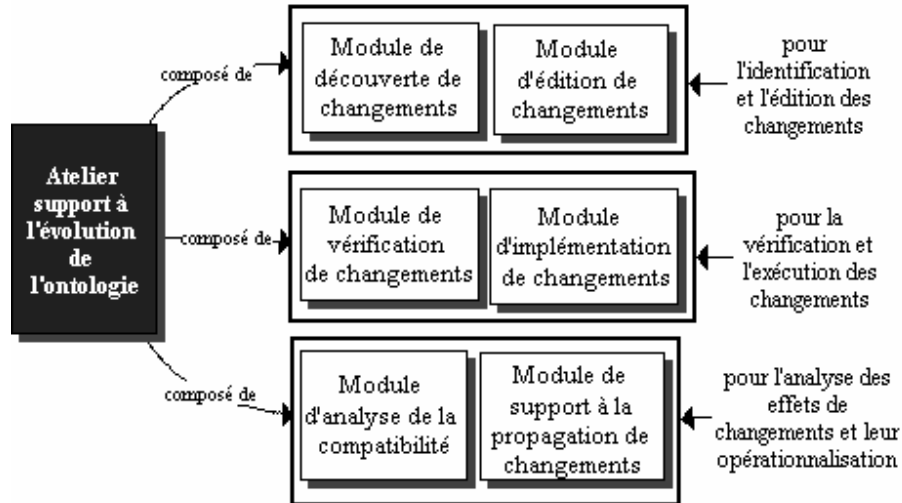


Figure 2 : Structure modulaire de l'atelier support à l'évolution de l'ontologie

Dans les paragraphes suivants nous présentons brièvement chaque étape du processus d'évolution de l'ontologie présenté dans la figure 1.

3.2.1 Télécharger l'ontologie à modifier

Les développeurs doivent télécharger une copie de l'ontologie à modifier. Nous la nommons *version précédente de l'ontologie* et nous la notons avec V_n .

3.2.2 Identifier les changements de l'ontologie

Dans cette étape les développeurs identifient les changements à apporter à l'ontologie selon deux méthodes :

- *identification descendante de changements explicites*. Il s'agit des changements imposés par la modification du domaine de définition ou celui d'utilisation de l'ontologie. De plus, l'ontologie, vue comme référentiel commun pour assurer l'agrégation sémantique d'un eLKMS composé des acteurs, des ressources et des activités pédagogiques, devrait permettre une rétroaction évolutive afin de prendre en compte de nouvelles caractéristiques exigées par l'évolution des connaissances

des acteurs ou le référencement des nouvelles ressources et activités pédagogiques.

- *identification ascendante des changements implicites*. Il s'agit des changements identifiés à partir de l'analyse de l'ontologie elle-même [16]. Cette identification est réalisée par le module de découverte de changements qui utilise un ensemble d'heuristiques⁷ définies par les développeurs.

Une fois les changements identifiés, qu'ils soient explicites ou implicites, les développeurs choisissent ceux qui leur semblent pertinents et produisent une liste de changements à apporter à l'ontologie.

3.2.3 Éditer les changements de l'ontologie

Le module d'édition fournit un ensemble des fonctionnalités pour permettre aux développeurs d'éditer les changements identifiés sous la forme d'une séquence des changements exécutables. Ces fonctionnalités devraient permettre l'édition des changements élémentaires et complexes. Un changement élémentaire est un changement primitif, qui ne peut pas être décomposé dans d'autres changements. Un changement complexe est composé des plusieurs changements élémentaires qui forment ensemble une seule entité logique. En s'inspirant de [17], nous avons développé deux taxonomies présentant des exemples de fonctionnalités pour permettre l'édition des changements élémentaires et complexes (figure 3 et figure 4).

Pour chaque changement dans l'ontologie, il est possible de générer différents scénarios de changements additionnels conduisant chacun à des états finaux différentes [14]. Par exemple, si un concept à l'intérieur d'une hiérarchie est supprimé, tous les sous-concepts peuvent être, soit supprimés, soit rattachés à d'autres concepts. Le résultat de l'étape d'édition consiste alors dans un rapport présentant la séquence des changements à apporter à l'ontologie, chaque changement ayant la forme⁸ d'un couple de type (changement édité, scénario des changements additionnels). De plus, chaque changement peut être annoté en utilisant un ensemble de métadonnées décrivant l'auteur et le but du

⁷ Un exemple d'heuristique : si un concept possède un seul sous-concept, alors ce concept peut être agrégé avec son sous-concept.

⁸ Au cas où aucun scénario n'est choisi pour un changement, le scénario considéré par défaut est exécuté.

changement, ainsi que le rôle sémantique de l'entité ontologique étant le sujet du changement.

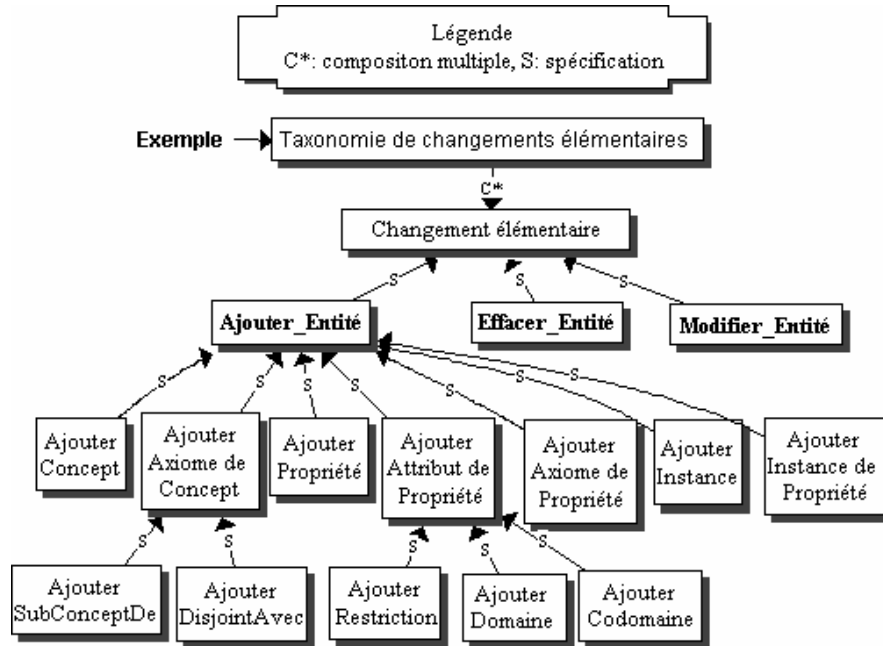


Figure 3. Fonctionnalités pour l'édition des changements élémentaires

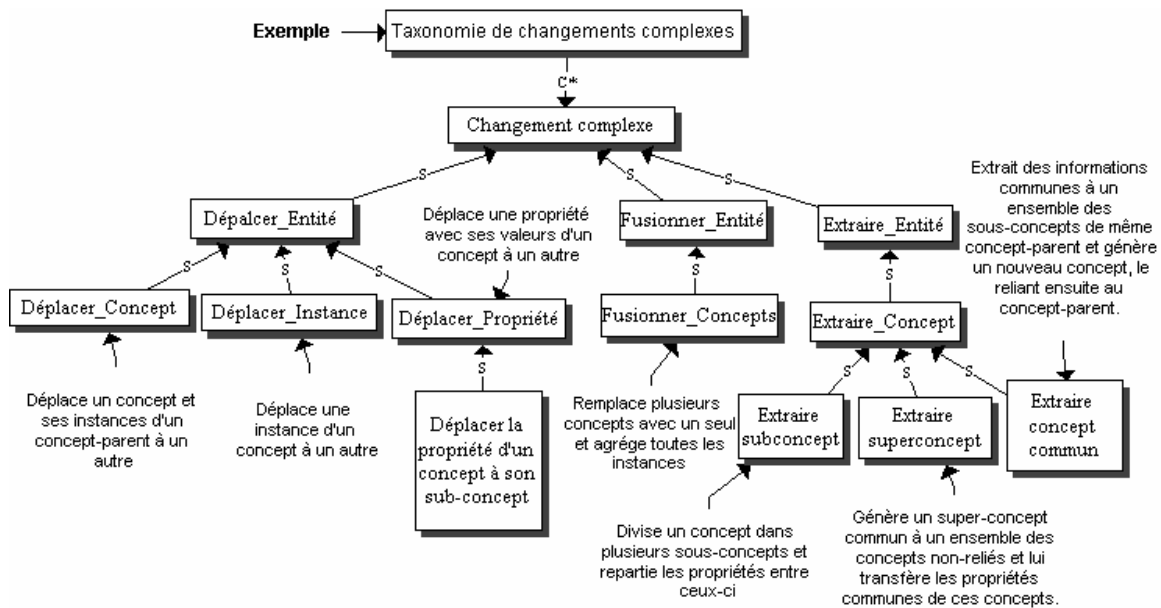


Figure 4. Fonctionnalités pour l'édition des changements complexes

3.2.4 Vérifier les changements de l'ontologie

Avant l'implémentation des changements il est nécessaire de vérifier leurs effets sur la consistance de l'ontologie (règle d'évolution 1). Cette vérification est réalisée principalement par le module de vérification qui assure :

- *La vérification de la consistance de changements.* Il s'agit d'identifier les changements qui invalident les contraintes structurelles de l'ontologie ainsi que ses axiomes. Au cas où un ou plusieurs changements invalident les contraintes de l'ontologie, les développeurs doivent retourner à l'étape 3 du processus afin de modifier la séquence des changements en conséquence. Concernant les changements qui invalident seulement les axiomes de l'ontologie une autre solution pour assurer la consistance de la nouvelle version de l'ontologie est celle de permettre au développeurs de modifier non pas les changements mais les axiomes invalidés afin de contourner les inconsistances.
- *La vérification de l'annotation des changements.* Il s'agit de vérifier le fait que tous les changements ont été annotés, dans l'étape d'édition, en utilisant un ensemble des métadonnées de type *Auteur du changement* _ *But du changement* _ *Rôle sémantique d'entité étant le sujet du changement*. Cette vérification peut prévenir les problèmes de compréhension de la signification d'une entité ontologique après l'évolution de l'ontologie⁹.

3.2.5 Approuver les changements de l'ontologie

Une fois les changements vérifiés, cette étape vise la validation consensuelle de la séquence des changements par les développeurs impliqués dans le processus d'évolution. Pour faciliter cette validation, il faudrait identifier les conséquences possibles de l'exécution des changements et les présenter aux développeurs [11].

3.2.6 Implémenter les changements de l'ontologie

Cette étape est réalisée par le module d'implémentation des changements qui :

⁹ Prenons comme exemple la situation où, dans V_n nous avons le concept 'Animal' avec son sous-concept 'Jaguar', et le concept 'Personne' relié au concept 'Jaguar' par la propriété 'Use'. L'effacement du concept 'Animal' fait en sorte que, dans la V_{n+1} nous restons qu'avec les concepts 'Personne' et 'Jaguar' reliés par la propriété 'Use'. Dans ce cas, il est difficile pour un lecteur externe, de conclure, sans aucun doute, que le concept 'Jaguar' est un animal et non une voiture (exemple adapté de[14]).

- Effectue une copie de la version V_n de l'ontologie et l'archive afin de permettre sa récupération;
- Exécute progressivement les changements spécifiés par les développeurs tout en leurs permettant d'annuler¹⁰ les changements en tout temps;
- Enregistre la séquence de changements et les métadonnées associées dans un journal qui préserve ainsi la trace du processus d'évolution ce qui facilite l'identification *à posteriori* des changements implémentés à V_n .

À la fin de cette étape, une *nouvelle version de l'ontologie* est obtenue et nous la notons par la suite avec V_{n+1} . Si un ou plusieurs développeurs veulent continuer à effectuer des changements, il peuvent retourner maintenant à l'étape 2 du processus, l'ontologie à modifier étant maintenant la nouvelle version obtenue à la fin de l'étape 6.

3.2.7 Analyser la comptabilité de la nouvelle version de l'ontologie

Pour faciliter l'analyse de la préservation des rôles de l'ontologie, le module d'analyse de la compatibilité identifie et classifie les changements implémentés à V_n pour obtenir V_{n+1} et fourni ensuite aux développeurs un rapport présentant les résultats de cette analyse.

3.2.8 Valider la nouvelle version de l'ontologie dans la communauté

Dans cette étape, les développeurs approuvent ou désapprouvent collectivement la nouvelle version de l'ontologie (règle d'évolution 3) avant de la rendre opérationnelle.

3.2.9 Opérationnaliser la nouvelle version de l'ontologie

Lors de cette étape, la version initiale de l'ontologie (V_n), se trouvant sur le Web, est remplacée avec la nouvelle version de l'ontologie (V_{n+1}). En fonction des résultats de l'analyse de compatibilité (étape 7), le problème le plus difficile est maintenant de préserver les rôles de l'ontologie pour cette nouvelle versions (règle d'évolution 2). Dans notre contexte, cela signifie la préservation de l'agrégation sémantique d'un eLKMS réalisée par le référencement des acteurs, activités et ressources pédagogiques à une même ontologie du domaine de l'apprentissage. Cette étape est réalisée par les développeurs à l'aide du module de propagation des changements.

¹⁰ Il s'agit d'annuler tous les effets de changements et de revenir à la version antérieure de l'ontologie ainsi qu'à l'étape 2 du processus.

4. Analyse de la compatibilité entre les versions d'une ontologie

Le processus d'évolution de l'ontologie peut causer des incompatibilités pouvant provoquer l'altération des rôles de l'ontologie [18]. Afin d'identifier ces incompatibilités, il faudrait analyser la séquence des changements exécutés pour passer de V_n (version initiale de l'ontologie) à V_{n+1} (version nouvelle de l'ontologie) et de les classifier en fonction de leur effet sur la compatibilité de la nouvelle version de l'ontologie.

4.1 Dimensions de l'analyse de la compatibilité

D'après [13], « defining what constitutes compatibility between different ontology versions becomes a salient issues since there are several dimensions to compatibility ». Dans cet article nous avons identifié trois dimensions à être considérées dans l'analyse de la comptabilité d'une ontologie : 1) la préservation des instances, 2) la préservation de la structure conceptuelle, et 3) la préservation de l'axiomatisation.

4.1.1 Préservation des instances de l'ontologie

La préservation des instances signifie qu'aucune instance n'est perdue dans le processus d'évolution de l'ontologie. Ce type de préservation est assuré si l'ensemble des extensions¹¹ de V_{n+1} est le même que celui de V_n ou un super ensemble de celui-ci. Nous présentons dans le tableau 2 les types de résultats possibles pour l'analyse de la préservation des instances.

Tableau 2. Types de résultats concernant la préservation des instances

Type de résultats de l'analyse	Situation d'occurrence pour le type de résultat	Exemples des changements provoquant le type de résultat
Préservation des instances sans ajout P_{IS} ¹²	L'ensemble d'extensions de V_n est le même que celui de V_{n+1}	- Ajouter_Concept - Ajouter_Propriété - Fusionner_Concepts - Diviser_Concept - Effacer, ajouter ou modifier l'annotation d'une entité ontologique

¹¹ L'extension d'un concept ou d'une propriété signifie l'ensemble des individus (*i.e.* les instances) qui sont les membres spécifiques du concept ou de la propriété.

¹² Nous notons avec P_{IS} , l'ensemble des changements qui assure la préservation des instances sans ajout. De même pour les autres types de préservation.

Préservation des instances avec ajout P_{IA}	L'ensemble d'extensions de V_n est élargi en V_{n+1}	<ul style="list-style-type: none"> - Ajouter_Instance - Ajouter_Instance de propriété
Non préservation des instances P_{IN}	L'ensemble d'extensions de V_n est réduit ou modifié en V_{n+1}	<ul style="list-style-type: none"> - Effacer_Concept (si le concept possède des instances) - Effacer_Instance - Effacer_Instance de propriété - Reclassifier une instance comme un concept - Ajouter <i>Disjoint_Avec</i> comme axiome pour un concept¹³

4.1.2 Préservation de la structure conceptuelle de l'ontologie

La structure conceptuelle de l'ontologie est utilisée comme un vocabulaire organisé pour permettre, dans notre contexte, le référencement et la gestion des éléments pédagogiques. Dans ce cas, une deuxième dimension de l'analyse de la compatibilité est la préservation de la structure conceptuelle de l'ontologie. Nous présentons, dans le tableau 3, les types de résultats possibles pour cette dimension de l'analyse.

Tableau 3. Types de résultats concernant la préservation de la structure conceptuelle

Type de résultats de l'analyse	Situation d'occurrence pour le type de résultat	Exemples des changements provoquant le type de résultat
Préservation de la structure conceptuelle sans ajout P_{CS}	L'ensemble de concepts de V_n est le même que celui de V_{n+1}	<ul style="list-style-type: none"> - Ajouter_Instance - Ajouter_Instance de propriété - Ajouter_Axiome - Effacer_Propriété - Effacer, ajouter ou modifier l'annotation d'une entité ontologique
Préservation de la structure conceptuelle avec ajout P_{CA}	L'ensemble de concepts de V_n est élargi en V_{n+1}	<ul style="list-style-type: none"> - Ajouter_Concept - Reclassifier une instance comme un concept
Non préservation de la structure conceptuelle P_{CN}	L'ensemble de concepts de V_n est réduit ou modifié en V_{n+1}	<ul style="list-style-type: none"> - Fusionner_Concepts - Diviser_Concept

¹³ Prenons l'exemple d'un changement qui déclare disjoint les concepts 'Professeur' et 'Tuteur'. Dans ce cas, les instances communes à ces deux concepts ne sont plus valides, elles sont perdues après l'application de ce changement.

Nous observons que, même si certains changements préservent les instances, ils ne préservent pas forcément la structure conceptuelle de l'ontologie (p. ex. Fusionner_Concepts) et vice-versa (Reclassifier une instance comme un concept). La même observation peut être appliquée à la préservation de l'axiomatisation de l'ontologie qui est la troisième dimension à prendre en compte dans l'analyse de la compatibilité.

4.1.3 Préservation de l'axiomatisation de l'ontologie

L'ontologie est aussi un système logique composé d'un ensemble d'axiomes qui contraint l'interprétation du contenu de l'ontologie, en spécifiant ainsi sa sémantique. Une troisième dimension à considérer dans l'analyse de la compatibilité est alors la préservation de l'axiomatisation de l'ontologie. L'axiomatisation de l'ontologie est préservée si tous les faits qui peuvent être inférés à partir de V_n peuvent encore être inférés à partir de la V_{n+1} . Nous présentons dans le tableau 4 les types de résultats possibles pour la préservation de l'axiomatisation de l'ontologie.

Tableau 4. Types de résultats concernant la préservation de l'axiomatisation

Type de résultats de l'analyse	Situation d'occurrence pour le type de résultat	Exemples dse changements provoquant le type de résultat
Préservation de l'axiomatisation sans ajout P_{AS}	L'ensemble des faits inférés à partir de V_{n+1} est le même que l'ensemble de faits inférés à partir de V_n	- Ajouter_Concept - Ajouter_Instance - Ajouter_Instance de propriété
Préservation de l'axiomatisation avec ajout P_{AA}	L'ensemble des faits inférés à partir de V_{n+1} est un super-ensemble de l'ensemble de faits inférés à partir de V_n	- Ajouter_Axiome - Ajouter_Propriété - Déplacer la propriété d'un concept à son super-concept
Non préservation de l'axiomatisation P_{AN}	L'ensemble des faits inférés à partir de V_{n+1} est réduit ou modifié par rapport à l'ensemble de faits inférés à partir de V_n	- Effacer_Propriété - Effacer_Axiome - Déplacer la propriété d'un concept à son sub-concept

Prenons un exemple pour la préservation de l'axiomatisation avec ajout. Supposons V_n où les deux concepts 'Professeur' et 'Tuteur' sont disjoints. Dans cette situation, un moteur d'inférence va conclure que toute instance du concept 'Professeur' est différente de toute instance du concept 'Tuteur'. Supposons maintenant V_{n+1} où nous avons déclaré

que le concept ‘Tuteur’ et équivalent¹⁴ avec le concept ‘Formateur’ et que le concept ‘Professeur’ est disjoint avec le concept ‘Formateur’. Dans ce cas, un moteur d’inférence utilisant V_{n+1} peut déduire, en plus de V_n , que toute instance du concept ‘Professeur’ est différente de toute instance du concept ‘Formateur’ est que l’ensemble des instances du concept ‘Formateur’ est exactement le même que celui du concept ‘Tuteur’.

4.2 Classification des changements de l’ontologie par rapport aux dimensions de l’analyse de la compatibilité

En prenant en compte les trois dimensions de l’analyse de la compatibilité nous pouvons classifier les changements exécutés pour passer de V_n à V_{n+1} .

- *changements compatibles* sont les changements qui préservent entièrement les rôles de l’ontologie. Généralement, l’ensemble des changements compatibles exécutés pour passer de V_n à V_{n+1} , est spécifié par l’intersection des changements assurant la préservation sans ajout selon les trois dimensions d’analyse (*i.e.* l’analyse de la préservation des instances, de la structure conceptuelle ainsi que de l’axiomatisation). L’ensemble des changements compatibles, que nous notons C_{ch} , peut être calculé de la manière suivante :

$$C_{ch} = (P_{IS} \cap P_{CS} \cap P_{AS})$$

- *changements rétrocompatibles*¹⁵. En prenant en compte les trois dimensions de l’analyse de la compatibilité, l’ensemble des changements rétrocompatibles, que nous notons BC_{ch} , peut être calculé de la manière suivante :

$$BC_{ch} = (P_{IA} \cup P_{CA} \cup P_{AA}) - (P_{IN} \cup P_{CN} \cup P_{AN}).$$

Les changements rétrocompatibles préservent les rôles de l’ontologie mais le nouveau contenu, introduit en V_{n+1} au moyen de ces changements, n’est pas utilisé pour élargir les rôles de l’ontologie, p.ex. pour référencer les éléments pédagogiques qui composent un eLKMS.

¹⁴ En OWL cette déclaration aura la forme *owl:equivalentClass* et affirme que toutes les instances d’un concept sont exactement les mêmes que les instance du concept équivalent.

¹⁵ Traduction de l’anglais « *backward compatible* ».

- *changements incompatibles* sont les changements qui ne préservent pas les rôles de l'ontologie. L'ensemble des changements incompatibles, que nous notons IC_{ch} , peut être calculé de la manière suivante :

$$IC_{ch} = (P_{IN} \cup P_{CN} \cup P_{AN})$$

Cependant, selon les rôles de l'ontologie, il est possible que les trois dimensions d'analyse ne soient pas toutes nécessaires pour analyser la compatibilité de la nouvelle version de l'ontologie. Par exemple, si nous considérons une banque de ressources pédagogiques référencées à l'aide de concepts d'une ontologie du domaine d'apprentissage, la préservation des instances n'est pas une dimension nécessaire de l'analyse de la compatibilité. Il faudrait alors calculer l'ensemble de changements selon les formules ci-dessus en considérant seulement la ou les dimensions nécessaires.

5. Propagation des changements pour assurer la préservation du référencement des éléments

Rendre utilisable la nouvelle version de l'ontologie ne signifie pas cependant qu'elle sera opérationnelle, c'est-à-dire que les rôles de l'ontologie seront préservés. Comme nous l'avons déjà montré auparavant, tous les changements incompatibles peuvent produire des inconsistances conduisant à la perte ou au dysfonctionnement de certains rôles de l'ontologie. Quant aux changements rétrocompatibles, les rôles de l'ontologie sont préservés mais le contenu de l'ontologie n'est pas tout utilisé. Il est alors possible que les développeurs désirent à considérer les connaissances nouvelles, ajoutées à l'ontologie, pour assurer l'élargissement de rôles de l'ontologie.

Dans notre contexte de recherche, l'ontologie est utilisée comme système commun de référencement pour les éléments pédagogiques¹⁶ qui composent un eLKMS. Pour préserver ce rôle de l'ontologie, nous proposons une approche qui vise à propager les changements, effectués pour passer de V_n à V_{n+1} , dans les éléments référencés par l'ontologie. La *propagation de changements* vise à apporter tous les éléments dépendantes à un état cohérent après l'évolution de l'ontologie [12].

¹⁶ Ces éléments sont les acteurs, les activités et les ressources pédagogiques.

La propagation de changements dans les éléments référencés par une ontologie signifie la modification des liens de références pour ces éléments afin de préserver leur référencement à l'ontologie évoluée, c'est-à-dire à la version V_{n+1} . Nous nommons *lien de référence*, le chemin¹⁷ qui relie un élément à une entité ontologique (p.ex. instance, concept) utilisée comme référence sémantique.

Prenons l'exemple d'un eLKMS dont les éléments sont référencés par des entités d'une ontologie se trouvant sur le Web. Dans ce cas, un lien de référence peut avoir la forme suivante "<http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine-2.0#Region>"¹⁸ où <http://www.w3.org/TR/2004/REC-owl-guide-20040210/> donne le chemin d'accès à l'ontologie, *wine-2.0#* donne le nom et la version (2.0) de l'ontologie et *Region* signifie le concept de l'ontologie qui est utilisé comme référence sémantique. En considérant les changements exécutés pour passer de V_n à V_{n+1} , nous pouvons rencontrer trois situations concernant la modification des liens de référence :

- Pour les changements incompatibles qui n'affectent pas les liens de référence des éléments car ces liens réfèrent à des entités ontologiques non touchés par ces changements, il faudrait modifier seulement le nom de la version. Par exemple, pour le lien "<http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine-2.0#Region>" il faut changer *wine-2.0* en *wine-3.0*.
- Pour les changements incompatibles qui affectent les liens de référence il faudrait modifier le nom de la version mais aussi le repère sémantique à l'intérieur de V_{n+1} . Prenons l'exemple d'un changement qui fusionne le concept '*Region*' avec le concept '*Département*' afin d'obtenir un nouveau concept '*Domaine*' dans la version *wine-3.0*, qui est la troisième version de l'ontologie *wine*. Dans ce cas, le lien de référence doit être modifié pour obtenir "<http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine-3.0#Domain>".
- Pour les changements compatibles et ceux rétrocompatibles, les liens de références des éléments ne sont pas affectés. Dans ce cas il faudrait changer

¹⁷ De l'anglais *path*.

¹⁸ Exemple d'URI adapté de OWL Guide 5. WebOnt, "Owl web ontology language guide and reference," 2004.

seulement le nom de la version dans le lien de référence. Toutefois, pour prendre en compte les nouvelles connaissances introduites dans la version V_{n+1} par les changements rétrocompatibles il faudrait définir, au besoin, de nouveaux liens de référence.

6. Conclusion

Dans cet article nous avons traité la problématique de l'évolution de l'ontologie dans un eLKMS distribué et dynamique. Afin d'assurer l'évolution de l'ontologie d'une manière consistante, nous avons proposé une méthodologie générique composée de neuf étapes principales qui répondent à un ensemble de règles d'évolution. Parmi ces règles, une de plus importante est celle demandant la préservation des rôles de l'ontologie après son évolution. Nous avons ainsi fait ressortir la nécessité d'avoir, comme partie intégrante du processus d'évolution de l'ontologie, une étape d'analyse de la compatibilité entre les versions d'une ontologie ainsi qu'une étape de propagation des changements afin d'assurer la préservation des rôles de l'ontologie utilisée comme référentiel commun pour les éléments pédagogiques qui composent un eLKMS. Ces étapes sont difficiles à réaliser dans le contexte du Web sémantique étant donné que « the most important flaw is the lack of detailed analysis of the effect of specific changes on the interpretation of data » [9]. En conséquence, un autre résultat fourni par cet article est une étude mettant en évidence les principales dimensions de l'analyse de la comptabilité ainsi que leur résultats possibles. En prenant en compte ces résultats, nous avons proposé une classification des changements exécutés pour passer de V_n à V_{n+1} ainsi qu'un procédé qui utilise cette classification afin de maintenir les liens de référence entre les éléments pédagogiques et les entités de l'ontologie utilisée comme références.

Références

1. T. Berners-Lee, J. Hendler and O. Lasilla, *The semantic web*, Scientific American **284** (5) (2001).
2. G. Paquette and I. Rosca, *Organic aggregation of knowledge objects in educational systems*, Canadian Journal of Learning and Technology **vol. 28** (2002), no. No. 3, 11-26.

3. J. Charlet, B. Bachimont and R. Troncy, "Ontologies pour le web sémantique," *Web sémantique*, J. Charlet, P. Laublet and C. Reynaud (Editors), Action_spécifique_32_CNRS/STIC, 2003.
4. T. Gruber, *A translation approach to portable ontology specifications*, Knowledge Acquisition **5** (1993), no. 2, 199-220.
5. WebOnt, "Owl web ontology language guide and reference," 2004.
6. G. Paquette, "Designing virtual learning centers," *Handbook on information technologies for education and training*, H. Adelsberger, B. Collis and J. Pawlowski (Editors), Springer-Verlag, Heidelberg, Germany, 2001, pp. 249-272.
7. D. Rogozan, G. Paquette and R. Hotte, *Dynamic approach for constructing a specific collaboration space among distant project groups*, E-Learn Conference, 2003, 2003, pp. 1752 -1755.
8. M. Klein and N. Noy, *A component-based framework for the ontology evolution*, Workshop on Ontologies and Distributed Systems, IJCAI-2003, 2003.
9. M. Klein and D. Fensel, *Ontology versioning for the semantic web*, International Semantic Web Working Symposium (SWWS), 2001.
10. D. Oberle, R. Volz, B. Motik and S. Staab, "An extensible ontology software environment," *Handbook on ontologies*, S. Staab and R. Studer (Editors), Springer Verlag, 2004, pp. 299-320.
11. L. Stojanovic and B. Motik, *Ontology evolution within ontology editors*, Knowledge Acquisition, Modeling and Management (EKAW), 2002.
12. L. Stojanovic, N. Stojanovic and S. Handschuh, *Evolution of the metadata in the ontology-based knowledge management systems*, Experience Management 2002.
13. N. Noy and M. Klein, *Ontology evolution: Not the same as schema evolution*, Knowledge and Information Systems **5** (2003).
14. L. Stojanovic, A. Maedche, B. Motik and N. Stojanovic, *User-driven ontology evolution management*, 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02), 2002.
15. G. Paquette, *Modélisation des connaissances et des compétences*, Presses de l'Université du Québec, Québec, 2002.
16. A. Maedche, B. Motik, L. Stojanovic, R. Studer and R. Volz, *Ontologies for enterprise knowledge management*, Intelligent Information Processing **March/April 2003** (2003).

17. J. Roddick, N. Craske and R. Richards, *A taxonomy for schema versioning based on the relational and entity relationship models*, 12th International Conference on the Entity-Relationship Approach: Entity-Relationship Approach, Springer, 1993.
18. Y. Ding, D. Fensel, M. Klein, B. Omelayenko and E. Schulten, "The role of ontologies in ecommerce," *Handbook on ontologies*, S. Staab and R. Studer (Editors), Springer Verlag, 2004, pp. 593-616.