



**HAL**  
open science

## Enrichissement d'ontologies : maintenance de la consistance et évaluation de la qualité.

Rim Djedidi, Marie-Aude Aufaure

### ► To cite this version:

Rim Djedidi, Marie-Aude Aufaure. Enrichissement d'ontologies : maintenance de la consistance et évaluation de la qualité.. 19es Journées Francophones d'Ingénierie des Connaissances (IC 2008), Jun 2008, Nancy, France. pp.249-260. hal-00416702

**HAL Id: hal-00416702**

**<https://hal.science/hal-00416702>**

Submitted on 14 Sep 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Enrichissement d'ontologies : maintenance de la consistance et évaluation de la qualité

Rim Djedidi<sup>1</sup>, Marie-Aude Aufaure<sup>1</sup>

<sup>1</sup>Département Informatique, Supélec Campus de Gif,  
Plateau du Moulon – 3, rue Joliot Curie – 91192 Gif sur Yvette Cedex  
{rim.djedidi, marie-aude.aufaure}@supelec.fr

**Résumé :** L'adaptabilité des ontologies à l'évolution des domaines qu'elles modélisent et à la dynamique des environnements dans lesquelles elles sont appliquées est un besoin fonctionnel essentiel à prendre en compte dans le cycle d'ingénierie ontologique. Plusieurs problématiques émanent de l'évolution d'ontologies : l'identification des besoins de changement, la spécification des changements, l'application des changements, la traçabilité des changements, la propagation des changements aux artefacts dépendants, etc. Dans cet article, nous nous intéressons particulièrement à la gestion des changements d'une ontologie et nous proposons une approche d'enrichissement d'ontologie dont l'objectif est d'automatiser et d'optimiser la validation des changements en assurant la consistance et la qualité de l'ontologie enrichie. La maintenance de la consistance est assurée à travers la proposition d'alternatives de résolution d'inconsistances. Un modèle de qualité est défini et appliqué pour évaluer l'impact des alternatives proposées sur la qualité de l'ontologie et guider la résolution des inconsistances tout en minimisant la dépendance à l'utilisateur.

**Mots-clés :** Evolution d'ontologies, Gestion de changements, Consistance OWL, Evaluation d'ontologies, Qualité d'ontologie.

## 1 Introduction

Les ontologies sont souvent utilisées dans des environnements dynamiques, multi-acteurs et distribués, elles ne peuvent être pensées comme une conceptualisation finie d'un domaine de connaissances délimité et stable. Les motifs de changements sont multiples, le domaine de définition peut évoluer ce qui implique des modifications dans l'ontologie pour rendre compte de cette évolution ; la réutilisation de l'ontologie par des applications différentes nécessite également son adaptation ; la conceptualisation même de l'ontologie peut être affinée à travers un processus de construction incrémental. Ainsi, pour répondre aux objectifs organisationnels et d'usage pour lesquels elles ont été définies, les ontologies doivent s'adapter aux besoins de changements.

Dans cet article<sup>1</sup>, nous présentons une approche d'enrichissement d'ontologies OWL selon laquelle, la validation des changements tient compte aussi bien de la consistance que de la qualité de l'ontologie évoluée. L'article est organisé comme suit : dans la section 2, nous détaillons l'approche d'enrichissement d'ontologies et ses différentes phases. La description du prototype implémentant l'approche est présentée en section 3. Avant de synthétiser les différents points de l'approche définie et présenter nos travaux futurs, une discussion et une comparaison avec les travaux existants sont présentées à la section 4.

## 2 Approche d'enrichissement d'ontologies

La gestion des changements d'une ontologie est une tâche cruciale. L'ingénieur d'ontologie doit contrôler tous les effets de changement, les résoudre et évaluer l'impact du changement sur l'ontologie. Il est indispensable de disposer d'une approche méthodologique d'enrichissement d'ontologies qui permet de spécifier formellement les besoins de changements et de guider et automatiser l'application des changements tout en assurant la consistance et la qualité de l'ontologie évoluée.

### 2.1 Spécification des changements

La spécification formelle des changements permet d'explicitier les changements à appliquer et d'indiquer les entités concernées ce qui permet de gérer et contrôler leur application. Elle dépend du langage de représentation de l'ontologie. Nous nous focalisons sur le langage OWL et plus précisément la couche OWL-DL qui représente aujourd'hui, le modèle fondamental du Web sémantique, particulièrement dans la conception d'ontologies. Il bénéficie des acquis des logiques de descriptions : une sémantique bien définie, des propriétés formelles compréhensibles en termes de complexité et de décidabilité, des algorithmes d'inférence connus et des raisonneurs DL existants qui permettent de vérifier la consistance des ontologies (Horrocks, et al., 2003).

Nous nous sommes référés au modèle OWL-DL pour définir des patrons de spécification formelle de changements (signature). L'idée est de catégoriser les changements, définir formellement leur signification, leur portée et leurs implications potentielles. Des travaux existants (Klein, 2004) ont déjà proposé une taxonomie des changements OWL en distinguant deux types d'opérations de changements : les opérations basiques et les opérations complexes. Les opérations basiques incluent des changements simples et atomiques appliqués à une seule entité de l'ontologie (granularité fine). Les opérations complexes correspondent à des mécanismes regroupant des opérations basiques pour former des unités logiques de changements riches et composites. Les changements basiques sont dérivés directement et de manière exhaustive du modèle OWL. Les compositions de changements complexes sont infinies.

---

<sup>1</sup> Ce travail est financé par l'Agence National de Recherche dans le cadre du Projet-RNTL DAFOE

Ainsi, nous avons considéré un premier noyau de changements basiques appliqués à des classes, des propriétés et des instances (voir table1) et nous avons défini un modèle de signature de changement *Ch-Sig* décrivant :

- Le type du changement selon la taxonomie des changements OWL ;
- Le type des entités concernées (classes, propriétés, instances) ;
- Les arguments c'est-à-dire les informations nécessaires à la gestion du changement (les entités référencées par le changement, les valeurs spécifiques au changement, etc.) ;
- Les contraintes à satisfaire (pré-conditions) pour que le changement puisse être appliqué (restrictions, cardinalité, domaine et co-domaine de propriétés, axiomes spécifiques comme la disjonction, etc.) ;
- Les inconsistances potentielles qui peuvent être probablement causées par ce type de changement. Elles sont inférées à partir des pré-conditions et peuvent être enrichie au fur et mesure à travers un processus d'apprentissage.

**Table 1.** Opérations de changements basiques.

Entité	Opérations
<b>Classes</b>	
Classe	ajout, suppression
Restriction	Ajouter, supprimer, modifier (contenu)
Type restriction	$\forall, \exists$
Cardinalité	Ajouter, supprimer, modifier
Relation « is-a »	Ajouter, supprimer, modifier
Disjonction	Ajouter, supprimer, modifier
<b>Propriétés</b>	
Propriété	Ajouter, supprimer
Domaine	Ajouter, supprimer, modifier
Co-domaine	Ajouter, supprimer, modifier
Relation « is-a »	Ajouter, supprimer, modifier
Inverse	Ajouter, supprimer, modifier
Symétrie	Définir, annuler
Transitivité	Définir, annuler
Type	Définir : datatype/object property
<b>Instances</b>	
Individual	Ajouter, supprimer

La spécification formelle du changement a un sens plus large qu'une simple formalisation du changement dans le langage de représentation de l'ontologie. Elle prépare les phases d'analyse et de résolution des effets de changement.

## 2.2 Processus de gestion des changements

Modifier l'interprétation d'un concept ou d'une relation de sorte que de nouvelles règles ou connaissances prennent effet, fait intervenir des connaissances inter-reliées et complexes et nécessitent la définition de mécanismes spécifiant comment les connaissances peuvent être changées et comment maintenir la consistance de ces connaissances. Les changements peuvent avoir plusieurs conséquences particulièrement sur la consistance de l'ontologie. Nous avons défini un processus de gestion de changements conduit à travers quatre phases : détection des inconsistances, proposition d'alternatives de résolution, évaluation de la qualité et application et validation finale des changements.

### 2.2.1 Phase de détection des inconsistances

Cette phase a pour objectif de vérifier la consistance et de délimiter la partie inconsistante de l'ontologie. Le changement formellement spécifié est appliqué à une version test de l'ontologie pour analyser ses effets. Nous commençons par localiser la sous-ontologie  $O'$  inconsistante minimale telle que  $O' \subseteq O$  et  $\forall O'' \subset O' \ O''$  est consistante. Deux niveaux de consistance OWL sont considérés : la consistance structurelle et la consistance logique.

La consistance structurelle se réfère aux contraintes du langage OWL-DL, c'est-à-dire l'utilisation de ses constructeurs. La consistance logique se réfère à la sémantique formelle de l'ontologie et à sa satisfiabilité dans le sens où elle est sémantiquement correcte et ne présente pas de contradictions logiques. OWL-DL est un langage centré axiome. Les concepts et les rôles possèdent des descriptions structurelles, élaborées à partir d'un certain nombre de constructeurs. Une sémantique est associée à chaque description par l'intermédiaire d'une interprétation du domaine (Horrocks et Patel-Schneider, 2004). La satisfaction de l'ontologie par une interprétation est contrainte par la satisfaction de tous les axiomes de l'ontologie (axiomes de concepts, de propriétés et d'individus).

Les contraintes logiques de OWL-DL étant nombreuses, nous nous focalisons pour le moment, sur un sous-ensemble de conditions liées aux axiomes de :

- Subsomption de concepts et de propriétés ;
- Disjonction de concepts ;
- Cardinalités : la cardinalité minimale par exemple, est vérifiée à travers la condition sémantique suivante :
 
$$\forall c \in C \ \forall p \in P \ \text{cardmin}(c,p) \Rightarrow \forall i \in I \ | \text{instprop}(p,i) | \geq \text{cardmin}(c,p)$$
- Domaine et co-domaine d'une propriété exprimés par des axiomes spécifiant les conditions nécessaires pour qu'une instance puisse être associée à une classe. Les assertions de domaine et co-domaine permettent d'appliquer des inférences sur les individus.

Les contraintes structurelle et une partie des contraintes logiques sont vérifiées par l'emploi du raisonneur Pellet qui supportent aussi bien le niveau terminologique (TBox) qu'assertionnel (ABox) de OWL-DL (Sirin et Parsia, 2004).

### 2.2.2 Phase de proposition d'alternatives de résolution

La résolution d'inconsistances permet de proposer à une ontologie donnée et un changement à appliquer, les alternatives potentielles de résolution des inconsistances causées par ce changement. Ces alternatives représentent des opérations de changements additionnels à appliquer pour maintenir l'ontologie dans un état consistant.

Tenant compte du principe de « continuité ontologique » qui stipule qu'une connaissance existante ne peut être infirmée (Xuan et al., 2006), nous tendons à minimiser les alternatives de suppression d'axiomes en proposant des alternatives de division de concepts, de fusion de concepts, de généralisation, de redistribution d'instances, etc. Cependant, si une suppression est nécessaire (elle peut d'ailleurs faire l'objet du changement lui-même et non seulement d'une alternative de résolution), elle est appliquée tout en vérifiant son impact sur l'ontologie. Certaines connaissances peuvent en effet, être réfutées à un moment donné, notamment lorsque l'ontologie est en cours de construction.

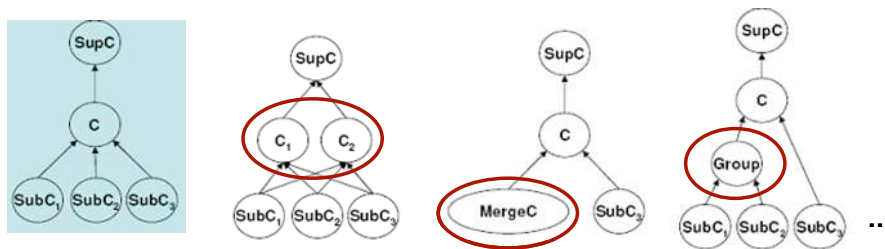


Fig. 1 – Exemple d'alternatives de résolution

Des patrons d'alternatives de résolution sont définis pour satisfaire un premier sous-ensemble de conditions de consistances logiques (voir section 2.2.3). Ces patrons sont ensuite adaptés aux spécificités du changement à appliquer, décrites par sa spécification formelle.

Un exemple d'alternatives de résolution peut être proposé pour illustrer la résolution d'une inconsistance liée à un axiome de disjonction : Si I une instance du concept C1, sous-concept du concept C, est assignée au concept C2, sous-concept de C et disjoint de C1, au lieu de supprimer l'axiome de disjonction ou l'axiome d'instanciation de I à C1, l'inconsistance de disjonction peut être résolue par les alternatives suivantes :

- L'instance I est redistribuée et assignée au concept C plutôt qu'à ces sous-concepts, ainsi la sémantique de l'instance I est gardée même si elle est moins précise et l'axiome de disjonction reste vrai ;
- Un nouveau sous-concept C3 est ajouté au concept C, l'instance I lui sera assignée, il représentera les sous-concepts de C auxquels l'axiome de disjonction n'est pas appliqué sans que ce dernier ne soit supprimé ;

- La même alternative que la précédente peut être proposée mais en assignant l'instance I uniquement au concept C3 nouvellement défini afin d'éviter une éventuelle redondance.

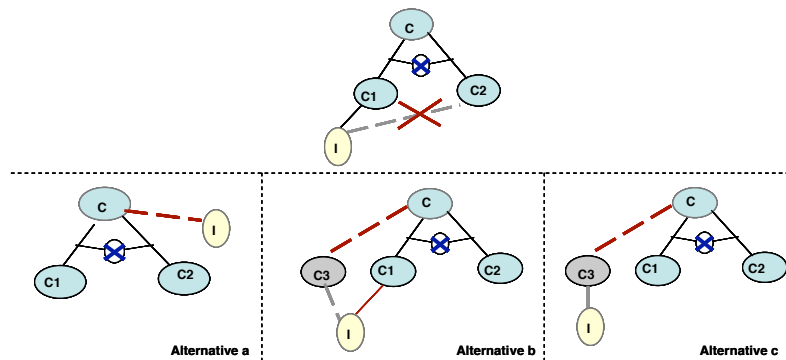


Fig. 2 – Les alternatives de résolution d'une inconsistance liée à la disjonction de concepts

Rappelons que les alternatives de résolution ont pour objectif de maintenir la consistance de l'ontologie enrichie. Ainsi, toute proposition d'alternatives implique la vérification et la résolution récursives des effets de ces alternatives sur la consistance globale de l'ontologie afin d'éviter en résolvant une inconsistance d'en causer une autre. Pour des raisons de synthèse, nous ne présenterons pas dans cet article, l'algorithme récursif de la résolution d'inconsistances.

### 2.2.3 Phase d'évaluation de la qualité

Plutôt que de présenter les différentes alternatives de résolution à l'ingénieur d'ontologie, nous proposons de guider le choix de l'alternative à appliquer en évaluant l'impact de chacune des alternatives sur la qualité de l'ontologie sur la base d'un modèle hiérarchique d'évaluation que nous avons défini (Djedidi et al., 2007). Le modèle de qualité tient compte de la structure et de l'usage de l'ontologie. Les critères de qualité considérés sont :

- Complexité : ce critère reflète la difficulté à comprendre la structure de l'ontologie ;
- Cohésion : elle tient compte des liens et chevauchements entre concepts (relations) ;
- Taxonomie : ce critère renseigne sur le rapport relations hiérarchiques / relations sémantiques. Il est inversement proportionnel à la richesse sémantique de l'ontologie ;
- Abstraction : ce critère mesure le niveau de généralisation/spécialisation des hiérarchies ;
- Modularité : la possibilité de découper l'ontologie en sous-hiérarchies indépendantes (modules). La modularité facilite l'enrichissement et la

maintenance de la structure et aussi la réutilisation de l'ontologie. Elle permet d'évaluer les deux aspects structure et usage de l'ontologie ;

- Complétude : Elle évalue le degré de couverture des sources représentatives du domaine et des propriétés pertinentes du domaine (conformité des désignations des concepts aux mots clés du domaine) ;
- Compréhension : ce critère tient compte de la facilité de compréhension de l'ontologie à travers l'annotation des primitives et la définition des concepts.

Différentes métriques d'évaluation de qualité ont été recensées dans la littérature. Nous nous sommes intéressés aux métriques quantifiables que nous avons redistribuées, sur les différents critères cités précédemment, et complétées pour la construction du modèle de qualité décrit par le tableau ci-après.

Certains critères peuvent être contradictoires (les changements qui augmentent la complexité de l'ontologie par exemple, améliorent la cohésion de sa structure), c'est pourquoi au début du processus d'enrichissement, l'ingénieur d'ontologie procède à la pondération des critères du modèle en attribuant un poids à chacun des critères en fonction de son importance par rapport au domaine modélisé et l'application utilisant l'ontologie.

**Table 2.** Description des critères et métriques évaluant la qualité de la structure et de l'usage de l'ontologie.

Critères	Métriques
<b>Structure</b>	
Complexité (P1)	Nombre moyen de Chemins de la racine vers un Concept. (NCP)
	Nombre moyen de descendants par Concept. (NPC)
	Nombre moyen de descendants par Propriété. (NPP)
	Nombre moyen de Relations par Concept. (NRC)
Cohésion (P2)	Nombre moyen de Relations par Concept. (NRC)
	Nombre de Concepts Racines (nombre de hiérarchies). (NRtC)
Taxonomie (P3)	Ratio des relations Is-a par rapport aux relations sémantiques. (H-ISA)
Abstraction (P4)	Profondeur Moyenne d'une hiérarchie. (DA)
Modularité (P5)	Nombre de Modules disjoints formant l'ontologie. (NM)



<b>Usage</b>		
Modularité	(P6)	Nombre de Modules disjoints formant l'ontologie. (NM)
Complétude	(P7)	Rappel (pourcentage des termes apparaissant dans le corpus représentatif du domaine, utilisés également comme noms de concepts dans l'ontologie). (REC)
		Précision (pourcentage des termes nommant les concepts de l'ontologie apparaissant également dans le corpus représentatif du domaine). (PREC)
Compréhension	(P8)	Pourcentage de Concepts Annotés. (AC)
		Pourcentage de Relations Annotés. (AR)
		Nombre moyen de Termes désignant un Concept. (NTC)

En reprenant l'exemple de la figure 2, l'alternative *a* n'altère pas la structure de l'ontologie, les alternatives *b* et *c* augmentent la complexité de celle-ci. Relativement à l'aspect usage, dans les alternatives *a* et *c*, l'instance *I* n'est plus rattachée à la classe *C1*, si le corpus représentatif de l'ontologie montre une importante assignation de cette instance au concept *C1*, le critère de complétude sera altéré. Le choix tiendra compte de ces évaluations et aussi des pondérations attribuées par l'expert aux critères concernés.

#### 2.2.4 Phase d'application des changements

Cette phase correspond à la validation finale du changement nécessaire et de ses changements dérivés tout en assurant la traçabilité des modifications appliquées à l'ontologie. Elle est optimisée par l'emploi de techniques d'évaluation de qualité permettant de guider la résolution des inconsistances et de minimiser la dépendance à l'utilisateur. Les évaluations de qualité sont comparées aux anciennes valeurs de qualité de l'ontologie initiale (sauvegardées dans le journal d'évolution). L'alternative qui préserve la qualité de l'ontologie, peut être automatiquement choisie et les changements directement validés et appliqués. L'expert n'interviendra que si toutes les alternatives ont un impact négatif sur la qualité. Il sera guidé dans sa décision, par l'indication de l'alternative la moins coûteuse pour la qualité.

L'analyse des changements et les résultats de maintenance de la consistance de l'ontologie sont appliqués à une version temporaire de l'ontologie qui peut être abandonnée si les changements sont finalement annulés, l'ontologie initiale sera alors préservée. Dans le cas contraire, une nouvelle version de l'ontologie est définie et l'historique des changements est sauvegardé dans le journal d'évolution.

Le journal d'évolution est une structure permettant de sauvegarder les changements appliqués à l'ontologie de domaine sous forme de séquences chronologiques d'information sur l'ensemble des traitements effectués et les résultats obtenus au long des phases de gestion de changements. Ce journal facilite le suivi de l'évolution, le retour arrière, la justification de l'enrichissement de l'ontologie, la

gestion des versions et dans une perspective future l'apprentissage de patrons d'inconsistances et de patrons d'alternatives de résolutions.

### 3 Système de gestion de changements

Un prototype du système de gestion de changement a été implémenté permettant de gérer les changements d'ontologies OWL tout en maintenant la consistance et la qualité. Le prototype permet de valider l'application d'un sous-ensemble prédéfini de changements formellement spécifiés.

L'architecture du système illustrée par la figure ci-après, inclut quatre principaux composants et cinq sources d'information. Le composant *détection des inconsistances* se base sur la représentation formelle du changement et les contraintes de consistance du modèle OWL pour localiser les inconsistances causées. Le composant *proposition d'alternatives de résolution*, basé sur ces mêmes ressources, propose des alternatives de résolution pour chacune des inconsistances détectées. Le composant *évaluation des alternatives* basé sur le modèle hiérarchique de qualité et les pondérations des critères définies par l'expert, permet d'évaluer l'impact des différentes alternatives proposées sur la qualité de l'ontologie. Le composant *application des changements*, tenant compte des résultats des évaluations permet de valider directement un changement si la qualité est préservée ou de fournir les résultats de résolution et d'évaluation à l'expert pour décider du changement. Un journal de l'historique des différentes étapes permet d'annoter et de garder la traçabilité des traitements effectués tout au long du processus.

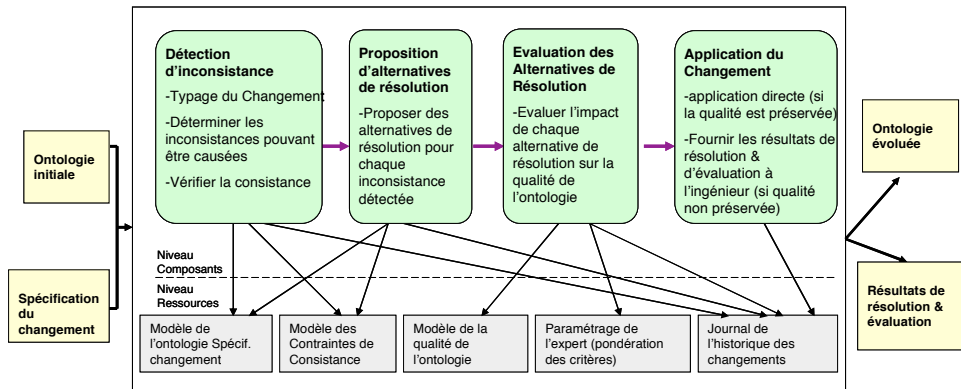


Fig. 3 – Architecture du système de validation de changements

#### 4 Discussion et travaux existants

L'analyse des effets de changements inclut non seulement la vérification de la consistance mais aussi la résolution des inconsistances détectées. Néanmoins, il n'y a pas suffisamment de travaux sur la maintenance de la consistance dans la littérature. La proposition la plus significative est présentée dans (Stojanovic et al., 2003) (Stojanovic, 2004). Les auteurs ont proposé un processus global de gestion d'évolution pour les ontologies KAON spécifiant la sémantique des changements et assurant la maintenance de la consistance. Ils présentent un modèle pour la sémantique des changements KAON et proposent des stratégies de résolution d'inconsistances en se basant sur les contraintes du langage KAON. Les stratégies sont présentées à l'utilisateur pour qu'il décide de leur application en se basant sur son expertise du domaine. Les principes de résolution d'inconsistances ont aussi été adaptés dans (Haase et Stojanovic 2005), aux ontologies OWL. Les auteurs ont introduit des stratégies de résolution basées sur les contraintes de OWL-Lite. La résolution des inconsistances logiques se base sur la détermination et la suppression des axiomes causant ces inconsistances selon deux approches : l'une générant le nombre minimal de changements à appliquer pour obtenir la sous-ontologie consistante maximale et l'autre permettant de localiser les inconsistances c'est-à-dire de retrouver la sous-ontologie inconsistante minimale. Les axiomes à supprimer sont ensuite présentés à l'utilisateur pour qu'il puisse décider et contrôler l'évolution de l'ontologie. Pour sélectionner les axiomes à supprimer, les auteurs proposent l'idée de définir des fonctions de sélection en s'inspirant des travaux sur l'incertitude et la réification et sur les recherches en ontologies contextuelles pour dégager des métadonnées décrivant les axiomes (confiance, degré de certitude, caractéristiques contextuelles, etc.). Dans notre approche, nous tendons à minimiser la suppression d'axiomes en proposant des patrons d'alternatives résolvant les inconsistances en fusionnant, divisant, généralisant ou spécialisant les concepts pour préserver les connaissances existantes.

L'application d'un changement peut altérer non seulement la consistance mais aussi la qualité de l'ontologie. Toutefois, les effets de changements sur la qualité de l'ontologie n'ont pas été étudiés dans la littérature bien que la notion de qualité d'une ontologie ait fait l'objet de plusieurs recherches (Hartmann et al., 2005) (Brank et al., 2005) (Supekar, 2005) (Yang et al. 2006). Selon cette perspective, nous proposons une approche complémentaire aux travaux en évolution d'ontologies, menant l'application des changements de manière systématique et optimisée. Ainsi, plutôt que de nécessiter l'intervention de l'expert dans la résolution des inconsistances, nous guidons le choix de l'alternative de résolution à appliquer en évaluant l'impact des alternatives sur la qualité de l'ontologie.

Une autre différence s'ajoute en comparaison avec les travaux de (Haase et Stojanovic, 2005) : l'emploi de techniques d'évaluation de qualité dans la résolution des inconsistances à travers un modèle d'évaluation basé sur des critères de qualité quantifiables. Dans (Haase et Stojanovic 2005), seuls quelques critères de qualité non-

quantifiables ont été évoqués, sous forme de conditions génériques de consistance définies par l'ingénieur pour assurer une bonne modélisation de l'ontologie telles que les méta-propriétés de la méthodologie OntoClean représentant les notions de rigidité, identité, unité et dépendance (Guarino et Welty, 2002). Par ailleurs, pour minimiser l'impact de suppression d'axiomes sur les ontologies, les auteurs se réfèrent à un ensemble de besoins spécifiques pouvant être définis par l'utilisateur (tels que la minimisation du nombre d'axiomes à supprimer, ou la définition d'un degré de pertinence pour chaque axiome pour guider le choix des axiomes à supprimer). Dans notre approche, de tels besoins peuvent toujours être considérés mais en amont, nous avons défini une couche d'évaluation de l'impact sur la qualité qui tient compte de critères quantifiables et indépendants de l'utilisateur.

Finalelement, notons que l'approche est centrée sur la validation des changements et non sur tout le processus d'évolution. Elle part de l'analyse d'un changement formellement spécifié et vise à l'appliquer tout en assurant la consistance et la qualité de l'ontologie. Par ailleurs, cette approche est principalement adaptée à un processus d'enrichissement d'une ontologie centralisée et ne tient pas compte dans sa version actuelle, de la dimension distribuée de l'environnement de l'ontologie.

## **5 Conclusion et travaux futurs**

Dans cet article, nous proposons une approche d'enrichissement d'ontologies dont l'objectif est d'optimiser et automatiser la gestion des changements tout en assurant la consistance et la qualité de l'ontologie évoluée. La maintenance de la consistance est assurée à travers la proposition d'alternatives de résolution d'inconsistance. Un modèle de qualité est défini et appliqué pour évaluer l'impact des alternatives proposées sur la qualité et guider la résolution des inconsistances tout en minimisant la dépendance à l'utilisateur. L'alternative préservant la qualité est choisie automatiquement. Elle représente les changements complémentaires à appliquer pour maintenir la consistance de l'ontologie et sa qualité. Les changements peuvent alors être validés directement. Par contre, si toutes les alternatives ont un effet négatif sur la qualité, l'intervention de l'expert est sollicitée. Le résultat d'évaluation offre une information complémentaire à l'expertise de l'utilisateur qui le guidera dans sa décision en lui indiquant l'alternative la moins coûteuse et l'aidera aussi dans la justification des coûts des changements par rapport à leur pertinence.

Actuellement, nous travaillons sur l'application du processus de gestion de changements d'ontologie pour l'enrichissement d'une ontologie OWL de pneumologie fournie par l'INSERM. Nous procédons à l'enrichissement de l'ontologie de pneumologie par extraction et réutilisation de connaissances issues des standards médicaux : GALEN-CRM, SNOMED CT et UMLS.

## Références

- BRANK, J. & GROBELNIK, M. & MLADENIC, D (2005). *A survey of ontology evaluation techniques*, Proceedings de la conférence data mining and data warehouses sikdd'05, Ljubljana, Slovenie.
- A. BANEYX & J. CHARLET & M. JAULENT, (2006). *Building an ontology of pulmonary diseases with natural language processing tools using textual corpora*. International Journal of Medical Informatics, Volume 76, Issue 2-3, Pages 208-215.
- DJEDIDI R. & ABBOUTE H. & MARIE-AUDE AUFAURE (2007). *Evolution d'ontologie : validation des changements basée sur l'évaluation*, Proceedings 1ères journées francophones sur les ontologies, Sousse, Tunisie.
- EDER J. & WIGGISSER K (2007). *Change Detection in Ontologies Using DAG Comparison*, Proceedings CAiSE, LNCS, pp. 21—35, Springer edition.
- GUARINO N. & C. WELTY (2002). *Evaluating Ontological Decisions with OntoClean*, Communication ACM, 45(2), pp 61-65.
- HAASE P. & STOJANOVIC L (2005). *Consistent Evolution of OWL Ontologies*, European Conference on Semantic Web, ECSW'05.
- HARTMANN J. & SPYNS P. & GIBOIN A. & MAYNARD D. & CUEL R. & SUAREZ-FIGUEROA M.C. & SURE Y. (2005). *Methods for ontology evaluation*, Knowledgeweb D1.2.3 deliverable.
- HORROCKS I. & PATEL-SCHNEIDER P. F. (2004). *Reducing OWL Entailment to Description Logic Satisfiability*. Journal of Web Semantics, 1(4).
- HORROCKS I. & PATEL-SCHNEIDER P. F. & VAN HARMELEN F. (2003). *From SHIQ and RDF to OWL: The Making of a Web Ontology Language*. Journal of Web Semantics, 1(1).
- KLEIN, M (2004). *Change Management for Distributed Ontologies*. Mémoire de thèse, Dutch Graduate School for Information and Knowledge Systems, SIKS Dissertation Series No. 2004-11.
- KLEIN M. & FENSEL D. & KIRYAKOV A. & OGNJANOV D. (2002). *Ontology versioning and change detection on the Web*, 13ème International Conference on Knowledge Engineering and Knowledge Management EKAW02.
- MAEDCHE, A. & STAAB, A (2002). *Measuring similarity between ontologies*. Proceedings de European Conference on Knowledge Acquisition and Management – EKAW Madrid, Spain, LNCS/LNAI 2473, Springer (2002) 251-263.
- PLESSERS P. & DE TROYER O (2005). *Ontology Change Detection using a Version Log*, Proceedings de International Semantic Web Conference ISWC'05.
- STOJANOVIC L (2004). *Methods and Tools for Ontology Evolution*. Mémoire de thèse, Université de Karlsruhe.
- STOJANOVIC L. & MAEDCHE M., STOJANOVIC N. & STUDER R (2003). *Ontology evolution as reconfiguration-design problem solving*, K-CAP03 Proceedings, ACM pp.162—171.
- SUPEKAR, K (2006). *A peer-review approach for ontology evaluation*, Proceedings de la 8ème conférence International. Protégé Conference, Madrid, Spain.
- SURE Y., TEMPICH C. (2004). *State of the art in ontology engineering methodologies*, SEKT informal deliverable 7.1.2, Institut AIFB, Université de Karlsruhe.
- XUAN D. N., BELLATRECHE L. PIERRA G. (2006). *A versioning management model for ontology-based data warehouses*, Proceedings of DaWak'06.
- YANG, Z., ZHAN, D., YE, C (2006). *Evaluation Metrics for Ontology Complexity and Evaluation Analysis*, IEEE International Conference on e-Business Engineering, ICEBE06.