

# Évolution de l'ontologie et gestion des annotations sémantiques inconsistantes

Phuc-Hiep Luong<sup>\*,\*\*</sup>, Rose Dieng-Kuntz<sup>\*</sup>, Alain Boucher<sup>\*\*</sup>

<sup>\*</sup>INRIA Sophia Antipolis  
2004 route des Lucioles, BP93, 06902, Sophia Antipolis, France  
Phuc-Hiep.Luong | Rose.Dieng@sophia.inria.fr  
<http://www.sop.inria.fr/acacia/>

<sup>\*\*</sup>Equipe MSI, Institut de la Francophonie pour l'Informatique  
42 rue Ta Quang Buu, Hai Ba Trung, Ha noi, Vietnam  
Alain.Boucher@auf.org  
<http://www.ifi.edu.vn>

**Résumé.** Les ontologies et les annotations sémantiques sont deux composants importants dans un système de gestion des connaissances basé sur le Web sémantique. Dans l'environnement dynamique et distribué du Web sémantique, les ontologies et les annotations pourraient être changées pour s'adapter à l'évolution de l'organisation concernée. Ces changements peuvent donc entraîner des inconsistances à détecter et traiter. Dans cet article, nous nous focalisons principalement sur l'évolution des annotations sémantiques en soulignant le contexte où les modifications de l'ontologie entraînent des inconsistances sur ces annotations. Nous présentons une approche basée sur des règles permettant de détecter les inconsistances dans les annotations sémantiques devenues obsolètes par rapport à l'ontologie modifiée. Nous décrivons aussi les stratégies d'évolution nécessaires pour guider le processus de résolution de ces inconsistances grâce à des règles correctives.

## 1 Introduction

Les connaissances organisationnelles sont considérées comme un capital important de l'organisation. De plus en plus, les organisations exploitent un système de gestion des connaissances afin de faciliter l'accès, le partage, et la réutilisation des connaissances organisationnelles. Dans la prochaine génération du Web, le Web sémantique visant une meilleure coopération entre humains et machines (Berners-Lee et al., 2001), le système de gestion des connaissances pourra être constitué à partir de trois composants principaux (i) des ressources hétérogènes, évolutives, distribuées sur l'Intranet et indexées en utilisant (ii) des annotations sémantiques exprimées avec le vocabulaire partagé fourni par (iii) des ontologies (Dieng-Kuntz, 2005).

Cependant, les organisations vivent dans un environnement dynamique et en cours d'évolution qui mène souvent à des changements de leur système de gestion des connaissances,

ces changements affectant l'ontologie parfois (Stojanovic et al., 2002). Les changements de l'ontologie de base doivent être propagés à toutes les ontologies dépendantes, aux instances ontologiques, aux logiciels utilisant cette ontologie modifiée et aux annotations sémantiques qui reposent sur le vocabulaire partagé de l'ontologie. En conséquence, de telles modifications influencent les activités et l'exécution du système de gestion des connaissances et elles ne sont pas encore bien résolues jusqu'à maintenant (Lindgren et al., 2002).

Dans cet article, nous présentons certains contextes qui peuvent affecter l'état consistant des annotations sémantiques en soulignant le cas où les modifications de l'ontologie entraînant les inconsistances sur ces annotations. Nous décrivons les stratégies d'évolution nécessaires pour le processus d'évolution de l'annotation sémantique et aussi une approche basée sur des règles permettant de détecter des annotations inconsistantes et de guider le processus de résolution de ces inconsistances en appliquant des règles de correction. Nous faisons d'abord un bref résumé des recherches actuelles sur l'évolution de l'ontologie et les systèmes de gestion des connaissances basés sur l'ontologie, l'influence des changements aux composants du système, etc. Ensuite, la section 3 présente (i) l'évolution de l'annotation sémantique en introduisant le scénario le plus fréquent dans lequel l'ontologie de base est modifiée, (ii) les stratégies d'évolution pour les parties dépendantes de l'ontologie ainsi que les annotations sémantiques, et (iii) son processus d'évolution. Dans la section 4, nous proposons une approche basée sur des règles de détection des inconsistances et des règles de correction des inconsistances. Avant de donner une conclusion dans la dernière section, nous discutons également dans la section 5 sur les points similaires entre notre travail et certaines études existantes.

## 2 Travaux connexes

La majorité des recherches dans le domaine des systèmes de gestion des connaissances, de l'ontologie ou des annotations sémantiques sont concentrées principalement sur des problèmes de construction. Peu de recherches font face aux changements et fournissent des facilités de maintenance du système de gestion des connaissances. Un travail de recherche intéressant sur l'évolution du système de gestion des connaissances est présenté dans (Lindgren et al., 2002). Cet article analyse deux types de changements : (i) changements fonctionnels et (ii) changements structurels qui peuvent apparaître dans des systèmes de gestion des connaissances.

Dans (Heflin, 2001), les auteurs précisent que les ontologies sur le Web doivent évoluer. Ils fournissent une nouvelle définition formelle des ontologies pour l'usage dans les environnements dynamiques et distribués, ils présentent également SHOE, un langage de représentation des connaissances sur le Web. Heflin (2001) aborde le support de multiples versions d'ontologies et la réutilisation d'ontologies mais il ne traite pas la propagation des changements aux ontologies distribuées et dépendantes. Certaines études sur la gestion des changements dans l'ontologie (Noy et Klein, 2004) (Stojanovic, 2004) s'inspirent de recherches sur l'évolution de schémas des bases de données (Roddick, 1996). Un système de gestion de versions d'ontologies est décrit dans (Klein et al., 2002), il permet l'accès aux données par différentes versions de l'ontologie. Cette approche est basée sur la comparaison de deux versions d'ontologies afin de détecter des changements. D'autre part, les auteurs de (Noy et Klein, 2004) et (Klein, 2004) ont proposé une plate-forme pour l'évolution d'ontologies mais leur approche n'analyse pas d'une manière détaillée de l'effet des changements spécifiques sur l'interprétation des données. Contrairement à cette approche de gestion de versions, (Stojanovic, 2004) présente une

approche d'évolution d'ontologie qui permet l'accès à toutes les données via la version de l'ontologie la plus récente. Les auteurs ont présenté un processus de six phases pour l'évolution d'ontologie avec certaines procédures de résolution de changement. Ils ont approfondi les deux phases (i) sémantiques de changement qui permet la résolution des changements d'ontologie d'une façon systématique en assurant la consistance entière de l'ontologie et (ii) la propagation de changement qui amène automatiquement les artefacts dépendants (par exemple, les instances basées sur l'ontologie, les applications et les ontologies dépendantes) dans un état cohérent après une modification de l'ontologie. Dans l'article (Haase et Sure, 2004), les auteurs ont récapitulé brièvement ces deux approches principales d'évolution et de versionnement de l'ontologie. (Rogozan et Paquette, 2005) a aussi présenté une approche combinée qui soutient le processus de l'évolution d'ontologie et de la gestion de versions d'ontologie en contrôlant l'historique des changements de l'ontologie. Toutefois, ces travaux mentionnés ci-dessus n'ont pas traité la propagation du changement aux annotations sémantiques relatives en cas de changements des ontologies de base.

Très peu d'approches ont étudié les problèmes de la propagation des changements ontologiques aux annotations sémantiques. Les auteurs dans (Stojanovic et al., 2002) ont proposé la plate-forme CREAM pour résoudre l'évolution des métadonnées basées sur l'évolution de l'ontologie. Cependant, cette approche ne présente qu'une proposition d'une plate-forme pour permettre la consistance des descriptions des sources de la connaissance en cas de changements de l'ontologie de domaine mais ne propose aucune technique pour la résoudre. Un autre travail sur l'influence de l'évolution d'ontologie sur les métadonnées par les contraintes relationnelles d'un système de base de données est également présenté dans (Ceravolo et al., 2004). Selon cette approche, une base de données relationnelle est utilisée pour stocker les descriptions en RDF(S) représentant les assertions basées sur l'ontologie et la structure d'ontologie elle-même. La maintenance d'ontologie peut être contrôlée en utilisant des triggers de base de données pour modifier automatiquement des domaines ou des co-domaines de propriété dans les assertions stockées.

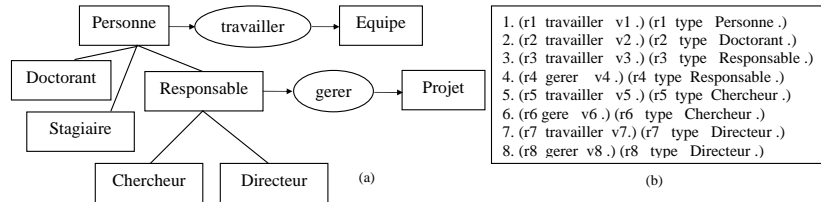
### 3 Évolution de l'annotation sémantique

Klein et al. (2002) a montré qu'une modification dans une partie de l'ontologie peut entraîner des inconsistances dans d'autres parties de la même ontologie, dans les instances basées sur l'ontologie ainsi que dans des ontologies dépendantes et dans les applications utilisant cette ontologie modifiée. En particulier, les changements de l'ontologie peuvent affecter les annotations sémantiques qui utilisent les concepts ou les propriétés définies dans cette ontologie de base. Inspirées des recherches sur l'évolution de schémas dans les bases de données (Roddick, 1996), sur l'évolution de l'ontologie (Stojanovic, 2004) et sur la gestion de versions de l'ontologie (Klein et al., 2002) (Klein, 2004), nous considérons *l'évolution de l'annotation sémantique comme un processus d'adaptation d'une annotation aux inconsistances générées à cause des changements sur l'ontologie ou sur l'annotation soi-même.*

#### 3.1 Contexte

Nous considérons des scénarios ci-dessous qui peuvent affecter l'état cohérent des annotations sémantiques :

## Gestion des annotations sémantiques inconsistantes



**FIG. 1** – Exemple d’une ontologie (a) et des annotations sémantiques basées sur cette ontologie (b)

- Scénario 1 : Le fournisseur de l’ontologie fait des modifications sur son ontologie de base. A cause de ces changements ontologiques, les annotations peuvent être affectées entraînant un état inconsistant.
- Scénario 2 : L’utilisateur change ses annotations lui-même sans référer à l’ontologie de base utilisée par ces annotations. Les annotations modifiées peuvent être inconsistantes avec l’ontologie de base correspondante.
- Scénario 3 : L’utilisateur importe les annotations à partir d’autres sources. Ces annotations importées et les annotations anciennes reposent sur une même ontologie. Il pourrait y avoir des annotations (ou des triplets) similaires qui décrivent la même ressource.
- Scénario 4 : L’utilisateur fait la migration de la base d’annotation à partir d’un autre formalisme de représentation (par exemple la migration de RDFS vers OWL-Lite). Il pourrait y avoir des inconsistances de syntaxe sur les annotations.

Cependant, le premier scénario semble le cas le plus rencontré dans la réalité. Dans cet article, nous allons nous focaliser sur un problème particulier : les changements de l’ontologie de base peuvent affecter l’état cohérent des annotations sémantiques qui reposent sur les termes définis dans l’ontologie de base. Nous reposons sur le langage RDF(S) présenté dans (Klyne et Carroll, 2004) pour modéliser l’ontologie et décrire un triplet ( $s p v$ ) en RDF dans l’annotation. Ce triplet représente une déclaration sur la ressource qui peut être exprimée comme "un sujet  $s$  a une propriété  $p$  dont la valeur est  $v$ ".

Examinons l’exemple d’une part de l’ontologie (c.f. Fig.1a) contenant des concepts *Personne*, étant le domaine de la propriété *travailler*, qui est le père des sous-concepts *Doctorant*, *Stagiaire* et *Responsable*. Le concept *Responsable*, ayant deux sous-concepts *Chercheur* et *Directeur*, est le domaine de la propriété *gerer*. Les concepts *Equipe* et *Projet* sont les co-domaines d’une manière équivalente des propriétés *travailler* et *gerer*. D’autre part, nous avons aussi certains triplets dans des annotations sémantiques basées sur cette part de l’ontologie (c.f. Fig.1b). Supposons que cette partie de l’ontologie soit modifiée en supprimant le concept *Responsable*, ses deux sous-concepts sont reconnectés au concept *Personne* et la propriété *gerer* reçoit désormais le concept *Directeur* comme domaine. En outre, les concepts *Doctorant* et *Stagiaire* sont fusionnés et remplacés par un nouveau concept *Etudiant*. Après avoir appliqué ces changements, nous obtenons la nouvelle version de l’ontologie (c.f. Fig.2a) dans laquelle certains éléments ont été changés par rapport à son ancienne version. Certains triplets deviennent maintenant inconsistants (c.f. Fig.2b) du fait de la perte des liens référence vers les termes correspondants dans l’ontologie avant sa modification. Nous allons analyser les causes

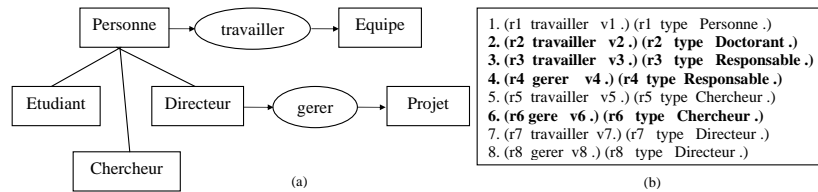


FIG. 2 – L'ontologie modifiée (a) et les annotations sémantiques devenant inconsistantes (b).

et les solutions pour les triplets inconsistants (e.g les triplets 2, 3, 4 et 6) dans la section 4.

### 3.2 Représentation des changements

Au cours de l'évolution, les changements doivent être identifiés et représentés dans les formats appropriés. Stojanovic (2004) a classé les trois niveaux de changements de l'ontologie : (i) changement élémentaire (ii) changement composite et (iii) changement complexe. De même, les opérations de l'ontologie sont aussi divisées selon deux dimensions (i) atomique/composite et (ii) simple/riche (Klein, 2004). Dans notre approche, nous étudions tous les changements de l'ontologie qui peuvent affecter la consistance de ses parties dépendantes, des autres ontologies dépendantes et particulièrement les changements ontologiques affectant la consistance des annotations sémantiques concernées. Nous avons construit une liste des changements nécessaires pour le processus d'évolution de l'ontologie et des annotations sémantiques contenant deux types de changements suivants :

- Changements élémentaires : les changements les plus fins et ne pouvant pas être décomposés (e.g. *Changer\_Nom\_Concept*, *Supprimer\_Concept*, *Supprimer\_Lien\_Domaine*, etc.)
- Changements composites : les changements pouvant être décomposés (e.g. *Fusionner\_Concepts*, *Diviser\_Concepts*, etc.)

### 3.3 Stratégies d'évolution pour les changements

Lorsqu'on effectue des modifications sur l'ontologie, non seulement les autres parties de cette ontologie peuvent devenir inconsistantes mais aussi des annotations sémantiques pourraient être influencées. Dans ce cas, la construction des stratégies d'évolution nous permettent de contrôler les inconsistances apparues. Ces stratégies d'évolution assurent que l'ontologie et les parties dépendantes restent cohérentes après des changements effectués sur l'ontologie. En outre, elles sont aussi responsables d'empêcher les changements illégaux. Stojanovic (2004) propose certaines stratégies d'évolution pour l'ontologie dans lesquelles elle définit les points de résolution et les stratégies élémentaires pour chaque cas de changement de l'ontologie. Cependant, ses stratégies d'évolutions ne couvrent que les effets des changements simples et l'auteur n'a pas encore abordé les stratégies d'évolution pour les annotations sémantiques.

Dans cette section, nous voudrions présenter un ensemble des stratégies d'évolution qui essaient de résoudre des inconsistances causées par les deux types de changement : simple et composite. Pour chacun des changements ontologiques qui pourrait affecter la consistance des annotations, nous avons construit une stratégie équivalente afin de corriger les inconsis-

## Gestion des annotations sémantiques inconsistantes

Stratégie d'évolution : pour l'ontologie évoluée	Stratégie d'évolution : pour les annotations concernées
<p>SO-1 : Traiter les instances du concept supprimé <math>c</math> : 4 options</p> <ol style="list-style-type: none"> <li>(1) supprimer toutes les instances du concept <math>c</math></li> <li>(2) rattacher toutes les instances du concept <math>c</math> au concept père <math>c2</math> de <math>c</math></li> <li>(3) rattacher toutes les instances du concept <math>c</math> au concept racine <math>c0</math> de l'ontologie</li> <li>(4) rattacher les instances du concept <math>c</math> au concept quelconque <math>cx</math> indiqué par l'utilisateur</li> </ol>	<p>SA-1 : Traiter le triplet contenant les instances du concept supprimé <math>c</math> : 4 options</p> <ol style="list-style-type: none"> <li>(1) supprimer ce triplet</li> <li>(2) si le concept père <math>c2 \in \text{domain}(p)</math> ou <math>c2 \in \text{co-domain}(p)</math> alors on remplace le nom du type <math>c</math> des instances par le nom du type <math>c2</math> dans le triplet. Sinon, on supprime ce triplet</li> <li>(3) si le concept racine <math>c0 \in \text{domain}(p)</math> ou <math>c0 \in \text{co-domain}(p)</math> alors on remplace le nom du type <math>c</math> des instances par le nom du type <math>c0</math> dans le triplet. Sinon, on supprime ce triplet</li> <li>(4) si le concept indiqué <math>cx \in \text{domain}(p)</math> ou <math>cx \in \text{co-domain}(p)</math> alors on remplace le nom du type <math>c</math> des instances par le nom du type <math>cx</math> dans le triplet. Sinon, on supprime ce triplet</li> </ol>
<p>SO-2 : Traiter les sous-concepts de <math>c</math> : 4 options</p> <ol style="list-style-type: none"> <li>(1) supprimer tous les sous-concepts de <math>c</math></li> <li>(2) rattacher les sous-concept de <math>c</math> au concept père de <math>c</math></li> <li>(3) rattacher les sous-concept de <math>c</math> au concept racine de l'ontologie</li> <li>(4) rattacher les sous-concept de <math>c</math> au concept quelconque <math>cx</math> indiqué par l'utilisateur</li> </ol>	<p>SA-2 : Traiter les triplets contenant les ressources du type <math>c</math></p> <ol style="list-style-type: none"> <li>(1) ne pas changer les triplets</li> </ol> <p><i>Noter : Il faut répéter les stratégies d'évolution pour tous les sous-concepts de <math>c</math> et pour les annotations sémantiques concernées</i></p>
<p>SO-3 : Traiter les propriétés concernées par le concept supprimé <math>c</math> : 3 options</p> <ol style="list-style-type: none"> <li>(1) enlever <math>c</math> du domaine de la propriété <math>p</math></li> <li>(2) si <math>c \in \text{domain}(p)</math> et il existe un concept <math>c2</math> ancêtre de <math>c</math> tel que <math>c2 \in \text{domain}(p)</math>, enlever <math>c</math> du domaine de la propriété <math>p</math></li> <li>(3) si <math>c \in \text{domain}(p)</math> et il n'existe pas un concept <math>c2</math> ancêtre de <math>c</math> tel que <math>c2 \in \text{domain}(p)</math>, enlever <math>c</math> du domaine de la propriété <math>p</math></li> </ol>	<p>SA-3 : Traiter les triplets contenant les ressources du type <math>c</math> et contenant la propriété <math>p</math> : 3 options</p> <ol style="list-style-type: none"> <li>(1) supprimer tous les triplets contenant les ressources du type <math>c</math></li> <li>(2) remplacer le nom du type <math>c</math> des ressources par le nom du type père <math>c2</math> de <math>c</math> dans l'annotation</li> <li>(3) supprimer les triplets contenant les ressources du type <math>c</math> et la propriété <math>p</math> dans l'annotation</li> </ol>

**TAB. 1** – Stratégies d'évolution proposées pour l'ontologie et les annotations dans le cas de changement de suppression d'un concept milieu

tances apparues dans les annotations sémantiques. Afin d'illustrer nos stratégies d'évolution, examinons un des cas de changement de la suppression d'un concept  $c$  au milieu de l'ontologie. Supposons que  $c2$  est le concept père de ce concept  $c$ ,  $c0$  est le concept racine,  $p$  est une propriété qui pourrait recevoir les concepts  $c$ ,  $c2$ ,  $c0$  comme son domaine/co-domaine. Les stratégies d'évolution sont décrites dans le tableau 1 (c.f. Tab.1).

### 3.4 Processus de l'évolution de l'annotation sémantique

Après avoir appliqué des changements sur une ontologie, cette ontologie a évolué vers une nouvelle version. Nous distinguons les deux cas d'évolution de l'ontologie qui peuvent influencer l'état consistant de l'annotation : (i) avec trace et (ii) sans trace des changements réalisés entre deux versions de l'ontologie.

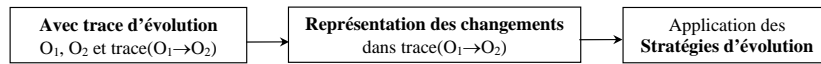


FIG. 3 – Évolution de l'ontologie avec trace entre deux versions.

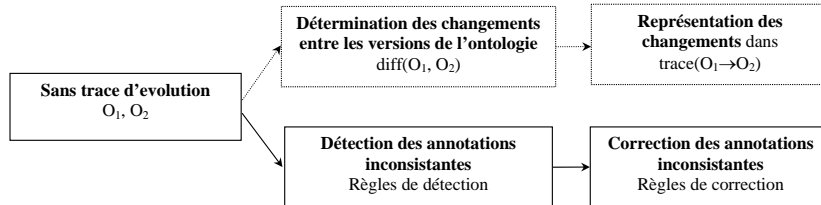


FIG. 4 – Évolution de l'ontologie sans trace entre deux versions.

Dans le cas où on peut garder la trace de changement  $trace(O_1 \rightarrow O_2)$  entre les deux versions  $O_1$  et  $O_2$  de l'ontologie (c.f. Fig.3), tous les changements réalisés ainsi que les résultats des opérations sont conservés dans un journal de changement. Ces changements réalisés pourraient être représentés d'une manière plus formelle selon notre classification de changement (simple et composite). Ensuite, nous appliquons des stratégies d'évolution correspondant à chaque changement ontologique afin de rétablir l'état consistant pour les annotations sémantiques influencées.

Le deuxième cas se passe souvent dans le contexte dynamique et distribué du Web sémantique, il n'est pas toujours possible de garder les traces entre des versions de l'ontologie (c.f. Fig.4). Dans ce cas, nous pouvons réutiliser les résultats des recherches existantes permettant de trouver les similarités et les différences  $diff(O_1, O_2)$  entre les deux versions  $O_1$  et  $O_2$  de l'ontologie ainsi que les changements réalisés entre ces versions (Klein et al., 2002) (Klein, 2004). Avec ces changements ontologiques détectés, nous pouvons suivre ensuite les procédures de résolution des inconsistances présentées dans le cas ci-dessus.

Cependant, nous proposons un autre processus permettant de trouver dans une base d'annotations les annotations concernées par les modifications de l'ontologie et particulièrement des annotations inconsistantes à cause des changements ontologiques. Nous utilisons Corese, un moteur de recherche sémantique développé par l'équipe Acacia (Corby et al., 2004), qui permet de faire des interrogations sur les bases d'annotation. Dans cette étape, Corese peut trouver les *annotations potentiellement inconsistantes* qui peuvent inclure des annotations consistantes ou inconsistantes concernées (une annotation est potentiellement inconsistante si elle est liée au changement ontologique mais sa contrainte de consistance n'a pas été vérifiée). Ces annotations seront ensuite vérifiées pour détecter les *annotations réellement inconsistantes* (une annotation réellement inconsistante viole la contrainte de consistance définie pour l'annotation) en utilisant des règles de détection d'inconsistance. Après avoir déterminé les annotations inconsistantes réelles, elles seront corrigées en appliquant les règles de correction d'inconsistance. Nous avons établi toutes les stratégies possibles qui résolvent la propagation des changements ontologiques vers leurs annotations sémantiques afin de conserver l'état consistant pour ces

annotations. Dans la section suivante, nous allons présenter ce processus de résolution des annotations sémantiques inconsistantes par une approche basée sur les règles.

## 4 Résolution des inconsistances de l'annotation sémantique : Une approche basée sur les règles

Notre approche basée sur les règles est construite à partir des contraintes de consistance qui doivent être satisfaites pour le modèle d'annotation. Nous avons défini certaines contraintes de consistance qui peuvent être considérées comme un accord cohérent pour les entités d'annotations sémantiques en ce qui concerne leur ontologie de base. En reposant sur ces contraintes, nous avons construit des règles qui permettent de détecter des inconsistances dans les annotations et de rétablir l'état consistant pour ces annotations.

### 4.1 Contraintes de consistance

Nous proposons ci-dessous des contraintes qui assurent l'utilisation d'une manière cohérente des termes (concepts, propriétés, domaine/co-domaine, etc) définis dans le vocabulaire de l'ontologie pour annoter les ressources.

- Contrainte sur le concept : Tous les concepts utilisés dans l'annotation doivent être définis avant dans l'ontologie.
- Contrainte sur la propriété : Toutes les propriétés utilisées dans l'annotation doivent être définies avant dans l'ontologie.
- Contrainte sur le domaine de propriété : Une ressource qui est le domaine d'une propriété dans l'annotation doit être compatible avec le domaine de la propriété correspondante définie dans l'ontologie.
- Contrainte sur le co-domaine de propriété : Une ressource qui est le range d'une propriété dans l'annotation doit être compatible avec le range de la propriété correspondante définie dans l'ontologie.
- Contrainte sur le type des données : Le type de données d'une valeur de propriété dans l'annotation doit être compatible avec la valeur de la propriété correspondante définie dans l'ontologie.

### 4.2 Détection des inconsistances de l'annotation sémantique

Cette phase de détection d'inconsistances nous permet de trouver les annotations inconsistantes qui deviennent obsolètes par rapport à l'ontologie modifiée. Elle est réalisée grâce à un ensemble des règles de détection d'inconsistances qui sont basées sur les contraintes de consistance. Soit  $C_O$  et  $P_O$  les ensembles des concepts et des propriétés de l'ontologie  $O$ , soit  $C_A$  et  $P_A$  les ensembles des concepts et des propriétés figurant dans l'annotation  $A$ . Tous les triplets dans l'annotation ont la forme  $(r \ p_t \ v \ .)$  avec la ressource  $r$  du type  $c_t$  ( $r \ type \ c_t \ .$ ). Nous utilisons aussi des prédicats  $domain(p, c)$  (respectivement  $range(p, c)$ ) pour exprimer que le concept  $c$  est le domaine/co-domaine de la propriété  $p$ . Nous avons construit des groupes de règles qui peuvent détecter les inconsistances concernant les concepts, les propriétés, les



domaine/co-domaine, les types des données (datatype). Dans le paragraphe suivant, nous présentons certaines règles qui peuvent être utilisées pour illustrer l'exemple mentionné au-dessus. Groupe 1 (règles de détection pour la ressource de concept) :

- R-1 :  $\forall$  triplet  $t$ , Si  $(c_t \in C_A)$  et  $(c_t \notin C_O)$  Alors *marquer(inconsistant)*
- R-2 :  $\forall$  triplet  $t$ , Si  $(c_t \in C_A)$  et  $(c_t \in C_O)$  Alors *marquer (OK)*

Groupe 2 (règles de détection pour la ressource de propriété) :

- R-3 :  $\forall$  triplet  $t$ , Si  $(p_t \in P_A)$  et  $(p_t \notin P_O)$  Alors *marquer(inconsistant)*
- R-4 :  $\forall$  triplet  $t$ , Si  $(p_t \in P_A)$  et  $(p_t \in P_O)$  Alors *marquer(OK)*

Groupe 3 (règles de détection pour la ressource qui est le domaine de propriété) :

- R-5 :  $\forall$  triplet  $t$ , Si  $domain(p_t, c_t) \in domain(P_A, C_A)$  et  $domain(p_t, c_t) \notin domain(P_O, C_O)$  Alors *marquer(inconsistant)*
- R-6 :  $\forall$  triplet  $t$ , Si  $domain(p_t, c_t) \in domain(P_A, C_A)$  et  $domain(p_t, c_t) \in domain(P_O, C_O)$  Alors *marquer(OK)*

Nous appliquons toutes les règles sur l'ensemble des annotations concernées par les modifications de l'ontologie. Grâce à la règle R-1, nous pouvons détecter les triplets 2, 3 et 4 qui deviennent inconsistants à cause de la perte des références vers les concepts *Doctorant*, *Responsable* de l'ontologie. La règle R-6 détecte le triplet 6 inconsistant parce que le lien de domaine entre le concept *Chercheur* et la propriété *gerer* a été supprimé dans la nouvelle version de l'ontologie.

Cependant, un problème se pose pour le processus de vérification des règles de détection d'inconsistances qui est l'identification du changement ontologique. Puisque nous ne savons pas quels changements ontologiques ont été exécutés, d'autre part chaque règle vérifiée peut correspondre à certains changements possibles. Donc, nous avons besoin d'un mécanisme d'identification des changements ontologiques exécutés. Par exemple, après avoir vérifié les règles R-1 et R-4, le résultat  $(c_t \in C_A)$  et  $(c_t \notin C_O)$  et  $(p_t \in P_A)$  et  $(p_t \in P_O)$  montre une inconsistance sur le concept, mais cela peut correspondre à plusieurs hypothèses parmi les suivantes :

- $c_t$  a été effacé dans l'ontologie, alors le nom du concept  $c_t$  peut-être remplacé par celui de son père
- $c_t$  a été renommé dans l'ontologie, alors le nom du concept  $c_t$  peut-être remplacé par le nouveau nom
- $c_t$  a été divisé en deux autres concepts le remplaçant dans l'ontologie, alors  $c_t$  a été supprimé
- $c_t$  a été fusionné avec un autre concept dans l'ontologie, alors le nom du concept  $c_t$  peut être remplacé par le nouveau nom fusionné.

Pour résoudre ce problème, nous proposons d'utiliser des heuristiques qui permettent de sélectionner une meilleure réponse parmi plusieurs changements possibles à identifier. Ci-dessous est un exemple d'une heuristique basée sur la situation de l'ensemble des instances du concept supprimé  $c_t$  dans la nouvelle version de l'ontologie qui permet d'identifier les types de changement exécutés.

- Si l'ensemble des instances du concept  $c_t$  ont été supprimées ou ont été rattachées au concept père de  $c_t$  Alors le concept  $c_t$  a été supprimé.
- Si l'ensemble des instances du concept  $c_t$  ne change pas et devient l'ensemble des instances du nouveau type de concept  $newc_t$  Alors le concept  $c_t$  a été renommé en  $newc_t$ .

## Gestion des annotations sémantiques inconsistantes

<pre>1. (r1 travailler v1.) (r1 type Personne.) 2. (r2 travailler v2.) (r2 type Doctorant.) 3. (r3 travailler v3.) (r3 type Responsable.) 4. (r4 gerer v4.) (r4 type Responsable.) 5. (r5 travailler v5.) (r5 type Chercheur.) 6. (r6 gere v6.) (r6 type Chercheur.) 7. (r7 travailler v7.) (r7 type Directeur.) 8. (r8 gerer v8.) (r8 type Directeur.)</pre>	<pre>1. (r1 travailler v1.) (r1 type Personne.) 2. (r2 travailler v2.) (r2 type Etudiant.) 3. (r3 travailler v3.) (r3 type Personne.) 4. supprimé 5. (r5 travailler v5.) (r5 type Chercheur.) 6. supprimé 7. (r7 travailler v7.) (r7 type Directeur.) 8. (r8 gerer v8.) (r8 type Directeur.)</pre>
(a)	(b)

FIG. 5 – Certains triplets inconsistants avant (a) et après (b) la mise à jour

- Si l'ensemble des instances du concept  $c_t$  a été augmenté et fait désormais partie de l'ensemble des instances du nouveau type de concept  $newc_t$ . Alors le concept  $c_t$  a été fusionné à  $newc_t$ .

Après avoir bien détecté et déterminé des types de changements ontologiques, nous pouvons appliquer des stratégies d'évolution pour résoudre des inconsistances dans les annotations sémantiques.

### 4.3 Correction des inconsistances de l'annotation sémantique

Après avoir rassemblé toutes les annotations sémantiques inconsistantes à partir d'un ensemble d'annotations potentiellement inconsistantes, nous devons corriger ces inconsistances en appliquant certaines règles de correction sur ces annotations. Ces règles guident la réalisation des stratégies d'évolution dans lesquelles nous avons indiqué comment propager la résolution de changements aux annotations inconsistantes afin de conserver une consistance globale (c.f. Tab. 1).

Revenons à l'exemple ci-dessus : après avoir détecté les inconsistances, nous trouvons que les triplets 2, 3, 4 et 6 sont dans un état inconsistant. Sachant que le changement *Supprimer\_Concept(Responsable)* entraîne la perte des liens des ressources dans les triplets 3 et 4 vers les termes correspondants dans l'ontologie, nous pouvons appliquer des règles ci-dessous :

- RC-1 : Si *SO-3(2)* est appliqué pour l'ontologie Alors exécuter *SA-3(2)* pour l'annotation. Cette règle va remplacer le nom du type *Responsable* des ressources par le nom du type père *Personne* dans le triplet 3 parce que le lien entre la propriété *travailler* et le concept *Personne* est encore gardé.
- RC-2 : Si *SO-3(3)* est appliqué pour l'ontologie Alors exécuter *SA-3(3)* pour l'annotation. Cette règle va supprimer triplet 4 parce qu'il n'existe pas le lien entre la propriété *gerer* et le concept *Personne* qui est le père du concept supprimé *Responsable*. Nous avons aussi construit des règles de correction des inconsistances dans les annotations correspondant aux autres types changements ontologiques (par exemple le type de changement *Fusionner\_Concept* concernant l'inconsistance du triplet 2).

## 5 Discussion

Considérant la résolution des inconsistances dans les annotations sémantiques dans le contexte des modifications de l'ontologie, le travail présenté dans cet article peut être comparé

à certaines études similaires présentées dans (Ceravolo et al., 2004) (Stojanovic et al., 2002). Notre proposition présente non seulement le processus de propagation des changements ontologiques aux annotations sémantiques mais indique également une approche basée sur les règles pour détecter des annotations inconsistantes et les procédures de correction guidés par les stratégies d'évolution pour résoudre ces inconsistances. Par rapport à l'approche basée sur les règles, notre travail diffère des règles d'identification présentées dans (Klein, 2004) pour trouver des changements complexes de l'ontologie. Contrairement à cette méthode qui est basée sur un ensemble de règles et d'heuristiques pour produire un changement complexe à partir d'un ensemble de changements élémentaires, notre approche se base sur les changements ontologiques exécutés stockés dans un journal d'évolution. Nos règles de détection ne traitent que les annotations sémantiques liées à ces changements au lieu d'examiner toute la base d'annotation. L'étape de détection des inconsistances sera réalisé automatique tandis que l'étape de correction des inconsistances aura besoin d'une intervention de l'utilisateur afin de choisir une meilleure solution pour chaque type d'inconsistance.

## 6 Conclusion

Nous avons présenté dans cet article les contextes d'évolution des annotations sémantiques dans un système de gestion des connaissances. Nous avons souligné le contexte le plus rencontré dans la réalité où les modifications de l'ontologie de base pourraient entraîner les inconsistances dans les annotations sémantiques. Le système de gestion des connaissances aurait besoin d'une capacité à s'adapter efficacement aux changements ainsi que les inconsistances apparues sur les ontologies et les annotations sémantiques. Pour cette raison, nous avons proposé une approche basée sur les règles permettant de contrôler l'évolution des annotations sémantiques quand les ontologies changent. Les inconsistances seront détectées et corrigées grâce à un ensemble des règles de détection et des règles de correction guidées par les stratégies d'évolution. Comme perspective pour ce travail, nous raffinerons cette approche basée sur les règles et étudierons certains algorithmes efficaces sur le processus de la correction et la validation de changement des annotations sémantiques. Nous nous concentrerons également sur le problème de gestion de versions permettant d'utiliser différentes versions des annotations sémantiques et des ontologies.

## Références

- Berners-Lee, T., J. Hendler, et O. Lassila (2001). The semantic web. *In Scientific American*, 35–43.
- Ceravolo, P., A. Corallo, G. Elia, et A. Zilli (2004). Managing ontology evolution via relational constraints. *In Lecture Notes in Computer Science 3215 Springer. Knowledge-Based Intelligent Information and Engineering Systems, 8th International Conference, KES 2004, Wellington, New Zealand*, 335–341.
- Corby, O., R. Dieng-Kuntz, et C. Faron-Zucker (2004). Querying the semantic web with the corese search engine. *In Proc. of the 16th European Conference on Artificial Intelligence (ECAI'2004), subconference PAIS'2004, Valencia, 22-27 August 2004, IOS Press*, 705–709.

- Dieng-Kuntz, R. (2005). Corporate semantic webs. In *Encyclopaedia of Knowledge Management*, D. Schwartz ed, Idea Publishing Group, September 2005, 67–80.
- Haase, P. et Y. Sure (2004). D3.1.1.b state-of-the-art on ontology evolution. Technical report, Institute AIFB, University of Karlsruhe. In SEKT/2004/D.3.1.1.b/v0.5.
- Heflin, J. (2001). *Towards the Semantic Web-Knowledge Representation in a Dynamic, Distributed Environment*. Ph.d. thesis, University of Maryland, College Park.
- Klein, M. (2004). *Change management for distributed ontologies*. Ph.d. thesis, Vrije Universiteit Amsterdam.
- Klein, M., A. Kiryakov, D. Ognyanov, et D. Fensel (2002). Ontology versioning and change detection on the web. In *Proceedings of the 13th European conference on knowledge engineering and knowledge management (EKAW-2002)*, Sigüenza, Spain, 197–212.
- Klyne, G. et J. Carroll (2004). Resource description framework (rdf) : Concepts and abstract syntax. Technical report, In W3C Recommendation. (<http://www.w3.org/RDF>).
- Lindgren, R., C. Hardless, K. Pessi, et U. Nuldén (2002). The evolution of knowledge management systems needs to be managed. *Journal of Knowledge Management Practice Volume 3, March*.
- Noy, N. et M. Klein (2004). Ontology evolution : Not the same as schema evolution. In *Knowledge and Information Systems*, 428–440.
- Roddick, J. F. (1996). A survey of schema versioning issues for database systems. In *Information and Software Technology 37*[7], 383–393.
- Rogozan, D. et G. Paquette (2005). Managing ontology changes on the semantic web. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, 430–433.
- Stojanovic, L. (2004). *Methods and tools for ontology evolution*. Ph.d. thesis, University of Karlsruhe.
- Stojanovic, L., N. Stojanovic, et S. Handschuh (2002). Evolution of the metadata in the ontology-based knowledge management systems. In *Proc. of the 1st German Workshop on Experience Management: Sharing Experiences about the Sharing of Experience*, 65–77.

## Summary

Ontologies and semantic annotations are two main components in a knowledge management system. In the dynamic and distributed Semantic Web environment, ontologies and the semantic annotations might be changed in order to adapt new changing requirements in system. These changes could impact to the system activities or even could bring about some inconsistencies in knowledge management system. They must be well controlled and managed. In this paper, we will focus mainly on the semantic annotations evolution by emphasizing the context where modifications of the underlying ontology bringing inconsistencies on these semantic annotations. We present a rule-based approach allowing to detect and extract inconsistencies in the obsolete semantic annotations which are related to the modified underlying ontology. We also describe all necessary evolution strategies to guide the resolution process of the inconsistent annotations by applying the inconsistency correction rules.