

---

# Types de changements et leurs effets sur l'évolution de l'ontologie

Najla Sassi — Wassim Jaziri

*Laboratoire MIRACL : Multimedia, InfoRmation Systems and Advanced Computing Laboratory*

*Institut Supérieur d'Informatique et de Multimédia de Sfax.  
sassinajla@yahoo.fr, wassim.jaziri@isimsf.rnu.tn*

---

RÉSUMÉ. *Les ontologies ont souvent une grande et complexe structure, dont le développement et l'entretien provoquent certains problèmes à la fois conceptuels et sémantiques. L'un des problèmes souvent rencontrés dans les environnements évolutifs est l'intégration et la prise en compte de nouvelles connaissances en réponse à des nouveaux besoins. Se pose ainsi le problème de la mise à jour de l'ontologie initiale et son adaptation aux nouveaux besoins afin de maintenir sa cohérence et mieux modéliser le domaine étudié. Nous étudions dans ce papier les types de changements pouvant intervenir dans les environnements évolutifs et leurs conséquences conceptuelles et sémantiques sur l'ontologie.*

ABSTRACT. *Ontologies often have large and complex structures which development and maintenance cause conceptual and semantic problems. A problem related to evolutionary environments is the integration and the taking into account of new knowledge in response to new operational and conceptual requirements. We must then update the initial ontology and adapt it to the new requirements in order to maintain its coherence and to better model the studied field. We study in this paper the types of evolution changes and theirs conceptual and semantic consequences on the ontology.*

MOTS-CLÉS : *Ontologie, évolution, intégration de nouvelles connaissances, types de changements.*

KEYWORDS: *Ontology, evolution, integration of new knowledge, types of change.*

---

## 1. Introduction

Conçues comme réponse aux problèmes posés par l'intégration et la prise en compte de la sémantique des informations au sein des systèmes informatiques, les ontologies sont devenues des outils incontournables pour la représentation et le traitement de l'information au niveau sémantique. Leur champ d'application ne cesse de s'élargir pour couvrir des domaines variés comme l'ingénierie des connaissances, la recherche et l'indexation d'information et plus récemment dans les systèmes d'information (Mhiri et al., 2005). Le but de l'utilisation des ontologies varie suivant les communautés :

- La communication entre machines et humains : intégrer du sens dans les informations manipulées par les machines.
- La communication entre machines : l'interopérabilité.
- La communication entre humains via des machines.

Dans la ligne de réflexion conduisant à nos activités de recherche, nous considérons les ontologies comme des représentations de connaissances (composants logiciels) s'insérant dans les systèmes d'information pour leur apporter une dimension sémantique qui leur faisait défaut jusqu'ici. Nous nous intéressons plus particulièrement à l'ontologie de domaine qui peut être définie comme une représentation (validée par les experts d'un domaine d'étude) plus ou moins formelle du monde basée sur l'identification des entités conceptuelles (concepts ou relations) pertinentes à ce domaine et de leur sémantique. Les ontologies de domaine se veulent des représentations de connaissances bâties au niveau conceptuel intégrant à la fois les connaissances terminologiques d'un domaine et l'expression de la sémantique de celui-ci, tout en conservant leur indépendance vis-à-vis des usages opérationnels spécifiques à une application donnée.

Nous nous plaçons dans le cadre d'environnements dynamiques où les données et les besoins sont susceptibles d'évoluer au cours du temps. Par conséquent, l'ontologie du domaine peut s'avérer inadaptée aux nouveaux besoins et ne plus être en mesure de modéliser le domaine étudié. Dans de tels environnements, l'ontologie du domaine d'étude doit évoluer pour prendre en compte les changements apparus. L'évolution de l'ontologie concerne la capacité de mettre à jour l'ontologie existante suite à l'apparition de nouveaux besoins et de maintenir son uniformité et sa cohérence (Maedche et al., 2003) (Stanjanovic et al., 2002). Cet aspect reste peu étudié dans la littérature malgré le bon nombre de recherches alignées autour de l'étude et de la construction de l'ontologie et qui permettent de représenter des ontologies statiques.

C'est dans cette perspective que nous cherchons à proposer une méthodologie supportant la dynamique de l'ontologie et à même de suivre les changements lors de la conception des systèmes d'information.

Après une présentation de l'état de l'art sur les méthodologies d'évolution de l'ontologie, nous présentons les causes de l'évolution et les types de besoins pouvant

apparaître suivant l'acteur concerné. Nous exprimons ces besoins d'évolution par un ensemble de changements élémentaires et composés. Nous étudions ensuite les conséquences des différents types de changement sur l'ontologie et la manière de les prendre en compte avant de conclure avec les perspectives de ce travail.

## 2. Etat de l'Art

La plupart des travaux autour de l'ontologie s'intéressent essentiellement aux outils de construction et de représentation mais peu à l'évolution de l'ontologie. Dans la littérature, il existe essentiellement deux méthodologies qui traitent l'évolution de l'ontologie : l'approche développée pour supporter l'évolution de l'ontologie et l'approche basée sur le versionnement.

La première approche se base sur la création d'une nouvelle ontologie pour s'adapter aux changements intervenants. L'ontologie créée peut être vue comme une nouvelle version évoluée de l'ontologie initiale (avant évolution) et est caractérisée par une durée de validité (jusqu'à l'apparition d'un nouveau changement). Dans cette approche, on ne conserve qu'une seule version de l'ontologie : la plus récente (Stanjanovic et al., 2002) (Giorgos et al., 2005) (Giorgos et al., 2006). Cette approche représente certaines limites. En effet, on ne peut pas récupérer l'ancienne ontologie (uniquement la dernière version est retenue), suite par exemple à une évolution de l'ontologie nécessitant le retour à une version antérieure. Dans cette approche, l'intérêt est plus porté sur les méthodologies d'évolution de l'ontologie plutôt que sur l'opérationnalisation des différentes étapes de son suivi. En plus, les auteurs ne proposent aucune étape d'analyse des effets des changements sur la relation de compatibilité entre l'ontologie évoluée et les éléments dépendants (ontologies, instances, applications).

Le versionnement de l'ontologie permet d'accéder aux données par différentes versions de l'ontologie dont le lien entre elles doit être établi avec précision afin de pouvoir les contrôler (Klein et al., 2001) (Klein et al., 2002) (Noy et al., 2004). L'établissement du lien de passage entre versions est une tâche délicate basée sur la conservation de l'ordre<sup>1</sup> exact des versions. La méthodologie de versionnement ne permet pas de supporter le processus d'évolution de l'ontologie et se limite à la gestion des versions d'une ontologie après son évolution.

Les approches présentées sont actuellement les seules ayant des indications méthodologiques pour supporter l'évolution de l'ontologie et elles restent limitées dans la mesure où elles ne fournissent ni une méthodologie complète ni un cadre intégrateur pour les éléments qu'elles proposent.

Par ailleurs, les différentes méthodologies développées dans la littérature ont contribué au développement de plusieurs outils pour aider l'utilisateur à construire

---

<sup>1</sup> Le non respect de l'ordre exact des versions entraîne la perte de la trace et de la connaissance sur l'historique des étapes menant d'une version à l'autre.

des ontologies mais peu d'entre eux traitent l'aspect évolution. Il existe des outils dédiés à l'approche d'évolution et d'autres au versionnement (Haase et al., 2004). Parmi les outils les plus connus, nous citons KAON (Karlsruhe Ontology and Semantic Web) (Maedche et al., 2003) (Karlsruhe, 2004), OntoView (Klein et al., 2002), PromptDiff (Noy et al., 2002) (Bruijn et al., 2003), SHOE (Olivier et al., 1999) (Heflin et al., 2000) (Haase et al., 2004), OntoManager (Stojanovic et al., 2003) et TextToOnto (Maedche et al., 2001). L'étude des principaux outils met en évidence le manque des fonctionnalités pour assurer le processus d'évolution de l'ontologie. Les outils proposés ne supportent pas l'étape d'identification des changements et d'analyse des effets des changements sur les liens entre les concepts, la sémantique et la compatibilité entre les versions de l'ontologie. Pour des raisons de simplicité, nous n'allons pas détailler ces différents outils en renvoyant le lecteur intéressé aux liens bibliographiques.

Dans l'approche que nous envisageons, notre premier objectif est d'ordre méthodologique visant à identifier et mettre en œuvre un processus de gestion de l'évolution de l'ontologie. Nous nous basons sur la combinaison, dans un cadre cohérent, des deux approches d'évolution et de versionnement et l'intégration de nouvelles étapes portant essentiellement sur l'identification des besoins et l'analyse des effets dérivés des types de changements sur l'ontologie.

### 3. Problème d'évolution de l'ontologie

Les systèmes d'information étant généralement évolutifs tant au niveau des besoins qu'au niveau des données. L'utilisation d'une ontologie comme référentiel sémantique dans la conception des systèmes d'information demande, entre autres, la prise en compte de son évolution. Partant de ce postulat, nous considérons l'ontologie comme un produit non stable, nécessitant une mise à jour pour la gestion de son caractère évolutif à travers la prise en compte des changements dans le domaine, dans la conceptualisation, ou encore dans la réutilisation pour des tâches différentes.

Nous considérons deux types d'évolution de l'ontologie : une *évolution nécessaire ou obligatoire* et une *évolution optionnelle*.

L'évolution nécessaire peut être la conséquence de l'évolution de l'environnement du domaine étudié, entraînant par là une incompatibilité de l'ontologie initiale au regard des besoins ou du domaine étudié. Les données de leur part, peuvent subir des changements au niveau de leur domaine de valeurs<sup>2</sup> (changement des connaissances du domaine) ou s'associer à d'autres données<sup>3</sup> suite à l'apparition de nouvelles connaissances. Il peut s'agir également de l'élargissement du domaine de connaissance pour intégrer un autre domaine ou une

---

<sup>2</sup> Exemple, certains concepts changent de sens au cours du temps.

<sup>3</sup> Apparition de nouveaux concepts ou relations.

partie d'un domaine. Les besoins des utilisateurs peuvent aussi évoluer suite à l'apparition d'un nouveau type / profil d'utilisateurs (les utilisateurs sont différents) ou suite à l'évolution des besoins et des connaissances des mêmes utilisateurs. L'évolution de l'usage de l'ontologie, c'est à dire le changement dans le contexte ou le rôle<sup>4</sup> de l'ontologie entraîne aussi une évolution obligatoire de l'ontologie.

Le deuxième type d'évolution (l'évolution optionnelle) est généralement la conséquence de la maturité du concepteur de l'ontologie, qui peut réviser l'ontologie pour améliorer sa structure conceptuelle ou sémantique malgré qu'elle réponde toujours aux attentes des utilisateurs.

Chaque type d'évolution reflète un besoin de mise à jour de l'ontologie et peut être exprimé par un ou plusieurs types de changements. Nous distinguons les changements élémentaires et composés pouvant avoir chacun des effets différents sur la structure conceptuelle et sémantique de l'ontologie. La section suivante s'intéresse à ces types de changements ainsi que leurs conséquences sur la mise à jour de l'ontologie.

#### **4. Les types de changements et leurs effets sur l'ontologie**

L'évolution d'ontologie est un processus compliqué, dû à la variété de sources et de conséquences des changements d'évolution et nécessite la prise en compte des effets secondaires de chaque changement sur l'ontologie. L'évolution d'ontologie ne peut donc être faite manuellement et le développement d'un système automatique s'avère de grande utilité afin de maintenir l'uniformité et la cohérence de l'ontologie. Cet aspect reste peu développé dans la littérature justifiant le manque d'outils et de systèmes opérationnels supportant les différentes étapes du processus d'évolution de l'ontologie.

La complexité de l'ontologie est due à son modèle qui est très riche et sa structure entrelacée. Chaque changement peut entraîner des effets secondaires touchant des concepts non directement concernés par le changement. Même lorsque les effets d'un changement sont mineurs, l'effet cumulatif de tous les changements, réalisant une demande d'utilisateur, peut s'avérer important (Stanjanovic, 2004).

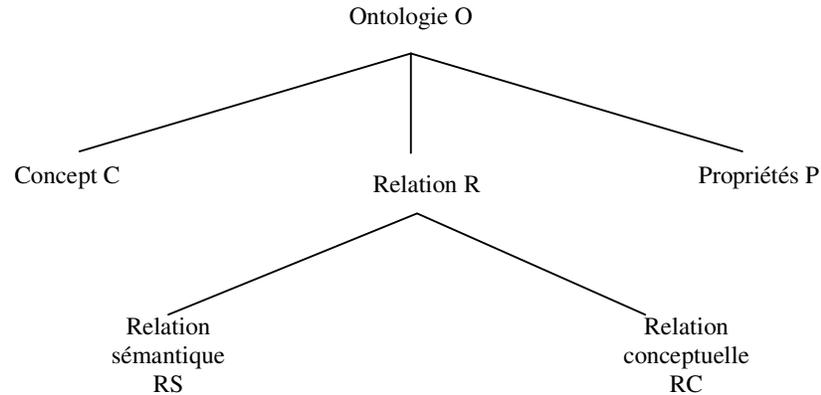
Plusieurs types de changements peuvent intervenir pour exprimer les causes d'évolution présentées dans la section précédente. L'analyse de la conséquence de ces changements nous permet de mieux prévoir les effets de chaque type sur les autres éléments de l'ontologie.

L'ontologie étant composée de concepts, de propriétés, de relations conceptuelles (relations simples, agrégation, composition, héritage) et sémantiques (synonymie, homonymie, équivalence, polysémie, ...) liant ces concepts, chaque

---

<sup>4</sup> Nous appelons rôle de l'ontologie le service assuré par l'ontologie (le but de son utilisation).

entité de l'ontologie peut être mise à jour par des transformations simples du type : ajouter et supprimer (figure 1).



**Figure 1.** Les différentes composantes d'une ontologie.

Tout besoin de changement est représenté par un type de changement (*opérateur*). Deux types de changements peuvent être identifiés : les changements simples ou élémentaires (ne peuvent pas être décomposés en changements plus simples) et les changements composés exprimant un enchaînement de plusieurs changements élémentaires formant ensemble une seule entité logique (Giorgos et al., 2006).

#### 4.1 Les changements élémentaires

Un changement élémentaire représente une opération primitive et non décomposable agissant généralement sur une seule entité de l'ontologie pour la modifier. Les changements élémentaires typiques sont l'ajout ou la suppression qui sont généralement de complexité faible.

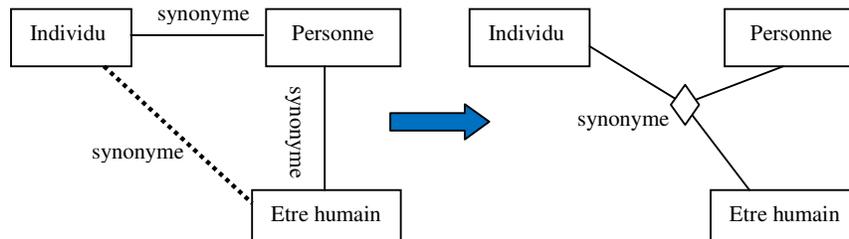
**DEFINITION 1.** — *Un changement élémentaire* est un changement de l'ontologie qui modifie (ajoute ou supprime) une seule entité du modèle de l'ontologie (Stanjanovic, 2004).

Nous identifions les types de changements élémentaires suivants :

- Add\_concept : Ajout d'un concept.
- Add\_property : Ajout d'une propriété.
- Add\_relationC : Ajout d'une relation conceptuelle.

- Add\_relationS : Ajout d'une relation sémantique.
- Copy\_concept : copier un concept.
- Remove\_concept : suppression d'un concept.
- Remove\_property : suppression d'une propriété.
- Remove\_relationC : suppression d'une relation conceptuelle.
- Remove\_relationS : suppression d'une relation sémantique.
- Rename\_concept : renommer un concept.
- Rename\_relationC : renommer une relation conceptuelle.
- Rename\_property : renommer une propriété.

Chaque type de changement peut générer des changements additionnels sur les autres parties de l'ontologie. Ces changements sont appelés changements dérivés. La prise en compte de ces changements dérivés peut nécessiter dans certains cas une optimisation de la représentation conceptuelle, ce que nous appelons les effets de bord. Exemple : l'ajout d'une relation de synonymie<sup>5</sup> entre 'Individu' et 'Etre humain' nécessite une optimisation de la représentation des relations entre les concepts : Individu, Personne et Etre humain (figure 2).



**Figure 2.** Exemple d'une optimisation de la représentation conceptuelle.

Nous présentons, dans le tableau 1, la syntaxe, la sémantique et les changements dérivés de chaque type de changement élémentaire relatif à un ajout.

Les types de changements élémentaires exprimant une suppression font l'objet d'un traitement plus spécifique. En effet, la suppression d'un concept doit être précédée, au préalable, par une analyse de ses effets sur les entités ontologiques (concepts fils, relations conceptuelles et sémantiques) liés à ce concept.

<sup>5</sup> Nous entendons, par synonyme de concepts, la synonymie entre les noms des concepts.

Syntaxe	Sémantique	Changements dérivés
Add_concept (C)	Ajouter un nouveau concept C	Ajout de nouvelles relations sémantiques en comparant les propriétés du nouveau concept C avec les propriétés des concepts déjà existants dans l'ontologie.
Add_property (C, P)	Ajouter une nouvelle propriété P au concept C	Ajout ou suppression de relations sémantiques.
Add_relationC (C1, C2, RC)	Ajouter la relation conceptuelle RC entre le concept C1 et le concept C2	Pas d'effet
Add_relationS (C1, C2, RS)	Ajouter la relation sémantique RS entre le concept C1 et le concept C2	Pas d'effet
Copy_concept <sup>6</sup> (C, Cnew)	Faire une copie du concept C dans Cnew	Le nouveau concept Cnew va être lié avec toutes les relations sémantiques et conceptuelles de C

**Tableau 1.** Types de changements élémentaires d'ajout.

La suppression d'un concept peut entraîner l'isolation d'un ou de plusieurs concepts (qui ne sont plus liés à d'autres concepts) qui doivent alors être supprimés ou reliés avec d'autres concepts. Toutes les relations conceptuelles ainsi que les relations sémantiques de types composition, agrégation, synonymie, homonymie, équivalence et polysémie reliant le concept à supprimer avec d'autres concepts sont à supprimer automatiquement de l'ontologie.

La relation d'héritage doit être étudiée avec précaution : il faut vérifier l'emplacement du concept à supprimer dans la hiérarchie des concepts :

- Cas où le concept est un concept feuille : le concept est supprimé avec toutes ses propriétés ainsi que la relation d'héritage le liant avec ses concepts super-concepts.
- Cas où le concept est au milieu d'une structure hiérarchique c'est à dire qu'il a

<sup>6</sup> Ce changement élémentaire est nécessaire pour effectuer certains changements composés (c.f. tableau 3).

des sous concepts et des super concepts : deux cas se présentent. (1) Supprimer toutes les relations d'héritage de ce concept avec ses super-concepts et ses sous-concepts. Les propriétés du concept sont conservées pour être immigrées vers ses sous-concepts en liant le sous arbre des sous-concepts avec tous ses super-concepts, (2) Supprimer le concept avec tous ses sous arbres.

– Cas où le concept à supprimer est au niveau le plus haut dans la hiérarchie. La suppression du concept s'accompagne par la suppression de toutes les relations d'héritage entre ce concept et ses sous-concepts avec propagation de ses propriétés vers ses sous-concepts.

Syntaxe	Sémantique	Changements dérivés
Remove_concept (C)	Supprimer le concept C avec toutes ses propriétés.	Supprimer les relations conceptuelles et sémantiques qui lient ce concept avec les autres concepts.
Remove_property (C, P)	Supprimer la propriété P du concept C.	Ajout ou suppression de relations sémantiques.
Remove_relationC (RC)	Supprimer la relation conceptuelle RC.	Pas d'effets.
Remove_relationS (RS)	Supprimer la relation sémantique RS.	Pas d'effets.
Rename_concept (C, nom_concept)	Modifier le nom du concept C par le nouveau nom nom-concept.	Suppression ou ajout de relations sémantiques d'homonymie, d'équivalence, polysémie et synonymie.
Rename_relationC (ancien-nom RC, nouveau_nom RC)	Modifier le nom d'une relation conceptuelle sans toucher à sa sémantique.	Pas d'effet
Rename_property (C, PA, PN)	Remplacer le nom de la propriété PA par le nouveau nom PN.	Ajout ou la suppression de relations sémantiques.

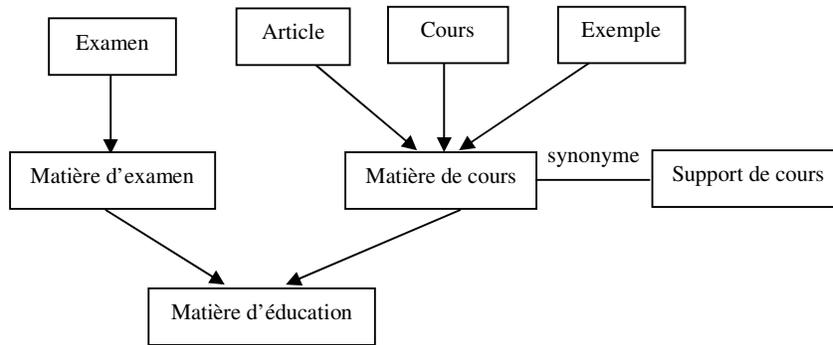
**Tableau 2.** Types de changements élémentaires de suppression.

Nous présentons, dans le tableau 2, la syntaxe, la sémantique et les changements

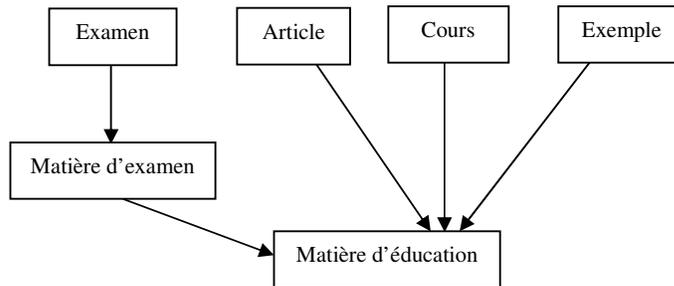
dérivés de chaque type de changement élémentaire relatif à une suppression.

Nous précisons que les types de changements `Remove_concept`, `Remove_relationC` et `Remove_relationS` peuvent entraîner des effets de bord comme l'apparition de concepts isolés qui doivent être pris en charge avec prudence.

Nous considérons l'exemple de la figure 3 (issu du domaine du e-Learning). Supposons qu'on n'a plus besoin du concept 'matière de cours' qui est un sous concept de 'matière d'éducation' et un super concept des concepts 'article', 'cours' et 'exemple'. La suppression du concept 'matière de cours' entraîne la suppression des relations d'héritage avec ses sous concepts et avec son super concept. La propagation des changements entraîne de nouvelles relations d'héritage entre les sous concepts de 'matière de cours' et son super concept. La relation de synonymie avec le concept 'support de cours' doit être aussi supprimé. Après application des changements dérivés, le concept 'support de cours' sera isolé et doit donc être supprimé. La figure 4 représente l'ontologie résultat après application du type de changement `Remove_concept` (Matière de cours).



**Figure 3.** Exemple d'une ontologie issu du domaine e-Learning.



**Figure 4.** Exemple d'application d'un changement élémentaire de suppression.

### 5.1 Les changements composés

Les changements élémentaires identifiés précédemment ne suffisent pas à eux seuls d'exprimer le niveau d'expression des besoins d'évolution. Souvent, l'intention des changements peut être exprimée à un niveau d'abstraction plus élevé, notamment par les utilisateurs dont le niveau d'expression ne reflète pas forcément l'enchaînement des traitements effectués au niveau système. Par exemple, si nous avons besoin de créer un super-concept commun à deux concepts C1 et C2, nous devons passer par les étapes suivantes :

1. Créer un nouveau concept Cnew ;
2. Définir le concept Cnew comme super-concept du concept C1 ;
3. Définir le concept Cnew comme super-concept du concept C2 ;
4. Créer une relation d'héritage entre C1 et Cnew ;
5. Créer une relation d'héritage entre C2 et Cnew.

Dans cet exemple, le type de changement 'créer un super-concept commun à deux concepts C1 et C2' est décomposé en une séquence d'opérations représentant chacune un changement élémentaire. Il est donc considéré comme un changement composé.

**DEFINITION 2.** — *Un changement composé* est un changement de l'ontologie qui peut être décomposé en plusieurs changements élémentaires (Stanjanovic, 2004).

Nous identifions les types de changements composés suivants :

- Fusion\_concepts : Fusion de concepts.
- Fusion\_relations : Fusion de relations liant les mêmes concepts.
- Fusion\_property : Fusion de propriétés.
- Split\_Concepts : Eclatement de concepts.
- Split\_relations : Eclatement de relations.
- Move\_up : Déplacement d'un concept vers le haut de la hiérarchie.
- Move\_down : Déplacement d'un concept vers le bas de la hiérarchie.
- Group\_concepts : Créer un super-concept commun entre deux concepts.

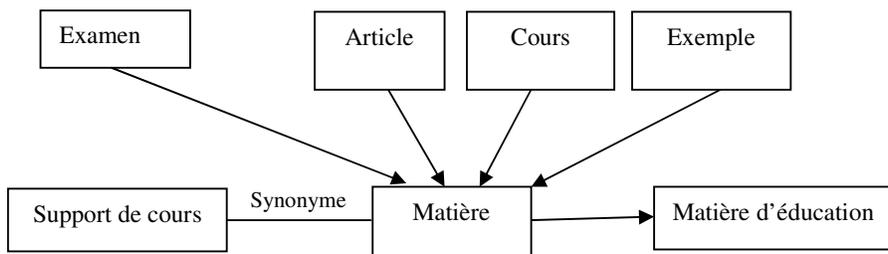
Syntaxe	Sémantique	Changements dérivés	Liste de changements élémentaires
Fusion_concepts (C1, C2, Cnew)	Fusionner les deux concepts C1 et C2 dans un seul concept Cnew.	Le nouveau concept Cnew prend comme propriétés l'ensemble des propriétés de C1 et C2 en éliminant les redondances et sera relié	- Ajouter un nouveau concept Cnew. - Ajouter à Cnew toutes les propriétés de C1 et C2. - Lier Cnew avec toutes

		par les relations conceptuelles et sémantiques qui relient les deux concepts C1 et C2.	les relations liant C1 et C2. - Supprimer C1 et C2.
Fusion_relations (R1, R2, Rnew)	Fusionner les relations R1 et R2 dans la même relation Rnew.	Les seules relations traitées ici sont les relations conceptuelles simples. Ce changement n'a pas d'effet sur la structure de l'ontologie.	- Ajouter une nouvelle relation Rnew. - Ajouter à Rnew toutes les propriétés de R1 et R2. - Lier R1 avec tous les concepts qui lient R1 et R2. - Supprimer R1 et R2.
Fusion_property (C, P1, P2, Pnew)	Fusionner dans le concept C les propriétés P1 et P2 dans une même propriété Pnew.	Ce changement n'a pas d'effet sur la structure de l'ontologie.	- Ajouter une nouvelle propriété Pnew à C. - Supprimer P1 et P2 de C.
Split_Concepts (C, C1new, C2new)	Dupliquer le concept C en deux concepts C1new et C2new en distribuant les propriétés de C entre eux.	Les concepts C1new et C2new seront liés par les mêmes relations liant C aux autres concepts.	- Créer deux nouveaux concepts C1new et C2new. - Distribuer les propriétés de C entre C1new et C2new. - Ajouter les relations liées par C à C1new et C2new. - Supprimer C.
Split_relations (R, R1new, R2new)	Dupliquer la relation R en deux relations R1new et R2new en distribuant les propriétés de R entre eux	Les relations R1new et R2new seront liés par les mêmes concepts que R.	- Créer deux nouvelles relations R1new et R2new. - Distribuer les propriétés de R entre R1new et R2new. - Lier R1new et R2new par les concepts de R. - Supprimer R.
Move_up (C)	Déplacer le concept C vers le haut de sa structure hiérarchique.	Attacher le concept C à tous les super-concepts de tous ses super-concepts.	- Copier le concept C à Cnew. - Attacher le concept Cnew aux super-concepts de tous les

			super-concepts de C. - Supprimer C. - Renommer Cnew par C.
Move_down (C)	Déplacer le concept C vers le bas de sa structure hiérarchique.	Attacher le concept C aux sous-concepts de tous ses super-concepts à l'exclusion de lui même.	- Copier le concept C à Cnew. - Attacher le concept Cnew aux sous-concepts de tous les super-concepts de C. - Supprimer C. - Renommer Cnew par C.
Group_concepts (C1, C2, C)	Créer un super-concept commun C pour les concepts C1 et C2	Ajout d'une nouvelle relation d'héritage.	- Créer un nouveau concept C. - Ajouter les propriétés communes de C1 et C2 à C. - Ajouter la relation d'héritage entre C1 et C2 et C.

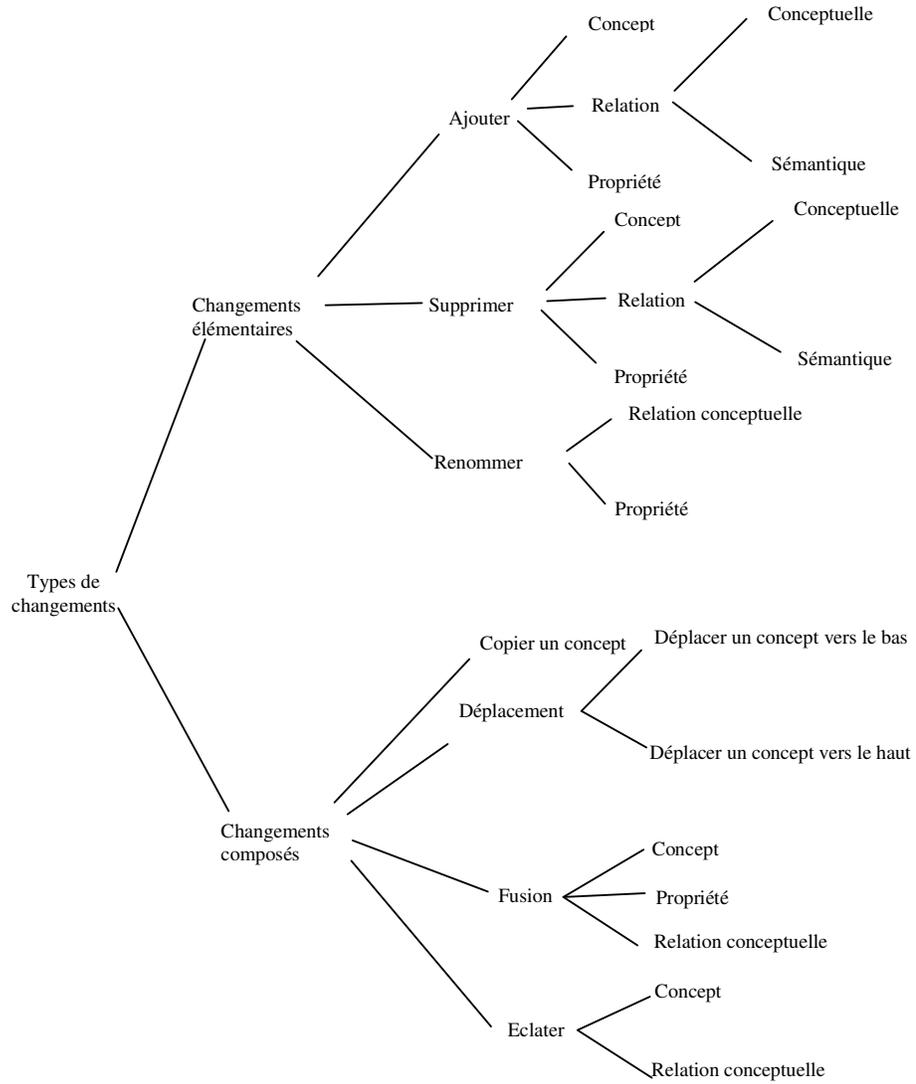
**Tableau 3.** Types de changements composés.

Nous partons toujours de l'ontologie de la figure 3. Supposons maintenant qu'on veut fusionner les concepts 'matière d'examen' et 'matière de cours' dans un seul concept nommé 'matière'. Nous commençons alors par l'ajout d'un nouveau concept 'matière' et le lier avec toutes les relations liant 'matière d'examen' et 'matière de cours' avec les autres concepts. Nous supprimons ensuite les deux concepts pour obtenir l'ontologie de la figure 5.



**Figure 5.** Exemple d'application d'un changement composé.

Nous présentons, dans la figure 6, la taxonomie des types de changements utilisés dans le processus d'évolution de l'ontologie.



**Figure 6.** Taxonomie des types de changements.

Les changements composés sont plus puissants puisqu'un concepteur d'ontologie n'a pas besoin de passer par chaque étape de la séquence des changements élémentaires pour réaliser le changement désiré. Par conséquent, les changements composés rendent l'évolution d'ontologie beaucoup plus facile, plus rapide et plus efficace. Ils sont plus facilement utilisables et compréhensibles car leur intention est explicite, contrairement à une suite de changements élémentaires ayant le même résultat. Ils permettent l'évolution de l'ontologie avec moins de pertes des données (Klein et al., 2003). Par exemple, le changement composé *Move\_up* (C) préserve les données instances, contrairement à la suite des changements élémentaires *Remove\_concept* (C) et *Add\_concept* (C).

## 5. Analyse des conséquences de l'évolution d'une ontologie

Nous avons vu précédemment les types de changements et leurs conséquences sur les éléments de l'ontologie. L'application de ces types de changements est susceptible d'apporter des modifications sur les instances et la structure conceptuelle de l'ontologie en passant d'une version VN à une version VN+1 évoluée. Nous étudions dans cette section ces modifications selon deux dimensions : la préservation des instances et la préservation de la structure conceptuelle.

**REMARQUE** — Nous considérons les instances comme étant indépendantes de l'ontologie et donc ne faisant pas partie des éléments de l'ontologie. A cet effet, les opérations de changements appliquées sur les instances n'étaient pas listées dans la liste des types de changements (tableaux 2 et 3).

### 5.1 Préservation des instances

La préservation des instances signifie qu'aucune instance n'est perdue lors du processus d'évolution de l'ontologie. Trois cas se présentent :

- Préservation sans ajout d'instances (**Ch<sub>pt</sub>**) : L'ensemble des instances de VN est le même que celui de VN+1. Ce cas peut apparaître suite à une fusion de concepts ou de relations, l'éclatement de concepts<sup>7</sup>, le déplacement (de concept, de propriété ou de relations), la suppression de propriété, l'ajout de propriété ou de relations.

- Préservation avec ajout d'instances (**Ch<sub>pat</sub>**) : L'ensemble des instances de VN

---

<sup>7</sup> L'éclatement d'un concept s'effectue en fonction de la valeur d'un ou plusieurs attributs caractérisant ce concept. Par exemple, l'éclatement du concept *Personne* en deux sous concepts *Etudiant* et *Enseignant* dépend de la valeur que prend l'attribut *profession* de la personne (qui peut prendre la valeur enseignant ou étudiant). Par conséquent, les instances de personne seront, après éclatement, réparties entre les deux nouveaux concepts *Etudiant* et *Enseignant*. La préservation des instances est vérifiée si l'attribut *profession* ne peut prendre que deux valeurs possibles : enseignant ou étudiant. Autrement, il y a perte des instances dont la valeur de l'attribut *profession* n'est ni enseignant ni étudiant.

est élargi dans VN+1. Ce cas peut apparaître suite à un ajout de concept ou de relation.

- Non préservation des instances ( $\mathbf{Ch}_{nPI}$ ) : L'ensemble des instances de VN est réduit ou modifié dans VN+1. Ce cas peut apparaître suite à une suppression de concept ou de relation (si la relation a des instances).

## 5.2 Préservation de la structure conceptuelle

La préservation de la structure conceptuelle signifie qu'aucun des éléments conceptuels<sup>8</sup> qui composent le vocabulaire de l'ontologie n'est perdu lors de l'évolution de l'ontologie. L'application d'un ou de plusieurs changements sur l'ontologie peut engendrer différents cas :

- Préservation de la structure conceptuelle sans ajout ( $\mathbf{Ch}_{PS}$ ) : L'ensemble des éléments conceptuels de VN est le même que celui de VN+1. Ce cas peut apparaître suite à la suppression de propriété ou le déplacement de concept etc.

- Préservation de la structure conceptuelle avec ajout de concept, de relation ( $\mathbf{Ch}_{PaS}$ ) : L'ensemble de éléments conceptuels de VN est élargi dans VN+1. Ce cas peut apparaître suite à un ajout de concept ou de relation.

- Non préservation de la structure conceptuelle ( $\mathbf{Ch}_{nPS}$ ) : L'ensemble des concepts de VN est réduit ou modifié dans VN+1. Ce cas peut apparaître suite à une fusion de concepts ou de relations, l'éclatement de concepts ou de relations<sup>9</sup>, la suppression de concept ou de relation.

En fonction des conséquences des changements sur l'ontologie, nous pouvons classer les changements exécutés pour passer de VN à VN+1 en 3 catégories :

– **Changement préserveur** : l'application de ce type de changement préserve les instances (aucune instance n'est perdue lors du processus d'évolution de l'ontologie) et la structure conceptuelle (aucun des concepts/relations qui composent le vocabulaire de l'ontologie n'est perdu lors de l'évolution de l'ontologie). Ces changements préservent donc entièrement les rôles de l'ontologie et peuvent être exprimés comme suit :

$$\mathbf{Ch}_P = \mathbf{Ch}_{PI} \cap \mathbf{Ch}_{PS}$$

– **Changement incrémental** : l'application de ce type de changement entraîne l'élargissement dans VN+1 d'au moins un ensemble parmi celui des instances ou éléments conceptuels de VN sans les réduire ou les modifier.

$$\mathbf{Ch}_I = (\mathbf{Ch}_{PaI} \cup \mathbf{Ch}_{PaS}) - (\mathbf{Ch}_{nPI} \cup \mathbf{Ch}_{nPS})$$

<sup>8</sup> Concepts et relations.

<sup>9</sup> Si, lors de l'éclatement d'un élément ontologique on perd l'original. Par contre, si on garde l'élément original et on crée un lien avec les concepts nouveaux, il s'agit plutôt d'un cas de préservation avec ajout.

– **Changement restructurant** : l'application de ce type de changement entraîne la réduction ou la modification (non incrémentale) dans VN+1 d'au moins un ensemble parmi celui des instances ou éléments conceptuels de VN. Ces changements ne préservent pas les rôles de l'ontologie.

$$\mathbf{Ch_R} = \mathbf{Ch_{nPI}} \cup \mathbf{Ch_{nPS}}$$

## 6. Conclusion et perspectives

Nous avons étudié dans ce papier les types de changements pouvant intervenir dans les environnements dynamiques et nécessitant une mise à jour de l'ontologie. Nous avons présenté des types de changements élémentaires et composés ainsi que leurs effets sur la préservation des instances et de la structure conceptuelle de l'ontologie. Cette tâche fait partie d'une méthodologie générale développée pour le suivi de l'évolution de l'ontologie et basée sur la création d'une nouvelle version mieux adaptée aux changements intervenants. La nouvelle version est une mise à jour de l'ontologie initiale et elle est caractérisée par une durée de validité se terminant suite à l'apparition d'un nouveau changement. Nous partons donc d'une version existante de l'ontologie, construite au préalable par le concepteur. Un lien de dérivation doit être introduit entre les versions de l'ontologie afin de suivre son évolution et de garder une trace sur l'historique des versions. La version la plus récente est appelée version courante, toutes les autres versions sont des versions historiques.

Cependant, la conservation d'un grand nombre de versions, notamment dans les environnements très évolutifs, devient très vite pénalisante (coût de mémoire et quantité d'informations manipulées). Nous orientons notre réflexion vers la conservation des versions les plus pertinentes. La pertinence reflète l'importance de l'information et de la sémantique contenue dans une version de l'ontologie. Lorsque le nombre de versions devient très important, un mécanisme doit être associé pour réorganiser les différentes versions en supprimant celles qui ne sont plus pertinentes. La suppression d'une version de l'ontologie est traduite par la suppression de tous ses éléments conceptuels qui ne sont pas propagées aux autres versions de l'ontologie. La liste des versions conservées à un moment donné représente une liste dont le premier élément est la version originale et le dernier est la version courante. Le reste des éléments représente les versions les plus pertinentes ordonnées chronologiquement.

Nos futurs travaux concernent donc trois axes principaux :

- Identifier et contrôler les différentes versions de l'ontologie en analysant les liens de passage entre elles.
- Déterminer les critères adéquats pour l'évaluation de la pertinence de chaque

version de l'ontologie.

– Développer un système d'aide à la conception des systèmes d'information intégrant les utilisateurs dans les phases d'expression et de validation des changements.

## 7. Bibliographie

- Bruijn J., Martin-Recuerda F., Manov D., Ehrig M., « *State-of-the-art survey on Ontology Merging and Aligning* », EU-IST Integrated Project (IP) IST-2003-506826, 2003.
- FZI Karlsruhe and AIFB Karlsruhe, « *KAON the Karlsruhe ontology and semantic web* », Framework-developer's guide for KAON 1.2.7, 2004.
- Giorgos F., Plexousakis D., « *Handling Ontology Change: Survey and Proposal for a Future Research Direction* », Rapport technique FORTH-ICS/TR-362, 2005.
- Giorgos F., Dimitris P., Grigoris A., « *Evolving Ontology Evolution* », Proceedings of the 32<sup>nd</sup> International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 06), 2006.
- Haase P., Sure Y., « *State of the Art on Ontology Evolution* », 2004. Web: <http://www.aifb.uni-karlsruhe.de/WBS/ysupublications/SEKT-D3.1.1.b.pdf>
- Heflin J., Hendler J., « *Dynamic Ontology on the Web* », Proceedings of the Seventeenth National Conference on Artificial Intelligence AAAI/MIT, pp. 443-449. CA, 2000.
- Klein M., Noy N., « *A component-based framework for the ontology evolution* », Workshop on Ontologies and Distributed Systems, IJCAI-2003, Mexico, 2003.
- Klein M., « *Versioning of distributed ontologies* », EU/IST Project WonderWeb, 2002.
- Klein M., Fensel D., « *Ontology versioning for the Semantic Web* », International Semantic Web Working Symposium, USA, 2001.
- Maedche A., Motik B., Stojanovic L., « *Managing Multiple and Distributed Ontologies in the Semantic Web* », VLDB Journal - Special Issue on Semantic Web, 12, 286-302, 2003.
- Maedche A., Volz R., « *The ontology extraction and maintenance framework text-to-onto* », In Proceedings of the ICDM'01 Workshop on Integrating Data Mining and Knowledge Management, 2001.
- Mhiri M., Mtibaa A., Gargouri F., « *Towards an approach for building information systems' ontologies* », FOMI'2005, 1st workshop Formal Ontologies Meet Industry, 9-10 June, Verona, Italy, 2005.
- Noy N., Klein, M., « *Ontology evolution: Not the same as schema evolution* », Knowledge and Information Systems, 6(4):428-440, 2004.
- Noy N., Musen M., « *PROMPTDIFF: A fixed-point algorithm for comparing ontology version* », 18<sup>th</sup> National Conference on Artificial Intelligence, Canada, 2002.
- Olivier D., Shahar E., Musen Y., Shortliffe M., « *Representation of change in controlled medical terminologies* », AI in Medicine, 15(1), 53-76, 1999.

Stanjanovic L., « Methods and Tools for Ontology Evolution », Thèse de doctorat de l'Université de Karlsruhe, 2004.

Stanjanovic, L., Stojanovic, N., Handschuh, S., « *Evolution of the Metadata in the Ontology based Knowledge Management Systems* », Experience Management, Germany, 2002.

Stojanovic N., Hartmann J., Gonzalez J., « *Ontomanager - a system for usage-based ontology management* », Proceedings of FGMLWorkshop, 2003.