

1. Introduction

Dans [8], l'évolution des ontologies est définie comme l'adaptation – dans le temps – d'une ontologie aux besoins de changement et la propagation cohérente des changements aux artefacts dépendants. Dans le contexte de l'ingénierie ontologique, la maintenance évolutive des ontologies s'intéresse aux différentes opérations effectuées sur les éléments de l'ontologie.

Nous proposons une approche basée en phase intra-développement sur (1) l'ingénierie inverse ou restructuration de chaque version de l'ontologie selon un modèle d'ontologie lexicale, (2) l'analyse préalable des relations et des sens de conductivité d'impact selon le modèle et (3) l'établissement des règles de propagation d'impact. Ce qui permet à chaque changement détecté de pouvoir effectuer une prédiction du chemin de propagation des flux d'impacts de modification dans l'ontologie. Elle est ainsi basée sur l'hypothèse selon laquelle une prise en compte générique et une bonne maîtrise des impacts d'évolution sur un système passe par une connaissance préalable des possibilités d'évolution dans le système, les sens des relations conductrices d'impact entre les composants de ce système. Nous associons ainsi à chaque phase de l'évolution d'une ontologie une couche supplémentaire modélisant les possibilités d'évolution. Cela permet de vérifier plus facilement si une ontologie est cohérente par rapport à son modèle, ce qui le cas si et seulement si, elle satisfait les contraintes spécifiées par ce modèle [10]. Il suffit ainsi de vérifier selon les contraintes d'équilibres posées au préalable dans le modèle si le nouvel état du système est cohérent par rapport au modèle de référence. Cela souligne l'importance de disposer d'un tel un modèle. Un état de l'art des approches gestion de l'évolution des ontologies est présenté dans la deuxième section.

2. Evolution d'ontologie

Avec l'avènement du web sémantique, il existe un grand intérêt et un renouveau dans la recherche sur la maintenance évolutive des ontologies. Comme énoncées dans [3], les questions essentielles dans ce contexte sont (1) de pouvoir détecter les changements ainsi que leur conséquence sur le système, (2) détecter les inconsistances générées à cause des changements ontologiques et enfin (3) résoudre les inconsistances.

L'analyse des impacts de modification peut être définie comme l'ensemble des activités visant à maîtriser les effets d'un changement donné. Ces changements peuvent être considérés comme provoquant les effets d'ondes ou le phénomène par lequel des modifications d'un composant entraînent des effets sur d'autres parties de l'ontologie. Il existe plusieurs approches de gestion de l'évolution des ontologies [10][5][1][9][7] et

[4] qui peuvent être caractérisées comme étant de type « what-if ». Dans [4] par exemple, le contrôle de l'évolution est basé sur la définition de patrons de gestion de changement permettant d'avoir une « bibliothèque » de scénarios de changement possible. Ainsi à la détection de chaque type de changement il est possible de proposer une procédure à suivre pour contrôler ses impacts et résoudre les inconsistances éventuelles. Dans [9], l'auteur propose une méthodologie d'évolution des ontologies basée sur la définition d'un langage de changement et un log de l'ontologie. Il est ainsi défini, dans le langage, plusieurs opérations de changements en termes de conditions devant être remplies avant et après l'opération de changement demandée. Dans [10] et de [5], on retrouve aussi le même type d'approche consistant à étudier avec la représentations OWL et RDF l'ensemble des éléments pouvant subir des impacts de modification résultant des conséquences de l'évolution d'un autre élément. Dans [1] et [7] on retrouve également une approche de gestion globale de l'évolution. L'un des problèmes posés par ces types d'approches est que l'analyse ne fonctionne que pour l'ensemble des scénarios définis, le système sera incapable de contrôler tous les effets de bords d'une modification non pris en compte. En plus, ce n'est pas impossible mais il est complexe de prévoir tous les scénarios de modification possible et effectuer un raisonnement logique pour un graphe ontologique dont les nœuds et les relations peuvent se compter en millions. De plus il conviendra, dans le cas d'espèces, de résonner sur des tests de satisfiabilités basés souvent sur la logique de description pour la vérification de la cohérence, ce qui est assez complexe pour les systèmes logiques. D'ailleurs, comme souligné dans [6], les problèmes de satisfiabilités pour certaines catégories de LDs sont dans le pire cas du problème au moins NP-difficile et EXPTIME-complet pour les LDs les plus expressives.

L'approche que nous proposons permet dans le contexte ontologique, en modélisant l'ontologie comme représentant un graphe orienté, de considérer que toute modification d'un nœud peut rendre inconsistant ses voisins. La propagation des impacts consiste dans ce cas à marquer, de proche en proche, les nœuds inconsistants. Ainsi chaque évolution d'un nœud donne le signal à un contrôle pour détecter les inconsistances sur ces voisins afin de les marquer au besoin. En plus, nous substituons aux problèmes de satisfiabilités ceux de recherche d'occurrence dans le modèle de restructuration lexical.

2.1. Structure d'ontologie lexicale ou à base lexicale

Dans les parties précédentes, nous avons argumenté et présenté les travaux qui vont dans le sens de la nécessité de disposer d'un modèle de l'évolution et de l'ontologie. Nous avons choisi d'utiliser les structures d'ontologies légères pour cela. Un modèle d'ontologie lexicale est formalisé comme la structure $O = \langle S, L \rangle$ où S est la structure et L le niveau lexical. Ainsi, la structure d'une ontologie O à laquelle est associée un lexique est le n-uplet $S := \{C, R, A, T, CAR_R, H^C, \sigma_R, \sigma_{CARR}, \sigma_A, \sigma_T\}$ où :

- C, A, T, CAR_R sont respectivement les ensembles contenant, les concepts de l'ontologie, les relations d'attribut, les types d'attribut et l'ensemble des caractéristiques des relations associatives que sont entre autres la synonymie, la transitivité, la symétrie et l'inversion et fonctionnelle;
- $R \subseteq (C \times C)$ est l'ensemble des relations associatives. Il permet de définir les types de relations sémantiques reliant les concepts de l'ontologie dans $(C \times C)$;
- H^C hiérarchie (taxonomie) de concepts : $H^C \subseteq (C \times C)$, $H^C(C_i, C_j)$ signifie que C_i est un sous-concept de C_j , pour les relations de subsomption entre les concepts de l'ontologie;
- $\sigma_R : R \rightarrow C \times C$ est la signature d'une relation associative. Nous noterons $\sigma_R(C_i, R_k, C_j)$ la signature de la relation associative R_k entre les concepts C_i et C_j ;
- $\sigma_A : A \rightarrow C \times T$ est la signature d'une relation d'attribut, T est composé des types simples. Elle est notée $\sigma_A(C_i, A_k, T_j)$ spécifiant la relation d'attribut entre un concept C_i et un attribut A_k ayant des valeurs de type T_j ;
- $\sigma_T : A \rightarrow T$ est la signature de la relation associant à un attribut A_k , le type T_j sous la forme $\sigma_T(A_k, T_j)$ spécifiant que l'attribut A_k est associé à des valeurs de type T_j ;
- $\sigma_{CARR} : R \rightarrow CAR_R$ est la relation spécifiant la caractéristique d'une relation associative. Nous allons ainsi noter une relation associative R_k transitive par la signature $\sigma_{CARR}(R_k, Trans)$.

Pour une structure d'ontologie S , un lexique L est défini comme le quadruplet:

$$L := \{L^C, L^R, F, G\} \text{ où :}$$

- L^C est l'ensemble dont les éléments constituent l'entrée lexicale des concepts ;
- L^R est l'ensemble dont les éléments constituent l'entrée lexicale des relations ;
- $F \rightarrow L^C$, la relation de référence pour les concepts tels que :
 - $\forall l_c \in L^C : F(l_c) = \{c \in C / (l_c, C) \in F\}$ et $\forall c \in C : F^{-1}(c) = \{l_c \in L^C / (l_c, c) \in F\}$
- $G \rightarrow L^R$, la relation de référence pour les relations telles que :
 - $\forall l_r \in L^R : G(l_r) = \{r \in R / (l_r, R) \in G\}$ et $\forall r \in R : G^{-1}(r) = \{l_r \in L^R / (l_r, r) \in G\}$

Nous pouvons ainsi distinguer cinq composants pouvant être cibles des modifications et cinq relations pouvant propager les modifications aux autres éléments. Remarquons qu'il est essentielle pour connaître le sens du flux d'avoir une connaissance à priori du sens de propagation des impacts pour chaque type de relation. La relation de subsomption entre deux classes C_i et C_j tel que $H^C(C_i, C_j)$ induit que C_i utilise la classe C_j et par conséquent toute modification de C_i n'influence pas à priori C_j mais toute modification de C_j à des conséquences sur C_i . Nous noterons dans ce cas que toute opération Δ de modification $\Delta(C_j) \mapsto \Delta(C_i)$. L'étude des différentes relations sémantiques proposées dans l'ensemble R en corrélation avec le langage OWL, nous permet d'avancer le *postulat* selon lequel on peut supposer que pour la plus-part des relations sémantiques définies telles que $\sigma_R(C_i, R_k, C_j)$ sont conductrices d'impact de modification dans le sens de C_j à C_i puisque dans de tel cas, à l'exception des relations d'opposition et d'équivalence c'est la classe C_i qui utilise C_j . Ainsi, nous avons un sens du flux défini tel que $\Delta(C_j) \mapsto \Delta(C_i)$. Nous allons dans la phase de restructuration de

l'ontologie veiller à ce que les relations soient composées selon ce modèle. Pour la relation d'attribut entre un concept de l'ensemble C_i et un attribut A_i de l'ensemble A tel que $\sigma_A(C_i, A_i, T_k)$, nous pouvons distinguer trois cas de figures possibles. Il est évident qu'une modification du concept C_i n'a pas d'influence sur l'attribut A_i ni son type T_k . Cependant une modification de l'attribut A_i entrainera des conséquences sur le concept C_i qui l'utilise, donc $\Delta(A_i) \mapsto \Delta(C_i)$. Il est aussi trivial qu'une opération sur le type entraînant des effets de bord aura des conséquences sur l'attribut et donc aussi sur le concept. Ainsi de manière indirecte une opération de modification sur un type entraînant des propagations d'impact de proche en proche $\Delta(T_k) \mapsto \Delta(A_i) \mapsto \Delta(C_i)$.

2.2. Définition et détection des changements

Il serait difficile, voir impossible d'aborder la problématique du contrôle de l'évolution d'une ontologie sans le constat d'un changement de sa structure. Il est donc nécessaire avant de développer une théorie sur les effets d'une opération donnée de pouvoir au préalable constater qu'une opération d'une nature donnée s'est produite.

L'évolution est modélisée par une fonction de transformation Δ entre deux phases ϕ_i et ϕ_j par la fonction de translation inter-phase $T_{ij} : \Delta(\phi_i) \mapsto \Delta(\phi_j)$. Le passage d'une phase ϕ_i à une autre ϕ_j est la conséquence d'une opération donnée sur un des composants. En générant un effet d'onde, on peut de manière imagée dire que le système était équilibré, l'opération l'a déséquilibré et les effets d'onde doivent être contenus et contrôlés pour un retour à l'état d'équilibre. A priori, nous pouvons définir trois ensembles d'assertions essentielles pour la réalisation d'une opération et le contrôle de la portée de ses effets d'onde :

- Ce qu'il faut obligatoirement pour que l'opération soit réalisée, par exemple avant de supprimer un concept il faut que ce concept existe ;
- Il y a des conditions devant être vérifiées après exécution de l'opération et qui participent à sa validation, par exemple dans le cas d'une suppression d'un concept le concept ne devra plus exister dans la structure de l'ontologie;
- Des conditions devant être vérifiées avant et après l'opération. Par exemple dans le cas de la suppression d'un concept, il ne devrait exister et subsister aucune forme de relation entre la classe cible et une autre classe ou attribut.

En fonction du type de composant cible et de la nature de l'opération, les ensembles d'assertions permettent de savoir s'il y a inconsistance et dans le cas échéant propager le flux d'impact aux composants concernés. La détection d'un changement correspondra ainsi à une opération et la nécessité de propagation des impacts de modification lorsque qu'une des conditions des assertions n'est pas satisfaite. Un changement est modélisé comme le triplet $\Delta : = \langle Op, args, Assert \rangle$ représentant l'opération, ses arguments et $Assert := \langle pre, inv, post \rangle$ les pré-conditions et post-conditions et les conditions invariantes.

Tableau 1: Assertions de base et générales

Id	Assertion	Signification
1	+C _i	$\exists (C_i \in C)$
2	+R _i	$\exists (R_i \in R)$
3	+A _i	$\exists (A_i \in A)$
4	+T _i	$\exists (T_i \in T)$
5	+CAR _{Ri}	$\exists (CAR_{Ri} \in CAR_R)$
6	-C _i	$\neg (C_i \in C)$
7	-R _i	$\neg (R_i \in R)$
8	-A _i	$\neg (A_i \in A)$
9	-T _i	$\neg (T_i \in T)$
10	-CAR _{Ri}	$\neg (CAR_{Ri} \in CAR_R)$
11	+H ^C (C _i , C _j)	$\exists (C_i \in C \wedge C_j \in C) / H^C(C_i, C_j)$
12	+H ^C (C _i , C _j)	$\exists (C_i \in C \wedge C_j \in C) / H^C(C_i, C_j)$
13	+σ _R (C _i , R _k , C _j)	$\exists (C_i \in C, C_j \in C \wedge R_k \in R) / \sigma_R(C_i, R_k, C_j)$
14	+σ _{CARR} (C _i , CAR _{Ri})	$\exists (C_i \in C \wedge CAR_{Ri} \in CAR_R) / \sigma_{CARR}(C_i, CAR_{Ri})$
15	+σ _T (A _i , T _k)	$\exists (A_i \in A \wedge T_k \in T) / \sigma_T(A_i, T_k)$
16	+σ _A (C _i , A _j , T _k)	$\exists (C_i \in C, A_j \in A \wedge T_k \in T) / \sigma_A(C_i, A_j, T_k) \wedge \sigma_A(A_i, T_k)$
17	-H ^C (C _i , C _j)	$\forall (C_i \in C) \wedge \forall (C_j \in C) : \neg H^C(C_i, C_j)$
18	-H ^C (C _i , C _j)	$\forall (C_i \in C) \wedge \forall (C_j \in C) : \neg H^C(C_i, C_j)$
19	-σ _R (C _i , R _k , C _j)	$\forall (C_i \in C) \wedge \forall (C_j \in C) \wedge \forall (R_k \in R) / \neg \sigma_R(C_i, R_k, C_j)$
20	-σ _{CARR} (C _i , CAR _{Ri})	$\forall (C_i \in C) \wedge \forall (CAR_{Ri} \in CAR_R) / \neg \sigma_{CARR}(C_i, CAR_{Ri})$
21	-σ _T (A _i , T _k)	$\forall (A_i \in A) \wedge \forall (T_k \in T) / \neg \sigma_T(A_i, T_k)$
22	-σ _A (C _i , A _j , T _k)	$\forall (C_i \in C) \wedge \forall (A_j \in A) \wedge \forall (T_k \in T) / \neg \sigma_A(C_i, A_j, T_k) \wedge \sigma_A(A_i, T_k)$
23	-H ^C (* , C _i)	$\forall (C_k \in C) : \neg H^C(C_k, C_i)$
24	-H ^C (C _i , *)	$\forall (C_k \in C) : \neg H^C(C_i, C_k)$
25	-σ _{CARR} (R _i , *)	$\forall (CAR_{Ri} \in CAR_R) : \neg \sigma_{CARR}(R_i, CAR_{Ri})$
26	-σ _T (A _i , *)	$\forall (T_k \in T) : \neg \sigma_A(A_i, T_k)$
27	-σ _A (* , A _i , T _j)	$\forall (C_k \in C) \wedge \forall (A_j \in A) \wedge \forall (T_j \in T) / \sigma_T(A_i, T_j) \wedge \neg \sigma_A(C_k, A_i, T_j)$
28	-σ _A (* , A _i , *)	$\forall (C_k \in C) \wedge \forall (T_j \in T) : \neg \sigma_A(C_k, A_i, T_j)$
29	-σ _A (C _k , * , *)	$\forall (A_i \in A) \wedge \forall (T_j \in T) : \neg \sigma_A(C_k, A_i, T_j)$
30	-σ _A (* , * , T _j)	$\forall (C_k \in C) \wedge \forall (A_i \in A) \text{ alors } \neg \sigma_A(C_k, A_i, T_j)$
31	-σ _R (C _i , * , *)	$\forall (C_k \in C) \wedge \forall (R_j \in R) \text{ alors } \neg \sigma_R(C_i, R_j, C_k)$
32	-σ _R (* , R _k , *)	$\forall (C_i \in C) \wedge \forall (C_j \in C) \text{ alors } \neg \sigma_R(C_i, R_k, C_j)$

Les opérations de changement peuvent être classées en deux catégories selon la taxonomie définie dans [5] : les changements de type simple et ceux de type complexe. En adaptant les définitions données dans [4], les changements basiques correspondent à des changements simples et indivisibles, qui ne modifient qu'une seule caractéristique de connaissances de l'ontologie. C'est le cas d'un changement de suppression d'une relation de subsomption. Les opérations de changement basiques présentées par tableau n°2 s'expriment par les assertions de base du tableau n°1. Le signe « + » devant un identifiant d'une assertion et qui précise que le fait posé (un critère ou état de la vérité de la proposition posée) doit être vérifié ou présent dans son ensemble d'appartenance à priori. Le signe « - » devant un identifiant d'une assertion négative et qui précise que le fait posé ne doit pas être vérifié ou doit être absent dans son ensemble d'appartenance à priori.

Les opérations de changements complexes correspondent à des changements riches, incorporant des informations sur leur implication et/ou composites. Ils renferment également des séquences logiques de changements basiques et/ou complexes. Nous

avons par exemple: étendre le co-domaine d'une propriété à la superclasse de la classe qui le spécifiait est un changement riche. Dans le cet article, nous ne nous sommes intéresserons qu'à la gestion des changements basiques qui constituent d'ailleurs le socle de ceux complexes.

Tableau 2:Opérations de base pour une phase ϕ_i

Id	Opérations simples	Pré-condition	Invariant	Post-condition
1	CréerConcept(C_i)	$-C_i$	$-H^C(*, C_i)$ $-H^C(C_i, *)$ $-\sigma_A(C_i, *, *)$ $-\sigma_R(C_i, *, *)$	$+C_i$
2	SupprimerConcept(C_i)	$+C_i$	$-H^C(*, C_i)$ $-H^C(C_i, *)$ $-\sigma_A(C_i, *, *)$ $-\sigma_R(C_i, *, *)$	$-C_i$
3	CréerRelationAssociative(R_i)	$-R_i$ $-\sigma_{CARR}(R_i, *)$	$-\sigma_R(*, R_i, *)$	$+R_i$ $+\sigma_{CARR}(R_i, CAR_{R_i})$
4	SupprimerRelationAssociative(R_i)	$+R_i$	$-\sigma_{CARR}(R_i, *)$ $-\sigma_R(*, R_i, *)$	$-R_i$
5	CréerAttribut(A_i, T_i)	$-A_i$	$-\sigma_A(*, A_i, T_i)$ $-\sigma_T(A_i, *)$ $+T_i$	$+A_i$ $+\sigma_T(A_i, T_i)$
6	SupprimerAttribut(A_i)	$+A_i$	$-\sigma_A(*, A_i, T_i)$ $-\sigma_T(A_i, *)$ $+T_i$	$-A_i$
7	CréerType(T_i)	$-T_i$	$-\sigma_A(*, *, T_i)$ $-\sigma_T(*, T_i)$	$+T_i$
8	SupprimerType(T_i)	$+T_i$	$-\sigma_A(*, *, T_i)$ $-\sigma_T(*, T_i)$	$-T_i$
9	AjouterRelationAttribut(A_i, C_i, T_k)	$-\sigma_A(C_i, A_i, T_k)$	$+A_i$ $+C_i$ $+T_k$ $+\sigma_T(A_i, T_i)$	$+\sigma_A(C_i, A_i, T_k)$
10	SupprimerRelationAttribut(A_i, C_i, T_k)	$+\sigma_A(C_i, A_i, T_k)$	$+A_i$ $+C_i$ $+T_k$ $+\sigma_T(A_i, T_k)$	$-\sigma_A(C_i, A_i, T_k)$
11	AjouterSubsompion(C_i, C_j)	$-H^C(C_i, C_j)$	$+C_i$ $+C_j$	$+H^C(C_i, C_j)$
12	SupprimerSubsompion(C_i, C_j)	$+H^C(C_i, C_j)$	$+C_i$ $+C_j$	$-H^C(C_i, C_j)$
13	CréerCaractRelation(CAR_{R_i})	$-CAR_{R_i}$	$+R_i$ $-\sigma_{CARR}(*, CAR_{R_i})$	$+CAR_{R_i}$
14	SupprimerCaractRelation(CAR_{R_i})	$+CAR_{R_i}$	$-\sigma_{CARR}(*, CAR_{R_i})$	$-CAR_{R_i}$
15	AjouterCaractRelation(R_i, CAR_{R_i})	$-\sigma_{CARR}(R_i, CAR_{R_i})$	$+R_i$ $+CAR_{R_i}$	$+\sigma_{CARR}(R_i, CAR_{R_i})$
16	SupprimerCaractRelation(R_i, CAR_{R_i})	$+\sigma_{CARR}(R_i, CAR_{R_i})$	$+R_i$ $+CAR_{R_i}$	$+\sigma_{CARR}(R_i, CAR_{R_i})$
17	AjouterRelationAssociative(R_k, C_i, C_j)	$-\sigma_R(C_i, R_k, C_j)$	$+C_i$ $+C_j$ $+R_k$	$+\sigma_R(C_i, R_k, C_j)$

18	SupprimerRelationAssociative (R _k , C _i , C _j)	+σ _R (C _i , R _k , C _j)	+C _i +C _j +R _k	-σ _R (C _i , R _k , C _j)
19	Re-Type(A _k , T _i , T _j)	+σ _T (A _k , T _i) -σ _T (A _k , T _j)	+T _i +T _j +A _k	+σ _T (A _k , T _i) -σ _T (A _k , T _j)
20	CréerRelation(R _k , T _i)	-R _k -σ _{CARR} (R _k , T _i)	-σ _R (*, R _k , T _i) +T _i	+R _k +σ _{CARR} (R _k , T _i)
21	AjouterRelation(R _k , T _L , C _i , C _j)	-σ _R (C _i , R _k , C _j)	+R _k +C _i +C _j +σ _{CARR} (R _k , T _L)	+σ _R (C _i , R _k , C _j)
22	SupprimerRelation(R _k , T _L , C _i , C _j)	+σ _R (C _i , R _k , C _j)	+R _k +C _i +C _j +σ _{CARR} (R _k , T _L)	-σ _R (C _i , R _k , C _j)
23	Re-DéfinirRelation(R _k , T _k , T' _k)	+σ _{CARR} (R _k , T _k) -σ _{CARR} (R _k , T' _k)	+R _k	-σ _{CARR} (R _k , T _k) +σ _{CARR} (R _k , T' _k)
24	DéplacerAttribut(A _i , T _k , C _i , C _j)	+σ _A (C _i , A _i , T _k) +σ _T (A _i , T _k) -σ _A (C _j , A _i , T _k) -σ _T (A _i , T _k)	+A _i +C _i +A _j +T _k	-σ _A (C _i , A _i , T _k) -σ _T (A _i , T _k) +σ _A (C _j , A _i , T _k) +σ _T (A _i , T _k)
25	ChangerSuperClasse (C _j , C _i , C _k)	-H ^C (C _i , C _k) +H ^C (C _i , C _j)	+C _i +C _j +C _k	-H ^C (C _i , C _k) +H ^C (C _i , C _j)

2.3. Propagation des flux d'impacts à priori et traçabilité

Nous l'avons précisé un peu plus haut, l'ontologie n'est plus cohérente dès lors qu'une des contraintes associées au modèle d'évolution n'est plus satisfaite. Chaque opération étant associée à un ensemble d'assertions, l'opportunité de propagation se décline en trois cas possibles. Ainsi, en se basant sur l'approche en contexte de maintenance évolutive des logiciels proposée dans [2] :

1. Si la *pré-condition* et l'*invariant* sont vérifiés, alors l'opération peut s'exécuter sans propagation d'impacts ;
2. Si la *pré-condition* n'est pas vérifiée, alors l'opération n'est pas vérifiée et il n'y a pas d'impact sur les composants ontologiques;
3. Si la *pré-condition* est vérifiée et que l'*invariant* n'est pas vérifié alors l'opération est exécutée et il y a un processus de propagation d'impact qui consistera à marquer les faits qui s'unissent avec l'invariant. C'est-à-dire les composants qui utilisent le composant responsable de la modification. C'est par exemple le cas de suppression d'un concept ayant des sous-concepts. L'assertion marquer consiste à identifier les composants et les relations qui seront affectés par une modification. *Marquer (Comp_i)* consiste à marquer un composant *Comp_i* affecté par une opération de modification, puis de marquer toutes les relations $Rel_i \in \{R, H^C, \sigma_R, \sigma_{CARR}, \sigma_A\}$ ayant comme source de flux d'impact ce dernier. *Marquer(Rel_i)* consiste à propager la modification au nœud source de la relation de *Rel_i*. Deux cas de figure peuvent se produire : si le nœud source d'impact de la relation *Rel_i* est le nœud modifié,

impliquant l'arrêt de la propagation ; sinon si le nœud source d'impact de la relation Rel_i est différent du nœud modifié, cela permet de propager l'effet de la modification au voisinage du nœud modifié. Pour un ensemble de faits la propagation ne peut être déclenchée plus d'une fois, ce qui évite les problèmes de récursivité infinie.

Le Tableau 3 présente les assertions qui permettront de propager le changement d'impact en cas d'inconsistance dû à une opération. C'est ce qui nous permet ainsi de proposer une analyse a priori des effets de chaque type de modification simple ou complexe mais aussi la prédication ou l'analyse à postériori des effets de modification. L'approche permet aussi de développer un scénario prédictif des tâches de maintenance.

Tableau 3: Assertions générales de base pour la propagation des impacts

Id	Assertions	Signification
33	marquerConcept(C_i)	$\forall (C_j \in C)$ Si $H^c(C_j, C_i)$ alors marquerRelation($H^c(C_j, C_i)$) $\forall (C_j \in C)$ et $\forall (R_k \in R)$ Si $\sigma_R(C_j, R_k, C_i)$ alors marquerRelation($\sigma_R(C_j, R_k, C_i)$)
34	marquerRelation(R_k)	$\forall (C_i \in C)$ et $\forall (C_j \in C)$ Si $\sigma_R(C_j, R_k, C_i)$ alors marquerRelation($\sigma_R(C_j, R_k, C_i)$)
35	marquerAttribut(A_k)	$\forall (C_i \in C)$ Si $\sigma_A(C_i, A_k, T_j)$ alors marquerRelation($\sigma_A(C_i, A_k, T_j)$)
36	marquerType(T_k)	$\forall (A_i \in A)$ Si $\sigma_A(A_i, T_k)$ alors marquerRelation($\sigma_A(A_i, T_k)$)
37	marquerType($CARR_k$)	$\forall (R_i \in R)$ Si $\sigma_{CARR}(R_i, CARR_k)$ alors marquerRelation($\sigma_{CARR}(R_i, CARR_k)$)
38	marquerRelation($(H^c(C_i, C_j))$)	Si (C_i non marqué) alors marquerConcept(C_i)
39	marquerRelation($(\sigma_R(C_i, R_k, C_j))$)	Si (C_i non marqué) alors marquerConcept(C_i)
40	marquerRelation($(\sigma_A(C_i, A_k, T_j))$)	Si (C_i non marqué) alors marquerConcept(C_i)
41	marquerRelation($(\sigma_T(A_i, T_k))$)	Si (A_i non marqué) alors marquerConcept(A_i)
42	marquerRelation($(\sigma_{CARR}(R_i, CARR_k))$)	Si (R_i non marqué) alors marquerConcept(R_i)
43	marquerRelation($(H^c(*, C_i))$)	$\forall (C_k \in C)$ Si ($H^c(C_k, C_i)$) et (C_k non marqué) alors marquerConcept(C_k)
44	marquerRelation($(H^c(C_i, *))$)	$\forall (C_k \in C)$ Si ($H^c(C_i, C_k)$) et (C_k non marqué) alors marquerConcept(C_k)
45	marquerRelation($(\sigma_{CARR}(*, CARR_{R_i}))$)	$\forall (R_k \in R)$ Si ($\sigma_{CARR}(*, CARR_{R_i})$) et (R_k non marqué) alors marquerRelation(R_k)
46	marquerRelation($(\sigma_T(A_i, *))$)	$\forall T_k \in T$, Si ($\sigma_T(A_i, T_k)$) et (T_k non marqué) alors marquerType(T_k)
47	marquerRelation($(\sigma_A(*, A_i, T_j))$)	$\forall C_k \in C$, Si ($\sigma_A(C_k, A_i, T_j)$) et (C_k non marqué) alors marquerConcept(C_k)
48	marquerRelation($(\sigma_A(*, A_i, *))$)	$\forall C_k \in C, \forall T_j \in T$: Si ($\sigma_A(C_k, A_i, T_j)$) alors Si (C_k non marqué) alors marquerConcept(C_k) Si (T_j non marqué) alors marquerType(T_j)
49	marquerRelation($(\sigma_A(C_k, *, *))$)	$\forall A_i \in A, \forall T_j \in T$: Si ($\sigma_A(C_k, A_i, T_j)$) alors Si (A_i non marqué) alors marquerConcept(A_i)
50	marquerRelation($(\sigma_A(*, *, T_j))$)	$\forall C_k \in C, \forall A_i \in A$: Si ($\sigma_A(C_k, A_i, T_j)$) alors Si (C_k non marqué) alors marquerRelation($\sigma_A(C_k, A_i, T_j)$) Si (A_i non marqué) alors marquerRelation($\sigma_T(A_i, T_j)$)
51	marquerRelation($(\sigma_R(C_i, *, *))$)	$\forall R_j \in R, \forall C_k \in C$: Si ($\sigma_R(C_i, R_j, C_k)$) alors Si (C_k non marqué) alors marquerConcept(C_k)
52	marquerRelation($(\sigma_R(*, R_k, *))$)	$\forall C_i \in C, \forall C_j \in C$: Si ($\sigma_R(C_i, R_k, C_j)$) alors Si (C_i non marqué) alors marquerConcept(C_i)

3. Conclusion

Nous avons proposé dans cet article un modèle de propagation des impacts de modifications d'une ontologie. Elle consiste à restructurer une ontologie suivant le modèle lexicale et un ensemble d'assertions établies à priori permet de connaître les conditions d'exécution d'une opération ainsi que les conditions devant être respectées avant et après exécution. L'impact de modification d'un composant sera ainsi propagé par les relations aux autres composants. Pour cela nous avons défini les opérations de base de modification en corrélation avec le modèle dévolution d'une ontologie lexicale.

Notre objectif de départ était d'effectuer un « binding » des approches de maintenance évolutives proposées en ingénierie logicielle, qui présentent des similarités avec celles proposées en approche ontologique. Nous projetons de réutiliser l'approche proposée pour le contrôle des opérations de changements complexes et ainsi l'étendre aux systèmes d'intégration de données basées sur les ontologies en l'appliquant aux dataweb sémantiques. Il conviendra aussi d'étudier la piste de l'évolution des ontologies sous l'angle de l'équilibre structurelle d'une part et sémantique d'autre part.

4. Bibliographie

- [1] Bloehdorn, S., Haase, P., Sure, Y. & Voelker, J. (2006). *Ontology evolution*. In J. Davies, R. Studer, & P. Warren (Ed.s), *Semantic Web Technologies, Trends and research in Ontology-based Systems* (pp. 51-70). John Wiley & Sons Publication, 2006.
- [2] Deruelle L.(2001)*Analyse d'impact de l'évolution des applications distribuées multi-langages et à bases de données hétérogènes*. Thèse de doctorat, ULCO, 2001.
- [3] Diop I., Lo M., Corby O.,(2010) *Problématique de recherche sur évolution d'un système d'intégration de dataweb sémantiques distribués*. Colloque Nationale sur la Recherche en Informatique et Automatique 2010. CNRIA'2010.
- [4] Jedidi, R.(2009) *Approche d'évolution d'ontologie guidée par des patrons de gestion de changement*. Thèse de doctorat de l'Université Paris-Sud XI Orsay. Novembre 2009.
- [5] Klein, M. (2004). *Change management for distributed ontologies*. Ph.D. Thesis, Dutch Graduate School for Information and Knowledge Systems. Germany, 2004.
- [6] Le Pham, A.,(2008) *De l'optimisation à la décomposition de l'ontologique dans la logique de description*. Thèse de doctorat, Université de Nice Sophia-Antipolis-21/01/2008.
- [7] Luong, P.H., Dieng-Kuntz, R. (2007). *A Rule-based Approach for Semantic Annotation Evolution*. The Computational Intelligence Journal, 23(3):320-338. Blackwell Publishing, Malden, MA 02148, USA, 2007.
- [8] Maedche A, Motik B, & Stojanovic L. (2003). *Managing multiple and distributed ontologies in the Semantic Web*. VLDB Journal, 12(4), 286–300. doi 10.1007/s00778-003-0102-4,2003.
- [9] Plessers, P. (2006). *An approach to web-based ontology evolution*, Ph.D. Thesis, University of Brussels, Belgium, 2006.
- [10] Stojanovic L. (2004). *Methods and Tools for Ontology Evolution*, Ph.D. Thesis, Karlsruhe University. Germany, 2004.