UNIVERSITÉ DU QUÉBEC À MONTRÉAL

LA FUSION DES ONTOLOGIES

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

KAIS SALHI

JUILLET 2014

REMERCIEMENTS

La recherche, quels que soient ses objectifs et le domaine dans lequel elle s'inscrit, exige de la part du chercheur un degré de motivation intrinsèque assez élevé. Mais en réalité, cette dernière demeure insuffisante si la recherche n'est pas en synergie avec un autre type de motivation d'une importance capitale: la motivation extrinsèque.

Bref, dans un mémoire ou dans une thèse de doctorat, le chercheur doit s'ingénier à s'enquérir du contenu de son travail, tout en prenant en considération l'apport d'autrui qui s'avère important, voire indispensable.

À ce titre, j'avoue personnellement que tout au long des différentes phases de mon présent mémoire, je fus encouragé et soutenu par bon nombre de personnes dont l'adhésion fut vraiment spontanée et fructueuse. Je veux donc:

- Exprimer d'abord ma gratitude à Monsieur ROGER NKAMBOU, mon directeur de projet, pour m'avoir pris en charge, pour son soutien scientifique et méthodologique, mais surtout pour sa patience et le temps qu'il a consacré dans les phases de lecture, de correction et de régulation de mon mémoire. Sincèrement, il est à noter que les apports et le suivi continu de Monsieur ROGER NKAMBOU ont largement contribué à la réalisation de mon mémoire.
- Exprimer également ma reconnaissance à Monsieur MOHAMED ROUANE-HACENE pour m'avoir aidé à surmonter tous les problèmes techniques et à combler les difficultés inhérentes au développement de mon projet. Je tiens également à adresser mes sincères remerciements à Monsieur PETKO VALTCHEV pour ses conseils techniques et méthodologiques.

- Adresser aussi mes sincères remerciements à tous mes collègues dans les laboratoires pour leur collaboration lors de la réalisation de mon travail, pour leurs idées et leurs mises au point proposées dans le parcours de la recherche.
- Remercier également tout le personnel de la bibliothèque de l'UQAM pour leur accompagnement constructif. En effet, ce personnel n'a épargné aucun effort pour répondre aux besoins demandés et mettre à ma disposition les différents documents de référence, ce qui a réellement facilité mes lectures personnelles et l'élaboration d'une bibliographie.
- Saluer tout le personnel de l'UQAM, et plus particulièrement celui du département d'informatique, pour m'avoir donné l'occasion de m'inscrire dans le domaine de la recherche et d'aller de l'avant dans mes études théoriques et pragmatiques.
- Présenter mille remerciements à mes parents et à mes deux sœurs qui n'ont de cesse de me doter de tous les moyens pour me permettre de fréquenter les grandes universités et d'aller toujours à la quête du savoir.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

RÉSUMÉ

La fusion des ontologies est une filière du domaine de la gestion des connaissances qui est en relation étroite avec d'autres domaines informatiques comme l'intelligence artificielle, le Web sémantique et le traitement du langage naturel. Cette filière prend une part de plus en plus accentuée dans la gestion des ontologies tout en considérant l'évolution rapide de la technologie des connaissances.

La naissance de la fusion d'ontologies a conduit à la menée de plusieurs travaux de recherches et le développement de différentes approches concrètes, mais qui présentent certaines faiblesses, notamment au niveau de l'analyse des données relationnelles. La majorité de ces travaux se concentrent principalement sur l'alignement et la détection des similarités, mais ils négligent les informations qu'on pourrait dégager à partir de l'analyse formelle et relationnelle des concepts.

Dans ce projet, nous proposons une approche de fusion d'ontologies, au sein de la plateforme INUKHUK, basée sur l'application d'analyses formelles et relationnelles des concepts (AFC et ARC). Ainsi, le principe de notre approche s'articule sur la factorisation des deux ontologies sources. Cette factorisation engendre une structure qui sera nettoyée à l'aide d'un outil d'alignement. Nous appliquons les analyses avec le moteur ARC sur la structure générée précédemment pour dégager un ensemble de treillis. Nous déduisons l'ontologie fusionnée à partir de l'ensemble de treillis dégagé. Avec notre approche, nous exploitons également la notion de la ré-ingénierie puisque nous factorisons, puis nous restructurons les ontologies.

Mots-clés: fusion d'ontologies, alignement d'ontologies, factorisation des ontologies, analyse relationnelle des concepts, analyse formelle des concepts, génération des treillis, restructuration des ontologies.

CHAPITRE I

INTRODUCTION

La construction des ontologies représente un espace multidisciplinaire comportant l'apprentissage machine, la représentation des connaissances, l'exploration des données et le traitement du langage naturel. Cela engendra la naissance de plusieurs travaux de recherche dans le domaine de l'ingénierie des ontologies, surtout sur le plan de la construction des ontologies ainsi que de la réutilisation de celles-ci. Cependant, les travaux de recherches sur la fusion des ontologies demeurent limités par rapport aux autres sous-domaines de l'ingénierie des ontologies. Or, la fusion d'ontologies est reconnue comme étant une opération essentielle intervenant à plusieurs niveaux de l'ingénierie des ontologies. Il est par exemple incontournable que la construction d'ontologies nécessite l'intégration de plusieurs modules ontologiques existants (par exemple: domaine biomédical et domaine d'analyse des données) (Lambrix et Tan, 2006) (Maiz, Boussaid et Bentayeb, 2008).

Les avantages offerts par les ontologies en termes de représentation de connaissances ont conduit à l'émergence explosive de centaines d'ontologies dans différents domaines, menant parfois à des redondances inutiles. De ce fait, il devient parfois primordial de fusionner deux ou plusieurs ontologies du même domaine tout en considérant les aspects lexical, sémantique, structurel et taxonomique.

Les ontologies sont devenues des modèles indispensables pour la représentation, le partage et la réutilisation des connaissances de domaine. L'une des motivations de la

fusion des ontologies tient à la construction d'une ontologie à partir de sources différentes. La fusion est déclenchée suite à un besoin qui pourrait nécessiter l'intégration de plusieurs ontologies modélisant les différentes parties d'un domaine.

La complexité des sources de données et leurs différentes distributions sont deux facteurs importants et considérables au niveau de la construction d'une ontologie. De ce fait, la mise en œuvre d'une approche de génération d'ontologie nécessite une opération efficace de fusion afin de permettre une construction par fusion.

1.1 Problématique

La construction des ontologies couvre plusieurs sous-domaines menant souvent à une multitude d'ontologies pour un même domaine, par exemple les ontologies Tourism et Travel qui couvrent tous les deux le domaine touristique. Mais certains besoins applicatifs peuvent parfois obliger à dégager une et une seule ontologie du domaine global. Il est alors important de mettre en place un engin (ou un outil) admettant la combinaison des connaissances des différentes ontologies à fusionner et ce, tout en considérant la cohérence et la sémantique de l'ontologie engendrée.

La fusion de deux ou plusieurs modèles ontologiques partiels en un modèle global est une opération plus complexe qu'une simple intégration puisqu'il y aura certaines vérifications syntaxiques, sémantiques et lexicales, ou encore une vérification de la cohérence du modèle global généré. Cependant, des composantes ontologiques similaires peuvent exister dans les modèles partiels, d'où le besoin de passer par une étape d'alignement. De plus, les deux modèles représentent des conceptualisations des sous-domaines sous-jacents, ce qui fait qu'une simple réunion des éléments de modélisation de ces sous-domaines pourrait inclure des incohérences ou des éléments superflus.

De ce fait, quelques travaux de recherche ont été proposés et certaines approches de fusion d'ontologies ont été développées. Néanmoins, ces approches comportent des limites que nous résumons comme suit:

- L'abstraction d'un processus initial d'alignement des deux ontologies pour pallier la redondance lexicale ou encore sémantique.
- L'inefficacité de l'algorithme ou du moteur de fusion des ontologies.
- La non-existence d'une rétro-ingénierie de l'ontologie résultante après la fusion des deux ontologies partielles.

1.2 Objectifs et méthodologie

Pour faire face aux problèmes soulevés dans la problématique, ce projet veut s'inspirer de la fusion des modèles (un processus relativement abouti en génie logiciel) pour proposer une méthode et un outil efficace pour la fusion des ontologies. Notre approche se fonde sur la manipulation d'un outil de factorisation et d'un outil d'abstraction pour la fusion des deux ontologies.

Elle comporte les étapes suivantes:

- Élaborer un module permettant de décortiquer les deux ontologies partielles en termes de concepts (nom, propriétés, super/sous-concepts, etc.), de relation, d'attributs, d'instances, de terme composé, etc. Cette étape vise la séparation de chaque élément ontologique pour chacune des deux ontologies.
- Implémenter un module qui permet de générer une structure FCR (Famille des Contextes Relationnelle) comportant des familles de contexte binaire utile pour une analyse formelle de la structure ontologique (par exemple: un contexte pour les concepts et un autre pour les relations).
- Créer une structure permettant de détecter la similarité au sein des deux ontologies partielles, en prenant en entrées deux ontologies en format OWL.

- Raffiner la structure FCR précédente avec ce que nous avons dégagé comme éléments similaires. La sortie de cette étape sera un RCFAligner, sans bruits et surtout propre (pas de redondance lexicale ou sémantique). RCFAligner sera une entrée de la prochaine étape de notre algorithme.
- Développer un module à l'aide de la technologie de l'ARC (Analyse Relationnelle de Concept) qui prend en entrée quelques contextes binaires de la structure FCR, pour produire en sortie un treillis pour chacun de ces contextes.
- Implémenter un module appelé ONTODesigner qui permet de générer l'ontologie fusionnée à partir des treillis de sorties de l'ARC. Cette étape est effectuée tout en manipulant les technologies de restructuration et d'extraction. À la fin de cette étape, nous obtiendrons une ontologie fusionnée.
- Réaliser un module de nommage pour pallier la multitude de noms des abstractions générées avec ARC. L'objectif de cette étape est de raffiner la qualité de l'ontologie finale et d'ouvrir des perspectives sur la nécessité d'une approche pertinente de nommage.
- La dernière étape de ce projet est de mettre en place l'évaluation nécessaire pour valider notre approche. De ce fait, nous avons eu recours aux métriques déjà implémentées au sein de la plate-forme INUKHUK (M. Rouane-Hacene *et al.*, 2011b) et nous avons également appliqué notre protocole de validation avec le choix de quelques ontologies. À la fin de cette étape, nous complétons notre méthodologie par une interprétation des résultats obtenus.

Notre contribution consiste essentiellement à la mise en place d'une approche de fusion basée sur la factorisation et la restructuration des deux ontologies. Pour y arriver, nous avons exploité l'outil AFC et son extension ARC. Par ailleurs, nous avons effectué des améliorations en réalisant un module de préalignement et en

montrant l'importance d'un nommage conceptuel pertinent dans la fusion des ontologies tel que détaillé dans les chapitres 4 et 5.

1.3 Plan du mémoire

En plus des chapitres d'introduction (premier chapitre) et de conclusion (dernier chapitre), notre mémoire contient quatre autres chapitres organisés de la manière suivante:

Le chapitre II présente le cadre conceptuel de notre projet. Nous y définissons les ontologies en tant qu'outils de représentation des connaissances et leurs différentes composantes. Par ailleurs, nous présentons les notions de base de notre outil de fusion des ontologies (Galicia, AFC, ARC, FCR) ainsi que la plate-forme INUKHUK sur laquelle est fondée le modèle et l'outil proposés.

Le chapitre III passe en revue la littérature des approches existantes sur la fusion des ontologies et de leurs limites. Nous y présentons aussi quelques outils de fusion des ontologies. De plus, nous analysons d'une manière plus profonde les approches basées sur les AFC. Nous introduisons à la fin de ce chapitre certaines métriques que nous utiliserons pour mesurer la qualité des ontologies fusionnées.

Dans le chapitre IV, nous présentons notre propre approche de fusion des ontologies ainsi que son processus de développement. Aussi, nous présentons les différents modules implémentés de notre algorithme de fusion. La dernière section de ce chapitre se concentre sur les potentielles améliorations de l'efficacité de notre algorithme de fusion des ontologies.

Dans le chapitre V, nous présentons l'évaluation effectuée pour valider notre travail ainsi que les expérimentations réalisées après chaque amélioration. Tous ces tests sont réalisés en nous basant sur un protocole de validation que nous présentons et en nous articulant sur les métriques de qualité développées au sein de la plate-forme

INUKHUK. Par ailleurs, une analyse des résultats permet de confirmer la qualité de l'ontologie fusionnée à l'aide de notre approche.

Ce mémoire se termine par une synthèse de nos contributions ainsi que par les limites et perspectives que nous aurons identifiées.

CHAPITRE II

LE CADRE CONCEPTUEL ET LES TECHNOLOGIES

2.1 Contexte

La mise en œuvre de notre travail nécessite l'usage de plusieurs techniques. Dans ce chapitre, nous aborderons les concepts fondamentaux du cadre conceptuel de notre approche de fusion d'ontologies basée sur AFC (l'Analyse Formelle de Concept) et ARC (l'Analyse Relationnelle de Concept). Les ontologies AFC et ARC constituent la couche de base (ou la première couche) de notre outil de fusion. Nous détaillerons également l'environnement Galicia qui supporte les concepts AFC et ARC, lequel environnement nous aide à les exploiter dans notre projet. Aussi, nous présenterons la plateforme de maintenance des ontologies INUKHUK au sein de laquelle notre projet sera implémenté. Donc, la deuxième couche de notre cadre conceptuel contient les environnements de développement et la troisième couche correspond à notre approche RCA-Merge. La figure 2.1 illustre notre cadre conceptuel de l'approche de la fusion des ontologies RCA-Merge.

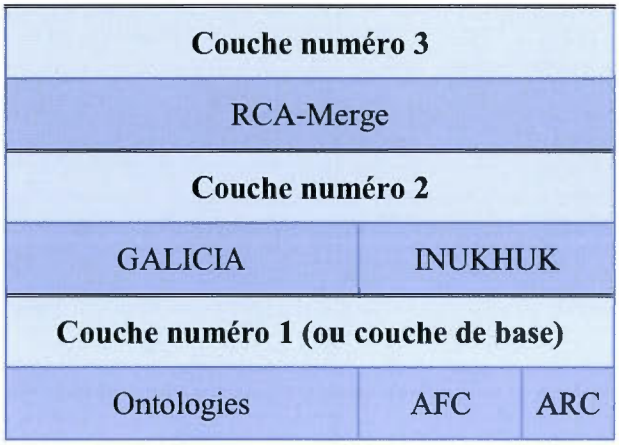


Figure 2.1 Le cadre conceptuel de notre approche RCA-Merge

Dans les sections suivantes de ce chapitre, nous détaillons les techniques mentionnées ci-dessus en tant que concepts fondamentaux.

2.2 Les ontologies

Le terme «ontologie» trouve son origine chez les philosophes pour indiquer la métaphysique et l'étude de l'existant. Mais au fil du temps, le terme a évolué pour se retrouver dans le domaine médical où l'ontologie désigne le traitement de la genèse des maladies. Finalement, la notion d'ontologie s'est retrouvée dans le domaine de l'informatique.

Le terme «ontologie» est utilisé en informatique pour désigner un formalisme de représentation des connaissances ou encore un modèle de connaissance. La définition la plus connue est celle de Gruber (Gruber, 1993): «une ontologie est une spécification explicite d'une conceptualisation» [notre traduction]. En d'autres termes, le modèle ontologique correspond à une représentation d'un domaine de la réalité à l'aide des objets et des relations entre ces objets. Aussi, une autre définition de W.N. Borst s'exprime ainsi: «une ontologie est une spécification formelle et explicite d'une conceptualisation partagée» [notre traduction] (Borst, 1997). Formellement, la dernière définition indique que l'ontologie se traduit par un langage compréhensible par la machine. Aussi, le terme sous-entend que les éléments de

l'ontologie (concepts, propriété, relation, axiome, fonction) sont présentés de façon explicite. La conceptualisation indique que l'ontologie représente une perception du monde à l'aide d'un modèle abstrait, et cette conceptualisation doit (normalement) être partagée par une communauté d'experts du domaine.

Par analogie, une ontologie est comparable à un modèle de données UML (le diagramme de classe) et ce, même si contrairement à un modèle UML, l'expressivité d'un modèle ontologique peut être beaucoup plus poussée.

Selon Gruber, une ontologie est modélisée en mettant en place cinq types d'éléments (Gruber, 1993):

- Les concepts qui sont représentés sous forme taxonomique. On retrouve également des super/sous-concepts ainsi que des concepts composés et élémentaires.
- Les relations qui sont similaires aux associations du diagramme de classe UML. En effet, les relations correspondent aux liens entre les concepts.
- Les instances qui désignent les objets.
- Les axiomes: l'utilisation des axiomes revient à accentuer la vérité.
- Les fonctions qui correspondent à une relation spécifique où le nième élément est présenté à partir des n-1. En d'autres termes, on a: $f:C_1 \times C_2 \times \ldots \times C_{n-1} \times C_n$.

En plus, Gruber (Gruber, 1995) a proposé quelques critères de base pour réaliser une ontologie, mais il mentionne que ces critères sont légèrement variables selon l'objectif de la conception de l'ontologie:

- La clarté: les termes utilisés au sein de l'ontologie doivent être clairs, compréhensibles et objectifs.
- La cohérence: une ontologie doit déduire des faits. Aussi, la cohérence doit exister sur le plan de la logique entre les concepts et les relations.

- L'extensibilité: une ontologie doit supporter certains ajouts et mises à jour prévus. Donc, la conception d'une ontologie doit être présentée de façon à ce qu'on puisse l'étendre sans avoir recours à une révision des anciens éléments.
- Le biais d'encodage minimal: la conceptualisation et le développement d'une ontologie doivent s'effectuer indépendamment des langages de codage et aussi indépendamment des symboles particuliers. Cela tient essentiellement à la diversité des sources de partage des ontologies.
- L'engagement ontologique minimal: une ontologie doit mettre l'accent sur le minimum d'engagement satisfaisant pour couvrir un domaine prédéfini. Gruber a présenté l'engagement ontologique minimal comme suit: «d'engagement ontologique s'articule sur la cohérence des termes» [notre traduction] (Gruber, 1995).

L'utilisation des ontologies s'étend dans plusieurs domaines, et surtout en intelligence artificielle puisqu'elle permet essentiellement de représenter les connaissances. La figure 2.2 illustre les domaines concernés par les ontologies. Comme on peut le voir dans cette figure, le Web sémantique est concerné. En effet, les ontologies permettent la structuration et l'ajout des données sémantiques (Giri, 2011).

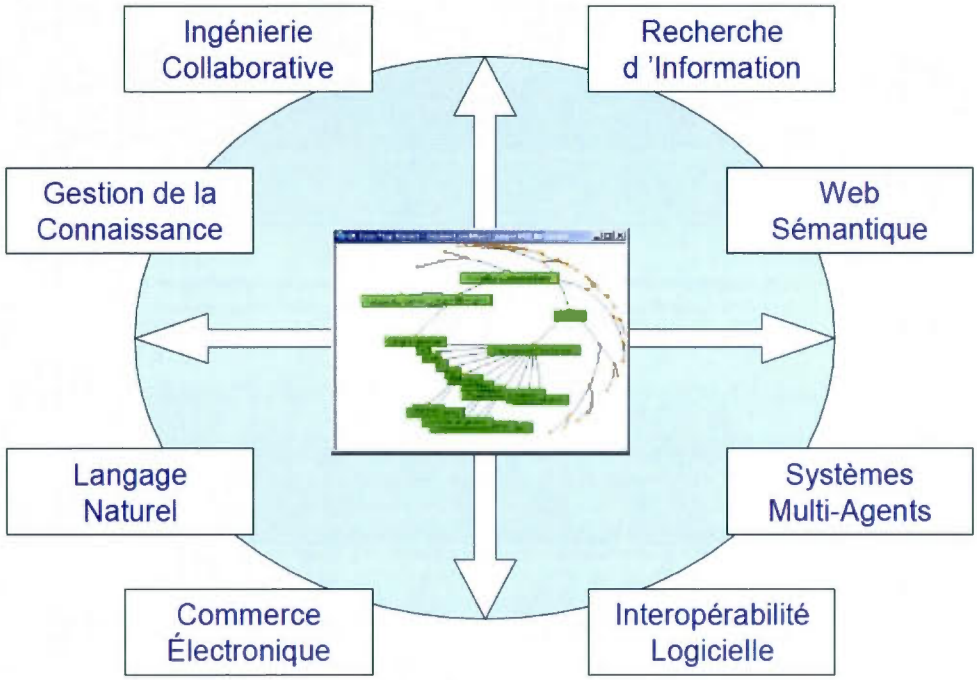


Figure 2.2 Les ontologies et les sous-domaines des TI[1]

2.3 L'analyse formelle de concept

L'analyse formelle de concept (AFC) (en anglais *Formal Concept Analysis*, ou FCA) est une analyse basée sur des notions mathématiques dans le but de générer une hiérarchie des concepts. L'analyse formelle des concepts vise essentiellement à générer des structures conceptuelles à partir des données et de les analyser (Peggy, Felix et Bernhard, 2013). Les technologie issues du domaine de l'AFC ont été utilisées dans différents domaines d'application comme la biologie, le génie civil, le génie logiciel, les systèmes d'information, etc (Ganter, Stumme et Wille, 2005).

Commençons tous d'abord par définir le concept qui est la clé de l'AFC. Un concept en informatique est une entité abstraite d'objets. Il est constitué principalement de

[1] http://www.lsis.org/espinasseb/Supports/ONTOWS-2010/Onto-2010-4p.pdf

deux composantes: la première est l'extension qui correspond généralement au nom du concept. La deuxième est l'intension qui désigne les attributs (propriétés) qui s'appliquent à tout objet de l'extension.

En effet, pour passer à la présentation des concepts formels, on doit admettre un modèle mathématique qui supporte les notions d'objet, d'attribut et de relation. Pour ce faire, le contexte formel est la structure appropriée. Selon (Wille, 2009), un contexte formel est défini comme suit: «un contexte formel est une structure K:=(G,M,I) où G et M sont des ensembles et I est une relation binaire entre G et M tel que $I \subseteq G \times M$, et les éléments de G et M sont appelés respectivement des objets et des attributs» [notre traduction]. Le tableau 2.1 représente un exemple de contexte formel.

Tableau 2.1 Exemple de contexte formel (Ganter *et al.*, 2005)

	natural	artificial	stagnant	running	inland	maritime	constant	temporary
tarn	X		X		X		X	
trickle	X			X	X		X	
rill	X			X	X		X	
beck	X			X	X		X	
rivulet	X			X	X		X	
runnel	X			X	X		X	
brook	X			X	X		X	
burn	X			X	X		X	
stream	X			X	X		X	
torrent	X			X	X		X	
river	X			X	X		X	
channel				X	X		X	
canal		X		X	X		X	
lagoon	X		X			X	X	
lake	X		X		X		X	
mere		X	X		X		X	
plash	X		X		X			X
pond		X	X		X		X	
pool	X		X		X		X	
puddle	X		X		X			X
reservoir		X	X		X		X	
sea	X		X			X	X	

En outre, Ganter et ses collègues (Ganter *et al.*, 2005) ont présenté les concepts formels du contexte formel en se basant sur quelques opérations de dérivation: «$X \subseteq G$ et $Y \subseteq M$. Donc, un concept formel d'un contexte formel K correspond à une paire (A,B) avec $A \subseteq G$ et $B \subseteq M$, A et B étant respectivement l'extension et l'intension du

concept formel (A,B)» [notre traduction]. La figure 2.3 représente le treillis des concepts (diagramme échiqueté et linéaire) (Ganter *et al.*, 2005).

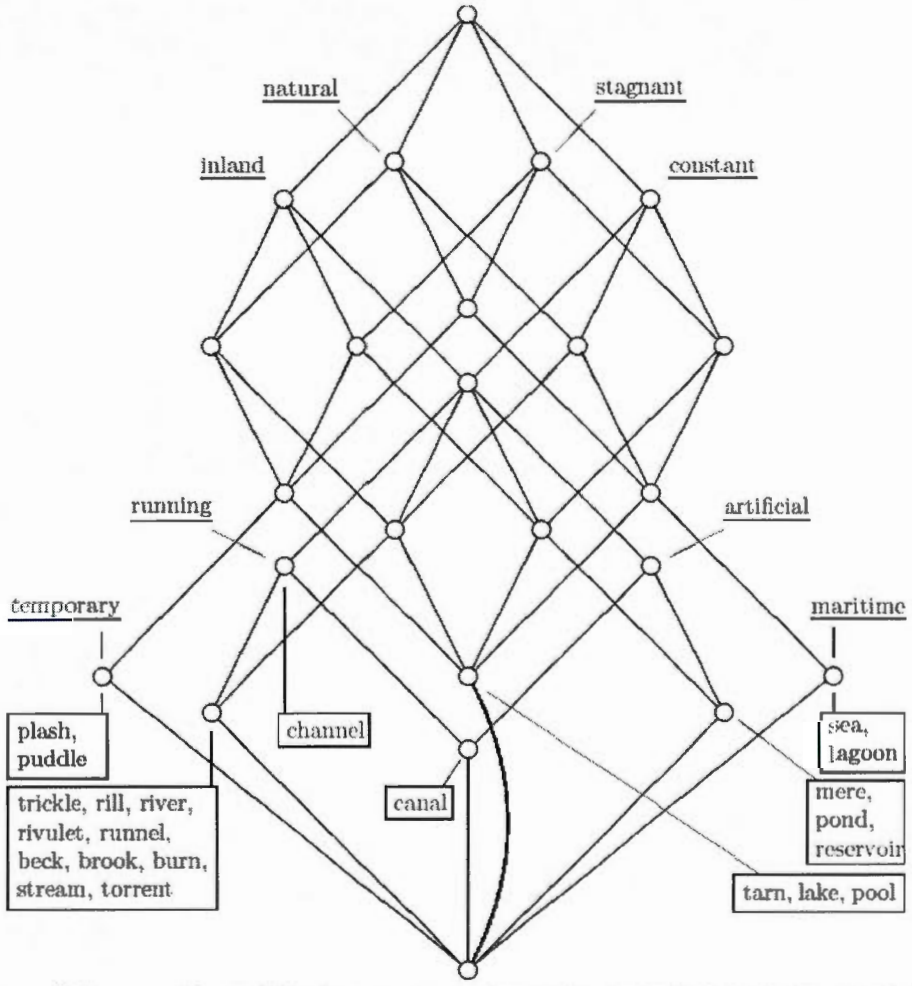


Figure 2.3 Un treillis des concepts du contexte formel de la figure ci-dessus (Ganter *et al.*, 2005)

De plus, l'AFC constitue un outil de restructuration, de génération et de modification des treillis de concepts. Aussi, l'utilisation de l'AFC se présente aux étapes de prédéveloppement telles que l'analyse des besoins et la récupération des données (Tilley, Cole, Becker et Eklund, 2005) (Macko, J., 2012).

2.4 L'analyse relationnelle des concepts

L'analyse relationnelle de concept (ARC) (en anglais *Relational Concept Analysis*, ou RCA) est une extension des mécanismes de base de l'AFC. L'ARC correspond essentiellement à traiter les données relationnelles, et plus précisément les multiples catégories d'individus (Rouane-Hacene, M., Huchard, Napoli et Valtchev, 2013). Aussi, l'ARC s'occupe de l'analyse des liens interindividuels; de ce fait plusieurs attributs seront progressivement déduits. Donc, l'ARC permet de maintenir efficacement les treillis de concepts avec les connaissances conceptuelles extraites (Rouane, Mohamed Hacene, Huchard, Napoli et Valtchev, 2007).

L'extension de l'AFC en ARC a été entreprise suite aux limites de l'AFC. Rouane-Hacene et ses collègues ont mentionné qu'il y a une absence d'intégration des relations entre les objets dans le processus d'analyse, et ce type de relation (ou lien) représente des données spécifiques appelées données relationnelles (Rouane, Mohamed Hacene *et al.*, 2007). L'ARC définit une méthodologie spécifique qui permet l'analyse conceptuelle des données relationnelles (M. Rouane-Hacene *et al.*, 2013).

Les paramètres d'entrées de l'analyse ARC sont orchestrés dans une structure appelée Famille de Contexte Relationnelle FCR (en anglais *Relational Context Family*, ou RCF). Comme son nom l'indique, la FCR est une structure qui contient un groupe de contexte binaire et un groupe de relation binaire; $R_k = O_i \times O_j$ avec O_i et O_j sont des objets de K_i et K_j. En considérant les ontologies, cette structure est transformée à un unique FCR contenant le contexte binaire K_i des entités (les concepts et les entités) de métamodèle ontologique et le contexte binaire des relations R_k (M. Rouane-Hacene *et al.*, 2013).

Par la suite, nous présenterons l'analyse relationnelle des données avec le moteur de l'ARC. Cette étape sera détaillée au chapitre 4 (4.2.2 avec RCFModeler et RCAEngine). La figure 2.4 représente un exemple de treillis des concepts finaux après l'analyse avec ARC.

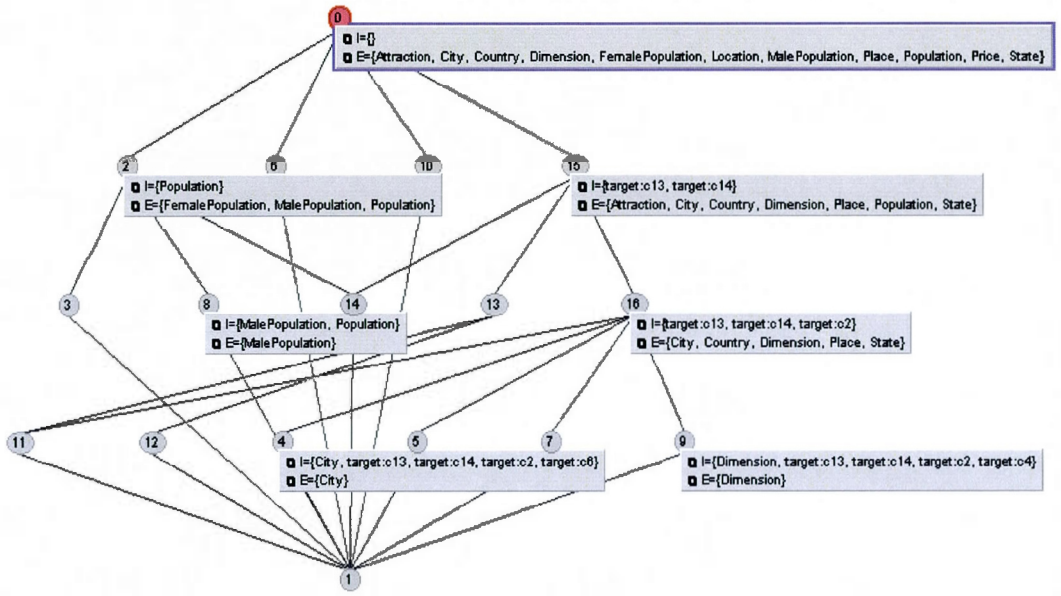


Figure 2.4 Exemple d'un treillis de concepts après analyse RCA

2.5 Galicia (*Galois Lattice interactive Constructor*)

L'efficacité de l'analyse formelle de concept (AFC) a bel et bien été prouvée dans le domaine de l'ingénierie des connaissances ainsi que dans le domaine de la fouille de données. Cependant, plusieurs travaux ont été réalisés pour permettre la construction et la manipulation des treillis et ce, à l'aide des algorithmes pertinents. Jusqu'ici, les chercheurs du domaine de l'AFC ont concrétisé leurs efforts pour développer des algorithmes et des outils de manipulation des treillis. Mais il fallait un environnement intégré pour le traitement des treillis. Galicia répond à ce besoin (Valtchev, Grosser, Roume et Hacene, 2003).

Galicia est une plateforme ouverte de construction, d'exploration et de maintenance des treillis (Valtchev *et al.*, 2003). Elle est conçue sous la base d'une architecture ouverte et d'une implémentation générique afin de maximiser l'adaptation des paramètres des problèmes. À cet effet, Galicia admet une conception de traitement et de description des données complexes – notons que FCR (Famille de Contexte

Relationnelle) correspond à un outil original de Galicia –, et la figure 2.5 montre un exemple de contexte binaire avec l'interface de Galicia. En ce qui concerne la visualisation des treillis de concepts, cet environnement (Galicia) offre un mode de visualisation en 2D et 3D, et la figure 2.6 montre l'interface de visualisation (Valtchev *et al.*, 2003).

Par ailleurs, Galicia comporte un ensemble d'algorithmes de maintenance et de génération des treillis tel que l'algorithme de Godin et ses collègues (Godin *et al,* 1993).

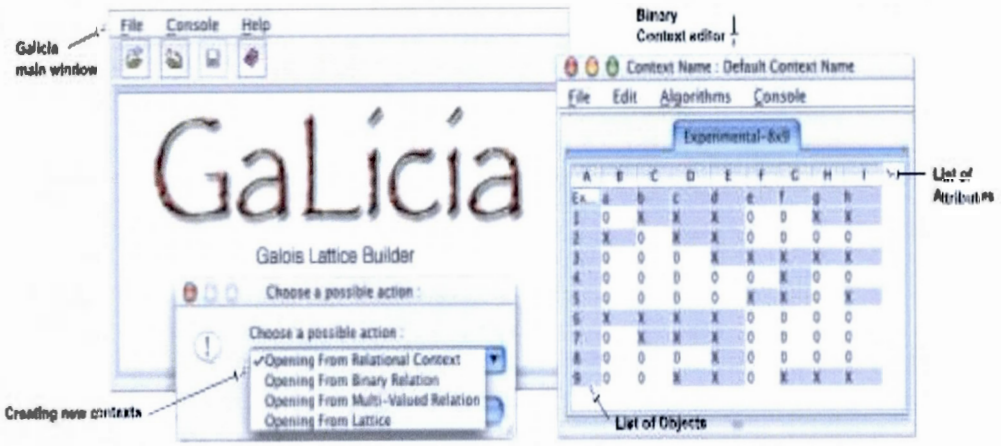


Figure 2.5 Exemple d'un contexte binaire et l'interface Galicia (Valtchev *et al.*, 2003)

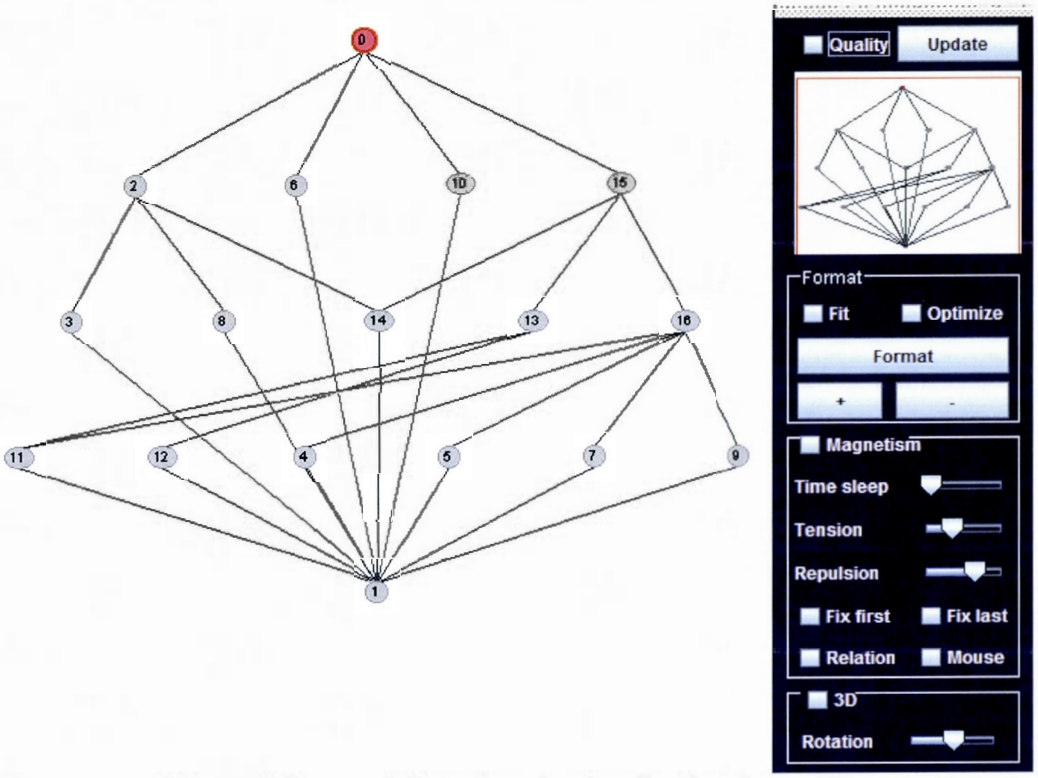


Figure 2.6 L'interface de visualisation des treillis

2.6 INUKHUK: une plateforme orientée service pour la maintenance des ontologies

La réalisation de notre travail fait partie du développement d'une plateforme orientée service appelé INUKHUK. Cette plateforme est basée sur l'analyse relationnelle des concepts. INUKHUK offre certains outils de traitement des ontologies. Parmi les services (ou outils) existants dans cette plateforme, on retrouve la construction, la fusion, la restructuration et la modularisation. En outre, le développement du système INUKHUK est réalisé à l'aide de l'environnement Galicia et ce, conjointement avec d'autres API et plateformes comme Simpack AlignApi, Wordnet, Jena (M. Rouane-Hacene *et al.*, 2011b).

Dans le cadre de notre projet, nous nous sommes intéressés au service de la fusion des ontologies. Cependant, comme l'illustre bien la figure 2.7, l'opération de fusion

touche directement ou indirectement certains autres services (M. Rouane-Hacene *et al.*, 2011b).

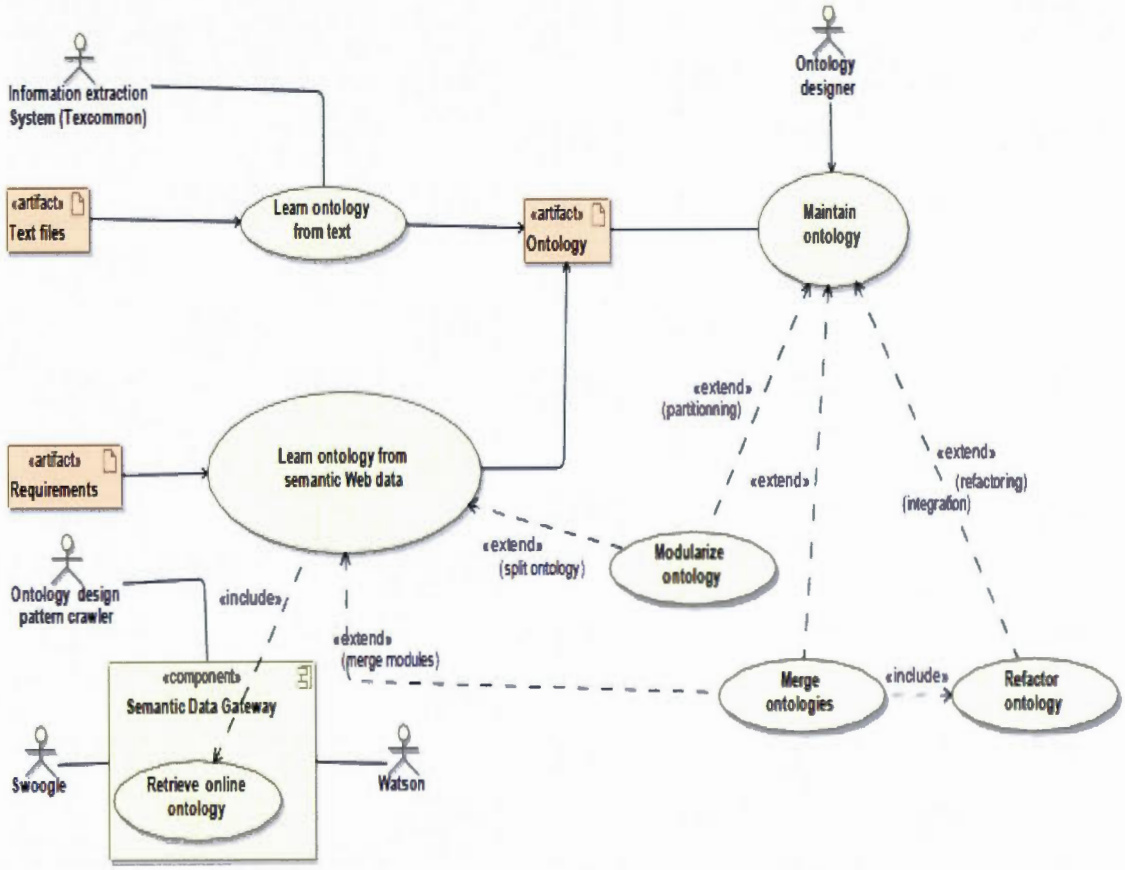


Figure 2.7 Les services de la plateforme INUKHUK (M. Rouane-Hacene et al., 2011b)

Dans ce chapitre, nous avons jusqu'ici abordé le cadre conceptuel général de notre approche. Nous avons également défini les notions fondamentales des éléments constituant le cadre conceptuel. Dans ce qui suit, nous nous intéresserons à l'état de l'art où nous présenterons quelques approches existantes et quelques métriques de mesure de qualité ontologique.

CHAPITRE III

LA FUSION DES ONTOLOGIES: ÉTAT DE L'ART

Ce chapitre présente une analyse critique de la littérature sur quelques approches générales de fusion des ontologies ainsi que sur celles basées sur l'AFC. Nous présenterons également les métriques de mesure de qualité existantes dans la plateforme INUKHUK dans le but d'évaluer les ontologies dégagées après la réalisation de notre approche.

3.1 Quelques approches de fusion d'ontologies

3.1.1 PROMPT

L'outil *PROMPT Suite* (Noy, Natalya F. et Musen, 2003) fait partie de la plateforme de gestion des ontologies Protégé-2000. PROMPT est constitué d'un ensemble de modules qui ont une grande importance dans les services d'alignement et de fusion. *PROMPT Suite* contient un outil de fusion des ontologies appelé iPROMPT, un outil appelé Anchor-PROMPT pour trouver les similarités entre les ontologies, un outil de comparaison des versions des ontologies appelées PROMPTDiff et un outil appelé PROMPTFactor qui permet de créer une ontologie factorisée. La figure 3.1 illustre l'infrastructure et les interactions entre les différents modules de *PROMPT Suite*.

Dans ce qui suit cette sous-section, nous nous intéresserons à l'algorithme d'iPROMPT puisqu'il représente le cœur de la fusion, la figure 3.2 illustrant ces étapes.

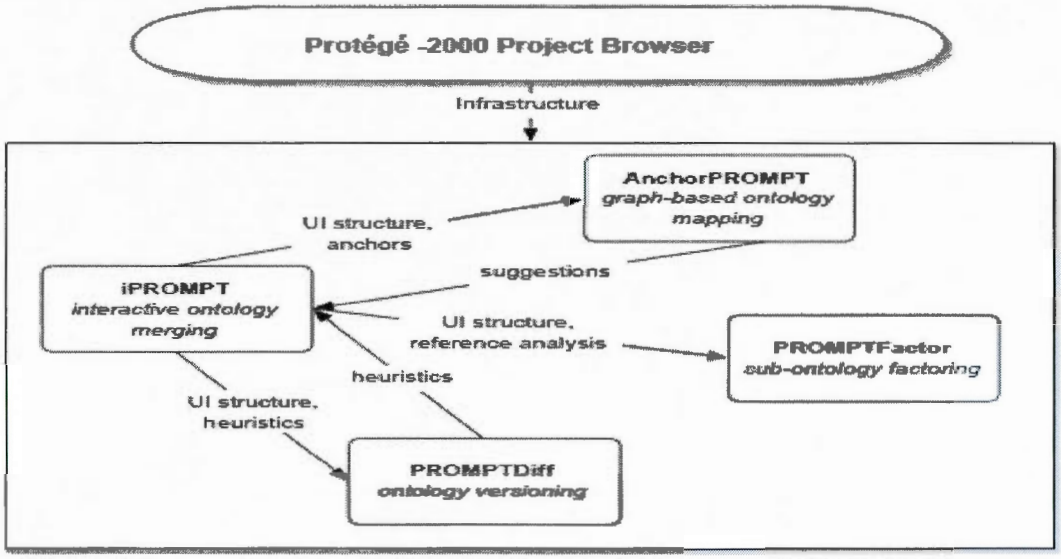


Figure 3.1 Les interactions entre les modules de *PROMPT Suite* (Noy, F. Natalya et Musen, 2003)

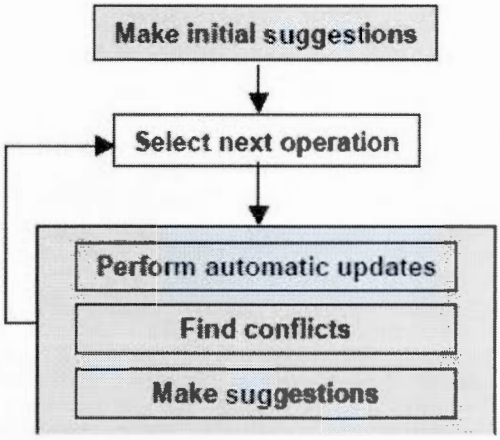


Figure 3.2 Les étapes de l'algorithme de iPROMPT (Noy, N. Fridman et Musen, 2003)

La première étape de l'algorithme prend en entrée deux ontologies et crée une liste de suggestions initiales des correspondances basées sur l'égalité lexicale des noms des concepts comme l'illustre la figure 3.3 (Noy, Natalya Fridman et Musen, 2000). Par la suite, l'algorithme passe à l'étape suivante où l'utilisateur exécute une opération de

son choix. Cette opération est une tâche dans l'algorithme qui s'effectue suite à l'intervention humaine. Le choix de l'opération est réalisé soit en sélectionnant l'une des suggestions, soit en spécifiant l'opération désirée à l'aide de l'environnement d'édition d'ontologie. La prochaine étape d'iPROMPT exécute automatiquement les modifications sous la base de l'opération choisie précédemment. Ensuite, iPROMPT génère de nouveau une liste de suggestions en s'articulant sur la structure de l'ontologie et sur les incohérences et les problèmes dégagés après l'exécution de l'opération. Finalement, iPROMPT propose des solutions pour ces problèmes (Noy, Natalya F. et Musen, 2003) (Noy, Natalya Fridman et Musen, 2000).

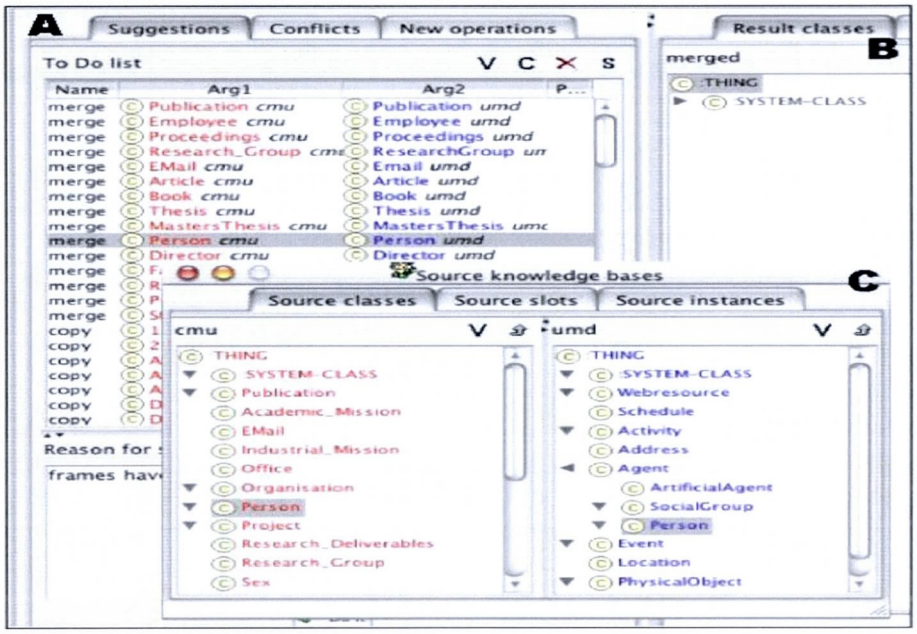


Figure 3.3 L'interface de génération des suggestions initiales (Noy, N. Fridman et Musen, 2003)

iPROMPT propose des opérations de base pour la fusion et l'alignement des ontologies (Noy, N. Fridman et Musen, 2003):

- La fusion de deux concepts – Pour fusionner deux concepts A et B, iPROMPT crée un nouveau concept M dans l'ontologie Om. Pour chaque super/sous-

concept C, son image C_i dans Om devient un super/sous-concept de M. Aussi, pour chaque attribut S attaché à A ou B, si S_i est une image de S dans Om, alors S_i sera attaché à M, sinon S sera copié dans Om. À la fin de chaque opération, iPROMPT relève les incohérences et les problèmes, et fournit une liste de suggestions correctives. En plus, les suggestions d'iPROMPT sont basées sur la sémantique des concepts et leurs positions dans l'ontologie fusionnée.

- La fusion de deux relations – Supposons qu'on a deux relations S_1 et S_2 à fusionner, iPROMPT crée une relation Sm (fusion de S_1 et S_2). Pour chaque concept C_i existant dans le domaine et le codomaine (en anglais: *range*) de S_1 et S_2, C_i est ajouté au domaine et codomaine de Sm dans Om.
- La fusion de deux instances – À partir de deux instances I_1 et I_2, iPROMPT crée une instance Im dans Om. Si les concepts C_1 et C_2 ont les instances I_1 et I_2 respectivement et que ces deux concepts n'ont pas d'image dans Om, alors C_1 et C_2 seront copiés dans Om. Si C_1 et C_2 ont des images dans Om et que les images qui existent dans Om sont différentes de celles qui seront copiées, il y aura alors fusion des images, mais l'utilisateur doit confirmer la fusion.

Le logiciel Protégé-2000 figure parmi les meilleurs logiciels de gestion des ontologies existantes. L'efficacité de ce logiciel repose sur l'efficacité des outils qu'il intègre (comme *PROMPT Suite* présenté précédemment). La fusion avec *PROMPT Suite* est très cohérente en termes d'interaction entre les sous-modules faisant partie intégrante de cet outil, et eu égard à la richesse de l'algorithme (incluant la détection des incohérences et la proposition des solutions). Cependant, le module iPROMPT présente quelques limites: (1) la semi-automatisation de l'algorithme de fusion, puisqu'il inclut une étape de sélection des opérations exécutées par l'utilisateur, une étape clé du processus de fusion, (2) on retrouve le même défaut pour certaines confirmations des tâches, et par conséquent, iPROMPT ne possède pas l'intelligence suffisante pour exécuter certaines tâches, (3) iPROMPT prend en considération la

structure de l'ontologie, mais ne prend pas en compte le traitement des relations entre les concepts ainsi que la pertinence des concepts ou la génération de nouvelles abstractions essentielles pour une meilleure structure ontologique.

3.1.2 SAMBO: système d'alignement et de fusion des ontologies biomédicales

SAMBO est un outil basé essentiellement sur un *framework* développé à l'aide des stratégies applicables sur les ontologies du domaine biomédical (Lambrix et Tan, 2006).

Le système SAMBO (Lambrix et Tan, 2006) prend en entrée deux ontologies de type OWL. La fusion passe par une phase d'alignement comportant deux étapes: l'alignement des relations et l'alignement des concepts. SAMBO est semi-automatique puisqu'il possède un système générateur de suggestions qui nécessite l'intervention humaine pour sélectionner le comparateur désiré (comme WordNet, terminologies lexicales, hiérarchie). Pour chacune des propositions, l'utilisateur doit juger si les termes sont équivalents. Dans ce cas, un nouveau nom sera créé; sinon, la suggestion sera rejetée. Aussi, à toutes les étapes, SAMBO offre une interface afin que l'utilisateur puisse visualiser l'ontologie sous forme d'arbre. L'étape d'alignement suit une stratégie bien définie illustrée dans la figure 3.4. Elle comporte plusieurs comparateurs (linguistique et terminologique, structurel basé sur les instances) ainsi que des comparateurs basés sur les restrictions (contraintes). Ces comparateurs sont combinés à une étape de la stratégie pour donner de meilleurs résultats en matière de calcul de similarité (Lambrix et Tan, 2006). À la fin de l'algorithme, des suggestions contenant les résultats d'alignement sont proposées à l'utilisateur qui peut soit les accepter, soit les rejeter. En cas d'acceptation, un vérificateur de conflits entre les relations d'alignement se déclenche pour détecter et corriger les incohérences. À la fin de l'algorithme, l'alignement est présenté sous forme d'un ensemble de relations entre les différents termes des ontologies d'entrées (Lambrix et Tan, 2006).

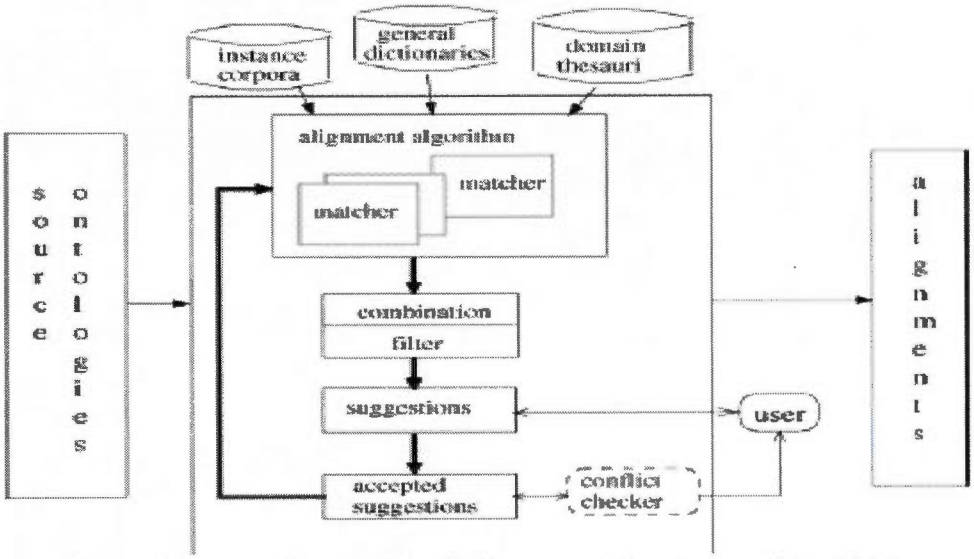


Figure 3.4 La stratégie d'alignement (Lambrix et Tan, 2006)

Après l'alignement, l'algorithme de fusion débute (figure 3.5). Le processus de fusion commence par la création d'une nouvelle ontologie en prenant en entrées les ontologies sources et la liste d'alignement finale. Le système fusionne les termes de la liste d'alignement. Ensuite, il effectue les changements supplémentaires tout en interagissant avec l'utilisateur pour la vérification. À la fin, l'algorithme de fusion prend les autres termes des deux ontologies et les copie dans la nouvelle ontologie. Par ailleurs, SAMBO utilise un raisonneur pour permettre à l'utilisateur de vérifier la cohérence et les satisfiabilité de la nouvelle ontologie (Lambrix et Tan, 2006).

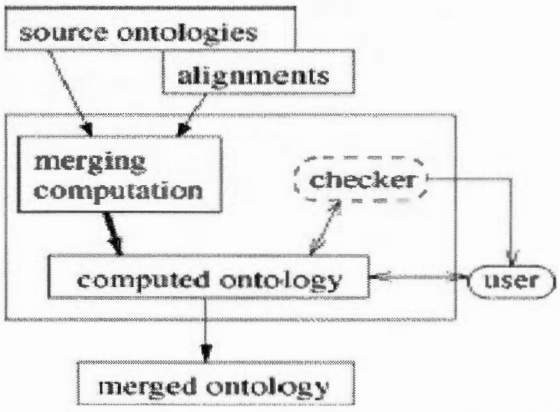


Figure 3.5 L'algorithme de fusion de SAMBO (Lambrix et Tan, 2006)

Le processus de fusion de SAMBO présente cependant quelques limites. La première limite concerne l'intervention humaine pour la vérification des conflits. Cela existe également dans l'alignement avec la présence d'une liste de suggestions destinées à l'utilisateur (comme dans PROMPT). La deuxième limite concerne l'absence de traitement des super/sous-concepts c'est-à-dire que l'algorithme ne traite pas la notion des super/sous-concepts (en d'autres termes, l'abstraction des concepts). De plus, à la fin de l'algorithme de fusion, il existe une forme de copier-coller des termes, dans l'ontologie finale, qui n'ont pas été alignés, et cela se réalise sans aucune vérification et sans aucun traitement sémantique. Donc, cela peut influer sur le domaine cible de l'ontologie fusionnée et créer une incohérence au sein de cette ontologie en termes de concept et de relation. Par exemple, si SAMBO copie un concept non pertinent dans l'ontologie fusionnée, il y aura alors une incohérence dans l'ensemble des concepts.

3.1.3 La classification hiérarchique pour la fusion automatique des ontologies

L'approche de fusion automatique des ontologies par classification hiérarchique est définie par O. Boussaid et ses collègues (Maiz, Boussaid et Bentayeb, 2008). Elle est basée essentiellement sur la hiérarchie des concepts. Le but principal de cette approche est d'enrichir le domaine des entrepôts des données. Cette approche

présentée dans la figure 3.6 peut prendre comme paramètres d'entrées plus que deux ontologies. Cette méthode de fusion repose sur quatre étapes (Maiz *et al.*, 2008):

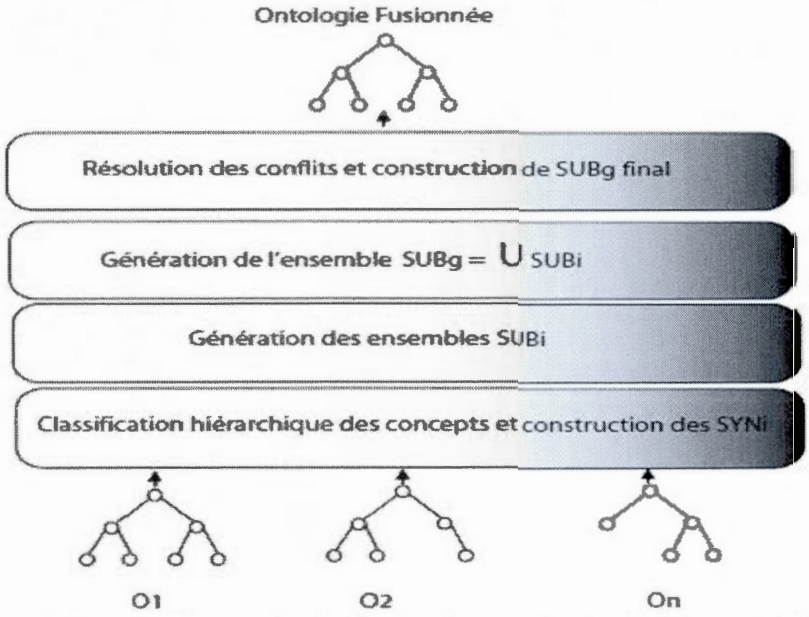


Figure 3.6 L'approche de fusion des ontologies (Maiz *et al.*, 2008)

La première méthode consiste à regrouper les concepts synonymes de différentes ontologies à fusionner. Chaque groupe de concepts équivalents correspond à une classe et par conséquent, le résultat est donc N classes noté SYN_i. Pour effectuer cette étape, un algorithme de classification hiérarchique est nécessaire ainsi qu'une technique de mesure de similarité (distance) pour connaître la similarité entre chaque paire de concept. La technique utilisée génère une matrice de similarité intégrée au sein de l'algorithme de classification. Par ailleurs, la matrice de similarité utilisée représente les concepts de différentes ontologies. Après la classification, s'ensuit la nommage de chaque classe avec un nom représentatif.

La deuxième étape qui suit la détermination des SYN_i comporte deux tâches: la première consiste à dégager le SUB qui est l'ensemble de toutes les paires (parent,

enfant) des ontologies. SUB_i est construit par un parcours normal des hiérarchies de chaque ontologie. En effet, chaque nœud parent trouvé sera pris avec son enfant, etc. Le résultat est un ensemble P de SUB_i. La deuxième tâche de cette étape consiste en la fusion par l'union de tous les SUB_i: $SUB_g = U_{i=1....p} SUB_i$. Cette opération peut engendrer beaucoup de redondance dans SUBg puisque la fusion est effectuée sans vérification lexicale ou sémantique.

La troisième étape consiste à raffiner l'ensemble SUB_g à l'aide des classes SYN_i. Ce raffinement s'effectue en remplaçant chacun des concepts par le concept général correspondant. Il en résulte des redondances au niveau des paires de concepts. Ces redondances sont éliminées en conservant une seule occurrence suite à un simple parcours de SUB_g.

La dernière étape correspond à la construction de l'ontologie finale. La génération de l'ontologie fusionnée se réalise de la façon suivante: (1) parcourir SUB_g jusqu'au concept racine, (2) chercher le deuxième élément de la paire qui possède le concept racine. [ce deuxième élément sera le fils direct de la racine et cette paire sera marquée], (3) continuer à chercher s'il y a un autre fils du concept racine, et s'il existe, on l'ajoute et on le marque. Il faut répéter l'opération (3) jusqu'à la dernière paire qui contient la racine comme premier élément. Ensuite, on prend le premier fils et on procède de la même façon jusqu'à ce qu'on arrive à marquer toutes les paires de concepts (Maiz *et al.*, 2008).

L'approche de fusion automatique des ontologies par classification hiérarchique est, de manière algorithmique, complète. Mais en se focalisant sur la méthode de fusion, on remarque qu'il existe quelques défauts dans cette approche. En premier lieu, il faut noter l'absence d'analyse sémantique pour comparer les concepts du point de vue sémantique, ce qui peut conduire à une augmentation des redondances. En deuxième lieu, une autre limite majeure tient à la non-exploitation des relations qui relient les

concepts. Donc, l'absence d'analyse relationnelle minimise la qualité ontologique du point de vue sémantique.

3.1.4 OWLDiff: un outil pratique pour la comparaison et la fusion des ontologies OWL

OWLDiff (Kremen, Smid et Kouba, 2011) a été introduit comme *plug-in* à Protégé-2000 et à NeOn Toolkit pour les enrichir avec des technologies de fusion et de comparaison. Aussi, il est implémenté en Java comme une technologie *open source*. Le but de cet outil est d'enrichir la gestion des ontologies et surtout d'effectuer leurs mises à jour. OWLDiff est un outil intéressant pour les utilisateurs qui modifient fréquemment les ontologies (Kremen *et al.*, 2011).

OWLDiff utilise essentiellement un utilitaire appelé Diff qui permet de vérifier les changements syntaxiques et sémantiques des deux ontologies OWL entrées comme arguments. L'une de ses ontologies est nommée par exemple *originale* et l'autre *mise à jour*. Par la suite, l'outil OWLDiff exploite un raisonneur pour vérifier si les deux ontologies d'entrées sont similaires. Aussi, OWLDiff offre une interface graphique pour visualiser les différences dans le cas où les ontologies ne sont pas similaires. Ainsi, ces différences sont bien celles qui doivent être mises à jour dans l'ontologie fusionnée (Kremen *et al.*, 2011).

L'outil de fusion de OWLDiff inclut en premier lieu une comparaison de deux ontologies à l'aide de deux algorithmes et ce, dans le but de mettre en relief les différences. Le premier algorithme permet de trouver les axiomes (ou les éléments ontologiques) qui manquent dans les ontologies et de fixer les axiomes qui ont été ajoutés ou modifiés. Le deuxième algorithme touche des aspects plus complexes, et il est divisé en deux étapes (Kremen *et al.*, 2011):

- La première étape est la Diff syntaxique et sémantique où l'algorithme construit quatre listes d'axiomes. Il ajoute tous les axiomes de l'ontologie

mise à jour dans une liste nommée UpdateRest. Aussi, pour l'ontologie originale, le même principe est appliqué en utilisant une liste d'axiomes appelé OrigRest.

- La deuxième étape correspond à une suite d'inférences. L'algorithme vérifie si les axiomes d'UpdateRest sont impliqués dans l'ontologie originale. Dans l'affirmatif, il les déplace dans une liste nommée PossiblyRemove, ce qui élimine la redondance qui peut existée dans l'ontologie originale. Par la suite, il y aura le même travail avec la liste OrigRest pour relever à la fin une liste nommée Présume qui contient des axiomes couverts dans l'ontologie mise à jour. La figure 3.7 montre un exemple de différences générées par OWLDiff:

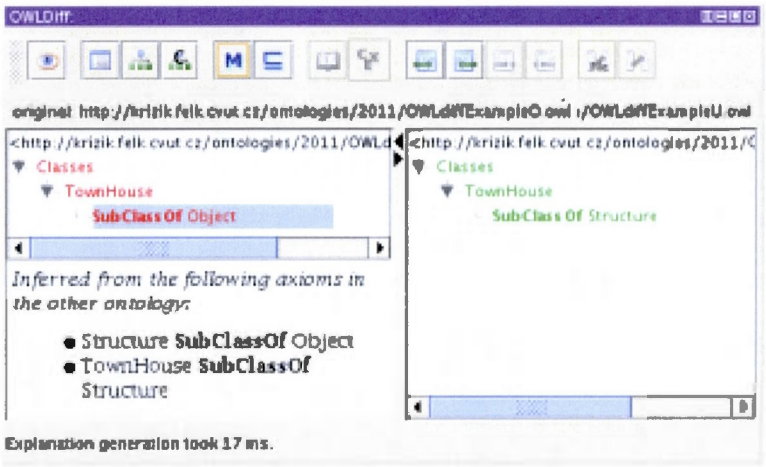


Figure 3.7 Exemple de génération d'une différence avec OWLDiff (Kremen *et al.*, 2011)

Après la détection des différences et la comparaison des deux ontologies, on en arrive à la phase de fusion. Cette phase se réalise à partir d'une interface où l'utilisateur contrôle tout. Donc, par défaut, l'ontologie fusionnée contient tous les axiomes de l'ontologie mise à jour et aucun axiome de l'ontologie originale. En effet, la fusion des ontologies est réalisée avec l'inclusion des axiomes de l'ontologie originale dans l'ontologie fusionnée et l'exclusion des axiomes de l'ontologie fusionnée. Par

conséquent, la fusion dépend de l'utilisateur puisque celui-ci gère toutes les opérations à ce niveau. La figure 3.8 illustre un exemple de fusion avec l'inclusion de «objet» et l'exclusion de «structure» (Kremen *et al.*, 2011).

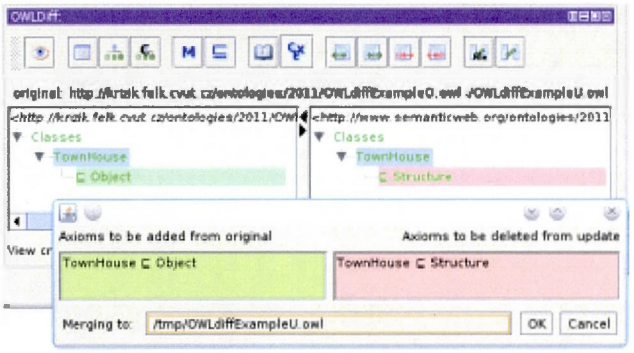


Figure 3.8 Exemple de fusion avec inclusion et exclusion (Kremen *et al.*, 2011)

OWLDiff est un outil intéressant en termes de comparaison des ontologies et de détection des similarités entre les différents éléments d'une ontologie. Cependant, OWLDiff présente quelques limites en tant qu'outil de fusion des ontologies. Avec OWLDiff, la fusion se réalise avec une intervention majeure de l'utilisateur, ce qui remet en cause l'automatisation et l'intelligence de l'approche. En outre, comme avec PROMPT, il y a absence totale de traitement des abstractions des concepts (le cas échéant) et de traitement des concepts pertinents, ces deux notions étant très importantes pour garantir la pertinence de l'ontologie fusionnée. Aussi, OWLDiff n'exploite pas les relations interconcepts, ce qui diminue le niveau de l'analyse sémantique des ontologies.

3.1.5 FCA-Merge: approche ascendante de fusion des ontologies

Comme son nom l'indique, l'approche FCA-Merge de Stumme et ses collèges (Stumme et Maedche, 2001) est basée sur l'analyse formelle des concepts. Le mécanisme de cette approche s'articule sur l'application spécifique des instances des deux ontologies d'entrées. La méthode de fusion prend comme entrées les deux

ontologies et un ensemble de documents D écrits en langage naturel. Ces documents doivent être pertinents pour les deux ontologies puisqu'ils seront utilisés par la suite pour l'extraction des instances. La figure 3.9 montre le processus de fonctionnement de FCA-Merge. Ce processus est composé de trois étapes: (1) l'extraction des instances et la génération des deux contextes, (2) l'application de l'algorithme de AFC et le calcul du treillis des concepts, et (3) la génération de l'ontologie fusionnée à partir du treillis des concepts (Stumme et Maedche, 2001).

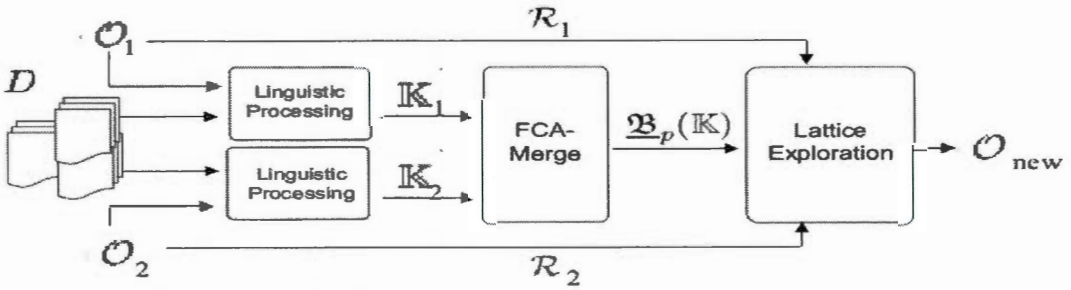


Figure 3.9 La démarche de FCA-Merge pour la fusion des ontologies (Stummeet Maedche, 2001)

FCA-Merge comporte trois étapes. La première étape correspond à l'analyse linguistique et à la génération de deux contextes formels pour chacune des ontologies sources. Dans le contexte formel k:= (G_i, M_i, I_i), l'ensemble des documents D sera pris comme objet (G_i:= D) et les concepts seront (M_i:= C_i). Mais le plus difficile est de générer les relations binaires I_i. La relation (g,m)$\in I_i$ doit avoir un document g qui contient une instance de m. De ce fait, la génération des relations binaires s'effectue à l'aide des techniques d'analyse linguistique comme SEMS (en anglais: *Saarbrücken Message Extraction System*) (Stumme et Maedche, 2001). Par exemple, «*si un concept hôtel est dans l'ontologie 1 et le document g contient l'expression hôtel, alors la relation sera (g, hôtel)*$\in I_1$» [notre traduction]. Aussi, le contexte formel admet la transitivité de la relation is-a; par exemple, «*si (g, hôtel)* $\in I_1$ *et hôtel est une accommodation, donc le document décrit aussi le concept accommodation (g,*

accommodation) $\in I_1$» [notre traduction]. La figure 3.10 illustre un exemple de deux contextes formels K_1 et K_2 dans le domaine du tourisme (Stumme et Maedche, 2001).

I_1	Vacation	Hotel	Event	Concert	Root
doc1	X	X	X	X	X
doc2	X	X	X		X
doc3	X		X		X
doc4	X	X	X	X	X
doc5			X	X	X
doc6		X			X
doc7		X			X
doc8	X		X	X	X
doc9	X	X	X		X
doc10	X	X	X		X
doc11	X	X	X	X	X
doc12		X			X
doc13		X	X	X	X
doc14	X	X	X		X

I_2	Hotel	Accommodation	Musical	Root
doc1	X	X	X	X
doc2	X	X		X
doc3	X	X	X	X
doc4	X	X	X	X
doc5			X	X
doc6	X	X		X
doc7	X	X		X
doc8	X	X	X	X
doc9	X	X		X
doc10	X	X		X
doc11	X	X	X	X
doc12	X	X		X
doc13	X	X	X	X
doc14	X	X		X

Figure 3.10 Exemple de deux contextes formels (Stumme et Maedche, 2001)

La deuxième étape correspond à la génération du treillis des concepts élagué à partir des deux *inputs* qui seront les deux contextes formels K_1 et K_2. Une étape de FCA-Merge correspond à éliminer les ambiguïtés entre les concepts puisqu'il peut y avoir une redondance du concept dans les deux ontologies. Par la suite, il y aura fusion de K_1 et K_2 pour obtenir un nouveau contexte formel K, et à partir de ce dernier, il y aura génération d'un treillis des concepts élagué. En plus, la construction du treillis des concepts élagué est effectuée à l'aide de l'algorithme TITANIC, lequel est cependant légèrement modifié pour déterminer l'élagage. L'avantage de TITANIC tient au fait qu'il utilise la notion d'ensemble de clés où chaque ensemble de clés décrit un concept formel. Dans FCA-Merge, l'ensemble des clés peut servir (1) à une indication sur le concept formel généré et (2) à une indication sur les noms des nouveaux concepts. La figure 3.11 montre un simple treillis des concepts élagué qui correspond à une partie de l'exemple de la figure 2.7 (Stumme et Maedche, 2001).

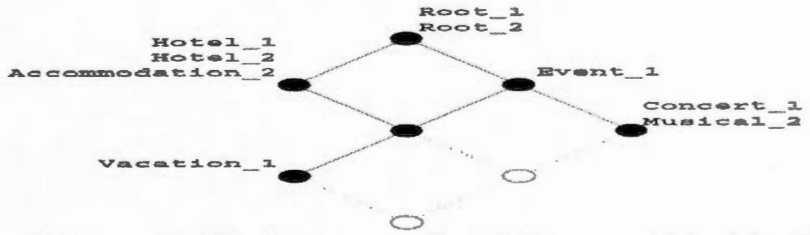


Figure 3.11 Treillis des concepts élagué (Stumme et Maedche, 2001)

La troisième et dernière étape représente la génération de la nouvelle ontologie à partir du treillis des concepts élagué. Alors que les deux étapes précédentes sont totalement automatiques, la troisième étape nécessite l'intervention humaine. Les concepts formels du treillis sont tous des candidats pour être des concepts ou des relations dans la nouvelle ontologie. Donc, FCA-Merge utilise quelques requêtes pour dégager ceux qui sont pertinents. Pour chaque concept formel, il existe quatre cas possibles pour vérifier s'il est pertinent et pour l'introduire dans l'ontologie fusionnée: (1) un concept formel comporte exactement un ensemble de clés de cardinalité 1, (2) un concept formel dispose de deux ou plusieurs ensembles de clés de cardinalité 1, (3) un concept formel ne dispose pas d'un ensemble de clés de cardinalité 0 ou 1, et (4) un concept formel comporte un ensemble de clés vide. Donc, la génération de l'ontologie finale exploite les cardinalités des ensembles de clés des concepts formels qui sont créés par l'algorithme TITANIC. Après le traitement des concepts formels, il s'effectuera une copie des relations de la première et de la deuxième ontologie dans l'ontologie finale, mais à ce moment-là, certains conflits et redondances peuvent survenir, le pouvoir de les résoudre revenant à l'utilisateur (Stumme et Maedche, 2001).

FCA-Merge présente quelques limites: (1) il n'existe pas d'analyse sémantique., FCA-Merge ne recourant en effet qu'à une analyse linguistique pour détecter la correspondance entre les concepts et les documents en langage naturel, (2) au niveau des contextes formels, il n'y a plus de traitement des relations des deux ontologies originales au sein de l'algorithme de fusion [en effet, le copier-coller des relations

risque de minimiser la définition du domaine cible de l'ontologie fusionnée], et (3) l'intervention d'un expert est requise pour régler certains problèmes. Avec FCA-Merge, l'intervention humaine se présente essentiellement au niveau de la résolution des conflits générés suite aux copier-coller des relations. On en conclut donc que l'algorithme de FCA-Merge est semi-automatique.

3.1.6 FCA-OntMerge: méthode de fusion des ontologies basée sur l'analyse formelle des concepts

FCA-OntMerge est une approche de fusion des ontologies développée par Guan-yu, Shu-peng et Zhao-Yan (2010). Elle est basée essentiellement sur l'analyse formelle des concepts. Selon Guan-yu *et al.* (2010), cette approche a résolu les problèmes sémantique, augmenté la flexibilité et minimisé l'intervention humaine. L'algorithme de FCA-OntMerge comporte quatre principales étapes: (1) convertir les deux ontologies sources dans le même format bien défini, (2) générer le (les) contexte(s) formel(s) à partir des ontologies sources, (3) faire la correspondance (en anglais: *mapping*) des attributs dans le contexte formel, et (4) fusionner les contextes formels et générer le treillis des concepts correspondant.

La première étape de l'algorithme configure les ontologies sous un seul format OWL. L'outil de gestion des ontologies protégé peut être utilisé pour unifier ces ontologies et ce, pour éviter la mauvaise structuration des ontologies (Guan-yu *et al.*, 2010).

La deuxième étape prend comme entrées les deux ontologies unifiées et génère les contextes formels. La génération des contextes formels s'effectue après utilisation de l'outil Jena (qui existe aussi comme API Java) qui permet l'analyse des ontologies. Avec Jena, il y aura extraction des concepts et des attributs qui seront respectivement des objets et des attributs dans les contextes formels. La figure 3.12 montre un exemple de deux ontologies et le double tableau 3.1 illustre les concepts formels correspondants. Au niveau de ces tableaux, lorsqu'un concept possède un attribut, on met 1 dans le cas correspondant, sinon, on met 0 (Guan-yu *et al.*, 2010).

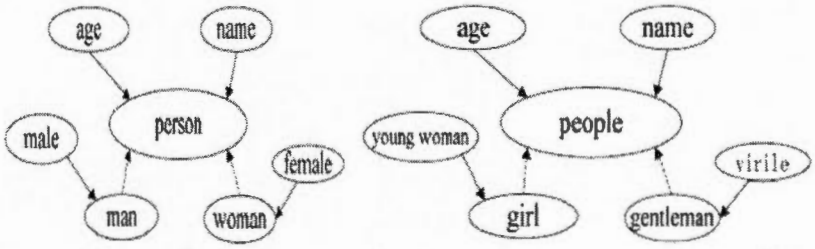


Figure 3.12 Exemple de deux ontologies: onto1 person.owl et onto2 people.owl (Guan-yu *et al.*, 2010)

Tableau 3.1 Les contextes formels de person.owl et de people.owl (Guan-yu *et al.*, 2010)

	name	*age*	*male*	*female*
person	1	1	0	0
man	1	1	1	0
woman	1	1	0	1

	name	*age*	*he*	*young_woman*
people	1	1	0	0
girl	1	1	0	1
gentleman	1	1	1	0

La troisième étape de l'algorithme de fusion détermine la correspondance entre les attributs des deux ontologies. Ce processus s'effectue à l'aide de WordNet. Le calcul de similarité des chaînes de caractère des attributs est défini selon la distance, c'est-à-dire le nombre d'insertions minimales, le nombre d'éliminations et de remplacements entre deux chaînes. Donc, si la similarité entre les deux chaînes est égale à 1, il y aura fusion des deux attributs. Aussi, en plus des méthodes de calcul des distances, FCA-OntMerge utilise WordNet pour enrichir la détection des similarités, plus précisément

avec les synonymes, les hyperonymes, les antonymes et les hyponymes, et à la fin de cette étape, il y a élimination des attributs redondants (Guan-yu *et al.*, 2010).

La quatrième et la dernière étape correspondent à la fusion des contextes formels. La fusion est réalisée en connectant les deux contextes formels et en utilisant un algorithme de fusion des treillis des concepts puisque le treillis des concepts offre la représentation d'un contexte formel. Le tableau 3.2 présente les contextes formels connectés pour donner un seul contexte formel. À partir de ce dernier contexte formel, il y aura une représentation du treillis des concepts finale qui détermine l'ontologie fusionnée (Guan-yu *et al.*, 2010).

Tableau 3.2 Un contexte formel généré suite à la connexion des deux contextes formels du tableau 3.1 (Guan-yu *et al.*, 2010)

	name	*age*	*male*	*female*	*youngwoman*
person	1	1	0	0	0
man	1	1	1	0	0
woman	1	1	0	1	0
people	1	1	0	0	0
girl	1	1	0	1	1
gentleman	1	1	1	0	0

Quelques critiques de cette approche: (1) il n'y a plus de traitement sur les concepts, il n'existe donc pas d'alignement des concepts ou d'élimination des redondances d'ordre syntaxique ou sémantique [plus précisément, FCA-OntMerge ne génère pas des abstractions, elle conserve seulement les abstractions existantes dans les ontologies sources], (2) il y a absence totale de traitement des relations, d'exploitation des positions des concepts dans leurs ontologies et de prise en compte de la structure des ontologies sources et de l'ontologie fusionnée, (3), une autre limite se présente à la fin: la transition du treillis de concept à une ontologie utilisable et complète.

3.1.7 FFCA: la fusion des domaines ontologiques basés sur le système de WordNet et sur les techniques d'analyse formelle des concepts flous

Dans notre projet, nous nous intéressons à la fusion des ontologies en général, c'est-à-dire que nous traitons les éléments ontologiques comme les concepts, les relations, les instances, les attributs, etc. Mais l'approche de FFCA exploite la notion des concepts flous pour fusionner deux domaines ontologiques. De ce fait, comme illustrée dans la figure 3.13, l'approche FFCA est composée principalement de trois étapes: (1) un prétraitement, (2) la fusion des ontologies en trois sous-étapes [fusion des concepts, alignement avec WordNet et alignement avec FFCA], et (3) la génération de l'ontologie floue (Chen, Bau et Yeh, 2011).

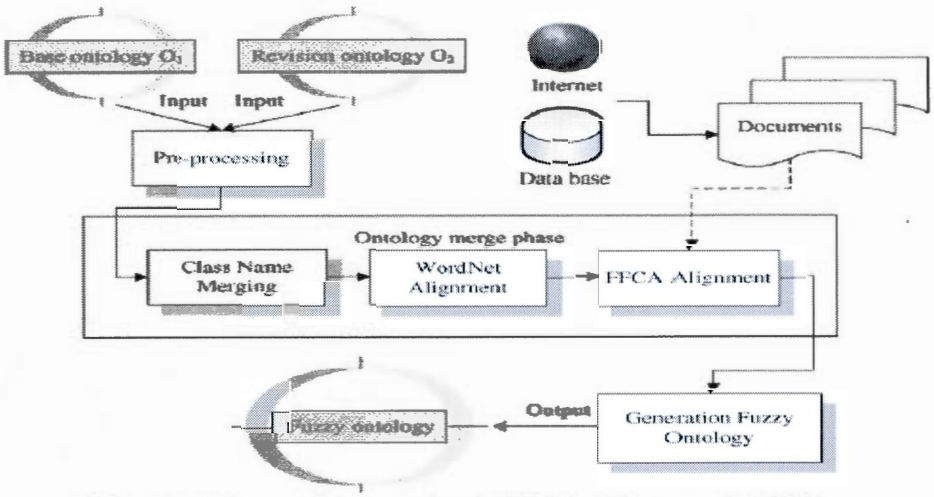


Figure 3.13 L'approche de FFCA (Chen *et al.*, 2011)

L'étape de prétraitement permet d'enrichir les concepts. Le prétraitement prend comme entrées deux ontologies: une ontologie de base O1 et une ontologie révisée O2. Le processus de prétraitement comporte deux phases. La première phase consiste à enrichir les concepts de l'ontologie source. Cela s'effectue à l'aide de WordNet où il y aura conservation des *synsets* et de la définition du concept pour pouvoir comparer les concepts dans les prochaines étapes. La deuxième phase consiste à dégager toutes les variantes des mots (par exemple: «la racine du verbe test inclut toutes les variantes: tests, testé et test» [notre traduction]). Cela peut être effectué

avec l'un des algorithmes appropriés. La figure 3.14 montre le cadre de travail de l'étape de prétraitement avec un exemple de concept flou (Chen *et al.*, 2011).

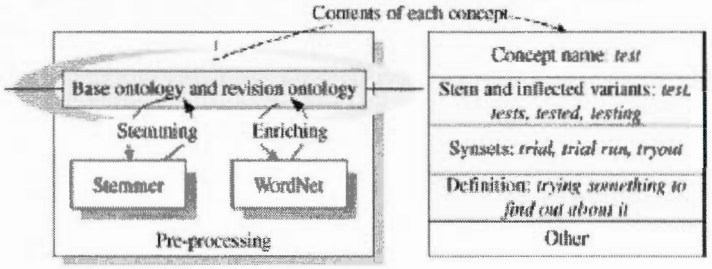


Figure 3.14 Le cadre de travail de l'étape de prétraitement (Chen *et al.*, 2011)

La première étape du processus de fusion détermine la fusion des noms des concepts pour les concepts qui ont les mêmes noms. En s'articulant sur WordNet deux concepts égaux revient à dire qu'ils ont les mêmes noms ou que leurs synset se chevauchent par exemple «si les deux concepts *test* et *tryout* ont des *synsets* qui se chevauchent et que leurs définitions d'équivalence satisfait la définition des concepts, le système va alors les fusionner» [notre traduction] (Chen *et al.*, 2011).

La deuxième étape de méthode de fusion définit l'alignement avec WordNet. À cette étape, les concepts qui restent de l'ontologie de révision seront utilisés pour définir les relations hiérarchiques des concepts dans l'ontologie de base. WordNet admet quelques définitions pour les hyponymes, les hypernoms et les descendants (en anglais: *sibling*). Par la suite, il y aura comparaison des concepts selon les définitions mentionnées ci-dessus du WordNet. Donc, le but principal d'utiliser WordNet est de mieux positionner les concepts dans l'ontologie de base. La figure 3.15 montre un exemple d'alignement avec WordNet (Chen *et al.*, 2011).

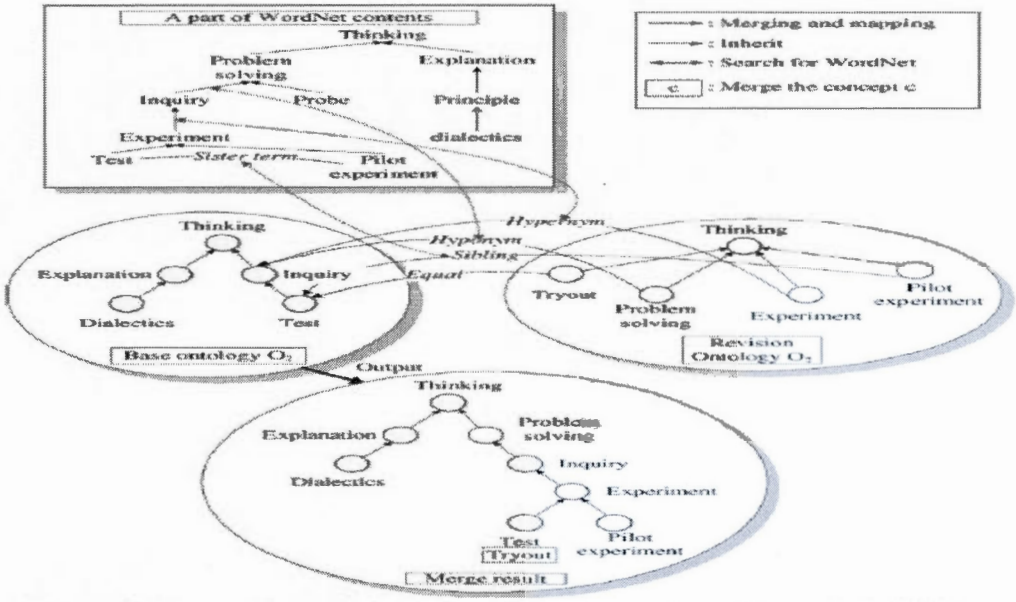


Figure 3.15 Exemple d'alignement avec WordNet (Chen *et al.*, 2011)

La troisième étape du module de fusion correspond à un alignement à l'aide de l'AFC (l'Analyse Formelle de Concept). L'application d'AFC dans l'approche de FFCA revient à faire face aux détails que WordNet n'a pas pu détecter. Donc, l'AFC traite les concepts qui restent après le passage par WordNet. En effet, la première phase du processus d'alignement de l'AFC sera la construction des contextes formels où les objets et les attributs seront respectivement les documents relevés du Web et les concepts. Ce phénomène ressemble à celui de FCA-Merge dans la sous-section avant précédente. Par la suite, il y aura fusion des deux contextes formels pour donner un seul contexte formel combiné. En plus – c'est là une information importante –, c'est que la valeur entre l'objet et l'attribut dans le tableau de contexte formel oscille entre 0 et 1, ce qui est différent par rapport à ce que nous avons présenté précédemment pour FCA-Merge et FCA-OntMerge qui utilisent les valeurs binaires. Donc, le calcul de cette valeur se réalise avec certaines formules qui exploitent les documents Web. Ensuite, après la fusion des contextes formels, il y aura un calcul de similarité entre les concepts formels du contexte formel fusionné (un calcul qui est aussi effectué avec une formule de calcul de similarité). La figure 3.16 montre le processus

d'alignement basé sur AFC. À la fin de ce processus, il y aura génération du treillis des concepts à partir du contexte formel fusionné et aligné (Chen *et al.*, 2011).

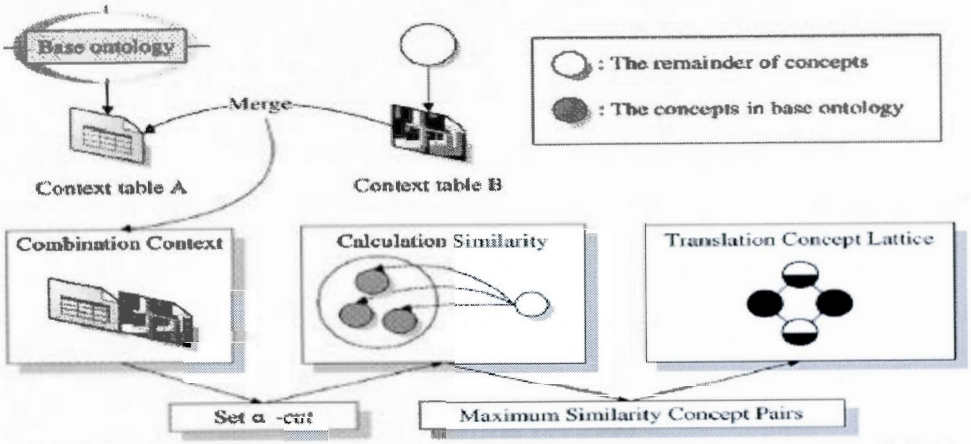


Figure 3.16 Le processus d'alignement avec FCA (Chen *et al.*, 2011)

La dernière étape de cette approche détermine la génération de l'ontologie floue, puisqu'elle contient des concepts flous. L'ontologie floue fusionnée est générée à partir du treillis des concepts. Mais la représentation de ce type de concept est différente du type général des concepts. Un concept flou contient: le super concept, la définition de ce concept, les attributs, les opérations, la valeur d'appartenance de chaque objet pertinent et des informations concernant la valeur entre les supers et les sous-concepts. La figure 3.17 représente les informations d'un concept flou et la figure 3.18 montre un exemple d'ontologie floue (Chen *et al.*, 2011).

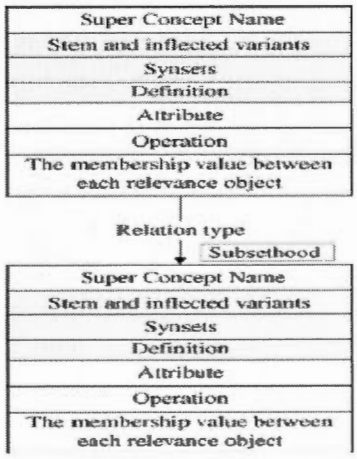


Figure 3.17 Les informations incluses dans un concept flou (Chen *et al.*, 2011)

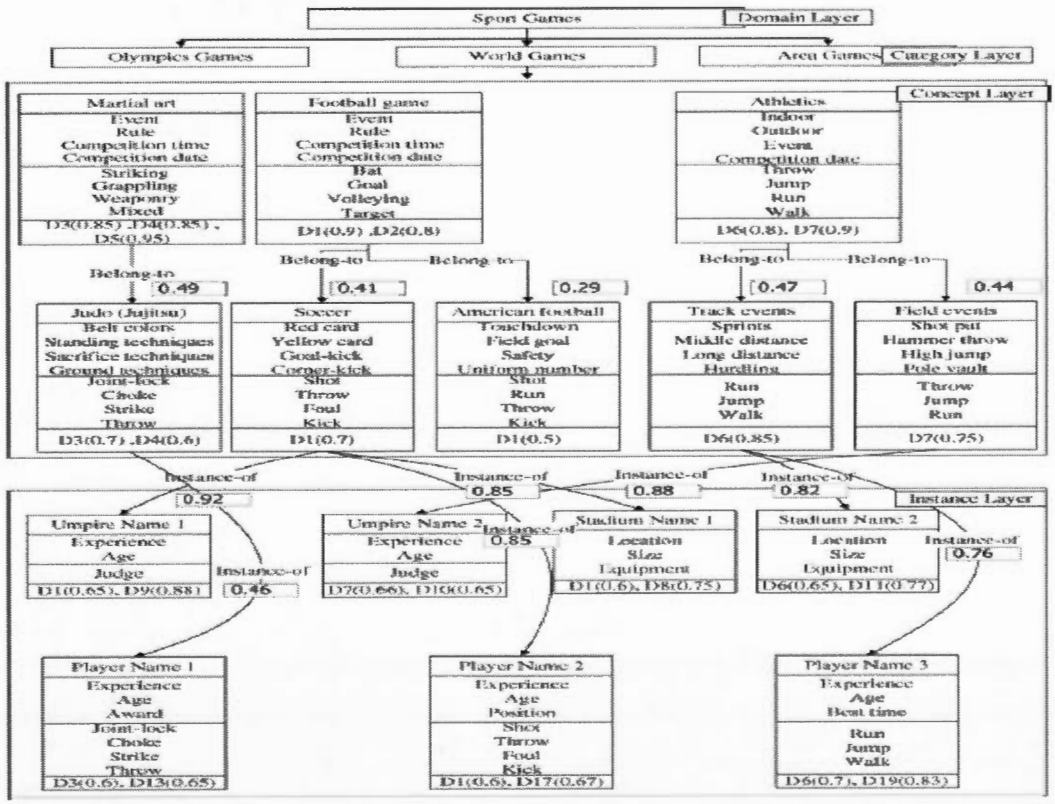


Figure 3.18 Exemple d'une ontologie floue (Chen *et al.*, 2011)

L'approche de FFCA utilise les concepts flous pour créer à la fin une ontologie floue. Mais dans notre projet, nous ne nous intéressons pas à ce type de concept. Ce qui

nous intéresse dans cette approche, c'est le processus de fusion. En effet, FFCA détermine des points forts au niveau de l'étape d'alignement. Cependant, FFCA présente quelques limites sur certains points: (1) à l'étape d'alignement, FFCA n'exploite plus les relations interconcepts des deux ontologies sources, et suite à cette lacune, la valeur de l'analyse sémantique diminue, (2) FFCA ne traite pas les éléments ontologiques pertinents, et plus précisément les concepts pertinents, et (3) on note l'absence de l'inclusion des attributs de chaque concept dans l'analyse sémantique et linguistique à l'étape d'alignement.

3.1.8 Synthèse des approches

Dans cette partie, nous illustrons une synthèse des approches présentées précédemment. Le tableau de synthèse 3.3 résume les avantages et les limites des approches. Par ailleurs, nous avons remarqué certaines limites communes, par exemple: l'absence de génération des abstractions (s'il y a lieu), le non-traitement des relations minimisant l'analyse sémantique, et parfois des copier-coller de certains éléments ontologiques sans passer par une analyse. De plus, nous avons constaté une intervention majeure et parfois excessive de l'utilisateur dans certaines approches. Ce dernier point est considéré comme une limite, eu égard à l'hétérogénéité de la logique humaine pour régler les conflits. C'est ainsi que pour remédier à ce problème, un standard intelligent (qui sera capable de prendre des décisions sans intervention humaine) sera nécessaire.

Tableau 3.3 Tableau de synthèse

	AVANTAGES	INCONVÉNIENTS
PROMPT	• Forte interaction avec les modules de *PROMPT Suite*. • Richesse de l'algorithme de détection des incohérences. • *PROMPT Suite* fournit une forte cohérence au sein de ses modules spécialisés.	• Intervention humaine excessive. • Absence de traitement des relations interconcepts. • PROMPT ne traite pas la notion des concepts pertinents. • Absence de génération des abstractions s'il y a lieu.
SAMBO	• Richesse de l'étape d'alignement avec des algorithmes d'analyse lexicale et sémantique. • Combinaison des différents algorithmes d'alignement pour offrir une seule structure contenant l'alignement final.	• Intervention humaine pour vérifier quelques conflits. • Absence de traitement des relations et de génération des abstractions si elles existent.
Classification hiérarchique pour la fusion automatique	• Aide à créer les entrepôts des données. • Aide à l'analyse des bases de données. • Analyse des hiérarchies ontologiques. • Prend de 1 à n ontologies en entrées. • Complétude de l'algorithme.	• Ignore l'aspect sémantique. • Génère beaucoup de redondances. • Absence de traitement des abstractions et des relations.
OWLDiff	• Vérification des changements syntaxiques et sémantiques des ontologies sources. • Alignement riche avec détection des différences et des similarités.	• Intervention humaine majeure. • Absence de traitement des concepts pertinents. • Absence de traitement des abstractions et des relations entre les concepts.
FCA-Merge	• Analyse des structures avec AFC. • Analyse sémantique des concepts renforcée avec des documents de langage naturel. • Traite les concepts pertinents.	• Absence d'analyse sémantique. • Copier-coller des relations sans étude sémantique. • Absence de traitement des abstractions.
FCA-ontoMerge	• Richesse avec des algorithmes et des outils de détection des similarités. • Exploite les concepts et leurs attributs pour la fusion.	• Non-élimination des redondances. • Absence de traitement des abstractions. • Inexistence d'une étape de transition treillis-ontologie. • Absence de traitement des relations interconcepts.
FFCA	• Utilisation de WordNet pour améliorer l'alignement et résoudre les conflits. • Exploitation de l'AFC pour enrichir l'alignement.	• FFCA ne traite pas les attributs des concepts. • Absence d'analyse relationnelle. • Absence de distinction entre les concepts pertinents /non pertinents.

Dans le chapitre suivant, nous présenterons notre approche RCA-Merge qui permet de remédier à la majorité des limites dégagées. Mais avant de détailler RCA-Merge, nous présenterons les métriques de qualité implémentées dans notre plateforme INUKHUK. Ces métriques seront la base de notre évaluation qui sera détaillée dans le chapitre 5.

3.2 Les métriques de qualité

Les ontologies sont apparues dans plusieurs domaines tels que le Web sémantique, l'intelligence artificielle, l'analyse des bases de données, les domaines médical et biomédical, etc. De ce fait, la complexité structurelle et l'hétérogénéité des ontologies entraînent une diversité au niveau de la qualité de ces dernières, ce qui ne facilite pas la tâche de l'utilisateur lorsque vient le temps de choisir des ontologies appropriées à ses besoins.

Dans la fusion d'ontologies, on assiste parfois à la construction entière d'une ontologie: création d'un nouveau graphe, apparition d'une nouvelle structure syntaxique et sémantique, génération de nouveaux concepts et de nouvelles relations, et création de nouvelles abstractions. Beaucoup de changements. De ce fait, une évaluation de la qualité de l'ontologie est nécessaire pour augmenter la réutilisabilité, pour valider la cohérence linguistique, sémantique et structurelle dans l'ontologie, ainsi que pour fournir à l'utilisateur des ontologies bien évaluées pour lui laisser l'espace de choisir ce qui convient à ses besoins.

Nous présenterons dans la section suivante notre système d'évaluation de qualité ontologique nommé OpenQAM qui est implémenté au sein de la plate-forme INUKHUK. Notre système d'évaluation comporte quatre "systèmes métriques" (qui sont basés essentiellement sur des formules mathématiques) que nous détaillons ci-dessous:

3.2.1 Les métriques d'ORME *et al.*

Selon ORME *et al.*, la validation de la qualité des données de l'ontologie influe directement sur la validation de la qualité du système qui utilise cette ontologie, et par conséquent, l'évolution des données de l'ontologie touche également l'évolution et la maintenance du système basé sur cette ontologie (Orme, Yao et Etzkorn, 2007).

Ainsi, la mesure de la qualité repose essentiellement sur l'exhaustivité et la stabilité des données de l'ontologie et ce, même en cas d'évolution, d'où la proposition de métriques pour mesurer la complexité et la cohésion des données au sein de l'ontologie (Orme *et al.*, 2007).

Tenant compte du nombre de propriétés (lequel nombre donne une indication sur la complexité d'une ontologie ainsi que la couverture des termes utilisés), ORME *et al.* tiennent également compte du nombre de sortants (c'est-à-dire les relations sortantes), de concepts racines et de concepts feuilles, de la mesure de la profondeur maximale de l'héritage, etc. (Orme *et al.*, 2007).

Voici la liste des mesures que nous avons retenues (Orme *et al.*, 2007):

- NOP (*Number Of Properties*): le nombre de propriétés.
- APC (*Average Properties per Class*):
  le nombre moyen de propriétés par concept.
- NOF (*Number Of Fanouts*) et NOFC (*NOF Class*):
  le nombre de sortants dans une ontologie et pour un concept donné.
- AFC (*Average Fanouts per Class*):
  le nombre moyen de sortants par classe.
- AFR (*Average Fanouts of Root Classes*):
  le nombre moyen de sortants des concepts racines.
- AFNL (*Average Fanouts of Non-Leaf classes*):
  le nombre de sortants des concepts non-feuilles.

- MaxDit (*Maximum Depth of Inheritance Tree*):
  retourne la profondeur maximale de l'arbre d'héritage.

3.2.2 Les métriques de TARTIR *et al.*

Les différentes mesures de qualité de TARTIR *et al.* sont toutes intégrées au sein d'une approche appelée OntoQA. C'est une approche qui analyse les éléments ontologiques (concept, relation, etc.) et qui les décrit par un ensemble bien défini de paramètres. Ces paramètres peuvent faire apparaître les principales caractéristiques d'une ontologie (Tartir, Arpinar, Moore, Sheth et Aleman-Meza, 2005).

Aussi, la qualité d'une ontologie peut être examinée sous plusieurs angles. Par exemple, les paramètres (ou mesures) de qualité peuvent être utilisés pour évaluer la qualité du schéma dans la modélisation d'un sous-domaine réel. De plus, la profondeur, la largeur, la hauteur de l'arbre d'héritage et l'équilibre de ce dernier peuvent contribuer à l'évaluation de la qualité. L'ontologie peut être également évaluée selon une autre dimension où il y aura vérification de la richesse et de la précision des concepts et des relations (Tartir *et al.*, 2005).

TARTIR et ces collègues ont proposé des méthodes au sein de l'approche OntoQA pour évaluer la qualité d'une ontologie en se basant sur les différentes dimensions mentionnées ci-dessus. Aussi, la contribution d'OntoQA fournit quelques mesures pour évaluer quantitativement la qualité des composantes (concepts, relations, attributs, etc.) de l'ontologie. Parmi les mesures de TARTIR *et al.*, on retrouve (Tartir *et al.*, 2005):

- RR (*Relationship Richness*): la richesse des relations.
- AR (*Attribute Richness*): la richesse des attributs.
- IR (*Inheritance Richness*): la richesse de l'héritage.
- CR (*Class Richness*): la richesse des classes.
- AP (*Average Population*): la moyenne de population.

3.2.3 Les métriques d'ALANI *et al.*

ALANI *et al.* proposent certaines mesures de qualité dans un prototype appelé AKTiveRank. Ce dernier permet d'évaluer et de classer les ontologies selon certains critères. Parmi les critères de base d'AKTiveRank, on retrouve la meilleure représentation du domaine de l'ontologie et la meilleure structuration des éléments ontologiques. Aussi, ce prototype applique des méthodes pour évaluer la couverture du contenu ou, en d'autres termes, pour évaluer la clarté du contenu de l'ontologie (Alani, Brewster et Shadbolt, 2006) (Alani et Brewster, 2005).

D'après ALANI et ses collègues, il est nécessaire d'évaluer les principaux éléments de l'ontologie. Tout cela sert à fournir une approche multidimensionnelle de classement et d'évaluation. En effet, AKTiveRank est défini comme un système expérimental de classement d'ontologies basé sur quelques mesures qui permettent d'évaluer les ontologies en termes de concepts et de relations. Donc, cela signifie que les ontologies peuvent être considérées comme des graphes sémantiques de concepts et de relations. Par conséquent, les mesures de similarité peuvent être appliquées pour explorer ces graphes conceptuels (Alani *et al.*, 2006).

Suivant les métriques d'ALANI *et al.*, nous avons choisi certaines mesures pour évaluer notre travail (Alani *et al.*, 2006):

- CMM (*Class Match Measure):* mesurer les concepts associés;
- DEM (*Density Measure):* mesurer la densité de l'ontologie;
- SSM (*Semantic Similarity Measure):*
  mesurer la similarité sémantique de l'ontologie;
- BEM (*Betweenness Measure):* mesurer les chemins entre les concepts.

3.2.4 Les métriques de STAAB *et al.*

STAAB *et al.* fournissent une nouvelle approche d'évaluation des ontologies et ce, en utilisant d'autres dimensions comme l'évaluation des termes lexicaux, le rappel

lexical et l'évaluation de la taxonomie. Aussi, ils essayent (STAAB *et al.*) de minimiser le plus possible les erreurs les plus courantes (Dellschaft et Staab, 2006).

Le présent travail se concentre sur deux axes essentiels: le premier concerne l'évaluation de la couche lexicale des termes, le deuxième concerne l'évaluation de la hiérarchie des concepts. En effet, la mesure lexicale des termes est souvent réalisée suivant une comparaison entre l'ontologie adaptée et l'ontologie de référence. Plus précisément, la comparaison lexicale des termes est basée sur la correspondance exacte des chaînes ou sur la distance d'édition entre deux chaînes. Donc, il est plus pertinent d'utiliser la deuxième méthode, car elle résout les petits problèmes d'orthographe et les erreurs de frappe (par exemple: «Bonjour» et «bonjour») (Dellschaft et Staab, 2006).

L'évaluation de la hiérarchie des concepts est définie par la représentation d'une famille de précision taxonomique et par des mesures de rappel taxonomique. À ce niveau, les mesures sont divisées en deux: (1) une mesure locale qui représente la similitude de deux concepts, cette mesure étant calculée sur la base des caractéristiques extraites de la hiérarchie des concepts, et (2) une mesure globale qui intègre la mesure taxonomique locale, cette dernière offrant une influence majeure sur les propriétés de la mesure globale. On peut alors conclure que la mesure locale compare les positions des deux concepts et que la mesure globale compare deux hiérarchies de concepts (Dellschaft et Staab, 2006).

Nous avons utilisé les mesures LP, LR, TP et TR pour estimer le rappel et la précision (Dellschaft et Staab, 2006).

- LP (*Lexical Precision*): la précision lexicale.

$$LP(Oc, Or) = \frac{|Cc \cap Cr|}{|Cc|}$$

Avec : Oc est l'ontologie évaluée et Cc les concepts de Oc
Or est l'ontologie de référence et Cr les concepts de Or.

- LR (*Lexical Recall*): le rappel lexical.

$$LP(Oc, Or) = \frac{|Cc \cap Cr|}{|Cr|}$$

- TP (*Taxonomic Precision*): la précision taxonomique.

$$TP(Oc, Or) = \frac{1}{|Cc|} \sum_{c \in Cc} \begin{cases} tp(c, c, Oc, Or) & \text{if } c \in Cr \\ 0 & \text{if } c \notin Cr \end{cases}$$

Avec : *tp* est un élément de calcul de similarité de deux concept en considérant la hiérarchie des concepts.

- TR (*Taxonomic Recall*): le rappel taxonomique.

$$TR(Oc, Or) = TP(Or, Oc)$$

- TF (*Taxonomic F-Measure*): définit la moyenne harmonique entre TP et TR.
- TF' (*Taxonomic F'-Measure*): définit la moyenne harmonique entre TF et LR.

3.2.5 Synthèse des métriques

Dans cette section, nous avons présenté les différentes métriques développées dans INUKHUK. De ce fait, ces métriques seront utilisées pour l'évaluation des ontologies issues de notre mécanisme de fusion des ontologies. Notre système d'évaluation comporte différentes approches comme l'approche fondée sur la comparaison à d'autres ontologies avec TARTIR *et al.*. Cette approche analyse les schémas d'ontologies à l'aide de l'ensemble de mesures que nous avons déjà citées à 3.2.2. L'approche de STAAB *et al.* est fondée sur la comparaison par rapport à une référence où il y aura une analyse lexicale ainsi qu'une analyse conceptuelle avec une comparaison de structures sémantiques. L'approche d'ALANI *et al.* est fondée sur

l'utilité, elle touche spécifiquement l'aspect sémantique de l'ontologie. Ces différentes approches nous permettront de mettre en évidence les aspects qualitatifs et quantitatifs pour déterminer la qualité de l'ontologie fusionnée.

Dans ce qui suit, nous présenterons notre processus de fusion d'ontologies RCA-Merge et nous détaillerons les étapes de développement avant et après l'amélioration tout en mentionnant les outils technologiques connexes qui nous ont aidés à réaliser RCA-Merge.

CHAPITRE IV

RCA-MERGE: LA FUSION DES ONTOLOGIES BASÉE SUR ARC

Dans ce chapitre, nous détaillerons notre approche sur le plan théorique en présentant le processus de fusion, les étapes du processus et les améliorations effectuées. Par la suite, nous aborderons le plan pratique avec des brèves définitions des technologies d'implémentation, nous détaillerons aussi les modules du processus de fusion et les ajouts réalisés au sein des améliorations.

4.1 L'approche de fusion des ontologies: RCA-Merge

Dans cette section, nous présentons notre approche RCA-Merge. Ensuite, nous aborderons l'ajout d'un module de préalignement dans notre approche, et à la fin, nous expliquerons l'importance du nommage dans la fusion des ontologies.

4.1.1 Le processus de base de RCA-Merge

Notre travail est fondé sur une approche de fusion des ontologies par le biais de l'ARC: l'Analyse Relationnelle des Concepts. Comme nous l'avons défini dans la section 2.4, l'ARC, offre un cadre formel et conceptuel pour supporter le processus global de la fusion. Dans notre approche, le but principal de l'ARC (Rouane, H. Mohamed, Dao, Huchard et Valtchev, 2007) est d'exploiter les liens (ou les relations) qui relient les concepts formels. De plus, l'ARC fournit une factorisation maximale des éléments d'un modèle (dans notre cas, un modèle est une ontologie) puisque tous les éléments d'une ontologie sont pris en considération dans la généralisation (comme les concepts, les propriétés, les relations, etc.). Il s'ensuit une meilleure abstraction

avec rassemblement des concepts non seulement à l'aide de leurs descriptions, mais également à l'aide des relations qu'ils partagent.

Le cadre de notre approche est réparti principalement sur trois modules comme présentés dans la figure 4.1:

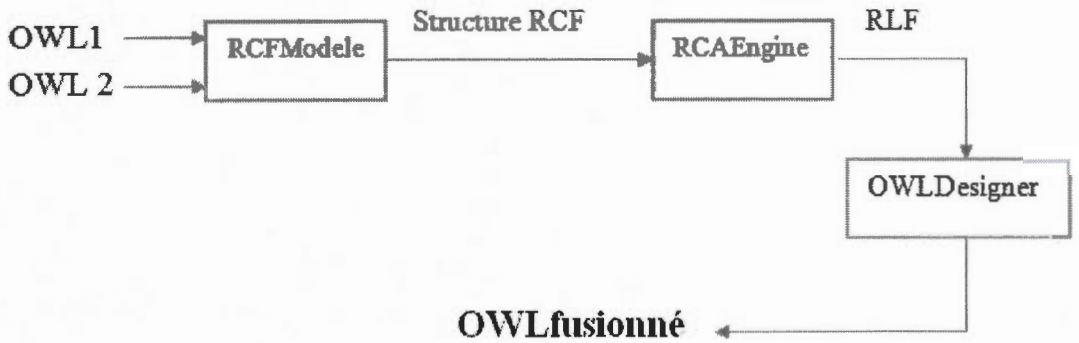


Figure 4.1 L'approche de RCA-Merge

Le premier module dans cette approche est le RCFmodeler qui prend en entrées deux ontologies en format OWL et génère en sortie une structure FCR. La FCR présentée dans la section 2.4 consiste à extraire les concepts et les relations des deux ontologies, et de les regrouper dans des contextes. Donc, le résultat sera une structure contenant un contexte pour les concepts, un contexte pour les rôles (relations), et des contextes pour les interrelations (*target*, domaine, source et range).

Le deuxième module correspond à un moteur ARC qui exploite la structure FCR générée précédemment pour construire deux treillis pour le contexte des concepts et le contexte des rôles, et ce, en mentionnant à chaque nœud les *extents* et les *intents*. Par conséquent, le résultat de l'exécution de ce module sera une structure FTR (Famille de Treillis Relationnelle), cette structure correspondant à un groupe de treillis.

Le troisième et dernier module consiste à construire l'ontologie fusionnée à partir de la FTR du deuxième module. En outre, ce module effectue l'extraction des concepts, des relations, des propriétés, la création des liens d'héritage, l'extraction des concepts

pertinents, etc. De plus, l'exploitation des API de Gallica et le filtrage des treillis nous ont beaucoup aidé à développer ce module. Par conséquent, la sortie de cette troisième étape donne une ontologie fusionnée en format OWL.

4.1.2 RCA-Merge avec préalignement

Comme mentionné dans la sous-section précédente, RCA-Merge est basé sur l'application de l'ARC. La première étape de cette application est de factoriser et de décortiquer tous les éléments des ontologies sources. De ce étape de factorisation résulte une structure FCR (Famille de Concepts Relationnelle). Parfois, les deux ontologies sources contiennent des éléments communs, que ce soit au niveau des concepts ou au niveau des relations. Alors, la structure FCR contiendra deux fois les éléments communs, conduisant ainsi à des redondances au sein de la structure résultante après la factorisation.

Ces redondances se propagent jusqu'au module ONTODesigner (module dédié à la génération de l'ontologie fusionnée). Comme effet collatéral, il deviendra par exemple difficile de détecter certaines abstractions au moment de la création des treillis, que ce soit dans le treillis du contexte des concepts ou dans le treillis du contexte des rôles (ou relations). Ainsi, la génération de la FTR avec des éléments redondants minimise la performance du processus de RCA-Merge et augmente le temps d'exécution de ce dernier.

Suite à une étude approfondie de cette étape du processus que nous avons menée, nous avons décidé d'ajouter un module à RCA-Merge appelé RCF-Aligner qui s'occupe de l'alignement et ce, afin d'éliminer des redondances au sein de la structure FCR. Ainsi, RCF-Aligner prend comme entrées la structure FCR et une autre structure appelée Align contenant toutes les similarités. Aussi, le module Align-API prend les deux ontologies sources en entrées pour produire à la fin la structure Align. Cette dernière est générée suite à la mise en œuvre d'une API développée en Java par

Euzenat (Euzenat, 2004) et appelée *AlignementAPI*. La figure 4.2 illustre l'intégration du module de préalignement dans RCA-Merge:

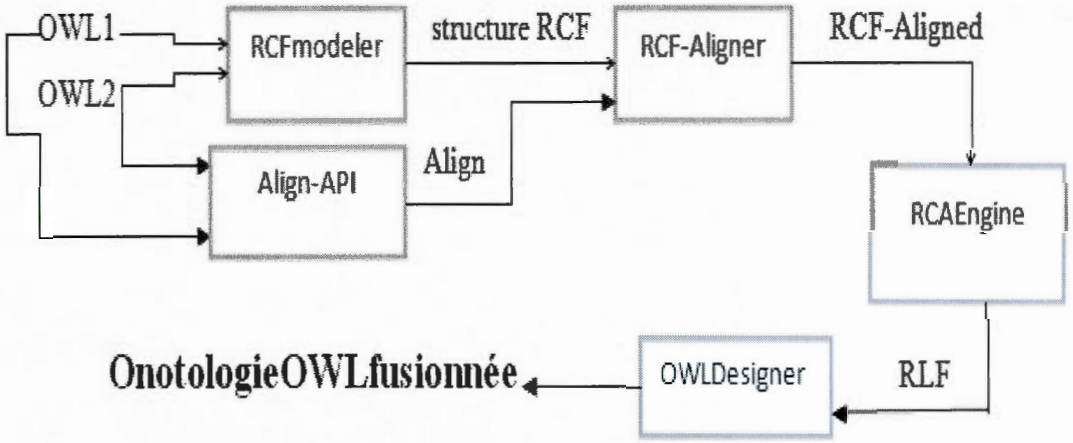


Figure 4.2 RCA-Merge avec préalignement

L'outil *Alignement API* nous permet de raffiner la structure FCR en termes de contenu, une opération effectuée avec les algorithmes du processus de cette API. Selon J. Euzenat (J. Euzenat, 2004), cette API peut être vue comme une extension d'*OWL-API*. *OWL-API*[2] est une API *open source* qui permet de construire et de manipuler des ontologies de type OWL. En plus, *AlignementAPI* est composé essentiellement de trois interfaces. La première interface décrit un alignement en particulier; elle contient les spécifications de l'alignement et la liste des cellules. Cette interface mentionne le niveau de format d'alignement, le type d'alignement, les deux ontologies sources et l'ensemble des correspondances. La deuxième interface définit la similarité particulière entre deux entités. Elle traite les éléments suivants: rdf:ressource (qui permet d'identifier cette correspondance), l'entité 1et l'entité 2. Quant à la troisième interface (appelée relation), contrairement aux deux interfaces précédentes, cette dernière ne contient pas de caractéristiques spécifiques. La figure A.9 de l'appendice A.4 montre un extrait de la structure Align.

[2] http://owlapi.sourceforge.net/

4.1.3 Rôle du nommage dans RCA-Merge

Parmi les points forts de RCA-Merge, on y trouve la génération des abstractions (si elles existent). Ainsi, il y aura construction des super-concepts sur la base des concepts qui existent dans les deux ontologies sources. Par conséquent, il est important d'attribuer au super-concept des noms pertinents de telle sorte que l'ontologie fusionnée demeure cohérente et compréhensible. Généralement, une approche de fusion d'ontologies (comme RCA-Merge) qui génère de nouveaux éléments ontologiques doit avoir une approche de nommage pertinente. De ce fait, la mise en œuvre d'une telle approche (c'est-à-dire le nommage) permet d'obtenir les bons noms. Par ailleurs, il sera intéressant de passer par des sources de données externes qui seront exploitées en collaboration avec le lexique et la sémantique de l'ontologie.

Parmi les sources de données externes, nous pouvons considérer WordNet et Wikipédia:

WordNet[3] est une base de données lexicale offrant une structure lexicale et sémantique des données. Elle fournit différents types de relation, par exemple l'hyperonyme qui nous permet de dégager des concepts plus supérieurs (ou plus communs). Elle offre également une relation appelée holonomie qui permet de chercher les différentes parties d'un concept. En effet, à partir de ces deux relations, nous pouvons exploiter les *extents* des concepts pour un meilleur nommage.

Wikipédia est aussi une source externe de données qui possède une structure pour répertorier les données. En outre, il existe DBpedia qui prend les données à partir de Wikipédia et les dispose sous forme de modèle structuré assez proche d'une structure ontologique. Par conséquent, nous pouvons exploiter ces deux sources de données dans RCA-Merge et elles peuvent être utilisées toutes les deux pour nous fournir les

[3] http://wordnet.princeton.edu/

termes les plus proches liés à une description de concept. De ce fait, elles peuvent suggérer les noms probables associés aux concepts formels découverts par l'ARC.

Pour valider l'importance du nommage dans RCA-Merge, nous avons procédé de la manière suivante: nous avons testé certaines ontologies fusionnées suite à l'exécution du RCA-merge, nous avons proposé des noms significatifs aux nouveaux éléments ontologiques tout en prenant en considération la cohérence de l'ontologie, et nous avons testé ces ontologies suivant les métriques de qualité de INUKHUK.

4.2 Implémentation du RCA-Merge

Dans cette section, nous présentons quelques technologies pertinentes utilisées pour le développement de l'outil RCA-Merge. Ensuite, nous détaillons les modules de base de cet outil, notamment le module RCF-Aligner.

4.2.1 Technologies d'implémentation

Jena

Jena4 est un framework open source développé en Java. Généralement, ce framework permet la manipulation des ontologies. Jena est composé de plusieurs API qui permettent de traiter des structures RDF et des ontologies OWL. Jena permet entre autres de charger des ontologies, d'extraire les éléments ontologiques (concepts, relations, propriétés, etc). De plus, RCA-Merge utilise Jena au niveau de la génération de l'ontologie fusionnée. Par exemple pour le chargement d'une ontologie, on utilise la bibliothèque *com.hp.hpl.jena.ontology.OntModel.* Donc, un exemple d'implémentation qui explique la facon de lire une ontologie avec jena. Onto est un variable de type OntModel, docURI est le URI de l'ontologie OWL et locationURL est le chemin de OWL s'il est enregistré chez l'utilisateur.

[4] http://jena.apache.org/documentation/ontology/

```
onto = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM,null);
onto.getDocumentManager().addAltEntry(docURI,locationURL);
onto.read(docURI);
```

WordNet

WordNet[5] rassemble les éléments lexicaux (noms, verbes, etc.) dans un groupe de synonyme cognitif (en anglais: *synset*). Le plus intéressant dans ce regroupement, c'est que les *synsets* sont reliés par des liens sémantiques et lexicaux. Par conséquent, parmi les objectifs de WordNet, on compte le traitement du langage naturel. Dans notre projet, nous avons utilisé Rita, une API au-dessus de WordNet permettant un usage simplifié de WordNet. L'utilisation de WordNet intervient après l'extraction des éléments ontologiques des deux ontologies à fusionner. En effet, le but est de dégager les relations de synonymie entre les éléments ontologiques.

Lucene

Lucene[6] est un moteur de recherche *open source* implémenté en Java. Lucene est utilisée généralement dans des applications pour indexer des documents et effectuer des recherches dans ces documents. Dans RCA-Merge, Lucene est exploité pour extraire des termes similaires à partir des mots composés. On l'utilise utilisé juste après l'extraction des éléments ontologiques. Le choix de la décomposition des mots composés permet d'améliorer l'analyse sémantique des termes de l'ontologie. De plus, Lucene favorise l'analyse relationnelle des concepts et à mener une meilleure analyse sur les plans lexical et sémantique.

---

[5] http://wordnet.princeton.edu/
[6] http://lucene.apache.org/

SimPack

SimPack[7] est un outil développé en Java pour la détection et la recherche des similarités entre les concepts ontologiques. SimPack est utilisé dans d'autres contextes comme la détection des similarités dans le code source (par exemple, le changement de nom des classes entre plusieurs versions d'un même logiciel) ainsi que la recherche de similarité entre des données structurées décrites en XML. Dans RCA-Merge, SimPack est exploité à l'étape d'évaluation d'ontologies où nous utilisons les métriques de qualités qui existent dans la plateforme INUKHUK. Certaines de ces métriques (comme celles qu'ALANI *et al.* exploitent) facilitent l'implémentation des calculs de différentes mesures. Par exemple, la mesure d'ALANI *et al.* (dite SSM) utilise SimPack pour mesurer la proximité entre deux concepts ontologiques.

Jung

Jung[8] (*Java Universal Network/Graph framework*) est une bibliothèque développée en Java permettant l'analyse, la visualisation et la modélisation des graphes. Jung permet la représentation des graphes en intégrant des entités et des relations. Cette API comporte plusieurs algorithmes de la théorie des graphes, l'extraction des données et l'analyse des graphes. En plus, Jung offre un cadre de visualisation pour faciliter la génération et le traitement des données. Dans le cadre de notre projet et dans le module ONTODesigner, Jung permet l'analyse des données extraites à partir du treillis (un treillis est un graphe) pour les représenter par la suite dans l'ontologie fusionnée. Jung intervient principalement dans la construction de l'ontologie fusionnée. Plus précisément, Jung aide à mettre en place les concepts et les relations ontologiques dans le graphe ontologique.

---

[7] https://files.ifi.uzh.ch/ddis/oldweb/ddis/research/simpack/index.html
[8] http://jung.sourceforge.net/

4.2.2 Les modules du processus de base de RCA-Merge

Comme il a été présenté dans 4.1.1, l'outil RCA-Merge est composé essentiellement de trois principaux modules qui existent déjà dans INUKHUK. Il est à noter que ces modules sont aussi partagés par d'autres services au sein d'INUKHUK, notamment le service de restructuration des ontologies (M. Rouane-Hacene, Fennouh, Nkambou et Valtchev, 2010):

Le module RCFModeler

RCFModeler (Rouane, H. Mohamed *et al.*, 2007) constitue le premier module de RCA-Merge. RCFModeler permet de transformer deux ontologies sources en une FCR, le format de données de base en ARC. RCFModeler chargent les éléments ontologiques et les place dans les contextes appropriés (un contexte par type d'élément ontologique). RCFModeler applique une série de transformations aux deux ontologies à fusionner.

La première transformation génère un contexte des concepts qui comporte les concepts des deux ontologies et les propriétés de chaque concept. Ainsi, le contexte obtenu est formé d'une part par les concepts ontologiques comme objets formels, et d'autre part par les noms des concepts et leurs propriétés comme attributs formels. Dans (Rouane, H. Mohamed *et al.*, 2007), les auteurs définissant les règles de représentation d'une ontologie en format FCR. Le ' 4.1 reprend quelques-unes de ces règles de transformation. En termes d'implémentation, l'extraction des concepts, des propriétés et des relations s'effectue à l'aide d'autres outils comme Jena, Lucene et WordNet. Aussi, ces outils aident au développement des méthodes de génération des contextes. Parmi les méthodes d'extraction les plus pertinentes, on retrouve la récupération des concepts (*get claasname()*), la récupération des relations entre les concepts (*get ObjectProperty()*) et la récupération des propriétés ou attribut conceptuel (*get DataProperties()*). De plus, pour la construction des contextes à la construction du contexte des concepts (*buildContext-Concept()*), on retrouve aussi

l'ajout des liens d'héritage entre les concepts dans le contexte des concepts (*addInheritanceLinksAmongConceptsToContextOfConcepts()*). La figure A.1 de l'appendice A.1 montre les méthodes développées pour l'extraction des éléments ontologiques et la construction du contexte des concepts et du contexte des rôles (ou relations) respectivement.

Tableau 4.1 Quelques règles de transformation d'une ontologie en un contexte formel

ONTOLOGIE	FAMILLE DE CONTEXTES RELATIONNELLE (FCR)
Attribut de concept	Une propriété du contexte des concepts
Relation d'incidence	1 s'il existe une relation et 0 sinon
Relation	Objet dans le contexte des rôles

La deuxième transformation est conduite par un mécanisme appelé *Scaling relationnel* (A.M. Rouane-Hacene, 2013) qui permet de transformer les liens interindividuels en un ensemble d'attributs relationnels. Plus précisément, ce mécanisme intègre les relations dans les contextes FCR. Dans ce dessein, nous nous sommes basé sur un métamodèle ontologique ODM. La figure 4.3 représente le métamodèle ODM. ODM propose le type d'élément ontologique à représenter par les contextes et les liens inter-contextes. Par conséquent, selon ODM, la structure FCR contient: (1) un contexte formel pour les concepts, (2) l'intégration des *DataProperties* dans le contexte des concepts, et (3) des contextes formels pour les relations binaires (domaine, codomaine, sources, destination). La figure 4.4 montre un exemple de deux ontologies et la figure 4.5 illustre la structure FCR correspondante. La figure A.3 de l'appendice A.1 représente le fichier d'extension .rcf correspondant. Par ailleurs, l'implémentation des relations intercontextes débute avec l'extraction du domaine et du codomaine à l'aide de l'outil Jena, et plus précisément par les méthodes *com.hp.hpl.jena.ontology.getDomain()* et *com.hp.hpl.jena.ontology.getRange()*. Ces

deux méthodes sont aussi exploitées pour construire les contextes codomaines et destinations.

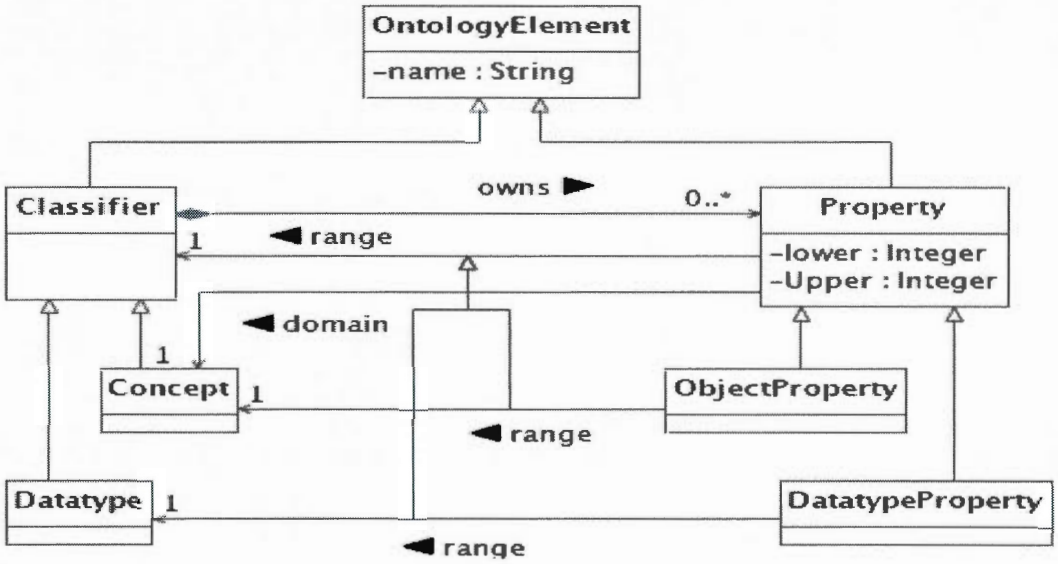


Figure 4.3 Le métamodel d'ontologie ODM (A.M. Rouane-Hacene, 2013)

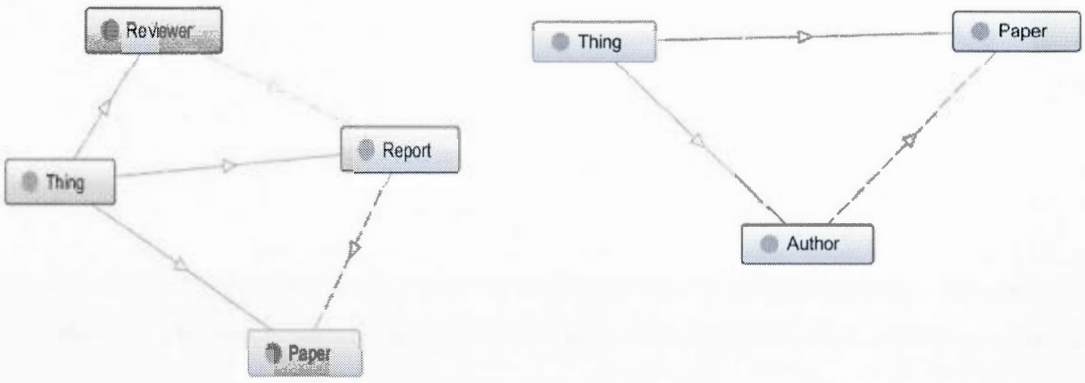


Figure 4.4 Deux ontologies à fusionner

	Reviwer	Report	Paper	Paper	Author	title	passw	name	login	email	contents	comments	affiliation	rank
Reviewer	1	0	0	0	0	0	1	1	1	1	0	0	1	0
Report	0	1	0	0	0	0	0	0	0	0	0	1	0	0
Paper	0	0	1	1	0	1	0	0	0	0	1	0	0	0
Paper	0	0	1	1	0	1	0	0	0	0	1	0	0	0
Author	0	0	0	0	1	0	0	1	0	1	0	0	1	1

Context des concepts

	rate	compose	write
rate	1	0	0
compose	0	1	0
write	0	0	1

Context des rôles

	rate	compose	write
Reviewer	0	1	0
Report	1	0	0
Paper	0	0	0
Paper	0	0	0
Author	0	0	1

Source

	rate	compose	write
Reviewer	0	0	0
Report	0	1	0
Paper	1	0	1
Paper	1	0	1
Author	0	0	0

Target

	Reviewer	Report	Paper	Paper	Author
rate	0	1	0	0	0
compose	1	0	0	0	0
write	0	0	0	0	1

Domaine

	Reviewer	Report	Paper	Paper	Author
rate	0	0	1	1	0
compose	0	1	0	0	0
write	0	0	0		1

Range

Figure 4.5 Les contextes correspondant aux ontologies de la figure 4.4

En récapitulant, la première étape de RCA-Merge passe par deux phases: (1) une phase d'extraction à l'aide d'un sous-module nommé OWLVisitor, ce dernier permettant de décortiquer les deux ontologies et de dégager tous les éléments ontologiques tout en utilisant Jena, Lucene et WordNet pour faciliter l'implémentation, et (2) la génération de la FCR dont les éléments constituant les contextes extraits par OWLVisitor. Ainsi, une bonne factorisation des deux ontologies engendre une bonne construction des contextes FCR. À la fin de cette étape, le résultat de RCFmodeler sera une entrée au deuxième module qui s'occupe de l'analyse des contextes générés dans la FCR.

Le module RCAEngine

L'analyse relationnelle de concepts s'appuie sur un algorithme appelé Multi-FCA. Cet algorithme construit, à partir d'une FCR, un ensemble de treillis appelé FTR (Famille de Treillis Relationnelle). Dans la FTR, chaque contexte est représenté par un treillis. FTR contient des concepts où les extensions englobent des individus qui partagent des liens avec d'autres individus. Ce partage est mis en évidence avec des attributs formels appelés aussi relationnels. Chaque attribut formel relationnel correspond à une relation entre deux concepts (M. Rouane-Hacene *et al.*, 2013).

La méthode Multi-FCA applique un processus itératif puisque l'échelle (*scaling*) relationnelle change l'état des contextes. Donc, une mise à jour des treillis est nécessaire pour pallier les changements et conserver les treillis dans un état cohérent. Le processus itératif de génération de FTR s'interrompt lorsqu'il atteint un point fixe. En d'autres termes, si un *scaling* relationnel est appliqué et qu'il n'offre aucun enrichissement d'aucun contexte, alors le point fixe est atteint. La figure 4.6 montre l'algorithme de traitement d'une structure FCR à l'aide de la méthode Multi-FCA (M. Rouane-Hacene *et al.*, 2013).

```
1: proc MULTI-FCA(
2: In: (K, R) an RCF, ρ a constructor mapping
3: Out: L array [1,...,n] of lattices)
4: p ← 0 ; halt ← false
5: for i from 1 to n do
6:    K_i^0 ← SCALE(K_i)
7:    L_i^0 ← BUILD-LATTICE(K_i^0)
8: while not halt do
9:    p = p + 1
10:    for i from 1 to n do
11:       K_i^p ← EXTEND-CONTEXT(K_i^{p-1}, ρ, L^{p-1})
12:       L_i^p ← UPDATE-LATTICE(K_i^p, L_i^{p-1})
13:    halt ← ⋀_{i=1,...,n} ISOMORPHIC(L_i^p, L_i^{p-1})
```

Figure 4.6 Algorithme de génération de treillis à partir d'un contexte FCR (M. Rouane-Hacene *et al.*, 2013)

L'algorithme de construction de treillis commence par une initialisation des structures de calcul. Il prend chaque contexte K_i^0 à partir de l'application de primitive *scale* (K_i). Par la suite, il y aura une construction des treillis L_i^0 à l'aide du primitif *Build-Lattice*. Ensuite, à partir de la ligne 8 de l'algorithme de la figure 4.6, chaque relation $r_k \subseteq O_i * O_j$, le treillis L_i^p du contexte de domaine K_i^p est amélioré à l'aide de treillis L_i^{p-1} du contexte de codomaine et ce, en utilisant le primitif *Extend-Context*. Par la suite, il y aura application du *Update-Lattice* (K_i^p,L_i^{p-1}) sur le treillis du contexte de domaine L_i^p. Finalement, l'algorithme prend fin lorsqu'il y aura un isomorphisme entre les treillis L_i^n et L_i^{n+1}, c'est-à-dire lorsqu'il n'y a pas de changement enrichissant entre l'itération n et n+1. L'appendice A.2 montre quelques fonctions pertinentes de RCAEngine (M. Rouane-Hacene *et al.*, 2013).

En résumant, cette phase importante de RCA-Merge applique le mécanisme de l'ARC à l'aide de la méthode Multi-FCA. Cette dernière prend en entrée FCR et le traite avec l'algorithme décrit précédemment. Cet algorithme contient lui-même quelques méthodes de construction et de mise à jour des treillis. Par conséquent, Multi-FCA donne en sortie un ensemble de treillis. La figure 4.7 représente un ensemble de treillis pour le contexte des concepts et le contexte des rôles de la figure 4.5. Par ailleurs, l'implémentation de RCAEngine est réalisée en grande partie à l'aide de Galicia. La famille des treillis relationnelle FTR résultante sera une entrée à l'étape suivante du processus de RCA-Merge.

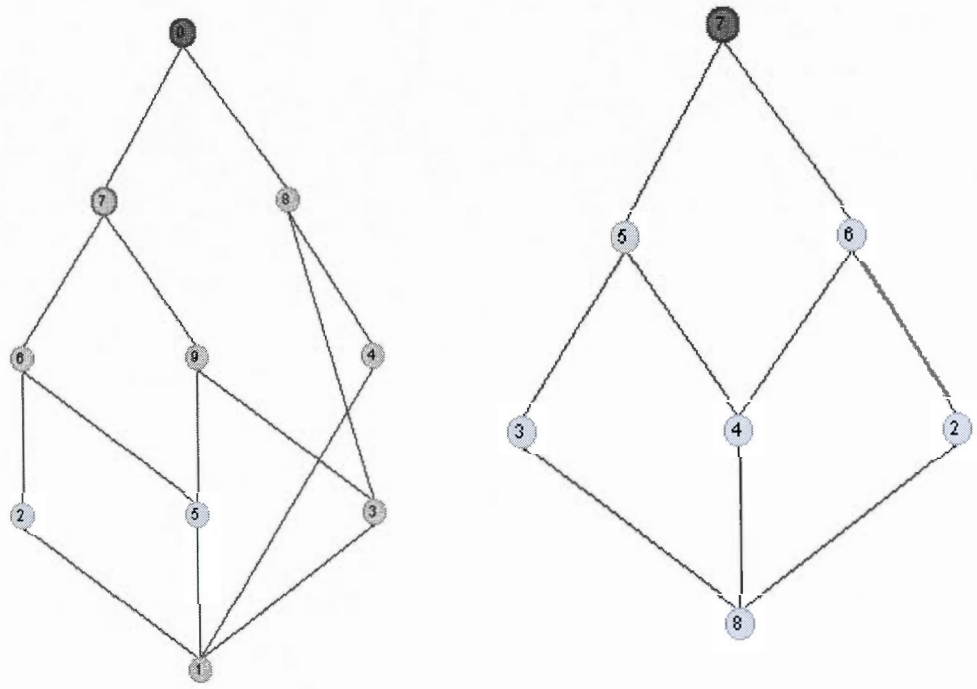


Figure 4.7 Treillis de contexte des concepts (à gauche) et treillis de contexte des rôles (à droite)

Le module ONTODesigner

Le module ONTODesigner prend en entrées les treillis produit par le module (une FTR) RCAEngine et produit en sortie une ontologie qui représente la fusion des deux ontologies en entrées. ONTODesigner applique une série de transformations guidées par le métamodèle d'une FTR pour créer un modèle ontologique. L'outil ONTODesigner possède une forte capacité pour générer une ontologie utilisable et ce, en mettant en œuvre la rétro-ingénierie des éléments des ontologies de départ ainsi qu'en appliquant des filtres sur les concepts et les abstractions suggérées par RCAEngine. En effet, ONTODesigner exécute des fonctions d'élagage sur les treillis de la FTR dans le but d'éliminer les concepts non pertinents en évaluant le degré de pertinence de chaque concept. Cette évaluation est basée sur quelques critères comme la présence de l'information clé du domaine correspondant (l'attribut d'un concept

ontologique est un exemple d'information clé). Par ailleurs, la rétro-ingénierie des éléments ontologiques vers une ontologie finale est basée sur un processus composé de quatre étapes (M. Rouane-Hacene *et al.*, 2010):

Étape 1 – Filtrer le groupe initial des concepts pertinents qui existe dans le contexte des concepts. Les concepts pertinents représentent généralement (dans la plupart des cas) les concepts initiaux des deux ontologies. Par exemple, tous les concepts feuilles (qui n'ont pas de descendants) qui apparaissent dans la figure 4.4 sont retenus dans l'ontologie fusionnée. La figure 4.8 illustre les concepts de l'ontologie fusionnée à partir des ontologies sources de la figure 4.4.

Étape 2 – Détecter les concepts pertinents supplémentaires (ou les abstractions). La détection de ces concepts supplémentaires est effectuée selon les informations contenues dans les intensions des concepts formels de la FTR. De ce fait, si l'intension contient des informations du domaine et qui est ou bien un attribut de concept ou bien un lien relationnel avec un autre concept pertinent, le concept formel qui possède cette intension est alors jugé pertinent. La figure 4.8 montre l'existence de concepts ontologiques pertinents supplémentaires qui correspondent aux abstractions.

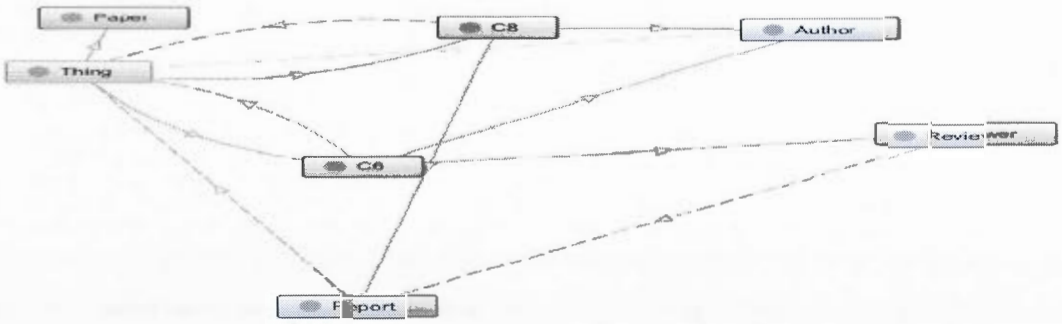


Figure 4.8 Les concepts de l'ontologie fusionnée

Étape 3 – Consiste à transformer les concepts pertinents dégagés lors des deux étapes précédentes en concepts ontologiques. De plus, les propriétés des concepts ontologiques et l'héritage sont tous transférés à l'ontologie finale. La figure 4.9

correspond au fichier OWL de l'ontologie fusionnée à partir des deux ontologies de la figure 4.4.

Étape 4 – Cette dernière étape permet d'attribuer des noms significatifs aux nouveaux concepts à l'aide de leurs descriptions. Le nommage des concepts est un sujet très complexe puisqu'il nécessite trop d'analyses structurelles, surtout sémantiques, pour trouver les noms exacts décrivant les nouveaux concepts ontologiques. En effet, quelques techniques ont été exploitées comme la technique qui consiste à combiner les termes dans la description des concepts ontologiques sans noms.

L'implémentation d'ONTODesigner s'appuie essentiellement sur les services de la plateforme Galicia. Toutefois, comme la sortie d'ONTODesigner est une ontologie fusionnée, il y a donc recours à l'API Jena. L'appendice A.3 montre certaines méthodes d'ONTODesigner (M. Rouane-Hacene *et al.*, 2010).

```
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
<owl:Class rdf:about="http://www.semanticweb.org/ontologies/2009/4/target.owl#Reviewer">
  <rdfs:subClassOf>
    <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2009/4/target.owl#C6"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="http://www.semanticweb.org/ontologies/2009/4/target.owl#Report">
  <rdfs:subClassOf>
    <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2009/4/target.owl#C8"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="http://www.semanticweb.org/ontologies/2009/4/target.owl#Author">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2009/4/target.owl#C8"/>
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2009/4/target.owl#C6"/>
</owl:Class>
<owl:Class rdf:about="http://www.semanticweb.org/ontologies/2009/4/target.owl#Paper"/>
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2009/4/target.owl#write">
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2009/4/target.owl#Author"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2009/4/target.owl#compose">
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2009/4/target.owl#Reviewer"/>
  <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2009/4/target.owl#Report"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2009/4/target.owl#composeWrite">
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2009/4/target.owl#C6"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2009/4/target.owl#rate">
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2009/4/target.owl#Report"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2009/4/target.owl#rateWrite">
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2009/4/target.owl#C8"/>
</owl:ObjectProperty>
```

Figure 4.9 Exemple de fichier OWL d'une ontologie fusionnée

En récapitulant, l'outil ONTODesigner prend en entrées des treillis FTR et génère en sortie une ontologie OWL. L'*ontologie fusionnée* est construite suivant la technique de rétro-ingénierie de deux modèles ontologiques initiaux. La rétro-ingénierie passe par un processus de construction. Ainsi, ce processus utilise des filtres pour extraire les concepts pertinents initiaux et dégager les nouvelles abstractions pertinentes. Le filtrage fait émerger des concepts significatifs dans l'ontologie de fusion. À la fin de la fusion des ontologies de départ, ONTODesigner s'occupe du nommage des nouveaux éléments ontologiques en utilisant des techniques mentionnées ci-haut.

Dans cette section nous avons présenté les différents modules qui composent l'outil RCA-Merge. Dans ce qui suit, nous présenterons les améliorations proposées pour

augmenter d'une part la performance du processus du RCA-Merge et d'autre part la qualité des treillis FTR, et par conséquent l'ontologie fusionnée.

4.2.3 Amélioration du RCA-Merge avec préalignement

Après la mise en œuvre et la validation du RCA-Merge (voir chapitre 5), nous avons constaté que nous pouvons augmenter la performance de cet outil. En effet, si les deux ontologies sources contiennent des éléments communs, la structure FCR contiendra alors des redondances. Il est donc nécessaire d'éliminer ces redondances.

Le module de préalignement, appelé RCFAligner, prend en entrées deux informations, c'est-à-dire la famille des contextes qui représentent les deux ontologies à fusionner (une FCR) ainsi qu'un fichier d'alignement des deux ontologies à fusionner. Cet alignement est obtenu grâce à l'usage de l'API d'alignement *AlignAPI* qui détecte les similarités dans les termes des ontologies à fusionner. La génération du fichier d'alignement se déroule comme suit (Euzenat, 2004):

- Considérer les deux ontologies à fusionner en tant que deux URI, et les initialiser.
- Fixer les paramètres de détection des similarités d'AlignAPI, comme StringFunction et SMODistance.
- Appliquer l'alignement en tenant compte des paramètres de détection des similarités et les URIs d'ontologies.
- Générer un fichier qui contient les alignements des relations et des concepts.

Après la préparation du FCR et du fichier d'alignement, l'outil RCFAligner exécute les étapes suivantes: (1) la lecture de la FCR pour pouvoir extraire les objets formels et les attributs formels à partir du contexte des concepts, (2) la comparaison des objets formels (attributs formels respectivement) entre eux avec *AlignAPI* afin d'éliminer les redondances au niveau des objets (attributs respectivement), et (3) par exemple, étant

donné les tableaux de la figure 4.10 qui représente un contexte de concepts ontologiques, on suppose que d'après l'alignement, le concept A2 est similaire à A1.

	A1	B	A2	C
A1	1	1	1	0
B	0	1	1	0
A2	1	0	1	1
C	0	0	1	1
D	0	1	0	1

Résultat après élimination du redondant A2

	A1	B	C
A1	1	1	1
B	1	1	0
C	1	0	1
D	0	1	1

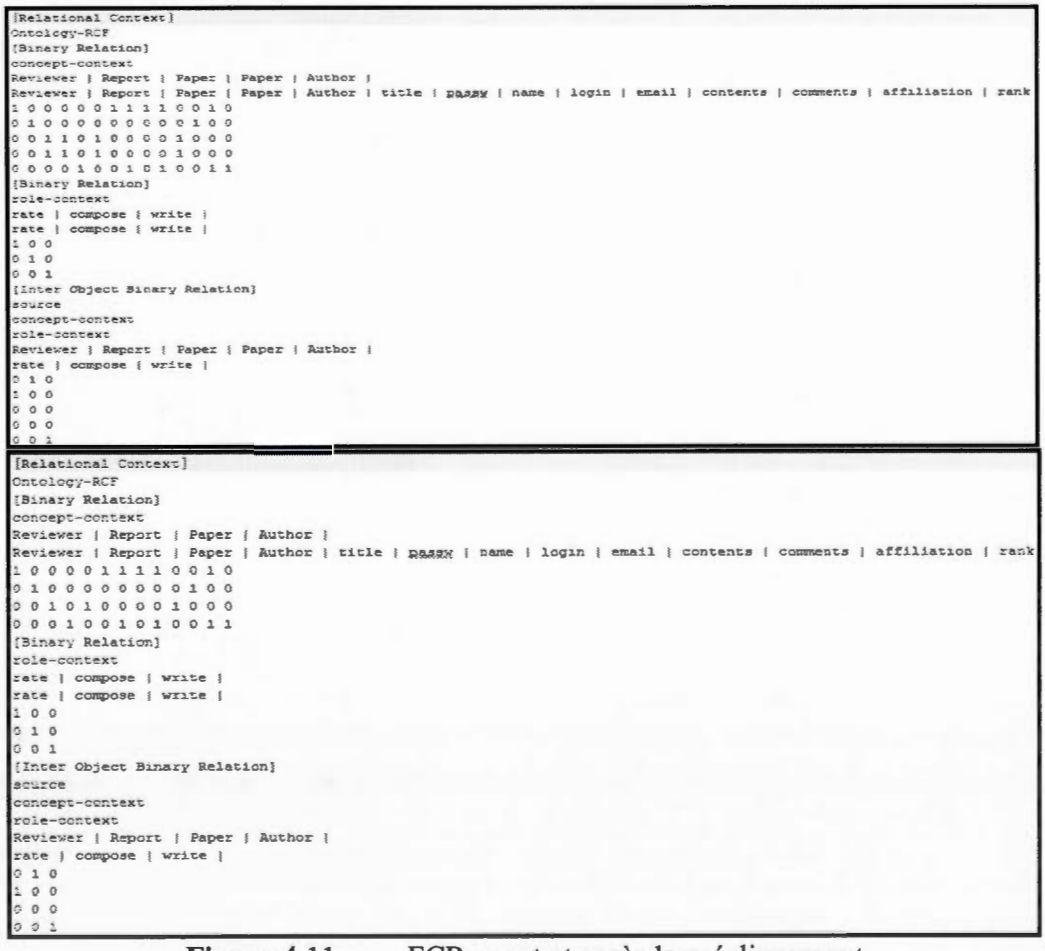
Figure 4.10 Exemple d'affectation des relations d'incidence

Par conséquent, RCF-Aligner élimine le concept A2. Toutes les relations d'incidence avec A2 sont remplacées par des relations avec A1. Ainsi, le concept C qui n'a aucune relation d'incidence avec le concept A1 dans le tableau de droite (mais il est en relation avec A2) se voit affecté par une relation d'incidence avec A1 dans le tableau de gauche. La dernière opération de RCF-Aligner tient à l'exportation de la nouvelle FCR, désormais dite RCF alignée. La figure 4.11 montre l'état d'une FCR avant et après le passage de RCF-Aligner pour les deux ontologies de la figure 4.4. L'appendice A.4 montre quelques méthodes du module de préalignement.

Le pseudo-code ci-dessous illustre les principales étapes du processus RCFAligner:

1. Générer FCR initiale
2. Générer le fichier d'alignement
3. Charger la FCR initiale()
4. Extraire les objets formels()
5. Extraire les attributs formels()
6. **Pour** i=0; **i<objetformel**.size(); i++
   a. Comparer les objets formels entre eux à l'aide d'Align()
   b. Éliminer les redondances
7. **Fin Pour**
8. Traiter les relations d'incidence
9. Écrire la RCF Alignée.

Dans ce chapitre, nous avons présenté RCA-Merge en tant qu'une nouvelle approche de fusion d'ontologies basée sur l'analyse relationnelle des concepts. Nous avons aussi détaillé les trois modules de base (RCFModeler, RCAEngine et ONTODesigner). Nous avons également montré l'amélioration en ajoutant le module de préalignement (RCFAligner). Par ailleurs, dans ce qui suit, nous présenterons l'étude expérimentale et la validation de RCA-Merge et RCA-Merge avec préalignement. Nous démontrerons également l'influence d'un nommage pertinent sur la qualité de l'ontologie. En plus, nous présenterons une étude comparative entre RCA-Merge (avec préalignement) et d'autres outils de fusion bien reconnus.



```
[Relational Context]
Ontology-RCF
[Binary Relation]
concept-context
Reviewer | Report | Paper | Paper | Author |
Reviewer | Report | Paper | Paper | Author | title | PaperX | name | login | email | contents | comments | affiliation | rank
1 0 0 0 0 0 1 1 1 1 0 0 1 0
0 1 0 0 0 0 0 0 0 0 0 1 0 0
0 0 1 1 0 1 0 0 0 0 1 0 0 0
0 0 1 1 0 1 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 1 0 1 0 0 1 1
[Binary Relation]
role-context
rate | compose | write |
rate | compose | write |
1 0 0
0 1 0
0 0 1
[Inter Object Binary Relation]
source
concept-context
role-context
Reviewer | Report | Paper | Paper | Author |
rate | compose | write |
0 1 0
1 0 0
0 0 0
0 0 0
0 0 1
```

```
[Relational Context]
Ontology-RCF
[Binary Relation]
concept-context
Reviewer | Report | Paper | Author |
Reviewer | Report | Paper | Author | title | PaperX | name | login | email | contents | comments | affiliation | rank
1 0 0 0 0 1 1 1 1 0 0 1 0
0 1 0 0 0 0 0 0 0 0 1 0 0
0 0 1 0 1 0 0 0 0 1 0 0 0
0 0 0 1 0 0 1 0 1 0 0 1 1
[Binary Relation]
role-context
rate | compose | write |
rate | compose | write |
1 0 0
0 1 0
0 0 1
[Inter Object Binary Relation]
source
concept-context
role-context
Reviewer | Report | Paper | Author |
rate | compose | write |
0 1 0
1 0 0
0 0 0
0 0 1
```

Figure 4.11 FCR avant et après le préalignement

CHAPITRE V

EXPÉRIMENTATION ET VALIDATION

Dans ce chapitre, nous présenterons les expérimentations effectuées sur RCA-Merge. La première partie sera consacrée aux tests internes, c'est-à-dire à des expérimentations effectuées sur le processus de base de RCA-Merge, puis sur RCA-Merge avec préalignement, ainsi qu'à une démonstration du rôle d'un nommage efficace dans RCA-Merge amélioré et ce, en s'appuyant sur les métriques d'OpenQAM.

Dans la deuxième partie, nous présenterons une étude externe, c'est-à-dire que nous comparerons RCA-Merge avec d'autres outils de fusion d'ontologies et ce, sur la base de OpenQAM de INUKHUK. Mais avant de passer aux expérimentations, nous présenterons notre protocole de validation qui sera utilisé pour les expérimentations interne et externe.

5.1 Protocole de validation

Pour bien mesurer notre approche de fusion d'ontologies, nous avons adopté un protocole de validation. Ce protocole consiste d'abord à récupérer une ontologie de référence (en anglais: *Gold Standard Ontology*) à partir du référentiel en ligne de Protégé-2000, et ensuite à partitionner cette ontologie de référence en deux sous-ontologies. Le partitionnement est effectué avec l'outil NÉON Toolkit[9] et donne lieu à deux ontologies nommées SO1 et SO2. Finalement, nous lançons notre processus de fusion avec les deux entrées SO1 et SO2 afin obtenir l'ontologie fusionnée en sortie.

[9] http://neon-toolkit.org/wiki/Main_Page

Nous effectuons une comparaison entre l'ontologie de référence et l'ontologie fusionnée. Cette comparaison s'appuie essentiellement sur le module OpenQam de la plateforme INUKHUK. OpenQam met en œuvre plusieurs métriques, notamment celles d'ALANI *et al.*, ORME *et al.*, TARTIR *et al.*, et finalement STAAB *et al.* (que nous avons présentés dans la section 3.2). Le but de ces métriques est de mesurer la qualité de conception d'une ontologie. Les métriques de Orme, par exemple, permettent de calculer le nombre de propriétés par classe (métrique APC) ainsi que le nombre de classes non-feuilles (NL).

5.2 Expérimentation et interprétation de RCA-Merge

En se basant sur les métriques présentées dans la section 3.2 sur l'état de l'art, nous effectuerons les tests nécessaires sur les résultats retournés par RCA-Merge et nous les interpréterons. À chaque étape d'expérimentation, nous utiliserons différentes ontologies afin de tester la robustesse de l'outil RCA-Merge. Le tableau 5.1 donne la nomenclature des métriques utilisées:

Tableau 5.1 Les mesures d'ORME *et al.*

MESURES	DÉFINITION DES MESURES
NOP	Retourne le nombre de propriétés dans une ontologie. Donne aussi une information sur la complexité de l'ontologie.
APC	Retourne le nombre de propriétés par concept. Ce nombre reflète la façon dont les concepts ont été définis.
NOFC	Calcule le nombre de sortants pour un concept donné.
NOF	Retourne le nombre de sortants dans une ontologie.
AFC	Calcule le nombre moyen de sortants par concept. Ce nombre offre une indication sur la complexité.
AFR	Calcule le nombre moyen de sortants des concepts racines. Le résultat reflète une indication sur la cohésion.
NL	Calcule le nombre de concepts non-feuilles.
AFNL	Calcule les sorties moyennes des concepts non-feuilles.
maxDit	Retourne la profondeur maximale de l'arbre d'héritage d'une ontologie.

Tableau 5.2 Les mesures de TARTIR *et al.*

MESURES	DÉFINITION DES MESURES
RRS	Retourne le pourcentage de la richesse des relations (proche de 0 s'il s'agit d'une relation is-a; proche de 1 s'il ne s'agit pas de relation is-a).
ARS	Retourne le nombre moyen des attributs par concept (une valeur élevée indique que le concept dispose un grand nombre d'attributs; une valeur inférieure indique que moins d'informations sont fournies dans ce concept).
IRS	Retourne le nombre moyen de sous-concepts par concept (IRS faible indique que l'ontologie est verticale; IRS élevé indique que l'ontologie est horizontale).
CR	Retourne la richesse des concepts (CR élevé si les données représentent bien les connaissances; CR faible si les données ne représentent pas bien les connaissances).
AP	Retourne la répartition moyenne des instances de tous les concepts.

Tableau 5.3 Les mesures d'ALANI *et al.*

MESURES	DÉFINITION DES MESURES
CMM	Évalue la couverture d'une ontologie pour des termes donnés. Cette mesure retourne la valeur de CMM.
DEM	Mesure la densité d'une ontologie. Cette mesure est calculée suivant le nombre de relations, de sous-concepts, de super-concepts, etc. DEM retourne la somme pondérée de chaque densité du concept adaptée.
SSM	Mesure la similarité sémantique de l'ontologie. Précisément, elle consiste à mesurer la distance la plus courte entre les concepts qui correspondent aux critères de recherche.
BEM	Étant donné un concept A, cette mesure permet d'évaluer le nombre de concepts impliqués dans le plus court chemin entre A et les autres concepts. La valeur retournée de BEM est la moyenne entre toutes les valeurs des concepts.

Tableau 5.4 Les mesures de STAAB *et al.*

MESURES	DÉFINITION DES MESURES
LP et LR	Reflètent à quel point les termes lexicaux couvrent le domaine cible. LP et LR sont généralement utilisés avec les mesures d'évaluation de la hiérarchie des concepts.
TP et TR	Calcule la précision sémantique uniquement pour les concepts communs des deux ontologies.
TF et TF'	La mesure TF est la moyenne harmonique globale de TP et TR. Une mesure TF élevée implique une qualité élevée de la hiérarchie des concepts. La mesure TF' est la moyenne harmonique de TF et LR, elle est donc influencée par le niveau lexical.

5.2.1 Avec les mesures d'ORME *et al.*

Pour minimiser l'écriture dans les tableaux, nous utiliserons les notations suivantes:

- Or: désigne l'ontologie de référence.
- Om: désigne l'ontologie fusionnée.

Les tableaux contiennent les pourcentages des mesures et l'appendice B.1 détermine les même tableaux en termes de valeurs (sauf la métrique de STAAB *et al.*, car elle est implémentée uniquement pour donner des pourcentage).

Tableau 5.5 Les résultats d'expérimentation de RCA-Merge en utilisant la métrique d'ORME *et al.*

	camera		univ-bench		Pizza	
	Or	Om	Or	Om	Or	Om
NOP	0,83	1,00	1,00	0,81	0,88	1,00
APC	1,00	0,84	1,00	0,83	0,93	1,00
AFC	0,47	1,00	0,69	1,00	0,49	1,00
AFR	0,28	1,00	1,00	0,63	0,52	1,00
AFNL	1,00	1,00	0,58	1,00	0,48	1,00
MaxDit	1,00	1,00	1,00	1,00	1,00	1,00
Total	4,59	5,84	5,28	5,27	4,32	6,00

Avec les mesures d'ORME *et al.*, l'ontologie de référence Or et l'ontologie fusionnée Om sont évaluées principalement sur le plan quantitatif. De ce fait, les deux mesures NOP et APC traitent respectivement le nombre de propriétés dans l'ontologie et le nombre de propriétés par concept. La mesure NOP est calculée suivant le nombre de relations (c'est-à-dire les *Objects Properties*), tandis que la mesure APC est égale à la mesure NOP divisée par le nombre de classes. Par conséquent, et puisque RCA-

Merge génère de nouveaux concepts pertinents (suite aux abstractions) et de nouvelles relations, la mesure NOP augmente donc. Mais si la mesure APC diminue, cela s'explique par le fait que le nombre de concepts ajoutés dans Om est supérieur à celui dans NOP.

En outre, les mesures AFC, AFR et AFNL traitent respectivement le nombre de sortants: d'abord par concept, ensuite dans les concepts racines et finalement dans les concepts non-feuilles. Le calcul de ces trois mesures est effectué de la manière suivante: la division du nombre de sortants par le nombre de concepts pour AFC, le nombre de concepts racines pour AFR et le nombre de concepts non-feuilles pour AFNL. D'après les résultats du tableau 5.5, on constate que ces trois mesures augmentent (dans la majorité des cas) dans Om par rapport à Or, et cette augmentation revient à l'augmentation de NOF (le nombre de sortants) d'Om. Mais parfois, des exceptions apparaissent suivant le nombre d'abstractions et le nombre de nouvelles relations.

5.2.2 Avec les mesures de TARTIR *et al.*

Tableau 5.6 Les résultats d'expérimentation de RCA-Merge en utilisant la métrique de TARTIR *et al.*

	camera		univ-bench		Pizza	
	Or	Om	Or	Om	Or	Om
RRS	1.00	0.72	1.00	0.81	1.00	0.44
ARS	1.00	0.70	1.00	0.00	0.00	0.00
IRS	0.47	1.00	0.83	1.00	0.39	1.00
CR	1.00	0.00	0.00	0.00	1.00	0.00
AP	1.00	0.00	0.00	0.00	1.00	0.00
Total	4.47	2.43	2.83	1.81	3.39	1.44

Les mesures de TARTIR *et al.* permettent l'évaluation du schéma ontologique quantitativement et qualitativement. En ce qui concerne la qualité, les mesures RRS et CR indiquent respectivement la richesse des relations et la richesse des concepts. La mesure RRS est calculée comme suit: RRS est égale au nombre de propriétés divisé par le nombre de propriétés augmenté du nombre de sous-concepts. Donc, puisqu'il y a une division par sous-concept (l'abstraction permet l'augmentation des sous-concepts), RRS diminue alors et tend vers zéro. Par conséquent, les relations d'Om sont de type is-a. Donc, ce sont des relations simples et claires. La mesure CR est calculée à partir des classes utilisées et ces dernières sont calculées suivant les instances, c'est-à-dire que si un concept possède une instance, il est donc inclus dans le calcul. CR est le nombre de concepts utilisés divisé par le nombre total de concepts. Si la valeur CR est égale à 0, cela signifie qu'il n'existe pas d'instances (comme dans le cas de l'ontologie camera). Aussi, la diminution de la valeur CR tient à l'augmentation du nombre de concepts pertinents (suite aux abstractions) dans Om, et par conséquent, la valeur CR diminue. D'autre part, cette métrique évalue la quantité, par exemple la mesure IRS qui offre une indication sur le nombre moyen de sous-concepts par concept. IRS est le nombre de sous-concepts divisé par le nombre de concepts. On constate que IRS augmente.

5.2.3 Avec les mesures d'ALANI *et al.*

Tableau 5.7 Les résultats d'expérimentation de RCA-Merge en utilisant la métrique d'ALANI *et al.*

	camera		univ-bench		Pizza	
	Or	Om	Or	Om	Or	Om
CMM	1.00	0.92	1.00	0.80	1.00	0.85
DEM	0.95	1.00	0.54	1.00	0.20	1.00
SSM	1.00	0.90	1.00	0.31	1.00	0.78
BEM	1.00	0.70	1.00	0.06	1.00	.037
Total	0.58	0.58	0.82	0.75	0.68	0.83

Les mesures d'ALANI *et al.* utilisées concernent l'évaluation sémantique d'une ontologie. CMM est une mesure qui permet de montrer si les termes recherchés sont bien couverts par l'ontologie. La métrique CMM est basée sur l'évaluation des concepts nommés, cela explique donc la diminution de sa valeur dans Om par rapport à Or, car les nouveaux concepts pertinents d'Om nécessitent une approche de nommage pertinente. La deuxième mesure DEM traite le détail des connaissances d'un concept en se basant sur ce qui l'entoure (comme l'entrée/sortie des relations, les super/sous-concepts). D'après les résultats du tableau 5.7, DEM augmente, et le détail des connaissances est donc parfait. La mesure SSM utilise la formule suivante: $SSM = 1/K \sum^{n-1} \sum^{n} ssm(Ci,Cj)$ avec ssm (Ci,Cj), le nombre minimal (ou la distance la plus courte) entre deux concepts Ci et Cj, k correspondant à un coefficient et n au nombre de concept. D'après nos résultats, SSM diminue dans Om, c'est donc que la similarité sémantique dans Om ne correspond plus aux termes recherchés. La dernière mesure BEM montre la représentation d'un concept dans l'ontologie. Par ailleurs, la génération d'un modèle ontologique Om à partir de deux autres modèles ontologiques sources ne donne pas le même modèle ontologique Or donc, le graphe Om ne correspond pas au Or, cela explique alors la diminution de la valeur de BEM dans Om.

5.2.4 Avec les mesures de STAAB *et al.*

Tableau 5.8 Les résultats d'expérimentation de RCA-Merge en utilisant la métrique de STAAB *et al.*

	camera	univ-bench	Pizza
Total Concepts in Reference Ontology (Or)	12	43	99
Total Concepts in Computed Ontology (Om)	17	42	104
Total Common Concepts in both Ontology	1	33	83
LP	0.64	0.78	0.79
LR	0.91	0.76	0.83
TP	1.0	1.0	1.0

TR	0.54	0.70	0.42
TF	0.70	0.82	0.59
TF'	0.79	0.79	0.69

Les mesures de STAAB *et al.* traitent essentiellement deux points importants. Premièrement, ces mesures concernent l'évaluation lexicale où il y aura utilisation des mesures de similarité pour détecter les similarités entre deux ontologies. Cette métrique exploite les noms des éléments ontologiques pour effectuer la comparaison. LP et LR sont les deux mesures de précision et de rappel. Selon le tableau 5.8, l'évaluation du contenu lexical d'Om donne des résultats acceptables (supérieurs à 60 %) dans les différents Om testés et ce, malgré l'absence de nommage efficace. Deuxièmement, il existe des mesures d'évaluation structurelle avec TP et TR où il y aura vérification de la précision et rappel des taxonomies. D'après le tableau 5.8, nous constatons que les taxonomies d'Or sont bien couvertes par les taxonomies d'Om, mais le rappel taxonomique TR est moindre suite au manque de nommage. Par conséquent, un nommage pertinent augmentera la qualité ontologique.

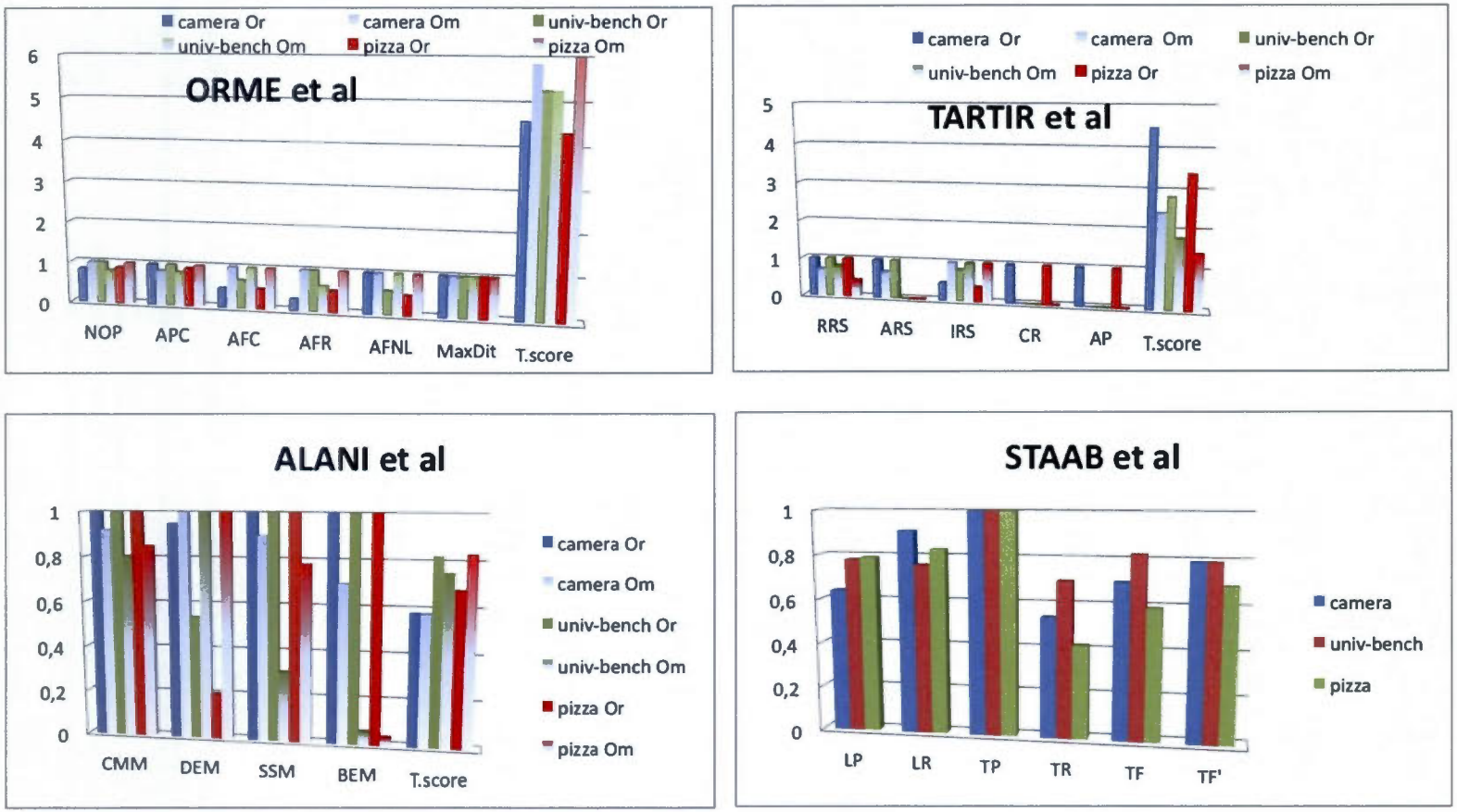


Figure 5.1 Synthèse graphique des résultats d'évaluation de RCA-Merge

5.3 Expérimentation et interprétation de RCA-Merge avec préalignement

Dans cette section, nous testons le module de préalignement ajouté à notre approche en utilisant les ontologies tourism.owl, camera.owl et pizza.owl. Oma désigne l'ontologie fusionnée en utilisant le module de préalignement.

Les tableaux contiennent les pourcentages des mesures et l'appendice C.1 détermine les même tableaux en termes de valeurs (sauf la métrique de STAAB *et al.*, car elle est implémentée uniquement pour donner des pourcentages).

5.3.1 Avec les mesures d'ORME *et al.*

Tableau 5.9 Les résultats d'expérimentation de RCA-Merge avec préalignement en utilisant la métrique d'ORME *et al.*

	tourism			camera			Pizza		
	Or	Om	Oma	Or	Om	Oma	Or	Om	Oma
NOP	1.00	0.76	0.82	1.00	1.00	1.00	0.88	1.00	1.00
APC	1.00	0.64	0.69	1.00	0.80	0.80	1.00	1.00	1.00
AFC	0.47	1.00	1.00	0.47	1.00	1.00	0.49	1.00	1.00
AFR	1.00	0.93	1.00	0.28	1.00	1.00	0.52	1.00	1.00
AFNL	0.80	1.00	1.00	1.00	1.00	1.00	0.48	1.0 0	1.00
MaxDit	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Total	4.27	5.34	5.52	4.59	5.84	5.80	4.32	6.00	6.00

Comme nous l'avons mentionné précédemment, les mesures d'ORME *et al.* évaluent l'ontologie sur le plan quantitatif. Après l'application de préalignement dans RCA-Merge, nous avons constaté que les pourcentages des mesures augmentent par rapport à Om. Par exemple, les mesures NOP et APC augmentent suite à la génération de plus d'abstractions. En d'autres termes, avec le préalignement, il y a une augmentation du nombre de nouveaux concepts pertinents (c'est-à-dire des nouveaux

concepts qui améliorent le contexte de l'ontologie) et cela entraîne une augmentation des relations et du nombre de sortants dans l'ontologie Oma. Donc, avec l'ajout de préalignement, RCA-Merge est capable de détecter d'autres concepts pertinents. En effet, l'élimination des redondances permet à RCAEngine de mieux analyser la structure FCR.

5.3.2 Avec les mesures de TARTIR *et al.*

Tableau 5.10 Les résultats d'expérimentation de RCA-Merge avec préalignement en utilisant la métrique de TARTIR *et al.*

	tourism			camera			Pizza		
	Or	Om	Oma	Or	Om	Oma	Or	Om	Oma
RRS	1.00	0.80	0.71	1.00	0.72	0.53	1.00	0.44	0.58
ARS	1.00	0.24	0.36	1.00	0.70	0.50	0.00	0.00	0.00
IRS	0.43	1.00	1.00	0.47	1.00	1.00	0.39	1.00	1.00
CR	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
AP	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00
Total	2.43	2.04	2.07	4.27	2.43	2.03	3.39	1.44	1.58

Les mesures de TARTIR *et al.* permettent d'étudier la qualité de l'ontologie en termes de quantité et de qualité. L'étude de la qualité aide à analyser la sémantique de l'ontologie. De ce fait, la mesure RRS (qui indique la richesse des relations) diminue dans Oma par rapport à Om et Or. Cela tient à l'augmentation des abstractions (donc, à l'augmentation des sous-concepts). En ce qui concerne la quantité, IRS mesure le nombre moyen de sous-concepts par concept. Donc, IRS d'Oma augmente en termes de valeur, non pas en termes de pourcentage.

5.3.3 Avec les mesures d'ALANI *et al.*

Tableau 5.11 Les résultats d'expérimentation de RCA-Merge avec préalignement en utilisant la métrique d'ALANI *et al.*

	tourism			camera			Pizza		
	Or	Om	Oma	Or	Om	Oma	Or	Om	Oma
CMM	1.00	0.91	1.00	1.00	0.84	0.92	1.00	0.85	0.85
DEM	1.00	0.78	1.00	0.95	1.00	1.00	0.20	1.00	1.00
SSM	1.00	0.62	1.00	1.00	0.90	0.30	1.00	0.78	0.70
BEM	1.00	0.00	0.05	1.00	0.70	0.18	1.00	.037	0.02
Total	1.00	0.64	1.00	0.58	0.58	0.83	0.68	0.83	0.82

La métrique d'ALANI *et al.* traite principalement la sémantique de l'ontologie. Avec la mesure CMM, nous constatons que le pourcentage d'Oma est supérieur à celui d'Om, ce qui signifie que l'ontologie Oma couvre bien les termes recherchés. La deuxième mesure DEM concerne la densité des concepts. D'après le tableau 5.11, on remarque que le pourcentage de DEM augmente (dans la majorité des cas) par rapport à Om, Oma possédant donc des connaissances claires et compréhensibles pour couvrir le domaine cible. La troisième mesure SSM définit la similarité sémantique dans l'ontologie. Selon les résultats dégagés, le pourcentage de SSM diminue dans Oma, mais cela permet de dire que le niveau de la similarité est acceptable. La dernière mesure BEM traite la centralité des concepts. Par ailleurs, RCA-Merge avec alignement et RCA-Merge sans alignement utilisent tous les deux la factorisation avec RCFModeler et la restructuration avec RCAEngine et ONTODesigner. Donc, le changement graphique des modèles ontologiques rend l'interprétation de BEM plus difficile, sans compter que les valeurs de cette dernière varient selon les graphes générés.

5.3.4 Avec les mesures de STAAB *et al.*

Tableau 5.12 Les résultats d'expérimentation de RCA-Merge avec préalignement en utilisant la métrique de STAAB *et al.*

	tourism		camera		Pizza	
	Om	Oma	Om	Oma	Om	Oma
Total Concepts in Reference Ontology (Or)	11	11	12	12	99	99
Total Concepts in Computed Ontology (Om)	11	13	17	20	104	106
Total Common Concepts in both Ontology	10	11	11	11	83	83
LP	0,90	0,84	0,64	0,45	0,79	0,77
LR	0,90	1,00	0,91	0,91	0,83	0,82
TP	1,00	1,00	1,00	1,00	1,00	1,00
TR	1,00	0,72	0,54	0,54	0,42	0,63
TF	1,00	0,84	0,70	0,70	0,59	0,77
TF'	0,95	0,91	0,79	0,79	0,69	0,80

Premièrement, STAAB *et al.* utilisent les noms au sein des ontologies pour effectuer la comparaison entre eux. La mesure LP diminue par rapport à Om, et cela tient à l'absence de nommage pertinent dans RCA-Merge. La mesure de rappel LR d'Oma présente une légère variation par rapport à Om. Cependant, malgré les hauts et les bas de LP et de LR, le contenu lexical d'Oma couvre bien le domaine cible. Deuxièmement, la mesure de la précision taxonomique demeure la même pour toutes les ontologies, tandis que le rappel taxonomique varie d'une ontologie à une autre selon la restructuration ontologique.

Les mesures de STAAB *et al.* utilisent les noms des éléments ontologiques, l'absence d'une approche de nommage pertinente influe donc sur l'évaluation lexicale et structurelle. Dans ce qui suit, nous présenterons le rôle du nommage dans RCA-Merge et nous déterminerons l'influence d'un nommage efficace sur le lexique, la sémantique et la structure de l'ontologie.

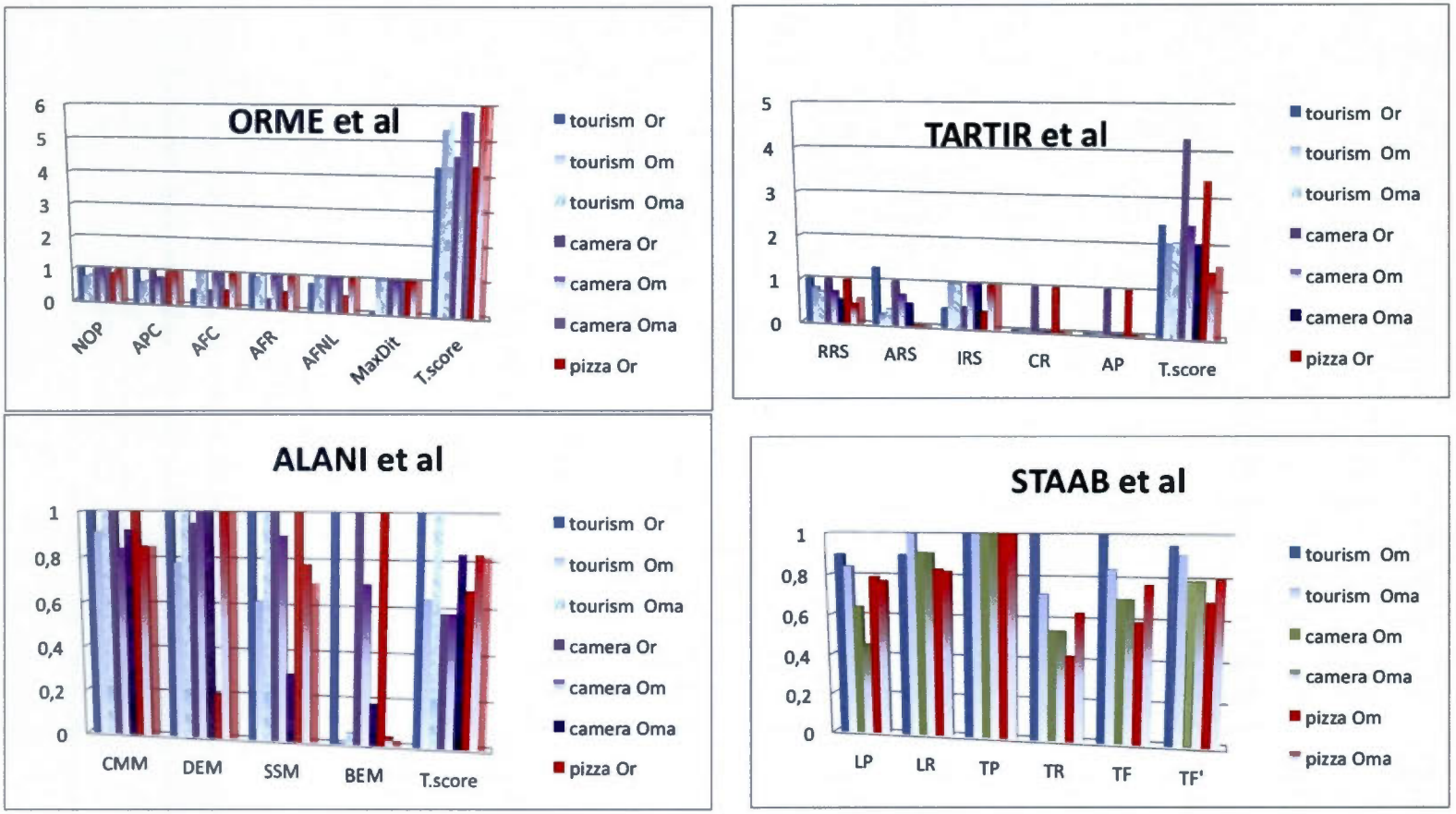


Figure 5.2 Synthèse graphique des résultats d'évaluation d'RCA-Merge avec préalignement

5.4 Rôle du nommage dans RCA-Merge

Pour accentuer le rôle du nommage dans RCA-Merge, nous avons procédé suivant l'approche experte. Selon cette approche, nous avons nommé les nouveaux éléments ontologiques générés après la fusion. Par la suite, en nous basant sur la métrique de STAAB *et al.* et d'ALANI *et al.*, nous évaluons la qualité de l'ontologie fusionnée. Le choix de ces métriques revient à évaluer le niveau lexical et structurel avec STAAB *et al.* ainsi que le niveau sémantique avec ALANI *et al.* En respectant notre protocole de validation, le premier test est effectué sur l'ontologie technical.owl. Les tableaux 5.13 et 5.14 montrent les résultats avec un nommage de niveau intermédiaire, où Oml désigne une ontologie fusionnée et nommée (*Ontology Merged and Labeled*).

Les tableaux contiennent les pourcentages des mesures et l'appendice D.1 détermine les même tableaux en termes de valeurs (sauf la métrique de STAAB *et al.*, car elle est implémentée uniquement pour donner des pourcentages).

Tableau 5.13 Démonstration du rôle du nommage en utilisant les mesures d'ALANI *et al.*

		CMM	DEM	SSM	BEM	Total
Technical	Oma	0.50	1.00	0.62	0.27	0.72
	Oml	0.56	1.00	0.82	0.15	0.79

Tableau 5.14 Démonstration du rôle du nommage en utilisant les mesures de STAAB *et al.*

		Total Concepts in Reference Ontology (Or)	Total Concepts in Computed Ontology (Om)	Total Common Concepts in both Ontology	LP	LR	TP	TR	TF	TF'
Tech-nical	Oma	16	20	9	0.64	0.56	1.00	0.77	0.87	0.68
	Oml	16	14	10	0.71	0.62	0.85	0.37	0.51	0.56

Après observation des résultats, on constate que le nommage permet d'augmenter les mesures LP et LR. Donc, sur le plan lexical, le nommage offre une ontologie de bonne qualité par rapport à Oma. De plus, avec la métrique d'ALANI *et al.*, nous remarquons que cela influe apparemment l'analyse sémantique de l'ontologie. Sur le plan sémantique, la génération d'une ontologie fusionnée de qualité tient donc à une meilleure analyse sémantique ainsi qu'à un nommage pertinent. D'autre part, les valeurs de TP et de TR diminuent dans Oml par rapport à Oma. La raison de cette diminution, c'est que ce nommage de niveau intermédiaire présente une faiblesse: il ne prend pas en considération la structure de l'ontologie. Finalement nous concluons qu'un nommage pertinent doit bien analyser le lexique, la structure et la sémantique de l'ontologie avant d'appliquer les termes choisis. Les tableaux 5.15 et 5.16 présentent les résultats d'un nommage complet ou parfait.

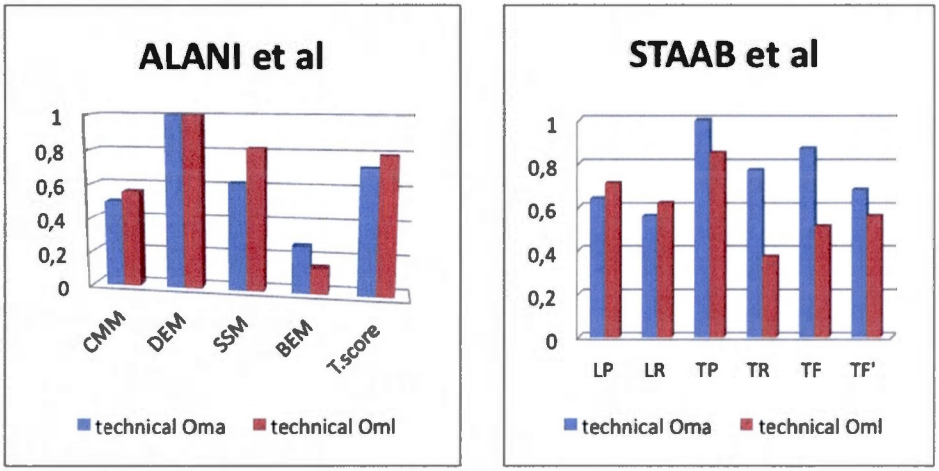


Figure 5.3 Synthèse graphique d'un nommage de niveau intermédiaire

Tableau 5.15 Démonstration du rôle d'un nommage complet en utilisant les mesures d'ALANI *et al.*

		CMM	DEM	SSM	BEM	Total
tourism	Oma	1.0	1.0	1.0	0.05	1.0
	Oml	1.0	1.0	1.0	0.07	1.0

Tableau 5.16 Démonstration du rôle d'un nommage complet en utilisant les mesures de STAAB *et al.*

		Total Concepts in Reference Ontology (Or)	Total Concepts in Computed Ontology (Om)	Total Common Concepts in both Ontology	LP	LR	TP	TR	TF	TF'
tourism	Oma	11	13	11	0.84	1.0	1.0	0.72	0.84	0.91
	Oml	11	13	11	0.91	1.0	1.0	1.00	1.00	1.00

D'après nos constatations des deux derniers tableaux, le nommage complet offre de meilleurs résultats dans les champs lexical, structurel et sémantique. Les mesures LP et LR donnent des valeurs parfaites, ce qui démontre la qualité lexicale de l'ontologie fusionnée. Aussi, le niveau structurel est à 100%, que ce soit pour la précision ou pour le rappel. En outre, les valeurs des mesures d'ALANI *et al.* avec Oml augmentent, ce nommage permet donc une analyse sémantique efficace.

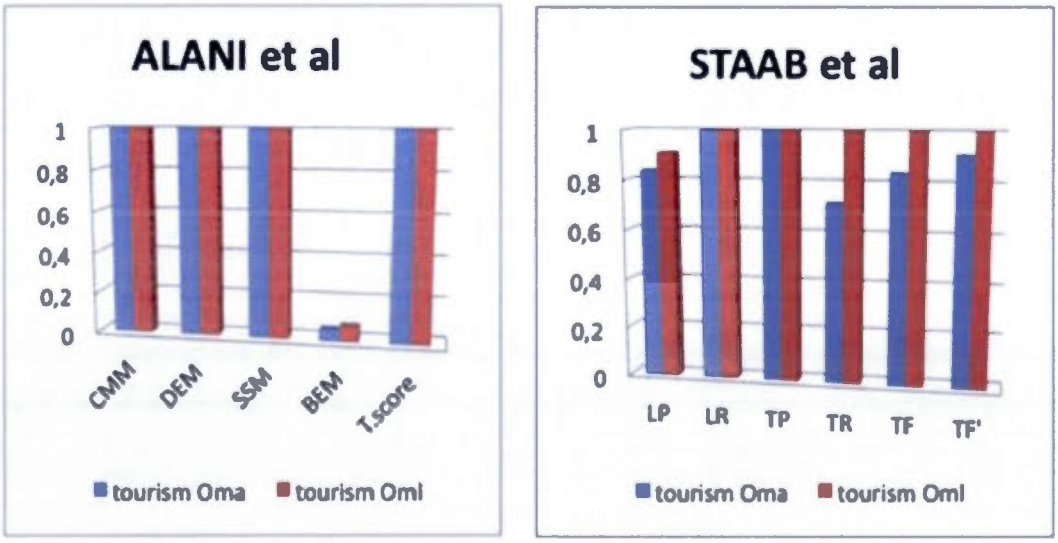


Figure 5.4 Synthèse graphique d'un nommage efficace

Dans les sections précédentes, nous avons montré et validé RCA-Merge et RCA-Merge avec préalignement, et démontré le rôle du nommage dans notre approche. Cela demeure donc à l'intérieur de notre approche puisqu'il n'y a aucune comparaison avec des approches (ou outils) extérieures. Cependant, dans ce qui suit, nous présenterons un autre volet de validation où nous aborderons une étude comparative de RCA-Merge avec d'autres outils de fusion d'ontologies.

5.5 Étude comparative de RCA-Merge avec d'autres outils

Dans cette section, nous comparerons notre outil RCA-Merge (c'est-à-dire RCA-Merge avec préalignement) avec les outils suivants: Neon Toolkit et Protégé[10]. Cette étude comparative est définie en suivant notre protocole de validation: nous prenons les deux sous-ontologies de chacune des ontologies utilisées dans la section 5.3 (tourism.owl, camera.owl et pizza.owl), pour ensuite effectuer la fusion à l'aide de RCA-Merge, Neon Toolkit et Protégé, et finalement comparé la qualité des ontologies fusionnées.

Pour minimiser l'écriture, nous utilisons les notations suivantes dans le tableau 5.17:

- Orc indique l'ontologie fusionnée à l'aide de RCA-Merge.
- On indique l'ontologie fusionnée à l'aide de Neon Toolkit.
- Op indique l'ontologie fusionnée à l'aide de Protégé.

Les tableaux suivants contiennent les pourcentages des mesures et l'appendice E.1 détermine les même tableaux en termes de valeurs (sauf la métrique de STAAB *et al.*, car elle est implémentée uniquement pour donner des pourcentage).

[10] http://protege.stanford.edu/

Tableau 5.17 Comparaison entre les trois outils en se basant sur les différentes métriques d'INUKHUK

La métrique d'ORME *et al.*									
	Tourism			Camera			Pizza		
	Orc	On	Op	Orc	On	Op	Orc	On	Op
NOP	0.82	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
APC	0.69	0.61	0.57	0.80	0.70	0.85	1.00	0.70	1.00
AFC	1.00	1.00	1.00	1.00	1.00	0.97	1.00	0.79	0.99
AFR	1.00	0.93	0.93	1.00	1.00	1.00	1.00	0.31	1.00
AFNL	1.00	0.83	0.83	1.00	1.00	0.76	1.00	0.38	1.00
MaxDit	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Total	5.52	5.38	5.34	5.80	5.70	5.59	6.00	4.20	5.99
La métrique de TARTIR *et al.*									
RRS	0.71	0.87	0.86	0.53	0.79	0.95	0.58	0.69	1.00
ARS	0.36	0.66	0.57	0.50	0.70	0.85	0.00	0.00	0.00
IRS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
CR	0.00	0.00	0.00	0.00	0.70	0.85	0.00	0.35	0.50
AP	0.00	0.00	0.00	0.00	0.70	0.85	0.00	0.70	1.00
Total	2.43	2.53	2.44	2.03	3.91	4.52	1.58	2.76	3.50
La métrique d'ALANI *et al.*									
CMM	1.00	1.00	1.00	0.92	1.0	1.00	0.85	1.00	0.99
DEM	1.00	0.96	0.96	1.00	0.8	0.90	1.00	0.99	1.00
SSM	1.00	1.00	1.00	0.30	1.0	0.69	0.70	0.98	1.00
BEM	0.05	0.67	0.67	0.18	1.0	0.87	0.02	0.93	1.00
Total	1.00	0.98	0.98	0.83	0.92	0.90	0.82	0.99	0.99
La métrique de STAAB *et al.*									
LP	0.84	0.78	0.78	0.45	1.00	1.00	0.77	0.70	1.00
LR	1.00	1.00	1.00	0.91	1.00	1.00	0.82	1.00	1.00
TP	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TR	0.72	1.00	1.00	0.54	1.00	0.87	0.63	0.98	0.98
TF	0.84	1.00	1.00	0.70	1.00	0.93	0.77	0.99	0.99
TF'	0.91	1.00	1.00	0.79	1.00	0.96	0.80	0.99	0.99

En analysant les résultats, nous remarquons que les mesures d'ORME *et al.* d'Orc sont toujours supérieures à celles d'On et Op, et cela se constate dans le score total. Cette augmentation est normale, car ces mesures analysent l'ontologie sur le plan quantitatif. Les valeurs augmentent chez Orc suite à la génération de nouveaux concepts pertinents (les abstractions) et de nouvelles relations. Par conséquent, le nombre de sortants (dont les concepts racines, les concepts feuilles, etc.) augmente. Par ailleurs, avec les deux autres outils, il n'y a pas de génération de nouveaux éléments ontologiques. De plus, une analyse quantitative et qualitative est définie avec TARTIR *et al.*, et nous constatons donc que la mesure IRS de l'ontologie Orc est toujours supérieure à celles d'On et d'Op, car le nombre de sous-concepts (issu à partir des abstractions) dans Orc augmente.

En outre, l'analyse sémantique s'accentue avec les mesures d'ALANI *et al.* Malgré l'absence d'une approche pertinente de nommage, nous constatons que la sémantique d'Orc est trop proche de celles d'On et Op. Alors, l'intégration d'un outil de nommage augmentera la quantité des ontologies fusionnées avec RCA-Merge, et cela a été prouvé dans la section 5.4 (rôle du nommage dans RCA-Merge, le tableau de la métrique d'ALANI *et al.*). Aussi, l'absence de nommage influe sur l'analyse lexicale et structurelle de STAAB *et al.* Sur les plans de la précision (LP et TP) et du rappel (LR et TR), l'ontologie Orc donne des résultats acceptables puisque toujours supérieurs à 70%. Mais en comparant avec On et Op, les valeurs d'Orc sont inférieurs. Cependant, dans la section 5.4 (tableau de métrique de STAAB *et al.*) et suite à un nommage efficace, nous avons prouvé l'augmentation de la qualité d'Orc.

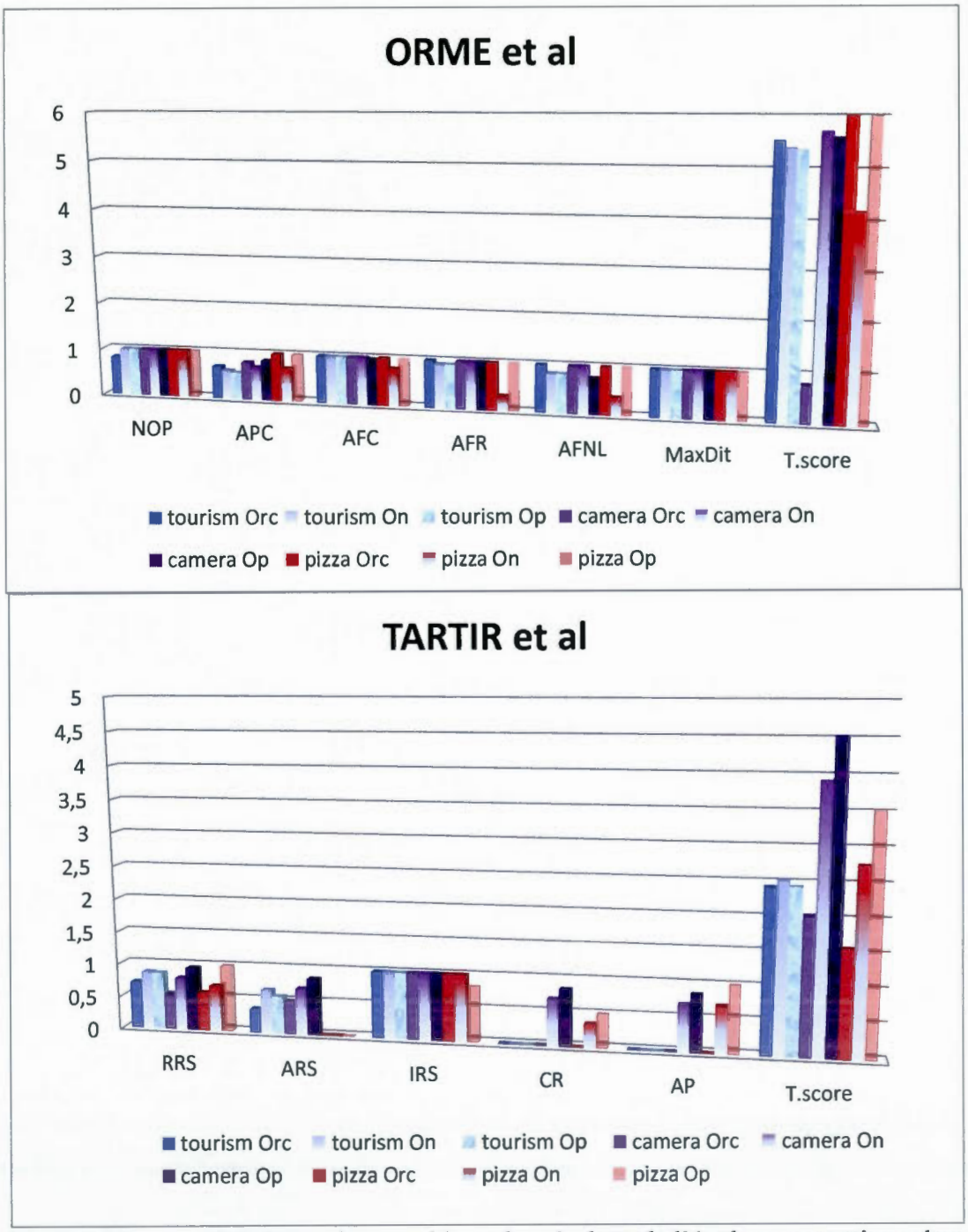


Figure 5.5 Représentation graphique des résultats de l'étude comparative selon ORME *et al.*, et TARTIR *et al.*

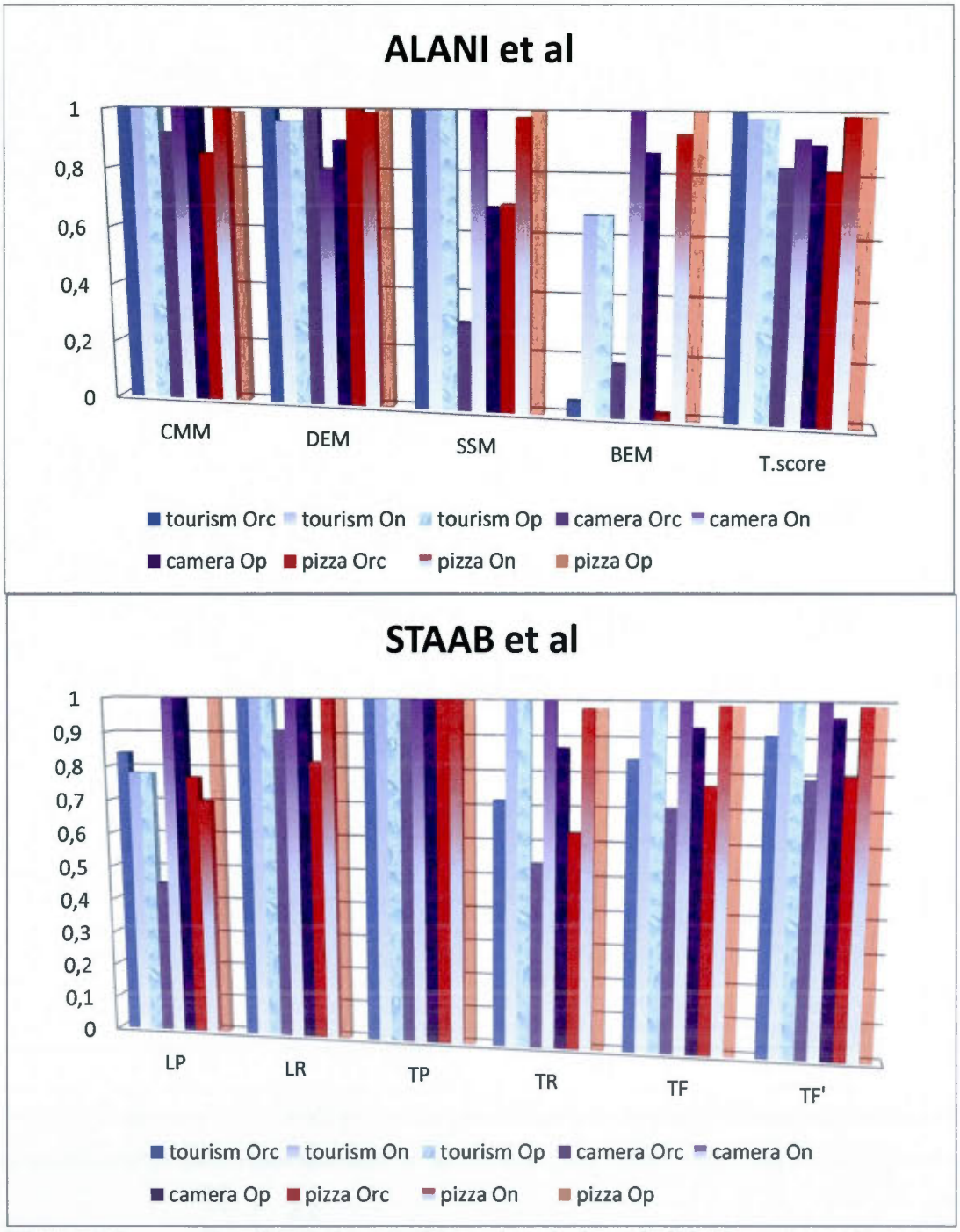


Figure 5.6 Représentation graphique des résultats de l'étude comparative selon ALANI *et al.*, et STAAB *et al.*

Dans ce chapitre, nous avons présenté deux sortes d'expérimentation. La première expérimentation concerne l'aspect interne de notre approche où nous avons évalué et validé le processus de base de RCA-Merge. Ensuite, nous avons expérimenté RCA-Merge avec préalignement pour démontrer l'amélioration réalisée. À la fin, nous avons montré l'importance du nommage dans RCA-Merge avec préalignement, surtout que notre approche de fusion génère de nouveaux éléments ontologiques qui doivent être nommés. La deuxième expérimentation s'est concentrée sur la comparaison de RCA-Merge (améliorée) avec d'autres outils reconnus (Neon Toolkit et Protégé). Eu égard à cette comparaison, nous avons remarqué que les valeurs générées suite à l'application RCA-Merge sont généralement acceptables malgré l'absence d'une approche de nommage efficace. Par ailleurs, notre approche détermine l'avantage de créer des abstractions et de nouvelles relations (suite aux analyses des données relationnelles ARC), ce qu'on ne peut faire avec Neon Toolkit et Protégé.

CHAPITRE VI

CONCLUSION

Dans notre travail, nous avons présenté une approche de fusion d'ontologies basée sur la factorisation et la restructuration de deux ontologies. Nous avons utilisé l'outil de AFC, et surtout son extension ARC, pour toucher un autre volet dans l'analyse conceptuelle. Avec notre approche, nous avons proposé un processus de base qui comporte principalement trois modules connectés l'un à l'autre en termes d'entrées/sorties. Le premier module RCFModeler génère la factorisation des deux ontologies sous la forme d'une structure FCR, le deuxième module analyse les données FCR pour donner un ensemble de treillis FTR, et le dernier module correspond à la restructuration des ontologies à partir de FTR pour générer l'ontologie fusionnée.

Nous avons amélioré RCA-Merge avec un module de préalignement qui intervient entre RCFModeler et RCAEngine afin de nettoyer la structure FCR générée. L'amélioration effectuée influe sur les treillis FTR et par conséquent sur l'ontologie fusionnée. Cette amélioration a enrichi RCA-Merge en renforçant le fonctionnement de RCAEngine.

Nous avons montré le rôle d'un nommage pertinent dans notre approche. RCA-Merge permet de générer de nouveaux concepts pertinents (suite aux abstractions) et de nouvelles relations, l'attribution de nouveaux noms à ces nouveaux éléments ontologiques étant donc essentielle pour améliorer la qualité de l'ontologie fusionnée. La mise en œuvre du nommage doit être réalisée à l'aide d'un outil vraiment pertinent

puisqu'il faut passer par l'analyse de quelques aspects comme la sémantique et la structure.

Une grande partie de notre travail a porté sur l'expérimentation et la validation. Nous avons procédé avec deux volets d'évaluation. Le premier volet s'est concentré sur l'aspect interne de RCA-Merge (dont nous avons évalué et validé son processus de base en tant que nouvelle approche basée sur l'ARC), nous avons expérimenté notre outil avec l'ajout d'un module de préalignement et nous avons démontré le rôle d'un nommage pertinent dans RCA-Merge amélioré. Le deuxième volet de l'expérimentation s'est quant à lui concentré sur l'aspect externe à l'aide d'une étude comparative de RCA-Merge par rapport aux deux autres outils (Neon Toolkit et Protégé) de fusion d'ontologies, mais ces deux outils ne permettent pas l'analyse relationnelle des concepts.

Nous avons réalisé tous les objectifs prévus de RCA-Merge et nous avons également intégré une étape de préalignement. Sur le plan de l'implémentation, nous avons concrétisé tout le processus de RCA-Merge et nous avons aussi implémenté le module de préalignement. Cependant, il serait avantageux de développer un outil efficace de nommage qui puisse interagir avec RCA-Merge.

Dans une perspective d'amélioration, nous planifierons l'implémentation d'un outil pertinent et une application complète de nommage des nouveaux éléments ontologiques. Nous prévoyons également l'utilisation de quelques sources de données externes comme WordNet (qui fournit une structure lexicale et sémantique des données) et DBpedia (qui offre une structure de données sous la forme d'un modèle ontologique). Cet outil de nommage sera intégré dans la plateforme INUKHUK en tant que module séparé pour pouvoir l'utiliser avec les autres services.

Notre projet constitue une avancée de plus dans l'ingénierie des ontologies, surtout sur le plan de la concrétisation de RCA-Merge, car peu de projets concrets existent.

Aussi, si RCA-Merge sur la base d'ARC soulève quelques problématiques, cela ouvre des perspectives intéressantes (comme le nommage efficace) au développement d'INUKHUK spécifiquement et à l'ingénierie des ontologies globalement.

BIBLIOGRAPHIE

Alani, H. et Brewster, C. (2005). Ontology Ranking Based on the Analysis of Concept Structures. K-CAP '05 Proceedings of the 3rd International Conference on Knowledge Capture, Actes du colloque, 2005: ACM New York, NY, USA.

Alani, H., C. Brewster, N. Shadbolt, I. Cruz, S. Decker, D. Allemang, C. Preist et D. Schwabe (dir.). (2006). *Ranking Ontologies with AKTiveRank. ISWC'06 Proceedings of the 5th International Conference on The Semantic Web*, Actes du colloque, 2006: Springer-Verlag Berlin, Heidelberg.

Bendaoud, R., M.R. Hacene, Y.Toussaint, B. Delecroix, A. Napoli, G. Flouris et M. d'Aquin (dir.). (2007). Text-based ontology construction using relational concept analysis. *International Workshop on Ontology Dynamics - IWOD to be held as part of the ESWC-07 conference*, Actes du colloque, 2007, Innsbruck, Austria.

Borst, W.N. (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse.* Universiteit Twente, The Netherlands. P.H.D.

Chen, R.-C., C.-T. Bau et C.-J. Yeh, (2011). Merging Domain Ontologies Based on the WordNet System and Fuzzy Formal Concept Analysis Techniques. *Applied Soft Computing, 11*(2), 1908-1923.

Dellschaft, K., S.Staab, I. Cruz, S. Decker, D. Allemang, C. Preist et D. Schwabe (dir.). (2006). *On How to Perform a Gold Standard Based Evaluation of Ontology Learning. ISWC'06 Proceedings of the 5th International Conference on The Semantic Web*, Actes du colloque, 2006: Springer-Verlag Berlin, Heidelberg.

Dou, D., D. McDermott et P. Qi, (2003). Ontology Translation on the Semantic Web. *Dans Meersman, R., Z. Tari et D.C. Schmidt (dir.), On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE* (Vol. 2888, p. 952-969): Springer Berlin Heidelberg.

Euzenat, J. (2004). An API for Ontology Alignment. Dans McIlraith, S. A., D. Plexousakis et F. v. Harmelen (dir.), *The Semantic Web – ISWC 2004* (Vol. 3298, p. 698-712): Springer Berlin Heidelberg.

Falleri, J.-R., G.Arévalo, M. Huchard, C. Nebut, P.W. Eklund, J. Diatta et M. Liquiere (dir.). (2007). Use of Model Driven Engineering in Building Generic FCA/RCA Tools. *Proceedings of the Fifth International Conference on Concept Lattices and Their Applications, CLA*, Actes du colloque, 2007, Montpellier, France: CEUR-WS.org.

Ganter, B., G. Stumme et R. Wille (2005). *Formal Concept Analysis Foundations and Applications*. (Vol. 10).

Giri, K. (2011). Role of Ontology in Semantic Web. *DESIDOC Journal of Library & Information Technology, Vol. 31*(issue 2), 116-120.

Godin, R. H. Mili et A. Paepcke (dir.). (1993). Building and maintaining analysis-level class hierarchies using Galois Lattices. *OOPSLA '93 Proceedings of the eighth annual conference on Object-oriented programming systems, languages, and applications* Actes du colloque, 1993, ACM New York, NY, USA.

Gruber, T.R. (1993). A Translation Approach to Portable Ontology Specifications. Dans Gaines, B. R. et J. H. Boose (dir.), *Knowledge Acquisition - Special issue: Current issues in knowledge modeling* (Vol. 5 Issue 2, p. 199-220): Academic Press Ltd. London, UK, UK.

Gruber, T.R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies, Volume 43* (Issues 5-6), 907–928.

Guan-yu, L., L. Shu-peng et Yan Zhao. (2010) Formal concept analysis based ontology merging method. Dans Communication présentée à/au Computer Science and Information Technology (ICCSIT) p. 279-282) Chengdu: 3rd IEEE International Conference on. Récupéré de http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5564899

Kalyanpur, A., D.J. Pastor, S. Battle, J. Padget, F. Maurer et G. Ruhe (dir.). (2004). Automatic Mapping of OWL Ontologies into Java. SEKE, Actes du colloque, 2004, Récupéré de http://www.bibsonomy.org/bibtex/245a0fa1ec83dd937ccbc6affe99fe15f/fparreiras

Kremen, P., M. Smid et Z. Kouba, (2011). *OWLDiff: A Practical Tool for Comparison and Merge of OWL Ontologies. DEXA '11 Proceedings of the 2011 22nd International Workshop on Database and Expert Systems Applications*, Actes du colloque, 2011: IEEE Computer Society Washington, DC, USA.

Lambrix, P. et H. Tan (2006). SAMBO-A system for aligning and merging biomedical ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web, 4*(3), 196-206. http://dx.doi.org/10.1016/j.websem.2006.05.003

Macko, J. (2012). Formal Concept Analysis as a Framework for Business Intelligence Technologies. Dans F. Domenach, D.I. Ignatov et J. Poelmans (dir.), *Formal Concept Analysis: 10th International Conference, ICFCA 2012, Leuven, Belgium* (Vol. 7278, p. 195-210): Springer Berlin Heidelberg.

Maiz, N., O. Boussaid et F. Bentayeb (2008) Fusion automatique des ontologies par classification hiérarchique pour la conception d'un entrepôt de données. *Dans Communication présentée à /au 8èmes Journées Francophones Extraction et Gestion des Connaissances p. 17-27) France*, Récupéré de https://www.rocq.inria.fr/axis/personnel/Marie-Aude.Aufaure/Actes%20Atelier%20SImSem.pdf#page=17

McGuinness, D.L., R. Fikes et J. Rice (2000). An Environment for Merging and Testing Large Ontologies. *In Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning* Actes du colloque, 2000, Breckenridge, Colorado, USA.

Noy, N.F. et M.A. Musen (2000). *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, Actes du colloque, 2000: AAAI Press.

Noy, N.F. et M.A. Musen (2003). The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies, 59*(6), 983-1024. http://dx.doi.org/10.1016/j.ijhcs.2003.08.002

Orme, A.M., H. Yao et L.H. Etzkorn (2007). Indicating ontology data quality, stability, and completeness throughout ontology evolution. *Journal of Software Maintenance and Evolution: Research and Practice, 19*(1), 49-75. http://dx.doi.org/10.1002/smr.341

Peggy, C., D. Felix et G. Bernhard (2013). *Formal Concept Analysis: 11th International Conference, ICFCA 2013, Dresden, Germany*. (Vol. 7880): Springer Berlin Heidelberg.

Rouane-Hacene, A.M. (2013). *Analyse relationnelle de concepts en action* PK-1140 201 President-Kennedy, Montreal Qc. Récupéré de http://www.latece.uqam.ca/en/ai1ec_event/seminaire-amine-m-rouane-hacene/?instance_id

Rouane-Hacene, M., S. Fennouh, R. Nkambou et P. Valtchev (2010). Refactoring Of Ontologies: Improving The Design Of Ontological Models With Concept Analysis. *Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on* Actes du colloque, 2010, Arras

Rouane-Hacene, M., M. Huchard, A. Napoli et P. Valtchev (2013). Relational concept analysis: mining concept lattices from multi-relational data. *Annals of Mathematics and Artificial Intelligence, 67*(1), 81-108. http://dx.doi.org/10.1007/s10472-012-9329-3

Rouane-Hacene, M., P. Valtchev et R. Nkambou (2011b). Supporting ontology design through large-scale FCA-based ontology restructuring. Dans Andrews, S., S. Polovina, B. Akhgar et R. Hill (dir.), *ICCS'11 Proceedings of the 19th international conference on Conceptual structures for discovering knowledge* (p. 257-269): Springer-Verlag Berlin, Heidelberg.

Rouane, M.H., M. Dao, M. Huchard et P. Valtchev (2007). *Aspects de la réingénierie des modèles UML par analyse de données relationnelles. lirmm-00163388, version 1*, Actes du colloque, 2007, Toulouse, France

Rouane, M.H., M. Huchard, A. Napoli et P. Valtchev (2007). A Proposal for Combining Formal Concept Analysis and Description Logics for Mining Relational Data Dans Kuznetsov, S. O. et S. Schmidt (dir.), *Formal Concept Analysis* (Vol. 4390, p. 51-65): Springer Berlin Heidelberg.

Stumme, G. et A. Maedche, (2001). FCA-MERGE: bottom-up merging of ontologies. *IJCAI'01 Proceedings of the 17th international joint conference on Artificial intelligence*, Actes du colloque, 2001, USA: Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Sun, H., W. Fan, W. Shen et T. Xiao (2010). Ontology fusion in HLA-based collaborative product development. *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, Actes du colloque, 2010, Istanbul

Tartir, S., I.B. Arpinar, M. Moore, A.P. Sheth et B. Aleman-Meza (2005) OntoQA: Metric-Based Ontology Quality Analysis. Dans Communication présentée à /au Proceedings of IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources Houston, TX

Tilley, T., R. Cole, P. Becker et P. Eklund (2005). A Survey of Formal Concept Analysis Support for Software Engineering Activities. Dans Ganter, B., G. Stumme et R. Wille (dir.), *Formal Concept Analysis* (Vol. Volume 3626, p. 250-271): Springer Berlin Heidelberg.

Valtchev, P., D. Grosser, C. Roume et M.R. Hacene (2003). Galicia: An Open Platform for Lattices *In Using Conceptual Structures: Contributions to the 11th Intl. Conference on Conceptual Structures (ICCS'03)* (p. 241--254): Shaker Verlag.

Wille, R. (2009). Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. Dans Ferré, S. et S. Rudolph (dir.), *Formal Concept Analysis* (Vol. Volume 5548, p. 314-339): Springer Berlin Heidelberg.

APPENDICE A

L'ARCHITECTURE DE RCA-MERGE AVEC PRÉALIGNEMENT

A.1 Le processus de base de RCA-Merge: RCFModeler

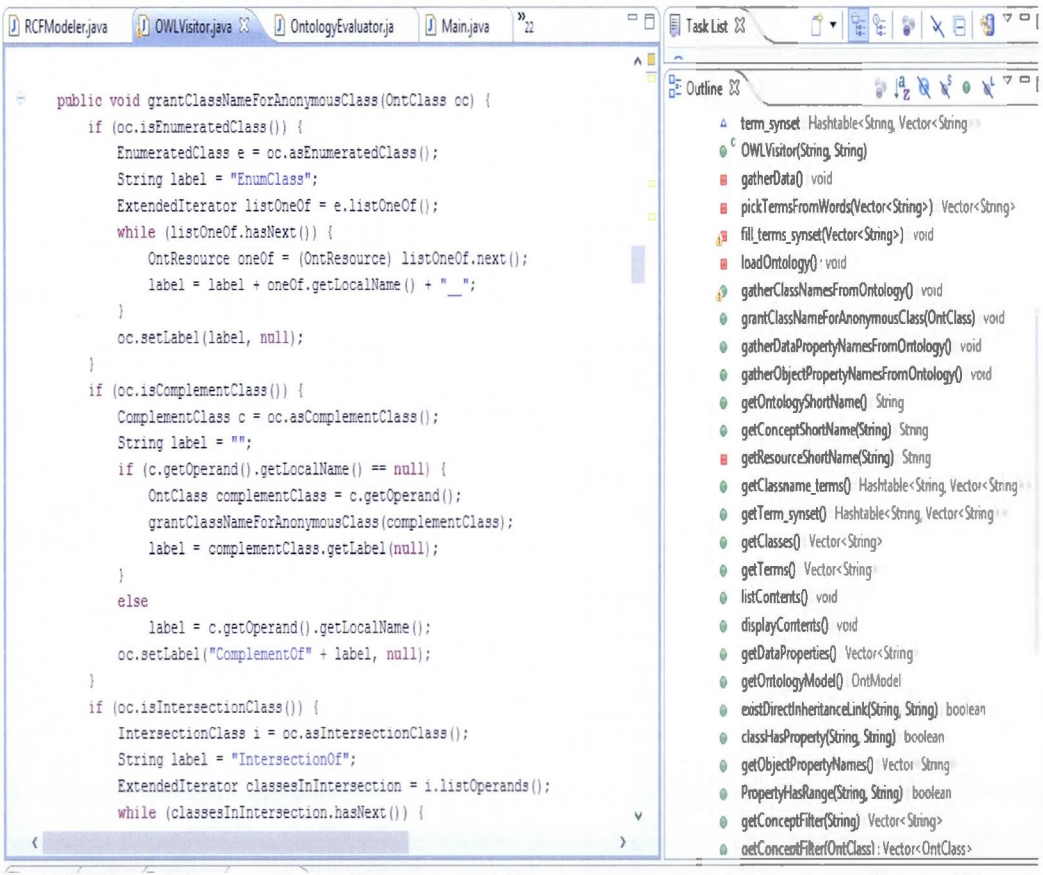


Figure A.1 Les méthodes pertinentes pour l'extraction des éléments ontologiques

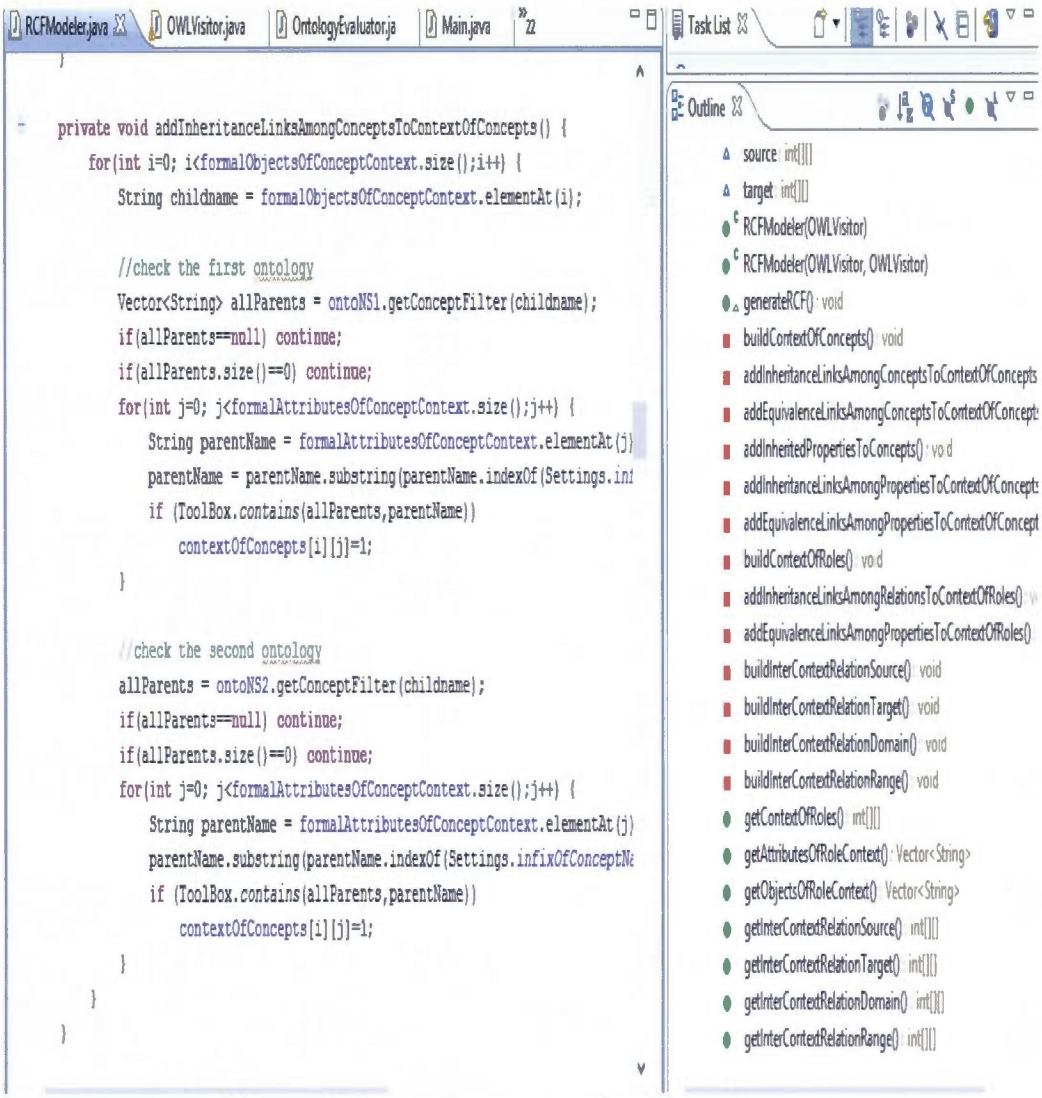


Figure A.2 La construction des contextes

```
[Relational Context]
Ontology-RCF
[Binary Relation]
concept-context
Reviewer | Report | Paper | Paper | Author |
Reviewer | Report | Paper | Paper | Author | title | passw | name | login | email | contents | comments | affiliation | rank |
1 0 0 0 0 0 1 1 1 1 0 0 1 0
0 1 0 0 0 0 0 0 0 0 0 1 0 0
0 0 1 1 0 1 0 0 0 0 1 0 0 0
0 0 1 1 0 1 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 1 0 1 0 0 1 1
[Binary Relation]
role-context
rate | compose | write |
rate | compose | write |
1 0 0
0 1 0
0 0 1
[Inter Object Binary Relation]
source
concept-context
role-context
Reviewer | Report | Paper | Paper | Author |
rate | compose | write |
0 1 0
1 0 0
0 0 0
0 0 0
0 0 1
[Inter Object Binary Relation]
target
concept-context
role-context
Reviewer | Report | Paper | Paper | Author |
rate | compose | write |
0 0 0
0 1 0
1 0 1
1 0 1
0 0 0
[Inter Object Binary Relation]
dom
role-context
concept-context
rate | compose | write |
Reviewer | Report | Paper | Paper | Author |
0 1 0 0 0
1 0 0 0 0
0 0 0 0 1
[Inter Object Binary Relation]
ran
role-context
concept-context
rate | compose | write |
Reviewer | Report | Paper | Paper | Author |
0 0 1 1 0
0 1 0 0 0
0 0 1 1 0
[END Relational Context]
```

Figure A.3 Fichier d'extension .rcf

A.2 Le processus de base de RCA-Merge: RCAEngine

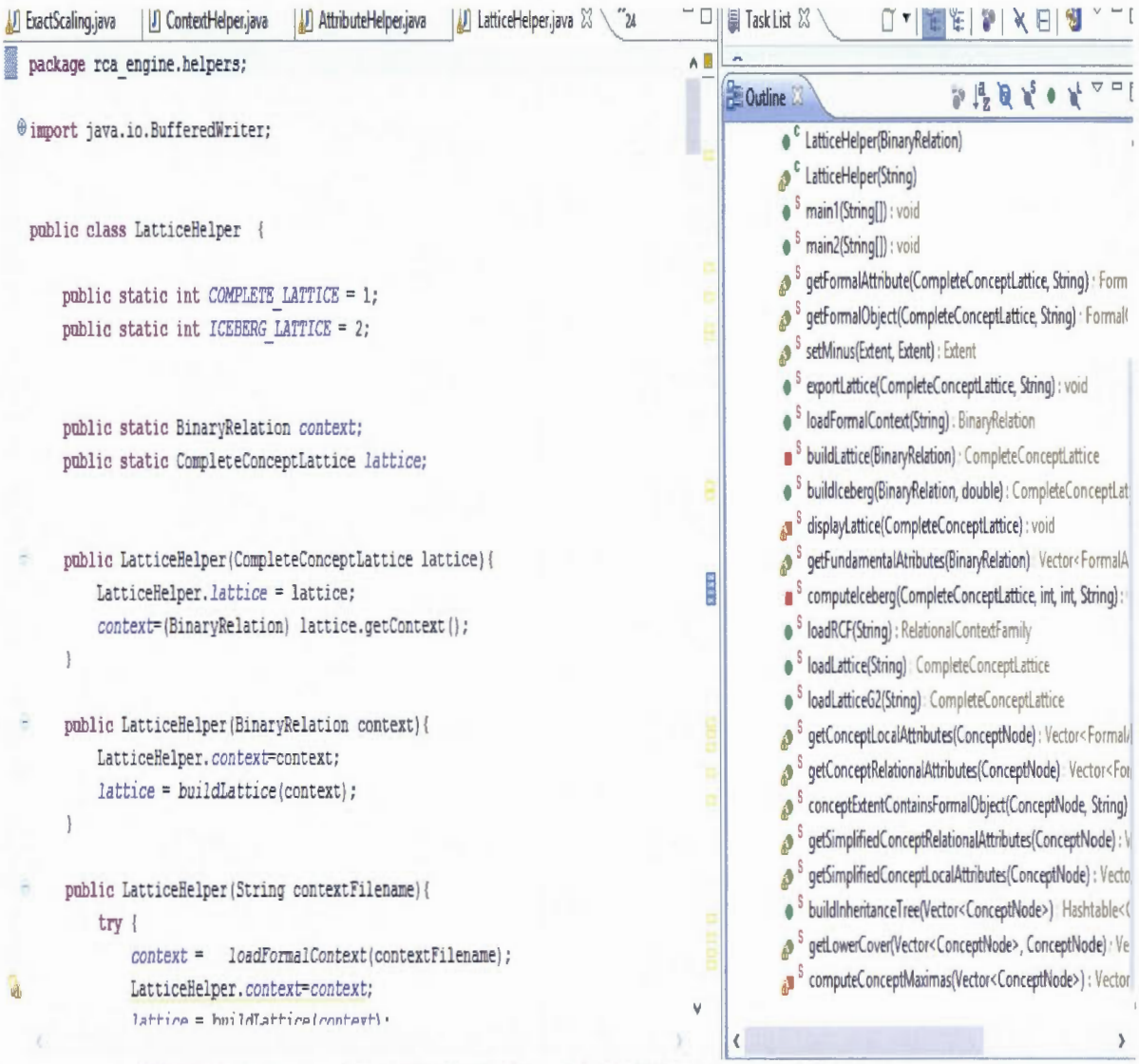


Figure A.4 La manipulation des treillis avec RCAEngine

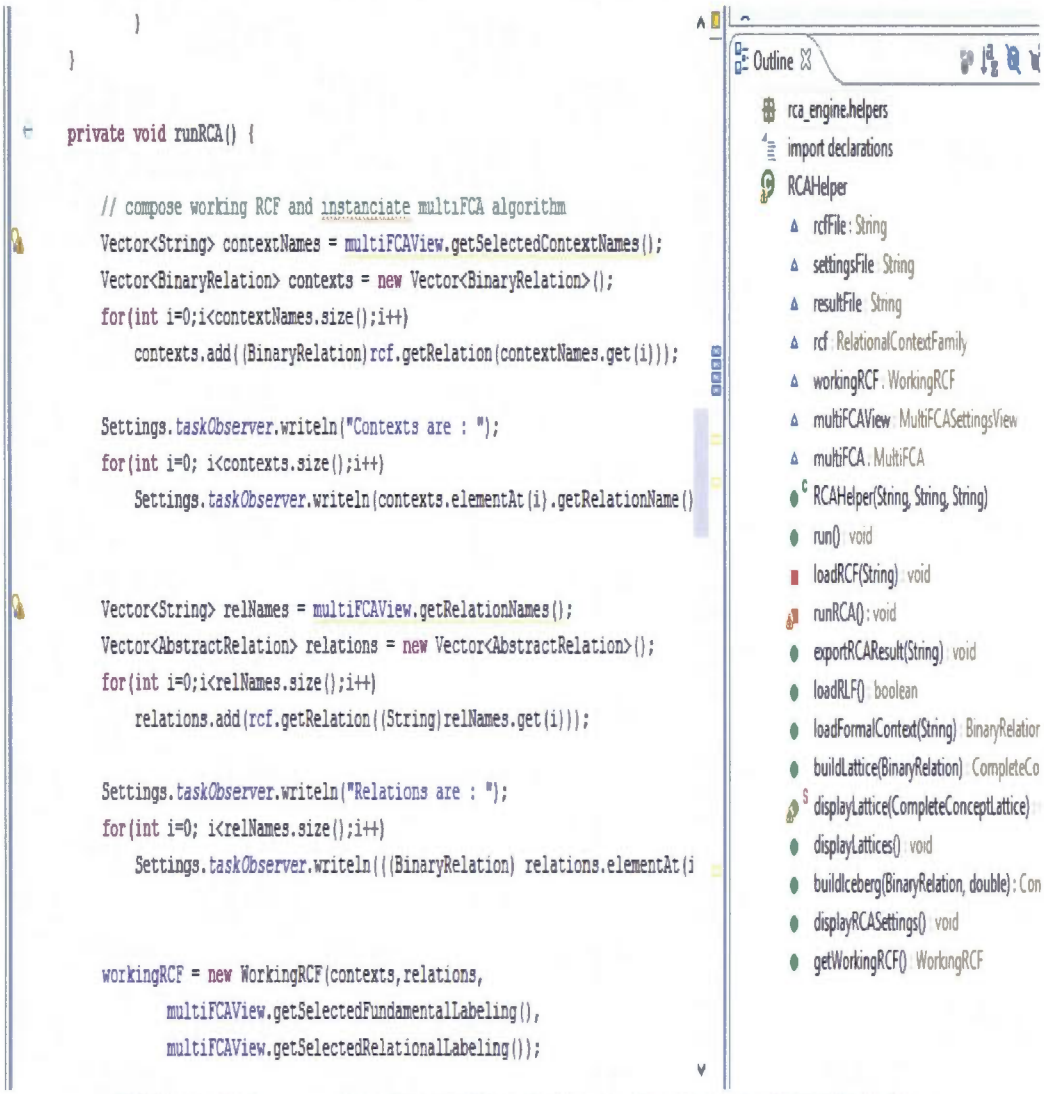


Figure A.5 Quelques fonctions pertinentes de RCAEngine

A.3 Le processus de base de RCA-Merge: ONTODesigner

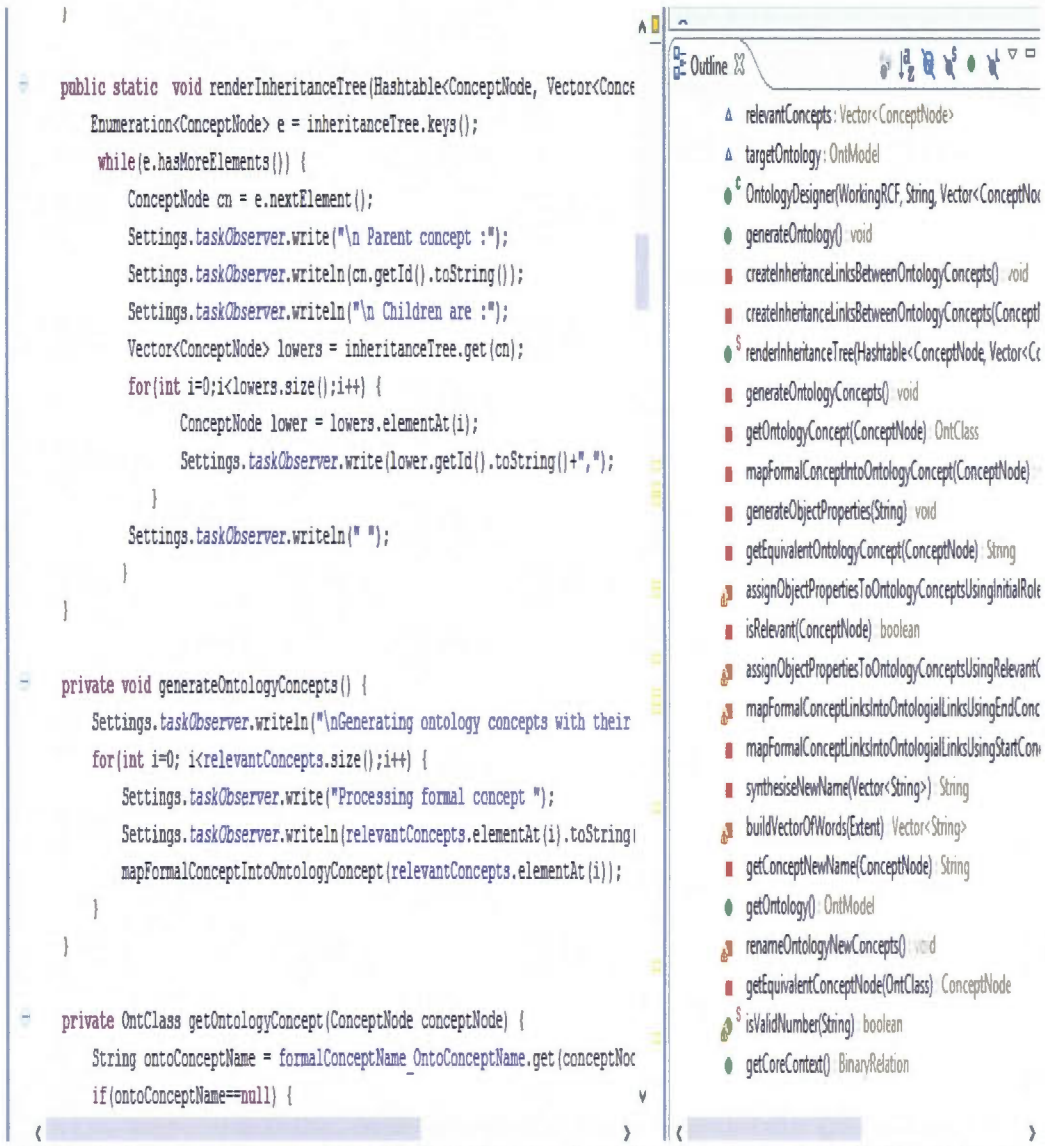


Figure A.6 Les méthodes les plus pertinentes de génération de l'ontologie fusionnée (ONTODesigner)

A.1 RCA-Merge avec préalignement

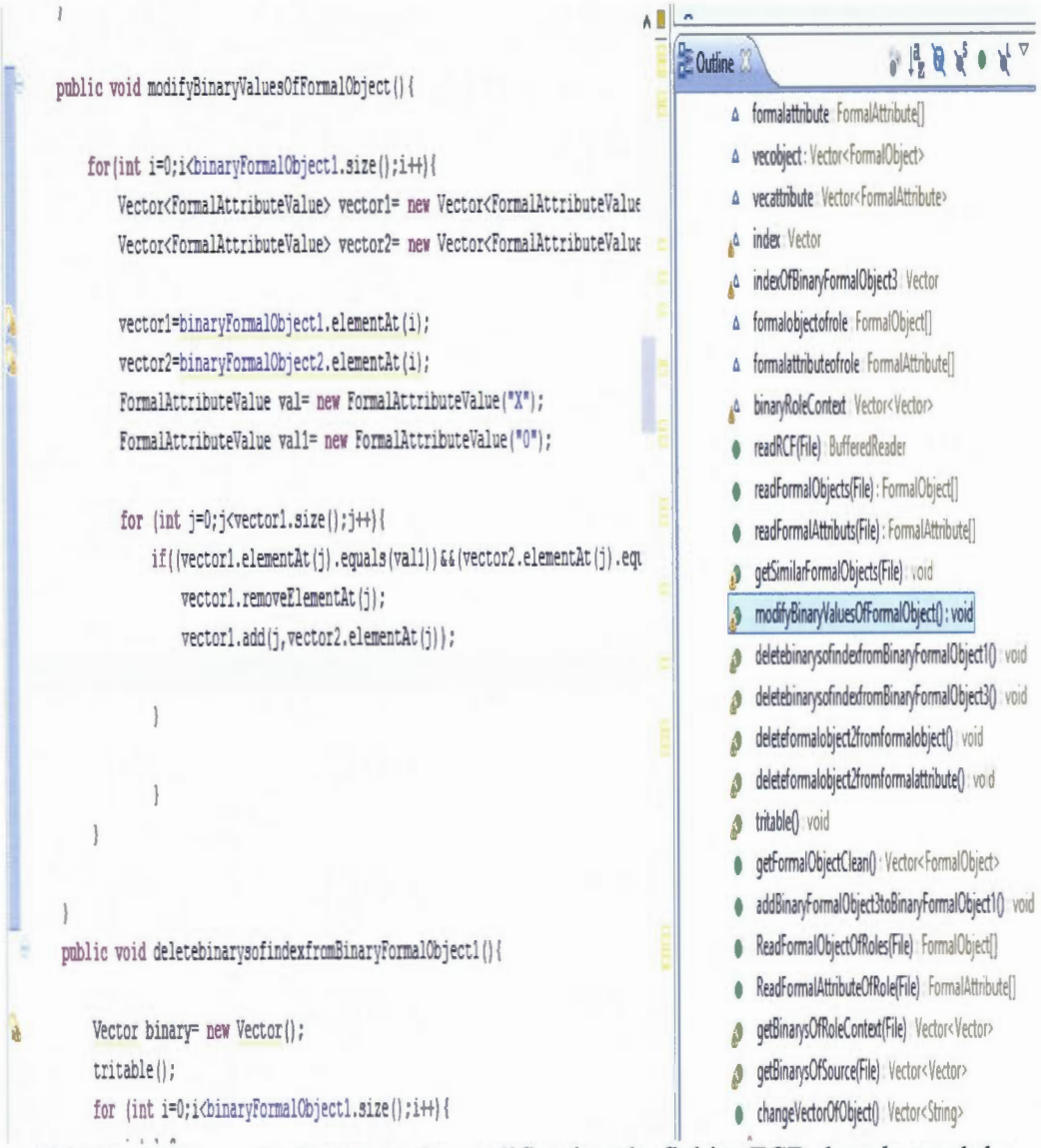


Figure A.7 La lecture et la modification du fichier FCR dans le module RCFAligner

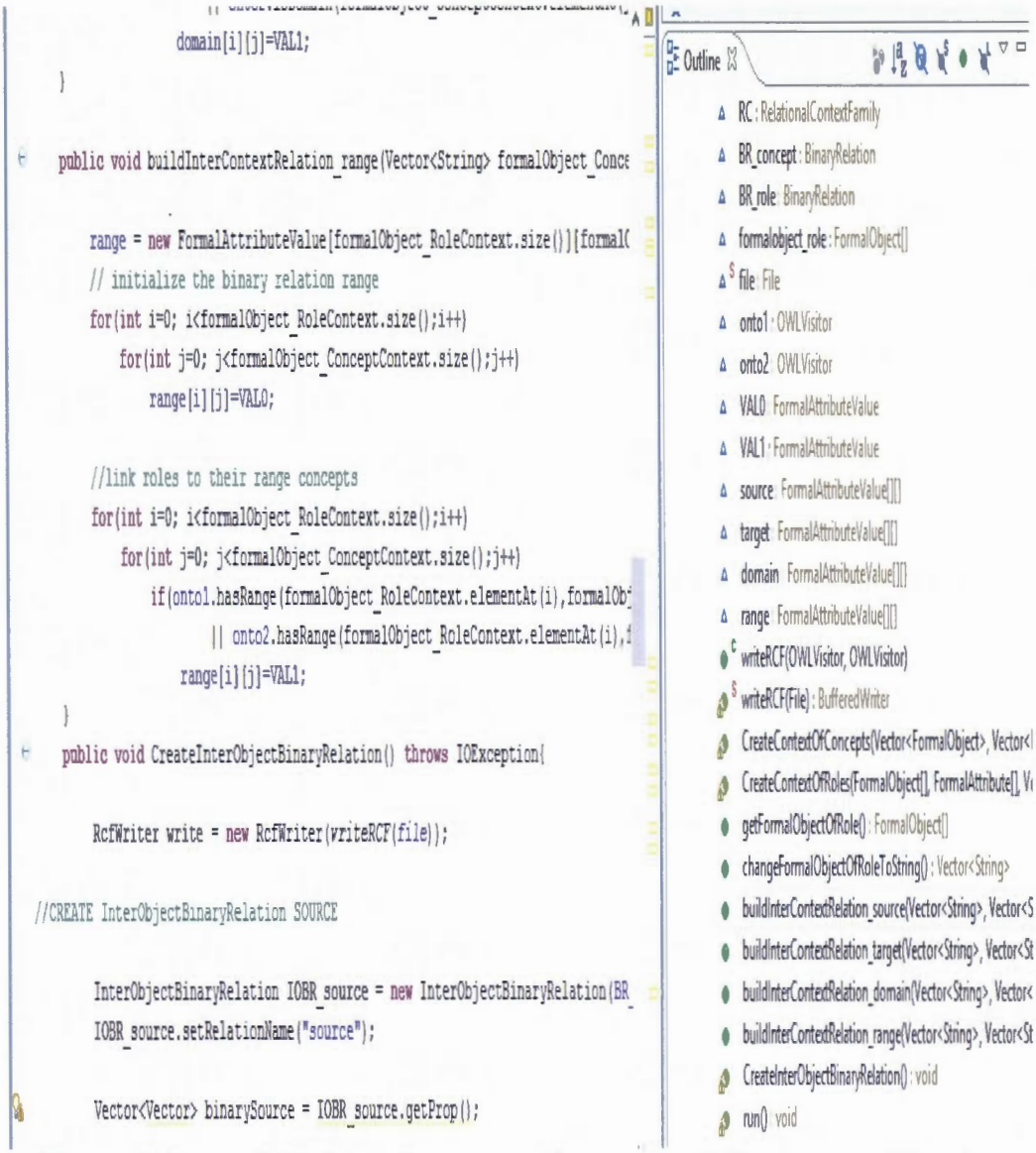


Figure A.8 L'écriture d'une nouvelle structure FCR après modification

```
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.co-ode.org/ontologies/pizza/2006/07/18/pizza.owl#DeepPanBase'/>
    <entity2 rdf:resource='http://www.co-ode.org/ontologies/pizza/2006/07/18/pizza.owl#DeepPanBase'/>
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.co-ode.org/ontologies/pizza/2006/07/18/pizza.owl#CaperTopping'/>
    <entity2 rdf:resource='http://www.co-ode.org/ontologies/pizza/2006/07/18/pizza.owl#CaperTopping'/>
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.co-ode.org/ontologies/pizza/2006/07/18/pizza.owl#VegetarianPizzaEquivalent1'/>
    <entity2 rdf:resource='http://www.co-ode.org/ontologies/pizza/2006/07/18/pizza.owl#Pizza'/>
    <relation>[redacted]</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.6612903225806451</measure>
  </Cell>
</map>
<map>
  <Cell>
    <entity1 rdf:resource='http://www.co-ode.org/ontologies/pizza/2006/07/18/pizza.owl#GreenPepperTopping'/>
    <entity2 rdf:resource='http://www.co-ode.org/ontologies/pizza/2006/07/18/pizza.owl#GreenPepperTopping'/>
    <relation>=</relation>
    <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
  </Cell>
</map>
```

Figure A.9 Extrait de la structure Align

APPENDICE B

VALIDATION DE RCA-MERGE

Tableau B.1 Les résultats de RCA-Merge selon ORME *et al.*

	camera		univ-bench		Pizza	
	Or	Om	Or	Om	Or	Om
NOP	15,0	18,0	32,0	26,0	8,00	9,00
APC	1,25	1,05	0,74	0,61	0,09	0,08
AFC	0,58	1,23	1,27	1,83	3,15	6,42
AFR	1,00	3,50	6,11	3,85	19,6	37,5
AFNL	3,50	4,00	5,00	8,55	15,0	30,68
MaxDit	2,00	2,00	3,00	3,00	6,00	6,00
Total	4,59	5,84	5,28	5,27	4,32	6,00

Tableau B.2 Les résultats de RCA-Merge selon TARTIR *et al.*

	camera		univ-bench		Pizza	
	Or	Om	Or	Om	Or	Om
RRS	0,75	0,54	0,48	0,39	0,08	0,03
ARS	0,66	0,47	0,16	0,00	0,00	0,00
IRS	0,41	0,88	0,79	0,95	0,84	2,12
CR	0,08	0,00	0,00	0,00	0,02	0,00
AP	0,16	0,00	0,00	0,00	0,05	0,00
Total	4,47	2,43	2,83	1,81	3,39	1,44

Tableau B.3 Les résultats de RCA-Merge selon ALANI *et al.*

	camera		univ-bench		Pizza	
	Or	Om	Or	Om	Or	Om
CMM	7,600	7,00	30,20	24,20	74,40	63,04
DEM	3,41	3,58	4,58	8,47	10,19	50,29
SSM	2,65	2,39	7,51	2,35	20,77	13,007
BEM	15,625	11,05	72,90	5,07	1840,16	70,82
Total	0,580	0,58	0,82	0,75	0,68	0,83

APPENDICE C

VALIDATION DE RCA-MERGE AVEC PRÉALIGNEMENT

Tableau C.1 Les résultats de RCA-Merge avec préalignement selon ORME *et al.*

	tourism			camera			Pizza		
	Or	Om	Oma	Or	Om	Oma	Or	Om	Oma
NOP	17,0	13,0	14,0	15,0	18,0	24,0	8,00	9,00	12,0
APC	1,54	1,00	1,07	1,25	1,05	1,00	0,09	0,08	0,11
AFC	0,18	0,38	0,61	0,58	1,23	4,60	3,15	6,42	6,43
AFR	0,66	0,62	1,60	1,00	3,50	18,5	19,6	37,5	38,27
AFNL	2,00	2,50	4,00	3,50	4,00	8,50	15,0	30,68	28,70
MaxDit	0,00	1,00	1,00	2,00	2,00	4,00	6,00	6,00	6,00
Total	4,27	5,34	5,52	4,59	5,84	5,80	4,32	6,00	6,00

Tableau C.2 Les résultats de RCA-Merge avec préalignement selon TARTIR et al.

	tourism			camera			Pizza		
	Or	Om	Oma	Or	Om	Oma	Or	Om	Oma
RRS	0,89	0,72	0,63	0,75	0,54	0,40	0,08	0,03	0,05
ARS	0,63	0,15	0,23	0,66	0,47	0,33	0,00	0,00	0,00
IRS	0,18	0,38	0,61	0,41	0,88	1,50	0,84	2,12	2,10
CR	0,00	0,00	0,00	0,08	0,00	0,00	0,02	0,00	0,00
AP	0,00	0,00	0,00	0,16	0,00	0,00	0,05	0,00	0,00
Total	2,43	2,04	2,07	4,27	2,43	2,03	3,39	1,44	1,58

Tableau C.3 Les résultats de RCA-Merge avec préalignement selon ALANI *et al.*

	tourism			camera			Pizza		
	Or	Om	Oma	Or	Om	Oma	Or	Om	Oma
CMM	7,39	6,80	7,39	7,60	6,40	7,00	74,40	63,40	63,30

DEM	1,72	1,36	3,23	3,41	3,58	4,00	10,19	50,29	50,24
SSM	0,36	0,22	0,87	2,65	2,39	0,86	20,77	13,00	100,0
BEM	8,36	0,00	0,38	15,62	11,05	2,81	1840,16	70,82	40,33
Total	1,00	0,64	1,00	0,58	0,580	0,83	0,680	0,830	0,820

APPENDICE D

RÔLE DU NOMMAGE DANS RCA-MERGE

Tableau D.4 Les résultats d'un nommage partiel selon ALANI *et al.*

		CMM	DEM	SSM	BEM	Total
technical	Oma	8,4	5,43	2,58	1,78	0,72
	Oml	9,4	8,22	3,42	1,00	0,79

Tableau D.5 Les résultats d'un nommage complet selon ALANI *et al.*

		CMM	DEM	SSM	BEM	Total
tourism	Oma	7,39	3,23	0,87	0,38	1,0
	Oml	7,39	2,84	1,02	0,46	1,0

APPENDICE E

ÉTUDE COMPARATIVE DE RCA-MERGE AVEC D'AUTRES OUTILS

Tableau E.1 Comparaison entre les trois outils en se basant sur les métriques de INUKHUK

La métrique d'ORME *et al.*									
	Tourism			Camera			Pizza		
	Orc	On	Op	Orc	On	Op	Orc	On	Op
NOP	14,0	18,0	17,0	24,0	15,0	15,0	12,0	8,00	8,00
APC	1,07	0,94	0,89	1,00	0,88	1,07	0,11	0,05	0,08
AFC	0,61	0,26	0,26	4,60	1,00	0,57	6,43	2,49	3,14
AFR	1,60	0,62	0,62	18,5	1,54	1,00	38,27	6,17	20,73
AFNL	4,00	1,66	1,66	8,50	4,25	2,66	28,70	5,77	15,55
MaxDit	1,00	1,00	1,00	4,00	3,00	2,00	6,00	6,00	6,00
Total	5,52	5,38	5,34	5,80	5,70	5,59	6,00	4,20	5,99
La métrique de TARTIR *et al.*									
RRS	0,63	0,78	0,77	0,40	0,60	0,71	0,05	0,06	0,08
ARS	0,23	0,42	0,36	0,33	0,47	0,57	0,00	0,00	0,00
IRS	0,61	0,26	0,26	1,50	0,58	0,42	2,10	0,87	0,83
CR	0,00	0,00	0,00	0,00	0,05	0,07	0,00	0,007	0,01
AP	0,00	0,00	0,00	0,00	0,11	0,14	0,00	0,03	0,05
Total	2,43	2,53	2,44	2,03	3,91	4,52	1,58	2,76	3,50

La métrique d'ALANI *et al.*									
CMM	7,39	11,2	11,2	7,00	10,0	8,80	63,3	74,4	73,8
DEM	3,23	1,70	1,70	4,00	2,52	2,86	50,24	10,17	10,25
SSM	0,87	0,67	0,67	0,86	2,87	1,95	10,0	20,37	20,86
BEM	0,38	4,75	4,75	2,81	31,76	13,53	40,33	171,58	185,87
Total	1,00	0,98	0,98	0,83	0,92	0,90	0,82	0,99	0,99