

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE**

UNIVERSITE MENTOURI DE CONSTANTINE

DEPARTEMENT D'INFORMATIQUE

Ecole Doctorale en Informatique de l'Est

Pôle de CONSTANTINE

Option : Intelligence Artificielle

Mémoire

Présenté en vue de l'obtention du diplôme de
magister en informatique

Thème :

***Une approche d'annotation sémantique à partir d'une
ontologie multi-points de vue***

Présenté par : M^{elle} DJAMA OUAHIBA

Dirigé par : Pr. ZIZETTE BOUFAÏDA

Soutenu-le : 20/06/2010 devant le jury composé de:

Président :

Pr. NACER EDDINE ZAROUB

Professeur, Université Mentouri de Constantine

Rapporteur :

Pr. ZIZETTE BOUFAÏDA

Professeur, Université Mentouri de Constantine

Examineurs :

Dr. NADIA ZEGHIB

Maître de conférence, Université Mentouri de Constantine

Dr. FAÏZA BELALA

Maître de conférence, Université Mentouri de Constantine

Remerciements

Je tiens à remercier en premier lieu dieu le tout puissant qui m'a donné le courage et la patience et qui a éclairé mon chemin pour achever ce travail dans les meilleures conditions.

Plus particulièrement je remercie madame Zizette Boufaïda Professeur à l'université de Constantine pour la confiance qu'elle m'a accordé, pour son encadrement continu avec patience et gentillesse, m'a fait profiter de sa grande expérience, ses remarques constructives, ses orientations, et ses conseils qui ont grandement contribué à améliorer la qualité de ce mémoire.

Je tiens à adresser mes vifs remerciements aux membres du jury pour s'intéresser au sujet et avoir accepté de lire ce mémoire.

Je tiens à remercier très sincèrement l'ensemble des membres de l'équipe SI & BC et en particulier Mounir Hemam pour son aide, ses remarques constructives, ses orientations, et ses conseils.

Je tiens à remercier profondément mes parents et mes frères qui m'ont toujours encouragé et poussé à réussir, pour tout ce qu'ils m'ont donné, leur aide éternel, et leur amour.

A tous ceux qui de près ou loin ont contribué à la réalisation de ce travail.

Résumé :

Dans ce travail, nous nous intéressons au problème de l'annotation sémantique des ressources dans une organisation hétérogène (constituée de plusieurs groupes de personnes diversifiés) en prenant en considération un aspect multi-points de vue. L'objectif est de proposer une approche d'annotation sémantique à partir d'une ontologie multi-points de vue, où les utilisateurs peuvent annoter une ressource selon leurs niveaux de connaissance et leurs points de vue personnels. Pour cela, nous nous inspirons de la méthode proposée dans [Sylvain 07] et nous l'adaptons de telle façon qu'elle puisse supporter la notion de point de vue. Cette méthode est motivée par un parallèle entre la structure d'une page web et la sémantique qui lui est associée. Ces deux dimensions de structure et de sémantique sont formalisées en logique de descriptions. En effet, les éléments de pages web formalisés comme des individus de structure sont annotés par des instances de concepts du domaine et les relations entre ces éléments sont formalisées par des instances de rôles. Dans le cadre de notre travail, l'annotation sémantique multi-points de vue est traitée comme un problème d'instanciation multiple. De ce fait, nous générons pour chaque point de vue, une structure de page propre à lui, puis nous associons une sémantique à chaque structure à l'aide d'une ontologie multi-points de vue.

Mots clés:

Annotation sémantique, ontologie, point de vue, logique de descriptions.

Abstract:

In this work, we address the problem of semantic annotation of resources in a heterogeneous organization (consisting of several diverse groups of people) taking into account a multi-views aspect. The objective is to propose an approach to semantic annotation from a multi-views ontology, where users can annotate a resource according to their levels of knowledge and their personal views. For this, we draw on the method proposed in [Sylvain 07] and we adapt it so that it can support the concept of point of view. This method is motivated by a parallel between the structure of a web page and the semantics associated with it. These two dimensions of structure and semantics are

formalized in logic descriptions. Indeed, elements of web pages as individuals formalized structure are annotated with instances of domain concepts and relationships between these elements are formalized by instances of roles. As part of our work, the semantic annotation multi-views and treated as a problem of multiple instantiation. Thus, we generate for each point of view, a page structure peculiar to him, then we associate a semantic to each structure using a multi- views ontology.

Keywords:

Semantic annotation, ontology, view, logic descriptions.

ملخص

في هذا العمل ، سنعالج مشكلة الشرح الدلالي لصفحات الانترنت في منظمة غير المتجانسة (تتألف من عدة مجموعات متنوعة من الأشخاص) مع الأخذ في الاعتبار وجهات النظر المتعددة الجانب. والهدف من ذلك هو أن نقتراح نهجا لشرح الدلالات بواسطة الانطولوجيا (علم الوجود) المتعددة وجهات النظر ، حيث يمكن للمستخدمين تعليم الصفحات وفقا لمستوياتهم المعرفية و وجهات نظرهم الشخصية. لهذا نستوحي فكرتنا من الطريقة المقترحة في [Sylvain 07] نكيفها بحيث يمكنها دعم مفهوم وجهة نظر. والدافع وراء اختيار هذه الطريقة هو من جانب التشابه بين بنية (هيكل) صفحة الويب والدلالات المرتبطة بها. هذين البعدين الهيكل ومعاني الكلمات موصوفين بواسطة المنطق. وشرح مدلولات عناصر صفحات الويب موصوفة كأفراد الهيكل في المنطق بواسطة مثيلات مفاهيم الحقل، و شرح مدلولات العلاقات بين هذه العناصر بواسطة مثيلات الأدوار. في عملنا، الشرح الدلالي متعدد الآراء يعالج على أنه مشكلة مثل متعددة. وهكذا ، فإننا نولد لكل وجهة نظر هيكل الصفحة الخاصة به ، ثم نقوم بإقران كل بنية بالدلالية باستخدام انطولوجيا متعددة وجهات النظر.

كلمات البحث :

الشرح الدلالي ، الانطولوجيا (علم الوجود) علم الوجود ، عرض أوصاف المنطق

Table des matières

Introduction générale	
1. Contexte de la recherche	1
2. Notre contribution	1
2. Plan du mémoire	2
<hr/>	
Chapitre I : Représentation des connaissances ontologiques	
<hr/>	
1. Introduction.....	5
2. Notion d'ontologie	6
3. Les composants de l'ontologie	7
3.1. Concepts	7
3.2. Relations	8
3.3. Axiomes	8
3.4. instances	8
4. Formalismes de représentation des ontologies	8
4.1. Graphes conceptuels	8
4.2. Frames	10
4.3. Logiques de description.....	11
5. Processus de construction d'ontologies	13
5.1. Conceptualisation	13
5.2. Ontologisation	14
5.3. Opérationnalisation	14
6. Cycle de vie d'une ontologie	14
7. Editeur d'ontologie.....	14
8. Langage de représentation d'ontologie	15
8.1. OWL	15
9. Système de raisonnement	17
10. Annotation sémantique	17
10.1. Processus d'annotation sémantique	18
10.2. Outils d'annotation sémantique	19
10.3. Evaluation des outils d'annotation sémantique	21

10.4. Approche d'annotation sémantique utilisant les instances concepts définis ...	21
10.4.1. Approche des classes d'annotation pour l'annotation sémantique	22
10.4.2. Un système automatique d'annotation sémantique de page web	22
10.5. Langages d'annotation sémantique	23
10.5.1. Les précurseurs	23
10.5.2. XML	24
10.5.3. RDF	28
11. Conclusion	29

Chapitre II : Représentation explicite de point de vue

1. Introduction	30
2. Notion de point de vue	31
3. Les propriétés de point de vue	32
4. Contexte et point de vue	33
5. point de vue en représentation des connaissances.....	34
5.1. Représentation des connaissances par objets	34
5.2. Représentation des connaissances par graphes	36
5.3. Représentation des connaissances par ontologies	37
5.3.1. Les composants d'ontologie multi-points de vue	38
5.3.2. Intérêt des ontologies multi-points de vue	40
5.3.3. Elaboration d'ontologie multi-points de vue	40
5.3.4. La spécification et la conceptualisation d'une ontologie multi-points de vue	41
5.3.5. Langages de représentation d'une ontologie multi-points de vue	43
5.3.6. Ontologie multi-points de vue et annotation sémantique	50
6. Conclusion	51

Chapitre III : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

1. Introduction	52
2. Outils et techniques utilisés dans l'approche proposée.....	53
2.1. XML	53
2.2. DOMXML	54

2.3. Ontologie de domaine	56
2.3.1. Développement de l'ontologie multi-points de vue	57
3. L'approche proposée	59
3.1. Notion de la régularité de structure	60
3.2. Schéma illustre les différentes étapes et activités de l'approche proposée	60
3.3. Description de l'approche proposée.....	62
3.3.1. Construire les modèles de structure	62
3.3.2. Formaliser les modèles de structures	65
3.3.3. Associer une sémantique à chaque structure.....	66
3.3.4. Annotation d'une nouvelle page appartenant en même domaine.....	69
4. Conclusion	70

Chapitre IV :Etude de cas : Application sur une page web du domaine d'immobilier

1. Introduction	71
2. Construction de l'ontologie de domaine	71
2.1. La spécification et la conceptualisation	71
2.2. La formalisation	76
2.3. Codification	77
3. Application de l'approche sur le domaine d'immobilier	79
3.1. Construire les modèles de structure	79
3.2. Formaliser les modèles de structures	81
3.3. Associer une sémantique à chaque structure	82
3.4. Annotation d'une nouvelle page appartenant en même domaine	87
4. Implémentation	92
4.1. Implémentation de l'ontologie en PHP5	92
4.2. Implémentation des concepts de structure.....	92
4.3. Implémentation des classes d'annotation.....	93
4.4. Annotation d'une nouvelle page appartenant en même domaine.....	93
4.5. Test.....	95
5. Conclusion	95

Conclusion et perspectives	96
Conclusion générale	96
Limite et difficultés	96
Perspective	97
Bibliographie	99

Introduction générale

Introduction générale

1. Contexte de la recherche

Afin d'avoir sur le Web, des ressources définies et liées, munies d'un sens explicitement interprétable par les processus logiciels, il est nécessaire d'associer à ces ressources des méta-données, décrivant leur contenu et leur fonctionnalité. L'annotation sémantique, est l'une des techniques majeures utilisée pour la création de méta-données sur le Web. Les ontologies, qui ont prouvé leur utilité dans de nombreux domaines d'application, peuvent aider à répondre à ce requis, en fournissant une représentation formelle des concepts relatifs à un domaine et des relations entre ces concepts et constituent une base commune pour la communication entre les différents acteurs de ce domaine. La définition d'une annotation sémantique sera simplement basée sur les éléments de l'ontologie à exploiter.

Dans ce travail, nous nous intéressons au problème de l'annotation sémantique des ressources dans une organisation hétérogène (i.e. constituée de plusieurs groupes de personnes diversifiés) en prenant en considération un aspect multi-points de vue. Un point de vue correspond, dans ce cadre, à la représentation des connaissances utiles à un groupe de personnes particulier, qui coexiste et collabore avec d'autres groupes de personnes. Chaque groupe de personnes a ses intérêts propres ; chacun regarde des propriétés et relations particulières des objets conceptuels du même univers de connaissances à considérer.

2. Notre contribution

L'objectif principal de ce travail est de proposer une approche d'annotation sémantique basée sur l'exploitation et l'instanciation d'une *ontologie multi-points de vue*. L'ontologie multi-points de vue, utilisée dans cette approche, permet de faire converger vers une même ontologie les différentes conceptualisations possibles du domaine modélisé selon des points de vue différents[Hemam 09b]. Par ailleurs, les annotateurs ou les utilisateurs finaux de cette ontologie peuvent faire des énoncés (i.e. des annotations) comme ils le souhaitent et selon leurs niveaux de connaissances et leurs

points de vue personnels. A chaque point de vue correspond une annotation sémantique locale ayant son propre langage et surtout sa propre interprétation vis-à-vis du contenu et de la fonctionnalité de la ressource considérée.

Pour atteindre notre but, nous adaptons l'approche proposée dans [Sylvain 07]. Cette dernière, prend en compte l'interaction entre la structure et la sémantique d'une page à annoter en les formalisant dans un langage commun fondé sur les logiques de descriptions. Ainsi, l'annotation sémantique est traitée comme un problème d'instanciation. En effet, les éléments de pages web, formalisés comme des individus de structure, sont annotés par des instances de concepts du domaine. Tandis que, les relations entre ces éléments sont formalisées par des instances de rôles. Dans le cadre de notre travail, l'annotation sémantique multi-points de vue est plutôt traitée comme un problème d'instanciation multiple. De ce fait, il est donc possible de générer, pour chaque point de vue à considérer, une structure de page propre à lui, puis d'associer une sémantique à chaque structure à l'aide d'une ontologie multi-points de vue. Pour illustrer notre approche, nous l'appliquons sur l'exemple du domaine de l'immobilier.

3. Plan du mémoire

La présentation de notre travail est divisée en deux parties :

Première partie (état de l'art) : chapitres 1 à 2

Il s'agit de présenter le cadre scientifique de l'étude menée. Toutes les notions, dont nous aurons besoin pour décrire notre approche, seront données dans cette partie.

Chapitre 1 : Représentation des connaissances ontologiques

Ce chapitre comportera des généralités sur l'ontologie tel que les définitions de la notion d'ontologie, les différents composants d'une ontologie ainsi que les formalismes de représentation des ontologies, la construction d'une ontologie, les éditeurs de l'ontologie, les langages de représentation d'une ontologie, aussi les systèmes de raisonnement sur l'ontologie. Enfin, le chapitre se terminera par la présentation de la

notion d'annotation sémantique comme une technique qui exploite l'ontologie, en insistant sur son apport et ses spécificités par rapport à l'annotation classique, puis nous avons présenté le processus d'annotation sémantique. En suite, nous allons exposer une synthèse des outils et approches d'annotation sémantique les plus importants. . Pour terminer, nous discutons sur les langages d'annotation sémantique

Chapitre 2 : Représentation explicite de points de vue

Dans ce chapitre, la notion de point de vue sera clarifiée et la représentation multi-points de vue des connaissances sera étudiée ainsi que les différents formalismes, issus de différents paradigmes de représentation des connaissances, seront présentés. Pour terminer, nous discutons sur le besoin de prise en compte de la notion de point de vue pour l'annotation sémantique.

Deuxième partie : chapitres 3 à 4

Cette partie sera consacrée à la description de notre approche.

Chapitre 3 : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

Dans ce chapitre, nous allons présenter, dans un premier temps, les principaux outils qui seront utilisés dans notre approche. Ainsi que la description en logique de descriptions d'une ontologie multi-points de vue. En suite, sur la base du modèle de représentation développé, nous proposons une approche d'annotation sémantique multi-points de vue, dans la quelle le formalisme de la logique de descriptions est utilisé comme mode de représentation unifié de la structure des documents, de l'ontologie et aussi de l'annotation sémantique du document.

Chapitre 4 : Etude de cas une application sur une page web du domaine d'immobilier

Comme il fallait choisir un domaine de connaissance dans l'objectif de concrétisation, nous avons opté pour l'annotation multi-points de vue d'une page web appartenant au

domaine d'immobilier. Nous illustrons dans un premier temps, nous montrons comment l'ontologie multi-points de vue « immobilier » a été conceptualisée avec deux points de vue : taille et financier puis formalisée et finalement traduite dans le langage OWL. Puis nous proposons un jeu d'implémentation de l'approche grâce au scripte PHP pour la programmation de l'arbre DOMXML, les classes d'annotation, et l'établissement des annotations sur la page. Ainsi que les possibilités d'implémentation de l'ontologie multi-points de vue avec l'outil protégé 2000.

Conclusion et perspectives

Le mémoire s'achève par une conclusion récapitulant la contribution de cette thèse. La conclusion est surtout l'occasion de présenter les limites, les difficultés, et les perspectives que nous jugeons importantes pour améliorer la solution proposée.

Chapitre I :

Représentation des connaissances ontologiques



Chapitre 1 : Représentation des connaissances ontologiques

1. Introduction

Avec l'apparition de l'intelligence artificielle la notion « connaissance » a été élargie dans l'informatique. Le passage de « information » à « connaissance » est lié à l'expérience de l'action. Donc la connaissance par définition c'est l'information et le mode d'emploi de cette information. [Charlet 03]. Pour que cette connaissance peut être exploitée par l'ordinateur il faut la représenter.

La représentation des connaissances est une discipline au cœur de la problématique des sciences cognitives et plus particulièrement de l'intelligence artificielle, dont le but est de proposer des outils méthodologiques et techniques permettant de représenter et manipuler, plus ou moins formellement, un ensemble de connaissances. Donc la représentation des connaissances désigne un ensemble d'outils et de technologies destinés d'une part à représenter et d'autre part à organiser le savoir humain pour l'utiliser et le partager par l'ordinateur.

Les applications modernes de l'informatique ont conduit à un usage généralisé des représentations de connaissances dans des contextes variés (recherche d'information, simulation, description des ontologies du web sémantique, ...). Dans le domaine des nouvelles technologies, la représentation formelle des connaissances s'est développée dans le domaine de l'intelligence artificielle. Dans une représentation formelle, les connaissances sont représentées par des objets logiques reliés par des propriétés, des axiomes et des règles. Ce type de représentation est utilisé dans les systèmes experts. Le développement du Web, et en particulier la perspective du Web sémantique a renouvelé le domaine en introduisant le terme controversé d'ontologie.

Dans ce chapitre, nous précisons les différentes définitions de la notion d'ontologie ainsi que les différents éléments dont elle est constituée. En suite, nous présentons les différents principes méthodologiques de construction d'ontologies. Finalement, nous discutons sur un champ d'exploitation d'une ontologie qui est l'annotation sémantique



en montrant le processus d'annotation sémantique, quelques outils et approches d'annotation sémantique.

2. Notion de l'ontologie

Il est difficile de définir ce qu'est une ontologie d'une façon définitive. Le mot est en effet employé dans des contextes très différents touchant la philosophie, la linguistique ou l'intelligence artificielle.

Le terme d'ontologie en informatique est apparu grâce notamment au projet ARPA Knowledge Sharing Effort [Gruber 91]. Les ontologies informatiques sont des outils qui permettent précisément de représenter un corpus de connaissances sous une forme utilisable par un ordinateur. C à d le terme ontologie est plus facilement associé à la définition d'un vocabulaire compréhensible par des machines. Ces vocabulaires/ontologies sont spécifiés avec assez de précision pour autoriser des relations précises entre des termes différents, c-à-d à la différence d'un vocabulaire, une ontologie cherche à représenter le sens des concepts et des relations qui les lient.

De ce fait, Gruber dans [Gruber 93] a donné une première définition pour les ontologies, dans le domaine de l'intelligence artificielle (IA) comme : « une ontologie est une spécification explicite et formelle d'une conceptualisation d'un domaine de connaissance ». Cette définition deviendra célèbre et restera la définition la plus citée dans les écrits en intelligence artificielle (Cf. Figure.1). La **conceptualisation** renvoie à un modèle abstrait d'un quelconque phénomène après en avoir relevé les concepts significatifs et les relations existantes entre ces concepts. Par **explicite**, il faut entendre que le type de concepts utilisés, ainsi que leurs contraintes d'utilisation, sont définies de façon explicite; quant à l'adjectif **formel**, il exprime le fait que l'ontologie doit être lisible par ordinateur [Gruber 95].

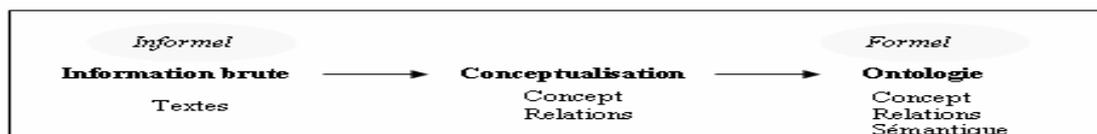


Figure.1 : L'ontologie comme spécification formelle d'une conceptualisation

Guarino et Giaretta [Guarino 95a] ont défini l'ontologie comme "une théorie logique qui rend compte partiellement mais explicitement d'une conceptualisation"(sens 1) ou «



synonyme de la conceptualisation » (sens 2). La conceptualisation étant définie comme "une structure sémantique intensionnelle qui capture les règles implicites contraignant la structure d'un morceau de réalité". L'ontologie est une représentation explicite partielle parce qu'elle se concentre sur les aspects de la conceptualisation nécessaires au bon fonctionnement du système. Donc les auteurs (dans [Guarino 95a] essaient de donner une clarification terminologique du terme « ontologie » utilisé dans le domaine de l'intelligence artificielle.

Guarino dans [Guarino 97] [Guarino 95b] cherche à clarifier ce qu'est une ontologie en énonçant que : «Les ontologies sont des spécifications partielles et formelles d'une conceptualisation commune».

Les auteurs de [Paolo 03] définissent les ontologies comme « des modèles partagés d'un domaine encodant une vue qui est commune à un ensemble de différentes parties »

3. Les composantes d'ontologie

Une ontologie ne peut être construite que dans le cadre d'un domaine précis de la connaissance, ne serait-ce que parce que beaucoup de termes n'ont pas le même sens d'un domaine à l'autre [Bachimont 00]. Les connaissances traduites par une ontologie sont à véhiculer à l'aide des éléments suivants:

3.1. Concepts

Un concept peut représenter un objet, une notion, une idée [Uschold 96]. Le **terme** est un élément lexical qui permet d'exprimer le concept en langue naturelle, il peut admettre des synonymes. La **notion** également appelée *intension* du concept, contient la sémantique du concept, exprimée en termes de propriétés et attributs, et de contraintes. L'ensemble **d'objets** appelé *extension* du concept, regroupe les objets manipulés à travers le concept ; ces objets sont appelés instances du concept. Il peut que des concepts partageant la même extension mais pas leurs intensions peuvent être désignés par le même terme. Ceci correspond à *des points de vue* différents sur un même concept [Falquet 01]. Par exemple, pour le domaine des voitures, d'un point de vue *mécanique* un véhicule peut être défini selon le type de moteur, le diamètre des roues, etc. tandis que dans d'un point de vue *vente*, les définitions vont plutôt contenir des informations sur le type de véhicule (tourisme, utilitaire, transport de groupes etc.).



3.2. Relations

Si certains liens conceptuels existant entre les concepts peuvent s'exprimer à l'aide de propriétés portées par les concepts, d'autres doivent être représentés à l'aide de relations autonomes [Furst 02]. Une relation permet de lier des instances des concepts. Elles sont caractérisées par un terme (voire plusieurs) et une signature qui précise le nombre d'instances de concepts que la relation lie, leurs types et l'ordre des concepts, c'est-à-dire la façon dont la relation doit être lue. Par exemple, la relation « écrit » lie une instance du concept « personne » et une instance du concept « texte », dans cet ordre.

3.3. Axiomes

Assertions acceptées comme vraies à propos des abstractions du domaine traduite par l'ontologie. Interviennent dans la définition des concepts ou des relations, dans l'inférence de nouvelles informations,

3.4. Instances (ou individus)

Constituent la définition extensionnelle de l'ontologie (pour représenter les éléments spécifiques)

4. Formalismes de représentation des ontologies

Trois grands paradigmes sont utilisés pour représenter des Connaissances en IA et de là les ontologies: Les Graphes Conceptuels, Les frames, et les logiques de description.

4.1. Graphes Conceptuels

Un graphe conceptuel est un outil symbolique utilisé dans la théorie des graphes conceptuels. Développée par John Sowa en 1984, cette théorie est un formalisme général de représentation de connaissances fondé sur la logique et basé sur la linguistique, la psychologie et la philosophie. Les graphes conceptuels peuvent être traduits dans une autre forme de logique comme la logique des prédicats. Ils ont été conçus dans l'objectif de développer un système de logique qui représente de façon plus simple et plus commode les structures du langage naturel. D'une manière formelle, un graphe conceptuel est un diagramme qui représente non la syntaxe mais la sémantique d'une phrase.



Représentation des connaissances

Les graphes conceptuels sont un Modèle de représentation de connaissances du type réseaux sémantiques, permet de représenter des connaissances sous forme d'un graphique (concepts et relations conceptuelles). Le but du formalisme des graphes conceptuels est de représenter la sémantique dans une forme qui soit précise, lisible et utilisable par un ordinateur. Le graphe conceptuel est basé sur la logique des prédicats du 1er ordre à deux niveaux :

- Niveau terminologique : où sont décrits les concepts, les relations, les instances,
- Niveau assertionnel : où sont décrits les faits, les règles et les contraintes sous forme de graphes où les sommets sont des instances de concepts et les arcs, des relations.

Dans le modèle des Graphes Conceptuels, un *support* représente la connaissance ontologique et les graphes conceptuels construits sur ce support représentent la connaissance assertionnelle relative à la connaissance ontologique. L'ontologie donc est une organisation hiérarchique de la connaissance sur un ensemble d'objets par leur regroupement en sous catégories suivant leurs caractéristiques essentielles [Gerbé 01].

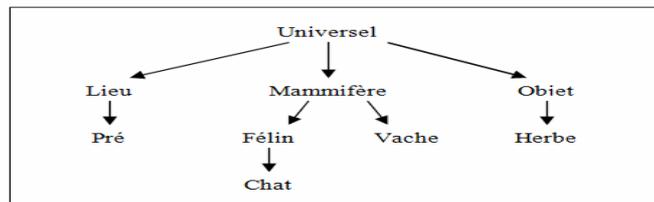


Figure2 : Exemple de hiérarchie de types de concepts

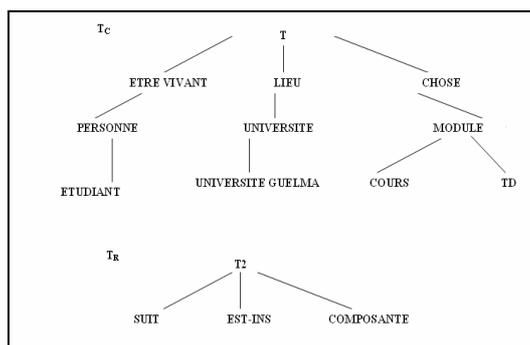


Figure3 : exemple d'un support conceptuel

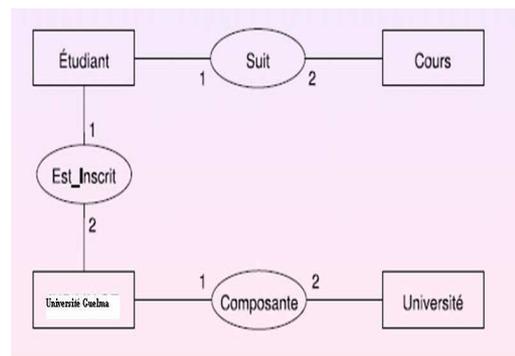


Figure4 : exemple d'un graphe conceptuel

En utilisant le vocabulaire défini dans le support défini dans la figure 3, le graphe conceptuel défini dans la figure4 peut être interprété par : « étudiant inscrit dans l'université Guelma suit un cours. ».



Mécanisme de raisonnement

Le mécanisme de raisonnement de base c'est la **projection**, il Permet de déterminer si un graphe est plus spécialisé ou plus général qu'un autre. La projection d'un graphe G1 dans un graphe G2, tous deux associés à un support est la donnée de deux fonctions :

Fr de l'ensemble des nœuds relations de G1 dans l'ensemble des nœuds relations de G2

Fe de l'ensemble des nœuds concepts de G1 dans l'ensemble des nœuds concepts de G2.

4.2. Frames

Les Frames ont été proposés par Minsky en 1975 comme une modélisation de base pour la représentation de connaissances dans le domaine de l'Intelligence Artificielle (IA). L'idée générale de cette nouvelle approche est résumée par Minsky : « Lorsqu'on rencontre une situation nouvelle, on sélectionne dans la mémoire une structure réelle qui est un frame (cadre). ».

Représentation des connaissances

Les frames Variante de Réseaux pour représenter les connaissances non procédurales dans un Système Expert. Elles sont utilisées pour représenter les connaissances sur des objets ou des événements types (oiseau type, repas type...). Un frame est une entité regroupant de manière structurée l'ensemble des connaissances relatives à un objet, un concept. Un frame est Composé d'un ensemble d'attributs (slots) qui peuvent porter différentes valeurs (facets) correspondant aux diverses notions relatives à la chose représentée (propriétés du concept), un attribut est à son tour décrit par des facettes, elles peuvent être descriptives ou comportementales. Un frame est doté d'instances. Donc il ressemble à une structure de données « enregistrement ». **Exemple illustre comment définir une chaise avec un frame** : Une chaise est une sorte de meuble, elle a quatre pieds, un dossier, et une couleur. Une chaise est confortable;

```
Frame Chaise
{ Style -du -dossier   $type enum
      $enum = { droit, rembourré }
  Nombre -de-pieds     $un entier
      $défaut    4
  Nombre -de-bras $un entier
      $intervalle 0..2 }
```



Les Frames sont organisés en classes et en sous-classes. Il y a donc ici une hiérarchie entre les classes qui contiennent les sous-classes. Les sous-classes héritent des propriétés (attributs) des classes, sauf s'ils ont des valeurs de propriétés qui sont en conflit avec celles des propriétés héritées. Les slots peuvent avoir des valeurs particulières. L'héritage peut être simple ou multiple.

Mécanisme de raisonnement

Les représentations par frames offrent deux types de mécanismes de raisonnement : les mécanismes globaux qui travaillent sur toute la base et les mécanismes locaux qui font des inférences au niveau d'un frame ou d'un attribut d'un frame. Les mécanismes globaux sont l'héritage, le filtrage et la classification et les mécanismes locaux sont les réflexes de contrôle et les réflexes de calcul [Drews 93].

4.3. Logiques de description

Les logiques de description aussi appelée logiques descriptives (LDs) forment une famille de langages de représentation de connaissance qui peuvent être utilisés pour représenter la connaissance terminologique d'un domaine d'application d'une manière formelle et structurée par des entités qui ont une description syntaxique à laquelle est associée une « sémantique ».

Les logiques de description ont été introduites par Brachman en 1979 et ont ensuite connu de nombreux développements [Napoli 97]. Elles sont issues de la logique des prédicats, des langages de frames et des réseaux sémantiques.

Le nom de logique de description se rapporte, d'une part à la description de concepts utilisée pour décrire un domaine et d'autre part à la sémantique basée sur la logique qui peut être donnée par une transcription en logique des prédicats du premier ordre. Dans ce sens, nous pouvons dire que les LDs ont une sémantique « descriptive » formelle.

Représentation des connaissances

Les logiques de descriptions permettent de représenter les connaissances d'un domaine à l'aide de concepts (classes d'individus), de rôles, et d'individus. Les concepts sont interprétés comme un ensemble dans un univers donné c à d entité générique d'un domaine exemple : Cours, Etudiant, Enseignant, Université,... Les rôles correspondent



aux "liens entre les éléments", ils sont interprétés comme des relations binaires sur un univers donné. Les individus correspondent aux éléments d'un univers donné. Les concepts et les rôles peuvent être primitifs ou définis. Les concepts (éventuellement les rôles) primitifs sont comparables à des atomes et servent de base à la construction des concepts définis (éventuellement les rôles définis). Un concept défini et un rôle défini possèdent une description structurée élaborée à partir de constructeurs.

Un LD est défini à partir d'un ensemble P_c de concepts primitifs (atomiques), un ensemble P_r de rôles primitifs (atomiques), des constantes T et \perp , et des règles de syntaxe suivantes :

$C, D \rightarrow$	T plus général	
	\perp	absurde (le plus spécifique)
	P	concept primitif
	$C \cap D$	conjonction de concepts
	$C \cup D$	disjonction de concepts
	$\neg C$	négation
	$\forall r.C$	restriction universelle (définit le co-domaine du rôle r)
	$\exists r.C$	restriction existentielle (il existe au moins un objet relié par le rôle r au concept C)
	$\leq n r.C$	cardinalité maximum
	$\geq n r.C$	cardinalité minimum
$r \rightarrow$	q	rôle primitif
	$r_1 \cap r_2$	conjonction de rôles
	$r_1 \cup r_2$	disjonction de rôles

Sémantique des concepts et des rôles

Une sémantique est associée à chaque description de concept et de rôle par l'intermédiaire d'une interprétation. Une interprétation \mathcal{I} se compose d'un domaine d'interprétation $\Delta^{\mathcal{I}}$ et d'une fonction d'interprétation $\cdot^{\mathcal{I}}$. Le domaine d'interprétation consiste en un ensemble d'individus. La fonction d'interprétation assigne à chaque concept atomique A un ensemble tel que $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, et à chaque rôle atomique R une relation binaire $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. La sémantique des logiques de description est définie comme suit :



$\Delta^{\mathcal{I}}$ est un ensemble non-vide.

- $\cdot^{\mathcal{I}}$ est une fonction assignant :
 - un élément $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ à chaque individu;
 - un sous-ensemble $C_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ à chaque concept atomique;
 - et une relation $R_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ à chaque rôle atomique.

$$\top^{\mathcal{I}} = \Delta_{\mathcal{I}}$$

$$\perp^{\mathcal{I}} = \emptyset$$

$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$

$$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$

$$(\neg C)^{\mathcal{I}} = \Delta_{\mathcal{I}} - C^{\mathcal{I}}$$

$$(\forall r.C)^{\mathcal{I}} = \{x \in \Delta_{\mathcal{I}} / \forall y : (x, y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$$

$$(\exists r.C)^{\mathcal{I}} = \{x \in \Delta_{\mathcal{I}} / \exists y : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$$

$$(\geq n r)^{\mathcal{I}} = \{x \in \Delta_{\mathcal{I}} / |\{y \in \Delta_{\mathcal{I}} / (x, y) \in r^{\mathcal{I}}\}| \geq n\}$$

$$(\leq n r)^{\mathcal{I}} = \{x \in \Delta_{\mathcal{I}} / |\{y \in \Delta_{\mathcal{I}} / (x, y) \in r^{\mathcal{I}}\}| \leq n\}$$

$$(r_1 \sqcap \dots \sqcap r_n)^{\mathcal{I}} = r_1^{\mathcal{I}} \cap \dots \cap r_n^{\mathcal{I}}$$

Mécanisme de raisonnement

Les concepts et les rôles sont organisés en hiérarchies sur lesquelles opèrent les processus de classification et d’instanciation, qui sont à la base du raisonnement terminologique. Le mécanisme de raisonnement de base des logiques de descriptions : classification de concepts, réalisée par un algorithme de classification, appelé le **classifieur**. Le **classifieur** prend une nouvelle description de concept et la place à l’endroit correct dans la hiérarchie. Pour trouver la place appropriée pour le nouveau concept, l’algorithme de classification détermine les **relations de subsomption** entre ce concept et les autres concepts de la hiérarchie.

La relation de *subsomption* organise concepts et rôles par niveau de généralité. C subsume D si C est plus général que D au sens que l’ensemble d’individus représenté par C contient l’ensemble d’individus représenté par D \Leftrightarrow hiérarchie de concepts et (parfois) hiérarchie de rôles.

5 Processus de construction d’une ontologie

Le processus de construction d’ontologies repose sur un enchaînement de trois étapes (conceptualisation, ontologisation, opérationnalisation) permettant de passer des données brutes à l’ontologie opérationnelle. Les données brutes, constituant un corpus (exprimé *a priori* en langage naturel), intègrent toutes les connaissances du domaine que l’on souhaite formaliser.

5.1. Conceptualisation

Cette étape consiste à identifier des connaissances contenues dans un corpus représentatif du domaine. Ce travail doit être mené par un expert du domaine, assisté par un ingénieur de la connaissance. Cette étape permet d’aboutir à un modèle informel, sémantiquement ambiguë et généralement exprimé en langage naturel.



5.2. Ontologisation

C'est une formalisation, autant que possible, sans perte d'information, du modèle conceptuel obtenu à l'étape précédente. Ce travail doit être mené par l'ingénieur de la connaissance, assisté de l'expert du domaine. Cette étape facilite sa représentation ultérieure dans un **langage** complètement **formel** et **opérationnel** [Fürst 03]. L'ontologisation peut être complétée par une étape d'intégration au cours de laquelle une ou plusieurs ontologies vont être importées dans l'ontologie à construire [Guarino 97].

5.3. Opérationnalisation

C'est une transcription de l'ontologie dans un langage formel (i.e. possédant une syntaxe et une sémantique) et opérationnel (i.e. doté de services inférentiels permettant de mettre en œuvre des raisonnements) de représentation de connaissances. Ce travail doit être mené par l'ingénieur de la connaissance.

6. Cycle de vie d'une ontologie

Les ontologies utilisées comme des composants logiciels dans des systèmes répondant à des objectifs opérationnels différents, leur développement doit s'appuyer sur les mêmes principes que ceux appliqués en génie logiciel [Fürst 02]. Les activités liées aux ontologies sont d'une part des activités de gestion de projet (planification, contrôle, assurance qualité), et d'autre part des activités de développement (spécification, conceptualisation, formalisation). S'y ajoutent des activités transversales de support telles que l'évaluation, la documentation, la gestion de la configuration [Blazquez 98]. Un cycle de vie inspiré du génie logiciel est présenté dans [Fürst 02]. chaque étape de ce cycle de vie pose des problèmes de recherche.

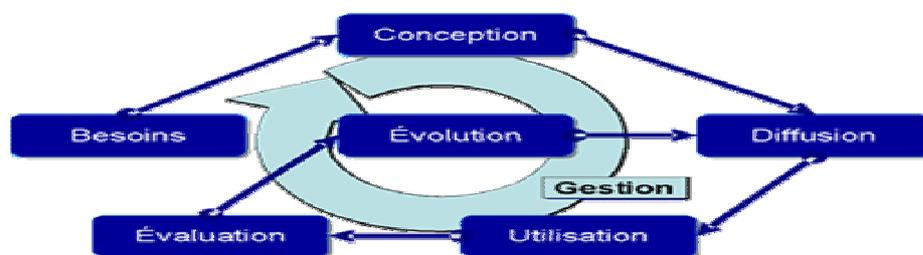


Figure 5 : Cycle de vie d'une ontologie

7. Éditeurs d'ontologies

Les éditeurs d'ontologie sont des environnements permettant de développer l'ontologie. Parmi les éditeurs qu'ils existent nous citons :



PROTEGE-2000¹ [Noy 01] : est un environnement graphique de développement d'ontologies développé par le SMI de Stanford. Le modèle de connaissances de PROTEGE est une hiérarchie de classes qui ont des attributs, qui peuvent avoir certaines propriétés. Une interface graphique dispense l'utilisateur de l'éditeur de toute manipulation du langage formel. L'architecture logicielle permet l'insertion de 'plugins' pouvant apporter de nouvelles fonctionnalités (par exemple, la possibilité d'importer et d'exporter les ontologies construites dans divers langages opérationnels de représentation ou encore la spécification d'axiomes).

OILed² [Bechhofer 01] : développé sous la responsabilité de l'université de Manchester, a été conçu pour éditer des ontologies dans le langage OIL. Il permet la création des hiérarchies de classes et spécialiser les rôles et utiliser avec l'interface les types d'axiomes les plus courants. Cet éditeur offre les services d'un raisonneur, FaCT, qui permet de tester la correction des définitions de classes et découvrir les subsomptions restées implicites dans l'ontologie.

8. Langages de représentation d'ontologies

Un langage d'ontologie est un langage formel permettant de représenter une ontologie, décrire les annotations, exploiter et raisonner sur les contenus des ressources. Parmi les langages existant dans la littérature nous citons OWL (Ontology Web Language):

8.1. Ontology Web Language (OWL)

Le langage OWL fournit des mécanismes pour créer tous les composants d'une ontologie : classes, instances, propriétés et axiomes. OWL repose également sur la syntaxe des triplets RDF et réutilise certaines des constructions RDFS. Comme en RDFS, les classes peuvent avoir des sous-classes, fournissant ainsi un mécanisme pour le raisonnement et l'héritage des propriétés. Par contre, en OWL, on distingue :

1) les propriétés objet (object property), i.e. les relations, qui relient des instances de classes à d'autres instances de classes. C'est l'équivalent des triplets RDF dont l'objet est une ressource.

¹ <http://protege.stanford.edu/index.shtml>

² <http://oiled.man.ac.uk/>



2) les propriétés type de données (datatype property), i.e. les attributs, qui relient des instances de classes à des valeurs de types de données (nombres, chaînes de caractères,...). C'est l'équivalent des triplets RDF dont l'objet est une valeur littérale.

Les axiomes fournissent de l'information au sujet des classes et des propriétés, spécifiant par exemple l'équivalence entre deux classes. Donc OWL permet de définir des ontologies comme un jeu de définition de classes, de propriétés et de contraintes. Toute classe définie dans une ontologie OWL est une sous-classe de owl:Thing.

OWL a été fractionné en trois langages distincts chacune étant une extension de la précédente :

- **OWL Lite** : Convient aux utilisateurs qui ont principalement besoin d'une hiérarchie de classification et de contraintes simples. Ce sous langage reprend tous les constructeurs de RDF (c'est-à-dire fournit des mécanismes permettant de définir un individu comme instance d'une classe, et de mettre des individus en relation), il utilise les mots-clés de RDFS (rdfs:subClassOf, rdfs:Property, rdfs:subPropertyOf, rdfs:range, rdfs:domain), avec la même sémantique, et il supporte les contraintes de cardinalité, mais ne permet d'utiliser que les valeurs 0 ou 1.
- **OWL LD** : Convient aux utilisateurs qui veulent le maximum d'expressivité, ce sous langage reprend tous les constructeurs d'OWL LITE, il permet tout entier positif dans les contraintes de cardinalité, et le tire son nom de sa correspondance avec les logiques de descriptions.
- **OWL FULL** : ce sous langage reprend tous les constructeurs d'OWL DL, et tous les constructeurs de RDF Schema. Il permet d'utiliser une classe en position d'individu dans les constructeurs.

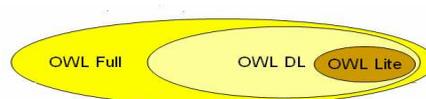


Figure6 : la relation entre les trois versions du OWL



9. Systèmes de raisonnement sur les ontologies

Les systèmes de raisonnement permettent la classification des nouveaux concepts intégrés dans l'ontologie. Ainsi de vérifier et corriger la classification dans une ontologie. Il existe plusieurs systèmes parmi eux nous citons :

RACER (Renamed Abox and Concept Expression Reasoner) [Haarslev 01] : est un système implémentant une logique de description. Prise en compte des représentations au format *DAML+OIL*. RACER permet le test de **satisfiabilité** d'un concept (vérifier qu'un concept C admet des instances) le test de **subsumption** de concepts (vérifier qu'un concept C est subsumé par un concept D), et le test d'**instanciation** (vérifier qu'un individu a est instance d'un concept C, si seulement si $a \in C$).

JENA : est une bibliothèque de classes Java qui facilite le développement d'applications pour le web sémantique. Elle permet de : manipulation de déclarations RDF, lecture et écriture RDF/XML, Notation, Stockage en mémoire ou sur disque de connaissances RDF, Langage d'interrogation d'une base RDF, Gestion d'ontologies : RDF-Sch.

10. Annotation sémantique

Une ressource web est une collection des informations concernant un domaine particulier. Ces informations peuvent présenter l'ambiguïté de la langue naturelle, la représentation de la sémantique des mots. Ces problèmes rendent l'utilisation de cette ressource difficile. Pour résoudre ces problèmes, il faut ajouter des informations explicatives sur la ressource. Cette tâche est appelée annotation. Une définition récente de l'annotation dans [Sandra 04] : « Une annotation est une note particulière attachée à une cible. La cible peut être une collection de documents, un document, un segment de document (paragraphe, groupe de mots, mot, image ou partie d'image, etc.), une autre annotation. À une annotation correspond un contenu, matérialisé par une inscription, qui est une trace de la représentation mentale que l'annotateur se fait de la cible ». Cette définition innove en mettant en avant le lieu du document annoté, appelé « cible ».

Afin d'avoir sur le web, des ressources définies et liées, munies d'un sens explicitement interprétable par les processus logiciels, mieux que être simplement interprétable par les humains, il est nécessaire d'associer à ces ressources des méta-données, décrivant leur



contenu et leur fonctionnalité. L'annotation sémantique est l'une des techniques majeures utilisées pour la création de méta-données sur le Web. L'annotation est dite *sémantique*, lorsqu'elle se base sur un modèle de connaissances déjà défini comme une ontologie. Cette dernière enrichit l'annotation en lui attribuant une sémantique et en la rendant utilisable comme telle par un agent logiciel.

En termes de documentation, les annotations sémantiques décrivent le lien entre les entités se trouvant dans le document et leurs descriptions sémantiques représentées dans l'ontologie. Elles permettent ainsi de désambiguïser le contenu du document pour un traitement automatique (ex. recherche documentaire, résumé...).

Les annotations sémantiques sont des annotations opérationnelles, car elles sont destinées à être traitées par des machines (par opposition aux annotations libres en langage naturel ou composées de symboles souvent tacites). Leur objectif majeur est de *désambiguïser* le document pour un traitement automatique. Les annotations sémantiques ne sont ni publiques (pas consultables directement, il faut disposer d'un éditeur spécialisé ou regarder le code source du document) ni privées (destinées des outils ayant accès à l'ontologie référencée) ce qui en fait un cas très particulier d'annotations.

10.1. Processus d'annotation sémantique :

Derrière le processus général d'annotation sémantique de documents par des ontologies se cache plusieurs phases qui font référence à des annotations de natures un peu différentes. Ces phases sont au nombre de trois :

Repérer : processus manuel ([**Handschuh 01**], [**Maedche 01**], [**Heflin 02**])... ou automatique [**Desmontils 02**] qui consiste à placer dans le document des références aux concepts de l'ontologie qu'il contient. Ces éléments sont considérés comme des méta-données. Le terme "Méta-donnée" est pris dans le sens "donnée sur une donnée". Il n'y a pas ajout d'information mais plutôt désambiguïstation ou formalisation (puisque l'information est présente mais pas directement opérationnelle).

Instancier : processus manuel ou automatique permettant de valider les attributs des concepts à l'aide des informations présentes dans le document (là encore, ce sont des méta-données).



Enrichir : processus manuel visant à ajouter des informations par l'intermédiaire des attributs de concepts qui n'ont pas pu être valus à la phase précédente.

Les deux premières étapes sont des étapes d'insertion de méta-données : il n'y a pas ajout d'information mais plutôt *localisation et caractérisation de l'information* déjà présente de manière intrinsèque. Par contre, la dernière est plutôt une étape d'annotation plus classique, car il y a *ajout d'information* : le document est enrichi d'information qui n'est pas explicitement présente dans le document (il y a plus-value informationnelle). Cette annotation est directement "formalisée" par des méta-données.

10.2. Outils d'annotation sémantique :

Un outil d'annotation sémantique est un outil logiciel qui permet d'insérer et de gérer des annotations sémantiques liées à au moins une ressource donnée. Dans le cadre du Web Sémantique, les outils d'annotation sémantique utilisent une ontologie, ou tout au moins un modèle formel, qui formalise et structure les annotations produites en fonction des concepts et des contraintes définis dans cette ontologie. Un outil d'annotation sémantique peut aussi être utilisé pour peupler une ontologie, i.e. pour instancier la base de connaissance contenant les instances de l'ontologie de référence.

Nous présentons ci-dessous quelques exemples d'outils. Nous n'avons considéré que les outils d'annotations de ressources textuelles, mais il existe également des outils ou projets s'intéressant également à l'annotation sémantique de ressources multimédia.

Annotea³

C'est un système de génération (manuelle) d'annotations RDF pour les pages web [Kahan 01] développé au sein du W3C. L'idée principale est de proposer à chaque utilisateur un outil lui permettant : (i) d'annoter un document en le consultant, (ii) de consulter toutes les annotations associées à un document et (iii) de typer les annotations en leur associant des métadonnées (par exemple dire que cette annotation est un commentaire ou un erratum pour le document) en se basant sur un modèle prédéfini en RDFS. Un langage permettant de décrire des ressources et les relations entre ces ressources.

³ Annotea a été intégré dans le navigateur web Amaya.



MnM

MnM fournit un environnement pour la génération semi-automatique d'annotations sémantiques associées aux documents Web [Vargaz-vera 02 a] [Vargaz-vera 02 b]. Il est basé essentiellement sur des techniques d'apprentissage et des méthodes de TALN (Traitement Automatique de la Langue Naturelle). Dans un premier temps, un corpus d'apprentissage est fourni aux utilisateurs afin de l'annoter manuellement en se basant sur une ontologie du domaine. Ce corpus est ensuite passé à l'outil Amilcare [Ciravegna 04a] qui génère un ensemble de règles d'extraction qui seront ensuite appliquées sur les autres documents pour générer automatiquement les annotations.

KIM

KIM [Popov 03] fournit une plate-forme de génération automatique d'annotations sémantiques et de recherche documentaire basée sur ces annotations. L'approche de KIM est basée sur l'extraction des entités nommées présentes dans le texte à annoter afin d'instancier les concepts d'une ontologie de haut niveau (KIMO) représentée en RDFS. Ces instances sont ensuite utilisées pour annoter les documents et pour enrichir la base de connaissances de KIM. Ces annotations permettent ainsi de faire des recherches documentaires plus précises en utilisant les restrictions sémantiques offertes par l'ontologie. Le processus d'extraction d'informations de KIM est basé sur l'outil GATE.

S-CREAM

C'est une plate-forme pour la création semi-automatique d'annotations sémantiques basées sur une ontologie [Handschuh 02]. Ce sont des travaux inspirés d'Annotea ont conduit à des systèmes d'annotation par rapport à une ontologie en DAML+OIL. Cette plate-forme fournit deux approches, toutes les deux implémentées dans l'outil OntoMat: La première est basée sur une phase d'apprentissage manuelle (similaire à celle de MnM), et la deuxième est basée sur une méthode originale nommée PANKOW [Cimiano 04] qui est entièrement automatique.

Beaucoup d'autres systèmes ont été proposés pour la génération (semi-)automatique d'annotations sémantiques [Uren 06]. Ils proposent, le plus souvent, des annotations en RDF (devenu un standard pour la représentation des méta-données).



10.3. Evaluation des outils d'annotation sémantique :

Pour les systèmes manuels, l'utilisation de langages d'ontologie implémentant les logiques de description comme DAML+OIL (S-CREAM,) et puis en OWL (SMORE [Kalyanpur 03]). Permet ainsi aux machines de raisonner sur les annotations produites. Cependant, l'annotation manuelle par un expert étant une source d'erreurs. La majorité ces systèmes s'intéressent surtout à la tâche d'instanciation de concepts et proposent des techniques d'instanciation de formes simples de relations pouvant exister entre ces concepts (synonymie, paronymie...), mais n'exploite pas les rôles entre les concepts d'ontologie, et donc ces systèmes ne traitent pas les concepts définis.

Les systèmes supervisés à partir d'apprentissage qui ont été proposés, notamment S-CREAM, et MnM traitent exclusivement les concepts primitifs. Pour instancier les concepts d'une ontologie de domaine, ils identifient des chaînes de caractères qui correspondent à ces concepts. Ils appliquent pour cela et pour l'apprentissage des techniques de Recherche et Extraction d'Information [Uren 06], ou le système d'extraction d'information Amilcare qui génère des règles d'extraction à partir d'un corpus de documents fourni en entrée. Cette technique permet de générer des instances de concepts mais n'exploite pas la structure globale du document, les relations entre les instances ne sont donc pas extraites. . Ou bien La structure de la page est en outre exploitée dans certaines méthodes, comme Kushmerick [Kushmerick 97] et Carne [Carne 07] pour identifier les relations entre les éléments. Cependant, pour pouvoir identifier des instances de concepts définis, il est nécessaire d'intégrer les connaissances au processus d'annotation.

Enfin, des systèmes non-supervisés comme Amardillo ou C-PANKOW visent à sortir totalement l'humain de la boucle d'annotation en exploitant la redondance de l'information sur le web. Ces systèmes sont toutefois inadaptés aux pages contenant plusieurs instances d'un même concept reliées à des instances d'un autre concept.

10.4. Approches d'annotation sémantique utilisant les instances des concepts définis

Ce sont des approches qui exploitent les concepts définis dans l'annotation sémantique contrairement aux outils précédents qui n'exploitent que les concepts primitifs dans l'annotation sémantique. Il existe beaucoup des ces approches mais nous citons :



10.4.1. Approche des Classes d'annotation pour l'annotation sémantique

L'approche « Classes d'annotation pour l'annotation sémantique⁴ » sur l'ontologie mono point de vue, proposé par Sylvain Tenier et Yannick Toussaint [Sylvain 07] propose la notion de classe d'annotation comme méthode d'annotation guidée par les connaissances. Les classes d'annotations prennent en compte l'interaction entre la structure et la sémantique des pages web en les formalisant dans un langage commun fondé sur les logiques de descriptions. Cette interaction entre la structure de la page et la sémantique présentée par l'ontologie de domaine permet d'exploiter les concepts définis pour l'annotation sémantique, de telles façons, les éléments de pages web sont annotés par des instances de concepts du domaine. Tandis que, les relations entre ces éléments sont annotées par des instances de rôles.

Évaluation de cette approche

L'approche « Classes d'annotation pour l'annotation sémantique » sur l'ontologie mono point de vue, proposé par Sylvain Tenier et Yannick Toussaint [Sylvain 07] prend en compte la dépendance entre les concepts, exprimée dans une ontologie sous forme de concepts définis. Cette approche désigne l'annotation à la fois comme processus et comme résultat de ce processus. Cette approche est motivée par un parallèle entre la structure d'une page web et la sémantique qui lui est associée. L'apport de cette approche porte sur l'annotation par des instances de concepts définis. Pour l'identification initiale des instances de concepts primitifs. De plus, les logiques de descriptions fournissent l'explication de l'annotation des pages. En outre, la maintenance des annotations devient un problème de satisfaction : si la structure change, l'instanciation échoue.

Avec le code HTML de la page, on se concentre seulement sur les balises de style de la page. Comment faire si la régularité de structure est sur l'organisation (ordre) des éléments de la page et pas des tous sur le style.

10.4.2. Un système automatique d'annotation sémantique de page web

La prise en compte d'informations sémantiques sur le domaine pour l'annotation d'un élément dans une page web à partir d'une ontologie [Sylvain 07 b] suppose d'aborder

⁴ Il existe également une version automatisée la création des classes d'annotation dans [Sylvain 07 a].



conjointement deux problèmes : (1) l'identification de la structure syntaxique caractérisant cet élément dans la page web, et (2) l'identification du concept le plus spécifique (en termes de subsumption) dans l'ontologie dont l'instance sera utilisée pour annoter cet élément.

Ce système permet de former un corpus d'apprentissage par un expert à la manière d'un navigateur web, qui permet de sélectionner un élément dans la page et de choisir dans l'ontologie le concept qui lui correspond afin d'apprendre à reconnaître l'ensemble des éléments à annoter afin d'identifier automatiquement, dans une page web, les éléments qui sont pertinents. Puis déterminer quels sont les concepts de l'ontologie les plus spécifiques possible, pour annoter chacun de ces éléments.

Evaluation de cette approche

L'originalité de cette approche est qu'elle permet de générer des instances de rôles en plus des instances de concepts de l'ontologie. On atteint les limites d'une méthode purement syntaxique. Lorsqu'un même élément peut être annoté par deux concepts différents, un raisonnement est effectué au niveau de l'ontologie pour déterminer le concept subsumant les deux concepts candidats. Dans cette approche, l'exploitation de la structure arborescente de la page présente toutefois certaines limites en fonction de la régularité de la page. Elle s'applique à des documents de type tabulaire contenant de multiples instances des concepts de l'ontologie.

10.5. Langages d'annotation sémantique

Différents langages de niveau de complexité croissante sont proposés afin de mieux exploiter, combiner et raisonner sur les contenus des ressources web.

10.5.1. Les précurseurs

Nous citons HTML-A, et SHOE.

- HTML-A

Une extension d'HTML, appelée HTML-A est utilisée pour insérer des annotations sémantiques dans les pages Web. Cette approche ne spécifiait pas le langage d'implémentation de l'ontologie de référence. Par contre, comme décrit dans [Benjamins 99], les agents logiciels du Web devaient ensuite savoir interpréter cette extension pour l'exploiter correctement, tel l'outil OntoAnnotate [Staab 01].



```
<html>
<head><Title>Le Clan coppola</Title>
<A ONTO="Personnalité:FFCoppola"/>
</head>
<body>
Francis Coppola naît le <A ONTO="Personnalité[dateNaissance=body]">7 avril 1939</A> à <A
ONTO="Personnalité[lieuNaissance=body]">Detroit</A>, dans le <A
ONTO="Personnalité[lieuNaissance=body]">Michigan</A>.
</body>
</html>
```

Figure 7: Exemple d'une annotation sémantique en HTML-A

- SHOE : Simple HTML Ontology Extension

L'approche de SHOE [Heflin 01] crée une extension d'HTML ayant pour objectif l'insertion d'annotations sémantiques pour la description des ressources du Web.

Au lieu d'utiliser l'attribut **ONTO** dans l'élément **A**, SHOE proposait d'utiliser un ensemble d'éléments prédéfinis tels **INSTANCE**, **CATEGORY**, **RELATION**, etc. Ces éléments étaient directement insérés dans le code HTML de la page Web annotée.

Les annotations devaient ensuite être interprétables par des agents Web. L'objectif de ce langage était de permettre aux agents de glaner de la connaissance au sujet de pages Web afin d'améliorer les mécanismes de recherche d'information et de fouille de données. Les ontologies SHOE décrivaient les relations entre les classes, appelées catégories en SHOE, et un ensemble de règles d'inférence simplifiées. Les concepts de l'ontologie de référence sont directement déclarés dans le corps du document et les annotations sont débarquées vis-à-vis du texte originel.

```
<html>
<head><Title>Le Clan coppola</Title></head>
<body>
Francis Coppola naît le 7 avril 1939 à Detroit, dans le Michigan.
<-INSTANCE KEY="FFCoppola">
<-USE-ONTOLOGY ID="People-Ontology" URL="http://www.elle.com/SHOE/people.html"
VERSION="1.0" PREFIX="people">
<-CATEGORY NAME="people.FFCoppola">
<-RELATION NAME="people.dateNaissance">
  <-ARG POS=1 VALUE="FFCoppola">
  <-ARG POS=2 VALUE="7 avril 1939">
</RELATION>
<-RELATION NAME="people.lieuNaissance">
  <-ARG POS=1 VALUE="FFCoppola">
  <-ARG POS=2 VALUE="Detroit">
</RELATION>
</INSTANCE>
</body>
</html>
```

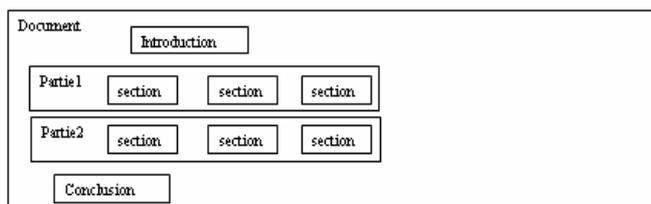
Figure 8. Exemple d'une annotation sémantique en SHOE

10.5.2. XML

L'eXtended Markup Language (XML) [Michard 99] est un langage de description et d'échange de documents structurés, issu de SGML (Standard Generalized Markup Language) et défini par le consortium Web. XML permet de décrire la structure arborescente de documents à l'aide d'un système de balises permettant de marquer les éléments qui composent la structure et les relations entre ces éléments. XML offre la



possibilité de créer ses propres balises. XML ne pose aucune contrainte sémantique sur la description des informations, il ne constitue donc pas un langage de modélisation d'ontologie à lui seul. En réalité les balises XML décrivent le contenu plutôt que la présentation (contrairement à HTML). Ainsi, **XML permet de séparer le contenu de la présentation**, ce qui permet par exemple d'afficher un même document sur des applications ou des périphériques différents sans pour autant nécessiter de créer autant de versions du document que l'on nécessite de représentations !



Correspond à :

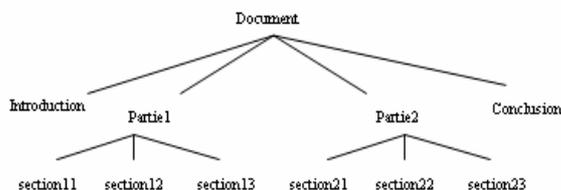


Figure9 : Exemple de représentation arborescente d'un document XML

XML fournit un moyen de vérifier la syntaxe d'un document grâce aux DTD (Document Type Definition). Il s'agit d'un fichier décrivant la structure des documents y faisant référence grâce à un langage adapté. Un document suivant les règles de XML est appelé document bien formé. Un document XML possédant une DTD et étant conforme à celle-ci est appelé document valide. La norme XML n'impose pas l'utilisation d'une DTD pour un document XML, mais elle impose par contre le respect exact des règles de base de la norme XML.

Il est essentiel pour le receveur d'un document XML de pouvoir extraire les données du document. Cette opération est possible à l'aide d'un outil appelé analyseur (en anglais parser, parfois francisé en parseur) comme DOM (Document Object Model) et API. SAX (Simple API for XML). Dans notre travail nous allons utiliser DOMXML. En effet, nous expliquons DOM dans ce qui suit.

Les avantages de XML sont: La lisibilité, une structure arborescente : permettant de modéliser la majorité des problèmes informatiques, universalité et portabilité,



déployable (il peut être facilement distribué par n'importe quels protocoles à même de transporter du texte, comme http), intégrabilité à des applications pourvue d'un parser (c'est-à-dire un logiciel permettant d'analyser un code XML), et extensibilité (un document XML doit pouvoir être utilisable dans tous les domaines d'applications).

Toutefois dans le cadre du Web Sémantique où l'on cherche à donner du sens à l'information, on se rend rapidement compte de la limite de XML. En d'autres termes *XML traite la structure syntaxique des documents mais ne permet de spécifier ni la sémantique d'une balise, ni le typage de son contenu, ni des relations normalisées entre les balises*. C'est la raison pour laquelle un nouveau langage a été défini : RDF (*Resource Description Framework*).

DOMXML :

DOM (*Document Object Model*, traduisez *modèle objet de document*) est une spécification du W3C (*World Wide Web Consortium*) définissant la structure d'un document sous forme d'une hiérarchie d'objets, afin de simplifier l'accès aux éléments constitutifs du document. Plus exactement DOM est un langage normalisé d'interface (*API, Application Programming Interface*), indépendant de toute plateforme et de tout langage, permettant à une application de parcourir la structure du document et d'agir dynamiquement sur celui-ci. Ainsi Javascript et ECMAScript utilisent DOM pour naviguer au sein du document HTML, ce qui leur permet par exemple de pouvoir récupérer le contenu d'un formulaire, le modifier, ... [**Apparo 98**]

DOM se divise en deux spécifications [**Apparo 98**]:

- La spécification **DOM level 1** (*DOM niveau 1*) se séparant en deux catégories
 - *Core DOM level 1*: La spécification pour les documents en général (dont XML)
 - *HTML DOM level 1*: La spécification retenant uniquement les méthodes applicables à HTML.

- La spécification **DOM level 2** ajoutant de nouvelles fonctionnalités comme la prise en compte des feuilles de style CSS dans la hiérarchie d'objets.



Document Object Model (DOM). La classe DOM (Document Object Model) XML fournit une représentation en mémoire d'un document XML. Le DOM vous permet de lire, de manipuler et de modifier un document XML par programme. Donc Le fait de déterminer qu'une page correspond à un modèle revient donc à vérifier qu'elle contient des sous-arbres équivalents à l'arbre servant de modèle. [Apparo 98].

- La classe **XmlDocument**, extension de **XmlNode**, prend en charge des méthodes destinées à exécuter des opérations sur le document [Apparo 98].

Les classes XmlNode et XmlDocument bénéficient toutes deux d'améliorations sur le plan des performances et des possibilités d'utilisation. Elles comprennent des méthodes et des propriétés qui permettent :

- d'accéder et de modifier des nœuds spécifiques au DOM, tels que les nœuds d'élément, de référence d'entité, et bien d'autres ;
- d'extraire des nœuds entiers, en plus des informations qu'ils contiennent, telles que le texte des nœuds d'élément.

- DOM dispose de plusieurs méthodes permettant de manipuler les fichiers XML. En voici les plus utilisés [Apparo 98]:

removeChild : qui permet de supprimer un noeud cible.

InsertBefore (nouveauNoeud, ancienNoeud) : qui permet d'ajouter un noeud *nouveauNoeud* avant le noeud déjà existant *ancienNoeud*.

appendChild (nouveauNoeud) : qui permet d'ajouter un nouveau noeud au noeud cible.

replaceChild(nouveauNoeud,ancienNoeud) : qui permet de remplacer un *ancienNoeud* par un *nouveauNoeud*.

Donc le DOM revêt un intérêt tout particulier pour la lecture de données XML et leur chargement en mémoire afin d'en modifier la structure, d'y ajouter ou d'en supprimer des nœuds, ou encore de modifier les données contenues dans un nœud tout comme dans le texte contenu dans un élément.



Remarque : Le noeud ne sera pas effacé ou échangé dans le fichier lui-même, mais uniquement dans l'arbre XML présent en mémoire.

Exploration des noeuds avec les propriétés DOM :

Pour explorer les noeuds en DOM, nous disposons de plusieurs propriétés [Apparo 98].

En voici les plus utilisés :

firstChild : qui retourne le premier noeud enfant du noeud cible.

lastChild : qui retourne le dernier noeud enfant du noeud cible.

nodeName : qui retourne le nom du noeud cible.

nodeValue : qui retourne la valeur du noeud cible.

childNodes : qui retourne les noeuds enfants du noeud cible.

parentNode : retourne le noeud parent du noeud cible.

nextSibling : retourne le noeud suivant du noeud cible.

previousSibling : retourne le noeud précédent du noeud cible....

10.5.3. RDF (Resource Description Framework)

RDF [W3C 99] est un standard de description de méta-données de puis l'année 97. RDF Utilise des graphes étiquetés orientés (RDF Model). Ces graphes sont constitués des triplets de forme : « sujet - propriété – objet » ou « ressource - propriété - valeur (ressource ou chaîne) ». RDF utilise les URIs (*Universal Resources Identifiers*) pour les pages (et parties de pages) mais aussi pour tout objet qui dispose d'un schéma URI (*documents, ...*).

RDF dispose d'une syntaxe XML et des espaces de nom XML [W3C 99]. RDF est un langage d'assertions et d'annotations. Les assertions affirment l'existence de relations entre des objets. Elles sont donc adaptées à l'expression des annotations associées aux ressources du Web [François 03]. (Principalement des pages Web), sans préjuger à priori sur des domaines d'applications particuliers, ni sur la sémantique de ces domaines. Le langage RDF fournit un moyen pour ajouter la sémantique à un document sans faire oppositions sur la structure du document [François 03].



RDF est un langage formel qui permet d'affirmer des relations entre des «ressources». Il sera utilisé pour annoter des documents écrits dans des langages non structurés ; RDF est muni d'une syntaxe, et d'une sémantique.

Quand les données de XML sont déclarées au format RDF, les applications peuvent comprendre une grande partie de la traduction des données. Il existe cependant un certain nombre de situations où l'on est en droit d'être plus exigeant. Par exemple, certaines fois, nous avons besoin de connaître des informations sur les ressources identifiées, comme par exemple ce qu'elles représentent exactement : Si nous avons une propriété représentant un auteur, nous pouvons en effet exiger que la valeur d'une telle propriété fasse référence à une personne (et non une voiture ou une maison). C'est pour cela que l'on associe à RDF le standard RDF Schéma (RDFS), qui est un peu l'équivalent des DTDs (*Document Type Definition*) pour le XML.

11. Conclusion

L'ontologie est une spécification formelle et explicite des termes ainsi que des relations que ces termes entretiennent entre eux. C'est donc un vocabulaire formalisé de termes et de relations liant ces termes. Ce vocabulaire est partagé par une communauté d'hommes ou de robots.

Les ontologies sont étudiées par les chercheurs travaillant en intelligence artificielle, sur la représentation de la connaissance, et maintenant, sur le Web sémantique. Ce qui traduit les espoirs qu'elles véhiculent et la diversité des types d'applications qui peuvent les intégrer sous des formes plus ou moins riches : recherche d'information sur le web ou dans des grandes bases documentaires, gestion documentaire (indexation, classification, référencement, capitalisation de connaissances, annotation ...etc.).

A travers ce que nous avons présenté dans ce chapitre, il ressort que la notion d'ontologie constitue l'une des approches les plus efficaces pour représenter les connaissances d'un domaine, mais l'existence de plusieurs façons d'appréhender ces connaissances selon différents points de vue donne lieu à la naissance d'une autre notion d'ontologie qui est l'ontologie multi-points de vue. Le chapitre suivant se focalise plus particulièrement sur l'intégration de la notion de point de vue en ingénierie ontologique.

Chapitre II :
***Représentation explicite de point de
vue***



Chapitre2 : Représentation explicite de point de vue

1. Introduction

Une ontologie capture et structure la connaissance dans un domaine et, ce faisant, elle capture la signification des concepts qui sont spécifiques pour ce domaine. Il existe généralement plusieurs façons d’appréhender les connaissances d’un domaine, la représentation des ontologies n’est donc pas une tâche facile. Ceci est dû principalement, à la difficulté de trouver des définitions consensuelles des concepts d’un domaine satisfaisant les définitions propres à chaque utilisateur, qui traduisent son point de vue sur le domaine.

La difficulté de représenter des ontologies est liée principalement à l’existence de plusieurs communautés d’utilisateurs qui peuvent s’intéresser au même domaine mais avec des points de vue différents. Ces communautés, évoluant dans un environnement pluridisciplinaire, coexistent et collaborent entre elles. Chaque communauté a ses intérêts propres et perçoit différemment les entités conceptuelles du même univers de connaissances à représenter. En effet la notion d’une ontologie multipoint de vue a été émergée.

Selon Hemam et al. [Hemam 09a], les deux notions ontologie et point de vue sont complémentaires, en effet l’ontologie représente les connaissances partagées par plusieurs utilisateurs et le point de vue représente les connaissances du domaine qui sont pertinentes selon un point de vue donné. Avec le couplage de ces deux notions on parle plutôt *d’ontologie multi-points de vue*. Cette dernière, confère à un même univers de discours plusieurs descriptions partielles telles que chacune est relative à un point de vue. Les ontologies multi-points de vue sont devenues essentielles aux communautés d’utilisateurs diversifiées qui doivent partager et échanger des informations dans un domaine donné.

Dans ce chapitre, nous présentons la notion de points de vue, les propriétés d’un point de vue et quelques travaux dans le domaine de la représentation des connaissances qui



prennent en compte cette notion. Pour terminer, nous discutons sur le besoin de prise en compte de la notion de point de vue pour l'annotation sémantique.

2. Notion de point de vue

Lorsqu'on parle du *point de vue* de quelqu'un dans le langage courant, il s'agit généralement de la *position* que prend cette personne concernant un sujet particulier. Ces deux termes (point de vue et position) relèvent de la même image : celle du lieu où se trouve la personne vis-à-vis du sujet et aussi de l'angle selon lequel elle le considère.

Un point de vue est plutôt lié à un type de personne (métier, âge, niveau de formation, etc.) ou d'utilisation (une même personne pourra avoir des points de vue différents en fonction de la tâche qu'elle cherche à accomplir). Un point de vue est donc un ensemble de définitions, chacune d'entre elles étant reliée à un concept différent **[Falquet 01]**.

Un point de vue correspond au point de vue que peut prendre une personne particulière à un moment donné. Dans un groupe des personnes, on trouve plusieurs définitions d'un concept. Chaque définition est associée à au moins un point de vue. Le même concept peut donc être défini plusieurs fois, selon des points de vue différents. Dans l'exemple ci-dessus, l'électron était défini selon le point de vue du physicien des hautes énergies, la définition qu'en donnera un chimiste sera différente, par exemple : "corpuscule électrique susceptible d'être arraché, capturé ou mis en commun entre atomes et molécules" alors que pour l'électronicien il s'agira de "plus petit porteur de charge matériel capable de se déplacer utilement dans les circuits électriques". Le point de vue peut également indiquer un niveau de lecture : étudiant, chercheur, journaliste scientifique, grand public.

Un point de vue correspond à un contexte ou à une situation, où des connaissances à propos d'un objet, d'un concept ou d'une entité sont exprimées et considérées comme valides et vraies selon ce point de vue. Un point de vue correspond aussi à une vue, où l'on examine des caractéristiques d'un concept ou d'une entité, qui sont considérées comme des caractéristiques pertinentes du concept ou de l'entité dans cette vue. Un point de vue peut provenir d'une personne, d'un groupe de personnes, d'une communauté dans une entreprise ou une organisation. On peut toujours atteindre un



certain niveau de consensus entre les gens dans un groupe, dans une organisation, contrairement à l'Internet, où l'hétérogénéité est vaste et incontrôlable. [Bach 06]

Dans [Benchikha 07] le terme de point de vue est défini comme “*une position conceptuelle mettant en liaison d'une part un acteur qui observe et d'autre part un monde qui est observé*”. Les acteurs peuvent observer un même univers de discours produisant des points de vue qui peuvent être considérés de différentes manières :

- **Points de vue uniformes:** dans ce cas, tous les acteurs ont la même vision de l'univers de discours et produisent des représentations équivalentes.

- **Points de vue complémentaires:** dans ce cas, chaque acteur voit une partie du monde observé. Chaque point de vue est une représentation partielle et cohérente du monde. Les différentes représentations qui découlent des différents acteurs sont alors complémentaires.

- **Points de vue comparables:** dans ce cas les acteurs produisent des représentations comparables au sens plus général/spécifique.

3. Propriétés d'un point de vue

Le point de vue d'une personne sur un objet définit un contexte ou une situation, où la personne exprime ses expressions sur l'objet donné. Ces expressions sont valides et vraies selon le point de vue [Bach 06]. Un point de vue a trois propriétés principales :

Propriété de filtrage. Un certain nombre des caractéristiques de l'objet sont considérés selon un point de vue. En effet selon la personne et le contexte dans lequel se trouve, certaines caractéristiques seront considérés comme pertinentes, et d'autres non. Par exemple du point de vue Finance, un locataire s'intéressera au prix. Par contre du point de vue Taille, un locataire s'intéressera plus à la surface et au nombre de pièces. Un point de vue est donc considéré comme un filtre sur les caractéristiques des objets, ne conservant que les plus pertinentes selon le point de vue.

Propriété de l'individu. Les énoncés d'une personne à propos des caractéristiques de l'objet sont considérés comme vraies pour cette personne-là, selon son point de vue. Il peut en ce sens être en contradiction avec un autre point de vue. Par exemple, du point de vue d'un locataire pour vacance, le meilleur appartement est l'appartement-à-coté-



mer. Ce point de vue pourra ne pas être partagé par un locataire permanent, ou le meilleur appartement est l'appartement-au-centre-ville. Dans cet exemple, locataire-vacance et locataire-permanent ont deux points de vue différents, dans le sens où leurs opinions s'opposent.

Propriété de représentation. Cette propriété peut être considérée comme une combinaison de deux propriétés précédentes, c'est une représentation de l'objet dans le monde réel sous les yeux d'une personne. En ce sens, un point de vue peut être une façon particulière de représenter et de définir les concepts d'un domaine. Par exemple, dans le domaine urbain, le concept rue est défini comme un support de la circulation par un ingénieur du trafic routier, et comme infrastructure de développement économique et social par un urbaniste.

4. Contexte et point de vue

La notion de contexte dans le domaine de la représentation des connaissances est très proche de la notion de point de vue. Le travail dans [Benerecetti 01] montre trois différentes formes d'utilisation de la notion de contexte dans la représentation des connaissances :

- (1) un contexte correspond à une partie, une vue partielle du domaine. La représentation des connaissances dans un contexte ne couvre donc qu'un sous-ensemble de connaissances dans le domaine ;
- (2) un contexte est une approximation. Les représentations des connaissances dans différents contextes ont différents niveaux d'approximation, différents niveaux de granularité ou d'abstraction ;
- (3) un contexte correspond à une perspective. Les représentations des connaissances dans différents contextes, donc différentes perspectives, dépendent des éléments extérieurs tels que le lieu, le moment...

Par ailleurs, d'après [Bach 06], la notion de point de vue peut être exploité pour :

- (1) découper des connaissances dans un domaine en plusieurs parties ;
- (2) séparer des connaissances de différents niveaux de granularité en différents points de vue ; ou bien (3) délimiter des connaissances selon leurs dépendances envers des éléments extérieurs par différents points de vue.



5. Point de vu en représentation des connaissances

Le point de vue est défini comme la matérialisation des besoins en information d'un groupe d'utilisateurs. Il détermine un univers du discours et une modélisation conceptuelle de cet univers. Il agit comme un processus d'abstraction complexe où:

- Seuls les objets et liens du monde réel pertinents pour le point de vue sont représentés. Cette étape est aussi appelée la délimitation de l'univers du discours. Chaque phénomène de l'univers du discours est décrit d'après le point de vue, ce qui induit les choix suivants:

- * le concept de représentation: un même phénomène peut ainsi être représenté comme un objet, une association ou un attribut selon le point de vue,
- * l'élaboration du type: parmi l'ensemble des propriétés que possède un phénomène, seules les propriétés d'intérêt pour le point de vue sont retenues,
- * parmi les valeurs possibles des propriétés sélectionnées, seules les valeurs correspondant aux objectifs du point de vue sont conservées.

Plusieurs travaux se sont intéressés à la représentation explicite de points de vue dans différents formalismes de représentation des connaissances.] On peut citer par exemple : *KRL* [Bobrow 77], *SHOOD* [Nguyen 92] et *TROPES* [Marino 93] en représentation des connaissances par objets et les travaux [Rivière 99], [Rivière 97], [Rivière 02] qui introduit les points de vue dans le formalisme des graphes conceptuels.

5.1. Représentation des connaissances par objets

Le modèle de la représentation des connaissances par objets repose principalement sur :

- (1) la notion de classe, qui représente un ensemble d'objets dans le monde réel ayant des caractéristiques communes et les classes sont organisées en une hiérarchie avec des relations de subsomption entre elles;
- (2) la notion d'héritage : les objets appartenant à une classe héritent de toutes les caractéristiques (attributs) de cette classe ; les sous-classes héritent de toutes les caractéristiques des classes ancêtres.



La notion de point de vue est intégrée dans ce modèle sous forme de multi-héritage : un objet (une instance) est associé (hérité) à plusieurs classes selon différents points de vue (appelés perspectives en KRL).

Plusieurs systèmes de représentation des connaissances par objets ont été construits en intégrant la notion de point de vue : KRL (Knowledge Representation Language) [Bobrow 77] ; SHOOD [Nguyen 92], TROPES [Marino 93]. Nous détaillons deux travaux suivants :

KRL (Knowledge Representation Language) [Bobrow 77] est l'un des premiers langages de représentation des connaissances à reconnaître qu'un objet peut être vu de plusieurs façons, selon le point de vue de l'observateur. Dans le formalisme KRL, la notion de point de vue (exprimée par le terme *perspective*) est représentée au niveau des instances. Un individu (appelée *unité individuelle*) a une première perspective qui est la classe la plus générale à laquelle il appartient, unité de type *Basic* et il peut avoir d'autres perspectives parmi les unités de spécialisation de celle-ci.

Le système TROPES [Marino 93] est un système de représentation des connaissances par objets (RCO) avec multi-points de vue. TROPES introduit les notions de concept¹, passerelle et point de vue. Ce dernier a un double rôle. Il permet de voir le concept selon un certain angle et de ce fait, seuls les attributs du concept pertinents pour le point de vue sont visibles.

Il permet aussi d'organiser les spécialisations du concept en une hiérarchie de classes significative pour le point de vue. Par ailleurs, la notion de passerelle est proposée afin de mettre en relation les classes d'un concept vu selon différents points de vue. TROPES a fourni une base de réflexion à d'autres projets.

C'est le cas notamment des travaux développés dans [Rivière 99] et [Rivière 97] et le système KASIMIR [D'Aquin 04]. Dans le premier cas, l'auteur utilise le formalisme des graphes conceptuels pour représenter et faire cohabiter différents points de vue d'experts sur un même sujet.

¹ TROPES ne considère pas, comme il est habituel de le faire en RCO et en LD, les termes de classe et de concept comme des synonymes.



Le système d'aide à la décision en cancérologie KASIMIR s'intéresse à la représentation multi-points de vue des connaissances contenues dans les référentiels (sortes de protocoles de décision médicaux) du domaine de cancérologie. Ce travail a d'abord été étudié dans un cadre de la représentation par objets [D'Aquin 04] puis a été implanté dans le cadre des logiques de descriptions distribuées et de C-OWL [D'Aquin 05b].

5.2. Représentation des connaissances par graphes

La notion de point de vue est intégrée dans le système VIEWS [David 87] sous la forme de vue. Le modèle proposé permet de représenter des objets structurés en décrivant plusieurs vues contenant des éléments communs. Une vue est définie par un réseau de parties, de relations et de contraintes. Cela permet de délimiter des relations qu'un objet peut posséder. Comme TROPES, ce système prédéfinit les vues (points de vue) via lesquelles un objet peut être examiné, les utilisateurs du système ne peuvent pas ajouter ou modifier ces vues.

Les travaux de Ribière [Ribière 99], [Ribière 97], et [Ribière 02] permettent d'utiliser le formalisme des graphes conceptuels [Sowa 84] pour représenter et gérer des points de vue explicitement dans une base des connaissances organisée selon de multiples points de vue. Le formalisme des graphes conceptuels proposé par Sowa se compose des éléments de base tels que des concepts, des relations, des types de concept, des types de relation.

Les types de concepts ou de relations sont organisés dans un treillis (une hiérarchie) avec des liens sous-type entre eux. Ce type de lien est exploité et étendu dans ces travaux pour représenter et intégrer la notion de point de vue. Ainsi, nous pouvons décrire qu'un type de concept est sous-type d'un autre type de concept selon un point de vue, et dans ce cas-là, ce dernier est appelé le type de concept basique, le premier est appelé le type de concept orienté (orienté point de vue).

Comme le modèle de ROME, ce modèle permet aussi d'exprimer des représentations multiples d'un concept : un concept instancie un type de concept basique et est relié à plusieurs concepts v-orientés via des liens de représentation. Comme le modèle de



TROPES, ce modèle permet d'avoir des passerelles entre des concepts définis selon de différents points de vue.

5.3. Représentation des connaissances par ontologie

La représentation des connaissances dans un domaine consiste à modéliser et formaliser les connaissances existantes dans ce domaine. Si une ontologie utilisée dans la tâche de représentation des connaissances, elle doit être construite de manière la plus consensuelle et la plus générale possible.

Les objets, les concepts dans l'ontologie doivent être bien sélectionnés et pertinents à représenter. Les termes représentant les objets, les concepts sont aussi bien choisis, ils devraient être clairs, non confus et leurs significations principales devraient correspondre aux objets, aux concepts à définir. De même, les caractéristiques de l'objet à décrire doivent aussi être soigneusement choisies parmi celles considérées comme les plus utiles et pertinentes dans le domaine. Tous ces choix et ces sélections sont effectués par des cogniticiens ou des ingénieurs d'ontologie [**Hemam 09b**].

Il existe généralement plusieurs façons d'appréhender les connaissances d'un domaine, c'est-à-dire différents points de vue (ou perceptions) selon lesquels ces connaissances peuvent être représentées. Ainsi, le même domaine peut avoir plus d'une ontologie, où chacune d'entre elles est décrite selon un point de vue ou une perception particulière [**Hemam 09a**].

En effet, dans une grande organisation, il peut exister plusieurs communautés ou groupes de personnes avec leurs propres *points de vue*. Ces points de vue dépendent du type de personne (métier, âge, niveau de formation, expérience, etc.) ou de l'utilisation (une même personne peut avoir différents points de vue en fonction de la tâche qu'elle cherche à accomplir) [**Falquet 01**]. Ainsi, les deux notions ontologie et point de vue sont complémentaires.

En effet l'ontologie représente les connaissances partagées par plusieurs utilisateurs et le point de vue représente les connaissances du domaine qui sont pertinentes dans un contexte donné. Avec le couplage de ces deux notions on parle plutôt d'**ontologie multi-points de vue**. Les ontologies multi-points de vue sont devenues essentielles aux



communautés d'utilisateurs diversifiées qui doivent partager et échanger des informations dans un domaine d'applications [**Hemam 08**].

5.3.1. Les composants d'ontologie multi-points de vue

Dans une ontologie multi-points de vue, nous représentons les connaissances communes² (concepts, rôles, et individus) pour tous les points de vue dans une représentation globale (neutre), puis nous représentons les connaissances de chaque point de vue dans une représentation locale, et enfin nous relierons les représentations locales par des passerelles. De ce fait, l'ontologie multi-points de vue obtenue, permet de faire converger vers une même ontologie les différentes conceptualisations possibles du domaine modélisé selon des points de vue différents [**Hemam 09b**].

Une représentation locale contient en plus des connaissances communes, des connaissances appropriées à lui selon le point de vue qu'elle représente. Ainsi un concept décrit par un ensemble des attributs dans le point de vue global, va avoir en plus d'autres attributs dans cette représentation locale, avec cette raison, à partir de ce concept, on obtient des concepts plus spécifiques que lui dans cette représentation. De cet fait il apparue des concepts, des rôles, et des individus vue seulement dans une représentation locale spécifique et pas dans les autres représentations locales.

Par exemple : dans le domaine d'immobilier, le concept Appartement décrit dans la représentation globale par l'attribut adresse (cet attribut est vu dans tous les points de vue locaux). Dans la représentation locale Taille, Appartement est décrit par les attributs : adresse, et nombre des pièces. A partir de l'attribut nombre de pièces, on peut obtenir les concepts Grand_Appartement, et Petit_Appartement qui sont plus spécifique que le concept Appartement, mais ces deux concepts vus seulement dans la représentation locale Taille, car l'attribut nombre de pièces est propre pour cette représentation locale. En effet, les composants d'une ontologie multipoints de vue sont [**Hemam 09a**] :

² Les connaissances vues dans tous les points de vue tous c à d les concepts les rôles, et les individus obtenus par une convention entre tous les représentants des points de vue existants



- **Concept global :** c'est un concept vu par l'ensemble des points de vue avec certaines propriétés³ communes. Ces dernières, sont visibles par tous les points de vue et constituent ce qu'on appelle la clé du concept global.
- **Concept local:** c'est un concept qui est vu et décrit localement selon un point de vue donné.
- **Rôle global:** c'est une relation entre deux concepts locaux définis dans deux points de vue différents.
- **Rôle local:** c'est une relation entre deux concepts locaux définis dans le même point de vue.
- **Hierarchie locale :** sous un point de vue l'ensemble des concepts locaux sont liés par la relation de *subsumption* (ou de généralité). Cette dernière, permet de les organiser en hiérarchie locale propre au point de vue. Par ailleurs, chaque concept racine (i.e. sommet de la hiérarchie locale) est associé à (subsumé par) un type de concept global.
- **Passerelle:** l'une des particularités de la représentation multi-points de vue est l'existence d'un canal de communication entre les différents points de vue. Ce canal de communication, appelé passerelle, permet de représenter des liens consensuels entre les concepts locaux de différents points de vue. Pour cela, trois types de passerelles sont distingués :
 - Passerelle d'inclusion : Cette passerelle exprime l'inclusion ensembliste entre l'extension⁴ d'un concept d'un point de vue, source de la passerelle, et celle d'un concept d'un autre point de vue, destination de la passerelle. En termes de logique, cette passerelle peut être vue comme une implication: être instance du concept source implique être instance du concept destination.
 - Passerelle d'équivalence : Exprime l'égalité entre deux concepts X et Y de deux points de vue différents, une telle passerelle est équivalente à deux passerelles d'inclusion
 - Passerelle unidirectionnelles: l'une ayant comme source le concept X et comme destination le concept Y et l'autre, l'inverse, ayant comme source le concept Y et comme destination le concept X.

³ Le terme propriété est pris au sens large et inclut les relations binaires entre concepts globaux et les relations unaires (attributs).

⁴ L'**extension** d'un concept désigne l'ensemble des instances auxquelles il s'applique



- Passerelle d'exclusion : Exprime un lien entre deux concepts, de deux points de vue différents, pour lesquels il ne sera pas possible, pour un même individu, d'appartenir en même temps aux deux ensembles d'instances correspondant à ces deux concepts.

- **Instanciation multiple:** le mécanisme d'instanciation multiple permet à un individu d'être une instance directe d'un ou de plusieurs concepts. Dans le contexte de ce travail, un individu possède la propriété suivante:

Propriété: un individu est une instance d'un concept global et une instance d'un ou de plusieurs concepts locaux définis dans un ou plusieurs points de vue.

Un individu possède donc une description de base (i.e. description globale) et peut être décrit partiellement selon un ou plusieurs points de vue.

5.3.2. Intérêt des ontologies multi-points de vue

Dans [Hemam 09b], le terme *d'ontologie multi-points de vue* est adopté, afin de mettre l'accent sur l'importance de la notion de point de vue pour :

- 1) résoudre le problème de la représentation multiple
- 2) avoir un meilleur accès et une meilleure visibilité des éléments ontologiques (concepts, rôles, individus)
- 3) tirer profit de la représentation multi-points de vue des connaissances pour permettre leur évolution.

Par ailleurs, pour prendre en compte la notion de point de vue, les auteurs dans [Hemam 09b] supposent que les différents points de vue sur un même univers de discours sont des visions partielles mais complémentaires. Leur union est une représentation complète et cohérente du monde.

5.3.3. Elaboration d'ontologie multi-points de vue

Les travaux sur la prise en compte des points de vue multiples au sein des ontologies ont pris deux directions principales. Dans la première, un point de vue est représenté dans une seule ontologie liée à d'autres ontologies relatives aux autres points de vue. La deuxième tend plutôt à intégrer les points de vue dans une même ontologie.

[Falquet 01] et [Falquet 02] présentent une construction de l'ontologie multi-points de vue où les concepts sont associés à plusieurs définitions formelles correspondant aux



différents points de vue sur les concepts en question. Les concepts sont organisés dans une hiérarchie et leurs places dépendent de leurs définitions. Le modèle proposé permet aussi d’avoir de multiples représentations des concepts et de multiples hiérarchies selon de différents points de vue.

5.3.4. La spécification et la conceptualisation d’une ontologie multi-points de vue

L’approche proposée dans [Hemam 08], et [Hemam 09b] est composée par les étapes suivantes :

Etape1 : Construction d’un glossaire global de termes (GGT)

Un terme peut être la représentation d’une entité pertinente du domaine appelée concept, ou d’une relation binaire qui lie deux concepts. Cette étape consiste à construire un GGT, ce dernier recueille les termes les plus importants et qui sont utiles et potentiellement utilisables

dans le domaine que l’on investit et associe à chaque terme identifié une description en langage naturel. Toutes ces sélections et ces descriptions des termes sont effectuées selon un point de vue neutre (i.e. un point de vue global) pour le domaine considéré.

Exemple :

Terme	Description
Appartement	Est un habitat composé d’un ensemble de pièces
Locataire	Est une personne qui prend à loyer un appartement
Agence	Est une entreprise proposant des services pour ses clients
habité-par	Un appartement est habité par un locataire
.....

Etape2 : Construction des ontologies locales

Pour chaque point de vue i, ils construisent une ontologie i locale selon la perception des experts par rapport au point de vue considéré. Une fois les concepts et relations identifiés par leurs termes, il faut en décrire la sémantique dans un langage semi-formel, en indiquant leurs attributs, leurs instances connues et les liens qu’ils entretiennent entre eux. Pour ce faire, nous suivons les activités suivantes :

Activité1 : Construction du glossaire local de termes local

Le glossaire local de termes (i.e. spécifique à un point de vue), consiste à ne retenir que les termes du domaine qui sont intéressants dans le cadre du point de vue visé. Les



experts du domaine, relatifs à un point de vue, peuvent faire de nouvelles propositions et signaler des termes qui ne sont pas apparus dans le glossaire global de termes.

Activité2 : Construction de la Hiérarchie Locale de Concepts

Une hiérarchie locale de concepts (HLC) organise un groupe de concepts entre eux sous forme d'une taxonomie en utilisant la relation de généralisation (i.e. classe/sous-classe).

Etape3 : Liaison des ontologies locales

Cette étape consiste, à lier les différentes ontologies locales des différents points de vue par des liens intermédiaires qu'on appelle passerelles.

Une passerelle décrit une règle entre un concept source (ou un ensemble de concepts

sources) et un concept cible de deux (ou plusieurs) points de vue différents.

- Exemple : tous les Grand_Appartement sont des Appartement_cher
- tous les Petit_Appartement sont des Appartement_pas_cher

On distingue les quatre types de passerelles suivants :

- **Passerelle d'inclusion unidirectionnelle** : Ce type de passerelle exprime l'inclusion ensembliste entre l'extension d'un concept d'un point de vue, source de la passerelle, et celle d'un concept d'un autre point de vue, destination de la passerelle. En termes de logique, la passerelle peut être vue comme une implication: être instance du concept source implique être instance du concept destination. Si on instancie le concept source, l'instance créée est automatiquement rattachée au concept destination.

- **Passerelle d'inclusion avec plusieurs sources** : Dans certains domaines d'application, l'instance doit appartenir à plusieurs concepts de différents points de vue pour que l'on puisse déduire son appartenance à un concept destination d'un autre point de vue. Dans ce cas, qui généralise le cas précédent, une passerelle est décrite par la liste de ses concepts sources des différents points de vue et le concept destination d'un autre point de vue.



- **Passerelle d'inclusion bidirectionnelle** : Une passerelle bidirectionnelle, entre deux concepts C et D de deux points de vue différents, représente l'égalité ensembliste ; une telle passerelle est équivalente à deux passerelles unidirectionnelles : l'une ayant comme source le concept C et comme destination le concept D et l'autre, l'inverse, ayant comme source le concept D et comme destination le Passerelle d'exclusion.

- Par opposition aux trois types de passerelles décrit précédemment, **une passerelle d'exclusion** exprime un lien entre deux concepts, de deux points de vue différents, pour lesquels il ne sera pas possible, pour une même instance, d'appartenir en même temps aux deux ensembles d'instances correspondant à ces deux concepts.

5.3.5. Langages de représentation d'une ontologie multi-points de vue

Les Logiques de Descriptions (LDs) [Paolo 03] forment une famille de langages de représentation de connaissances pouvant être utilisée pour représenter la connaissance terminologique d'un domaine d'application. Elles sont fondées sur les notions de *concepts* correspondant à des "classes d'éléments", de *rôles* correspondant aux "liens entre les éléments" et d'*individus* correspondant aux éléments d'un univers donné.

BORGIDA et al proposent une Logique de Descriptions Distribuée (LDD) [Borgida 03], qui généralise la logique de descriptions, avec une sémantique à modèles locaux pour formaliser des ontologies disjointes reliées par des ponts sémantiques. Les auteurs affirment qu'il n'existe pas une vue globale d'un univers de discours, mais une correspondance entre les différentes conceptualisations hétérogènes. Une *règle de pont* est proposée dans ce cadre pour contraindre cette correspondance. Bien que la proposition originale n'ait considéré que le lien de subsumption entre les concepts, le modèle a été étendu dans [Bouquet 04], par l'introduction de cinq liens sémantiques appelés *mapping contextuels* puis ils ont proposé une

Un langage d'ontologie MVP-OWL permettant de représenter une ontologie multi-points de vue dans le cadre du web sémantique proposé par [Bach 06]. MVP-OWL est basé sur le langage d'ontologie OWL recommandé par le W3C', et est basé sur le modèle multi-points de vue. MVP-OWL est une extension de OWL DL, par l'ajout de



nouvelles primitives permettant d'exprimer les concepts (classes), les faits, les annotations à propos des objets réels en prenant en compte la notion de point de vue. MVP-OWL modifie la sémantique prédéfinie des constructeurs de OWL DL pour pouvoir prendre en compte des expressions du point de vue.

Pour le besoin de représentation d'ontologies multi-points de vue, dans [Hemam 09a], ils ont représenté dans la logique de descriptions, les notions suivantes : **(pas une extension de la logique de description)** :

- **Ontologie multi-points de vue**: c'est une description multiple d'un même univers de discours selon différents points de vue. Elle est définie par un quadruplet $O = \langle C^G, R^G, Vp, M \rangle$, où :

C^G est l'ensemble des concepts globaux,

R^G est l'ensemble des rôles globaux,

Vp est l'ensemble des points de vue et

M est l'ensemble des passerelles.

- **Point de vue**: c'est une description partielle d'un univers de discours selon une perception particulière. Un point de vue est défini par un triplet $VP_K = \langle C^L, R^L, A^L \rangle$, où :

C^L est l'ensemble des concepts locaux,

R^L est l'ensemble des rôles locaux et

A^L est l'ensemble des individus locaux.

- **Concept global** : la Syntaxe d'un concept global est : Soit $S = \{vp_1, \dots, vpk, \dots, vp_m\}$ l'ensemble des noms de points de vue. Un concept global noté par $C^{\hat{o}}$, peut être formé en utilisant les constructeurs booléens (conjonction, disjonction) et les constructeurs de restriction globaux⁵ décrits dans la table 1 suivante :

Constructeurs de restrictions globales	Descriptions
	Définit un nouveau concept dont toutes ses instances sont

⁵ Permet à un rôle d'avoir plusieurs cardinalités et plusieurs domaines de valeurs selon des points de vue différents



$\forall_{vp_1, \dots, vp_k} R.C$	reliées, via le rôle R , seulement aux instances du concept C dans les points de vue vp_1 à vp_k
$\exists_{vp_1, \dots, vp_k} R.C$	Définit un nouveau concept dont toutes ses instances sont reliées, via le rôle R , à au moins une instance du concept C et seulement dans les points de vue vp_1 à vp_k
$\leq_{vp_1, \dots, vp_k} n R$ $\geq_{vp_1, \dots, vp_k} n R$	Spécifie la cardinalité minimale ou maximale du rôle R dans les points de vue vp_1 à vp_k

Table 1. Constructeurs de restriction globaux.

- **Concept local** : la Syntaxe d'un concept local est : Soit $vp_i \in S$. Un concept local, noté $vp_i: C$, peut être défini au moyen de la syntaxe suivante:

$vp_i: C \rightarrow (\text{Concept-Global}) \mid (\neg C) \mid (C \sqcap C) \mid (\exists R. C) \mid (\forall R. C) \mid (\geq n R) \mid (\leq n R) \mid (R. \{a, b \dots\})$

- **Rôle global** : la Syntaxe d'un rôle global est : Un rôle global, noté par $R^{\hat{}}$, peut être défini selon la forme suivante: $R^{\hat{}}(vp_i: C, vp_j: D)$ où R est un nom de rôle global, C et D sont deux concepts locaux définis dans deux points de vue différents.
- **Rôle local** : la Syntaxe d'un rôle local est : Un rôle local, noté $vp_i: R$, peut être défini selon la forme suivante: $vp_i: R(C, D)$ où R est un nom de rôle local défini dans le point de vue VP_i , C et D sont deux concepts locaux définis dans le même point de vue VP_i



- **Estampille:** nous adaptons le mécanisme d'estampillage⁶ utilisé dans [Hemam 08] pour permettre la multi représentation des concepts. Dans notre approche, une estampille (i.e. label) permet de reconnaître pour chaque élément ontologique (i.e. concept, rôle, individu) le point de vue auquel il appartient.

- **Hierarchie locale :** La Syntaxe d'une relation de subsomption est : Sous un point de vue VP_i , une hiérarchie locale, notée vp_i/H , est définie par le triplet $(C^L, \partial, \sqsubseteq)$ où :
 1. C^L est l'ensemble des concepts locaux,
 2. ∂ est une fonction de C^L dans C^G qui associe chaque concept racine (i.e. le plus général) S de C^L à un seul concept global $C^{\hat{o}}$ de C^G ,
 3. \sqsubseteq est la relation de subsomption utilisée pour exprimer explicitement un lien d'ordre direct selon les deux formes suivantes :

$vp_i: D \sqsubseteq vp_i: C$ où C et D sont deux concepts locaux définis dans le même point de vue VP_i ,

$vp_i: S \sqsubseteq C^{\hat{o}}$ où S est le concept le plus général défini dans le point de vue VP_i et $C^{\hat{o}}$ est un nom de concept global.

- **Passerelle**

Deux types de passerelles sont distingués:

Une passerelle unidirectionnelle représente une relation d'inclusion.

Une passerelle bidirectionnelle représente une relation d'égalité ou bien une relation d'exclusion.

Une passerelle (uni ou bidirectionnelle) s'exprime de quatre manières :

$$vp_i: X \xrightarrow{\hat{o}} vp_j: Y \quad (\text{Inclusion}) \quad (1)$$

⁶ Les estampilles sont des concepts de modélisation utilisés afin de différencier les représentations multiples d'une même réalité [Falquet 01] .



Exprime l'inclusion ensembliste entre l'extension⁷ d'un concept d'un point de vue, source de la passerelle, et celle d'un concept d'un autre point de vue, destination de la passerelle. En termes de logique, cette passerelle peut être vue comme une implication: être instance du concept source implique être instance du concept destination.

$$vp1: X_1 \sqcap \dots \sqcap vpk: X_k \xrightarrow{\circ} vpj: Y \text{ (Inclusion avec plusieurs sources)}$$

(2)

Dans certains domaines d'application, l'individu doit appartenir à plusieurs concepts de différents points de vue pour que l'on puisse déduire son appartenance à un concept destination d'un autre point de vue. Dans ce cas, une passerelle est décrite par la liste de ses concepts sources des différents points de vue et le concept destination d'un autre point de vue.

$$vpi: X \xleftrightarrow{=} vpj: Y \text{ (Inclusion bidirectionnelle)}$$

(3)

Exprime l'égalité entre deux concepts X et Y de deux points de vue différents, une telle passerelle est équivalente à deux passerelles d'inclusion unidirectionnelles: l'une ayant comme source le concept X et comme destination le concept Y et l'autre, l'inverse, ayant comme source le concept Y et comme destination le concept X.

$$vpi: X \xleftrightarrow{\perp} vpj: Y \text{ (Exclusion bidirectionnelle)}$$

(4)

Exprime un lien entre deux concepts, de deux points de vue différents, pour lesquels il ne sera pas possible, pour un même individu, d'appartenir en même temps aux deux ensembles d'instances correspondant à ces deux concepts.

▪ **Instanciation multiple.**

La sémantique de ces notions est fournie par une interprétation globale I_G , un ensemble d'interprétations locales $I_{vp} = \{I_1, \dots, I_k, \dots, I_m\}$, et un ensemble $\{r_{ij}\}_{i \neq j}$ de relations de domaine :

⁷ L'extension d'un concept désigne l'ensemble des instances auxquelles il s'applique



Interprétations locales : Pour chaque point de vue $\mathbf{VP}_K = \{C^L, R^L, A^L\}$, nous associons une interprétation locale $I_k = (\Delta^k, \cdot^k)$ où Δ^k est un ensemble appelé domaine d'interprétation locale et où \cdot^k est appelée fonction d'interprétation locale. Cette fonction associe pour chaque concept local $A \in C^L$ un sous-ensemble A^k de Δ^k , pour chaque nom de rôle local $r \in R^L$ un sous-ensemble R^k de $\Delta^k \times \Delta^k$ et pour chaque individu local $a \in A^L$ un élément de Δ^k .

Interprétation globale : Une interprétation globale $I_G = (\Delta^I, \cdot^I)$ consiste en un domaine d'interprétation globale $\Delta^I = \Delta^{I_1} \cup \dots \cup \Delta^{I_k} \dots \cup \Delta^{I_m}$, et en une fonction d'interprétation globale \cdot^I qui fait correspondre à chaque concept global $C^{\hat{0}} \in C^G$ un sous-ensemble de Δ^I et à chaque rôle global $r^{\hat{0}} \in R^G$ un sous-ensemble de $\Delta^{I_i} \times \Delta^{I_k}$ où $i \neq k$.

Relation de domaine : Une relation de domaine r_{ij} de Δ^{I_i} à Δ^{I_j} , définit comment deux points de vue différents (\mathbf{VP}_i et \mathbf{VP}_j) interagissent et elle est nécessaire pour évaluer la satisfiabilité des passerelles. Nous utilisons la notation fonctionnelle $r_{ij}(a)$ pour dénoter l'ensemble $\{a' \in \Delta^{I_j} \mid \langle a, a' \rangle \in r_{ij}\}$, le couple $\langle a, a' \rangle$ appartenant à r_{ij} signifie que à partir du point de vue \mathbf{VP}_j , l'élément a de Δ^{I_i} , correspond à l'élément a' de Δ^{I_j} . Par ailleurs, pour chaque sous-ensemble A^{I_i} de Δ^{I_i} nous utilisons $r_{ij}(A^{I_i})$ pour dénoter $\bigcup_{a \in A^{I_i}} r_{ij}(a)$. En d'autres termes, $r_{ij}(A^{I_i})$ est une notation pour l'interprétation du concept local A , défini dans le point de vue \mathbf{VP}_i , telle que considéré dans le domaine d'interprétation du point de vue \mathbf{VP}_j .

Satisfiabilité des passerelles : Les passerelles sont interprétées, en utilisant la relation de domaine, selon les définitions suivante :

1. $\langle I_i, r_{ij}, I_j \rangle \models \text{vp}_i: X \xrightarrow{\hat{0}} \text{vp}_j: Y$, Si $r_{ij}(X^{I_i}) \subseteq Y^{I_j}$
2. $\langle \{I_1, \dots, I_k\}, r_{ij}, I_j \rangle \models \text{vp}_1: X_1 \sqcap \dots \sqcap \text{vp}_k: X_k \xrightarrow{\hat{0}} \text{vp}_j: Y$,
Si $r_{1j}(X_1^{I_1}) \cap \dots \cap r_{kj}(X_k^{I_k}) \subseteq Y^{I_j}$
3. $\langle I_i, r_{ij}, I_j \rangle \models \text{vp}_i: X \xleftrightarrow{=} \text{vp}_j: Y$,
Si $r_{ij}(X^{I_i}) \subseteq Y^{I_j}$ et $r_{ji}(Y^{I_j}) \subseteq X^{I_i}$
4. $\langle I_i, r_{ij}, I_j \rangle \models \text{vp}_i: X \xleftrightarrow{\perp} \text{vp}_j: Y$,



$$\text{Si } r_{ij}(X^i) \cap r_{ji}(Y^j) = \emptyset$$

Les auteurs dans [Hemam 09a] illustrent leur approche à travers un simple exemple de modélisation et qui concerne la représentation du domaine de l'immobilier. Dans cet exemple, trois points de vue sont considérés: Taille, Finance et Localisation désignés respectivement par vp_1 , vp_2 et vp_3 . Par ailleurs, le niveau global est simplifié à un unique concept global *Appartement*.

Concept global

$$\text{Appartement}^{\hat{\circ}} \equiv (\forall_{vp_1} \text{ Nbr_pièces.Number}) \sqcap (\forall_{vp_2} \text{ Loyer.Number}) \sqcap$$

$$(\forall_{vp_1, vp_2, vp_3} \text{ Adresse.String}) \sqcap (\geq_{vp_1, vp_2, vp_3} 1 \text{ Adresse}) \sqcap (\leq_{vp_1, vp_2, vp_3} 1 \text{ Adresse})$$

Définit un concept global avec un attribut Nbr_pièces selon vp_1 , un attribut Loyer selon vp_2 et un attribut Adresse selon les trois points de vue vp_1 , vp_2 et vp_3

Concept local

$$vp_1: \text{Petit_Appartement} \equiv \text{Appartement}^{\hat{\circ}} \sqcap (\text{Nbr_pièces. } \{1, 2\})$$

Définit un concept local, dans le point de vue vp_1 , comme étant un appartement et dont la valeur de l'attribut Nbr_pièces est dans l'ensemble $\{1, 2\}$.

Relation de subsomption

$$vp_2: \text{HLM} \sqsubseteq vp_2: \text{Appartement_pas_cher}$$

Exprime un lien de subsomption entre deux concepts locaux définis dans le même point de vue. En effet, sous le point de vue vp_2 , tous les HLM sont des appartements pas chers.

$$vp_2: \text{Appartement_pas_cher} \sqsubseteq \text{Appartement}^{\hat{\circ}}$$

Exprime un lien de subsomption entre le concept local Appartement_pas_cher, défini sous le point de vue vp_2 , et le concept global Appartement^{hat{circ}}

Rôle local

$$vp_2: \text{habite_par}(\text{Appartement_cher}, \text{Locataire_Riche})$$

Définit un rôle local entre deux concepts locaux définis dans le même point de vue vp_2

Rôle global

$$\text{habite}^{\hat{\circ}}(vp_2: \text{Locataire_Riche}, vp_3: \text{Appartement_CentreVille})$$

Définit un rôle global entre deux concepts locaux définis dans deux points de vue différents (vp_2 et vp_3)



Passerelle unidirectionnelle/bidirectionnelle

vp₂: HLM \longleftrightarrow vp₃: Appartement_Banlieue

Exprime que les deux concepts locaux, définis dans deux points de vue différents, sont équivalents.

En effet, tous les HLM sont dans la banlieue et tous les appartements de banlieue sont des HLM

vp₁: Plus_TroisPièce \sqcap vp₃: Appartement_CentreVille $\xrightarrow{\circ}$ vp₂: Appartement_cher

Signifie que tous les appartements de plus de trois pièces qui se trouvent au centre-ville sont des appartements chers

5.3.6. Ontologie multi-points de vue et annotation sémantique

Lorsque nous voulons annoter sémantiquement une ressource dans une organisation hétérogène (i.e. constituée de plusieurs groupes de personnes diversifiés) en prenant en considération un aspect multi-points de vue, nous pouvons utiliser une ontologie multi-points de vue. Les annotateurs ou les utilisateurs finaux de cette ontologie peuvent faire des énoncés (i.e. des annotations) comme ils veulent et selon leurs niveaux de connaissances et leurs point de vue visé. A chaque point de vue correspond une annotation sémantique locale ayant son propre langage et surtout sa propre interprétation vis-à-vis au contenu et à la fonctionnalité de la ressource considérée.

Donc cette ressource va être annotée en plusieurs façons selon chaque point de vue. Elle va avoir une annotation globale selon le point de vue global, et plusieurs annotations locales selon les différents points de vue à considérer. De ce fait, pour un point de vue spécifique, seulement les éléments de la page concernant ce point de vue vont être annotés par des instances des concepts connus dans ce point de vue et les autres éléments ne vont pas être annotés dans ce point de vue.

Actuellement, il n'existe pas, à notre connaissance, des travaux qui s'intéressent au problème de l'annotation sémantique à partir d'une ontologie multi-points de vue. Mais dans [Bach 06] il y a un peu des discussions sur ce sujet. Ils ont considéré que les descriptions d'un objet réel sont également comme des annotations. Dans un système employant le modèle multipoints de vue (MVP), les annotations sont stockées dans une base d'annotations, sous forme des quadruplets $\langle IS, IP, IO, VP \rangle$. Par exemple, $\langle Voiture_de_Rose, aCouleur, Couleur_Verte, vp_Khaled \rangle$ est une annotation stockée dans la base d'annotations. Dans ce système, ils ont utilisé un squelette en MVP-OWL



pour décrire un objet réel (individu). Dans chaque bloc `<vp:EQUIV> ... </vp:EQUIV>`, ils écrivent un ensemble de descriptions à propos de l'objet réel selon un point de vue particulier. Les descriptions dans un point de vue doivent être cohérentes mais elles peuvent être contradictoires avec celles décrites dans un autre point de vue. D'autres descriptions peuvent être écrites dehors les blocs `<vp:EQUIV> ... </vp:EQUIV>`, dans ce cas, elles sont considérées comme décrites selon le point de vue général (globale).

6. Conclusion

Dans le but de permettre la construction et l'exploitation d'un Web sémantique dans une organisation hétérogène où l'on a besoin de résoudre les problèmes concernant l'hétérogénéité, nous avons présenté la notion de point de vue et comment introduire cette notion dans le domaine de la représentation des connaissances pour construire une ontologie multi-points de vue.

Parmi les champs d'utilisation de la notion d'ontologie est l'annotation sémantique. Lorsque nous voulons annoter sémantiquement une ressource dans une organisation hétérogène, nous pouvons utiliser une ontologie multi-points. Ainsi, nous avons montré le besoin d'introduire la notion de point de vue dans l'annotation sémantique d'une ressource. Ceci afin de résoudre le problème d'interprétation sémantique d'une ressource dans un domaine hétérogène.

L'objectif du chapitre suivant est de proposer une approche d'annotation sémantique basée sur l'exploitation et l'instanciation d'une ontologie multi-points de vue.

Chapitre III :

***Proposition d'une approche
d'annotation à partir d'une ontologie
multi-points de vue***



Chapitre3 : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

1. Introduction

Dans ce travail, nous nous intéressons au problème de l'annotation sémantique des ressources dans une organisation hétérogène (i.e. constituée de plusieurs groupes de personnes diversifiés) en prenant en considération un aspect multi-points de vue. Un point de vue correspond, dans ce cadre, à la représentation des connaissances utiles à un groupe de personnes particulier, qui coexiste et collabore avec d'autres groupes de personnes. Chaque groupe de personnes a ses intérêts propres ; chacun regarde des propriétés et relations particulières des objets conceptuels du même univers de connaissances à considérer.

L'objectif principal de ce travail est de proposer une approche d'annotation sémantique basée sur l'exploitation et l'instanciation d'une ontologie multi-points de vue. Cette dernière, permet de faire converger vers une même ontologie les différentes conceptualisations possibles du domaine modélisé selon des points de vue différents [Hemam 09b]. Par ailleurs, les annotateurs ou les utilisateurs finaux de cette ontologie peuvent faire des énoncés (i.e. des annotations) comme ils le souhaitent et selon leurs niveaux de connaissances et leurs points de vue personnels. A chaque point de vue correspond donc une annotation sémantique locale ayant son propre langage et surtout sa propre interprétation vis-à-vis du contenu de la ressource considérée.

Pour développer une approche pour ce type d'annotation, nous nous sommes basés sur la notion de classe d'annotation qui est proposée et utilisée dans [Sylvain 07] comme méthode d'annotation guidée par les connaissances. Les classes d'annotations prennent en compte l'interaction entre la structure et la sémantique des ressources à annoter en les formalisant dans un langage commun basé sur les logiques de descriptions. De ce fait, cette technique peut être utilisée pour construire une structure propre à chaque point de vue. Ensuite, chaque structure va interagir avec la sémantique concernant le point de vue qu'elle représente.



Chapitre3 : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

Dans ce qui suit, nous allons présenter, dans un premier temps, les principaux outils qui seront utilisés dans notre approche. Ainsi que la description en logique de descriptions d'une ontologie multi-points de vue. En suite, sur la base du modèle de représentation développé, nous proposons dans la section 3 une approche d'annotation sémantique multi-points de vue, dans la quelle le formalisme de la logique de descriptions est utilisé comme mode de représentation unifié de la structure des documents, de l'ontologie et aussi de l'annotation sémantique du document.

2. Outils et techniques utilisés dans l'approche proposée

Dans cette section, nous présentons tous les outils et les techniques utilisés dans notre approche, ainsi que le principe de développement d'une ontologie multi-points de vue de domaine.

2.1. XML

Dans l'approche de classe d'annotation [Sylvain 07], la structure de la page est modélisée en logique de description à partir d'une représentation sous forme d'arbre DOM du code HTML de la page. Dans notre approche proposée, nous modélisons la structure de la page en logique de description, mais ceci à partir d'une représentation sous forme d'arbre DOMXML.

En effet, avec le code HTML, on se concentre uniquement sur le style des balises et non pas sur l'organisation des éléments de la page concernée (i.e. comment ces éléments sont regroupés et organisés sur la page).

Par exemple : un appartement est un élément composé d'une adresse, nombre des pièces, et loyer. Donc il faut préciser dans la page qu'un appartement est le regroupement d'une adresse, d'un nombre de pièces et d'un loyer. De tel sorte ces derniers constitués les attributs de l'objet appartement. Dans ce cas le code XML est plus approprié, car il propose des balises personnalisées permettant de représenter tel situation. Ainsi, nous représentons l'élément appartement dans la balise personnalisée `< appartement >`. Puis entre les balises `< appartement >` et `</ appartement >` nous représentons les éléments adresse, nombre des pièces, et loyer. De telle sorte, chaque



Chapitre3 : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

élément est représenté à son tour dans une balise personnalisée. L'exemple suivant montre cette situation :

```
<Appartement>
<Adresse> 112, 6 rue benbais 25000, 4ème étage</Adresse>
<Nbrpiece> 4 </ Nbrpiece >
<Loyer > 4000 DA</ Loyer >
</Appartement>
```

2.2. DOMXML

Dans cette section, nous allons expliquer les composants d'un arbre DOM avec un exemple que nous l'utiliserons dans notre étude de cas. Ainsi nous rémentionnerons les méthodes et les propriétés que nous voulons les utiliser dans notre approche.

En prenant l'exemple de la page web « Immobilier », qui contient une liste de locataires. Chaque locataire, qui est identifié par un nom, loue un ou plusieurs appartements. Par ailleurs, un appartement, qui est identifié par une adresse, ce caractérise par un nombre de pièce, ainsi qu'un loyer.



Figure1 : la page web

```
<Racine>
<Location>
<Personne>
<Nom>Héman</Nom>
<Salaire>40000DA</Salaires>
</Personne>
<Appartement>
<Adresse>appartement numero 112, etage4, 6rue benbais25000</Adresse>
<Nbrpieces>1</Nbrpieces>
<Loyer >2000</Loyer>
</Appartement>
<Appartement>
<Adresse>appartement numero 10, etage1, constantine25000</Adresse>
<Nbrpieces>3</Nbrpieces>
<Loyer>8000DA</Loyer>
</Appartement>
</Location>
<Location>
<Personne>
<Nom>Mohamed</Nom>
<Salaires>14000DA</Salaires>
</Personne>
<Appartement>
<Adresse>appartement numero 11, etage4, Ain S'mara 25000</Adresse>
<Nbrpieces>2</Nbrpieces>
<Loyer>4000</Loyer>
</Appartement>
</Location>
</Racine>
```

Figure2 : l'extrait de code XML



Chapitre3 : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

La structure arborescente de la page web définie par le Document Object Model (DOM) est représentée dans la figure3 :

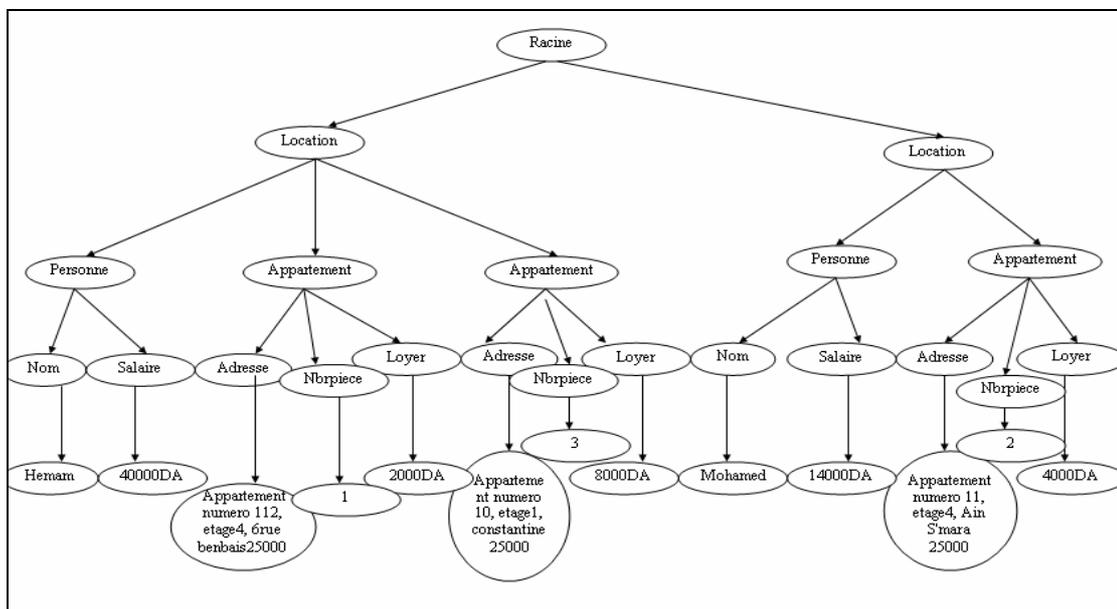


Figure3 : arbre DOM de la page « Immobilier »

Au sein de la structure d'un document XML, chaque cercle de cette illustration représente un nœud, appelé objet **XmlNode**. L'objet **XmlNode** est l'objet de base de l'arborescence DOM. Les nœuds : « **Hemam** », « **14000 DA** », « **Appartement numéro 11, étage4, 6 rue benbais 25000** », « **1** », « **2000 DA** », « **Appartement numéro 10, étage1, Constantine 25000** », « **3** », « **8000 DA** », « **Mohamed** », « **40000 DA** », « **Appartement numéro 11, étage4, Ain S'mara 25000** », « **2** », et le nœud « **4000 DA** », représentent respectivement le texte des nœuds d'élément. **Nom**, **Salaire**, **Adresse**, **Nbrpieces**, **Loyer**, **Adresse**, **Nbrpieces**, **Loyer**, **Nom**, **Salaire**, **Adresse**, **Nbrpieces**, **Loyer**, Les autres nœuds (Location, Personne et Appartement) sont les nœuds d'élément. Le nœud Racine représente la racine de la page.

DOM dispose de plusieurs méthodes permettant de manipuler les fichiers XML. Dans le cadre de notre travail, nous utilisons seulement la méthode **removeChild** qui permet de supprimer un nœud cible. En effet, DOM permet la lecture de données XML et leur chargement en mémoire afin d'en modifier la structure, d'y ajouter ou d'en supprimer des nœuds, la modification donc s'effectuée sur l'arbre XML présent en mémoire et pas



Chapitre3 : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

sur le fichier lui-même. Cette propriété nous permet de faire sortir plusieurs structures virtuelles pour notre document, sans changer la structure réelle.

Pour explorer les noeuds en DOM, nous disposons de plusieurs propriétés [Apparo 98]. Dans le cadre de notre travail, nous utilisons seulement *firstChild* : qui retourne le premier noeud enfant du noeud cible, et *nextSibling* : qui retourne le noeud suivant du noeud cible.

2.3. Ontologie de domaine

L'ontologie doit guider la génération d'individus de domaine à partir des chaînes de caractères identifiées dans une page web et permettre la génération d'instances de rôles entre les individus reliés [Sylvain 07]. Cette ontologie est une ontologie multipoints de vue. Dans le cadre de notre travail, cette ontologie doit vérifier les propriétés suivantes :

- Il faut distinguer deux types de représentations : Représentation globale selon le point de vue neutre (convention entre tous les points de vue), qu'on peut la nommer l'ontologie globale qui contient tous les concepts et les rôles obtenus par une convention entre toutes les représentants des points de vue existants, et plusieurs représentation locales selon chaque point de vue, qu'on peut les nommer les ontologies locales. Où chacune d'elle contient en plus des connaissances vues dans le point de vue global, des connaissances appropriées pour le point de vue qu'elle représente.

Ainsi pour notre travail, chaque représentation locale, ainsi que la représentation globale est définie avec les concepts et rôles suivants :

- Les concepts appropriés d'un point de vue sont hiérarchisés dans ce point de vue par la relation de subsomption dans une hiérarchie H. Un sous ensemble de H est l'ensemble des concepts *primitifs*. Un concept primitif est défini par une relation de subsomption avec un autre concept primitif. Certain concept ont des attributs, et parmi ces attributs il y a un ensemble des attributs constituant la clé. Cette dernière permet d'identifier le concept et d'identifier une instance de ce concept dans une page web. Notant que les attributs de cet ensemble sont visibles dans tous les points de vue. Par



Chapitre 3 : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

exemple, "Numero", "Adresse", et "Etage" appartient à l'ensemble des attributs du concept *Appartement*, ces attributs sont visibles dans le point de vue global, taille, et financier, donc cet ensemble constitue la clé du concept *Appartement*.

- Les attributs correspondent à des caractéristiques, des spécificités particulières, attachées à un concept et qui permettent de le définir de manière unique dans le domaine. Leurs valeurs sont littérales, i.e. de type primitif, comme une chaîne de caractères ou un nombre entier.
- Les rôles définissent les relations binaires entre concepts. Soient C, D deux concepts de cette ontologie, l'existence d'une relation entre C et D est formalisée comme suit : $C \underset{r}{\subseteq} D$. Dans cette ontologie, r est unique pour deux concepts donnés,
- Les concepts définis sont définis par un ensemble de rôles ou bien par un ensemble des attributs.
- Les concepts communs (vue dans le point de vue global) sont plus généraux que les concepts locaux.

2.3.1. Développement de l'ontologie multi-points de vue

Afin de construire notre ontologie, nous nous basons sur l'approche proposée dans [Hemam 08], [Hemam 09b].

A) La spécification et la conceptualisation

Nous commençons par la construction d'un glossaire global de termes (GGT) on recueille les concepts et les rôles communs¹ entre tous les points de vue, puis en utilisant ce glossaire pour construction des ontologies locales, en premier temps nous construirons un glossaire local de termes local, où on regroupe les concepts et les rôles vus dans chaque point de vue local. En suite, nous construirons pour chaque ontologie locale une hiérarchie locale de concepts vus dans le point de vue local qui cette ontologie locale le représente. Enfin nous relierons chaque concept local d'une ontologie

¹ Vues dans tous les points de vue.



Chapitre3 : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

local par un concept local d'une autre ontologie locale par un des types des passerelles vus dans le chapitre2 afin d'établir des liaisons des ontologies locales

B) Formalisation

Nous utilisons le formalisme de la logique de description afin de formaliser notre ontologie, tout en représentant les concepts, les rôles, et les individus, ainsi que les notions concernant le point de vue comme il est indiqué dans [Hemam 09a] (les détails dans le chapitre 2),

- Une ontologie multi_points de vue est un quadruplet $O = \langle C^G, R^G, Vp, M \rangle$, où :

C^G est l'ensemble des concepts globaux,

R^G est l'ensemble des rôles globaux ($C^G \cup R^G$: représentation globale),

Vp est l'ensemble des représentations locales (points de vue) et

M est l'ensemble des passerelles.

- Une représentation locale est défini par un triplet $VP_k = \langle C^L, R^L, A^L \rangle$, où :

C^L est l'ensemble des concepts locaux,

R^L est l'ensemble des rôles locaux et

A^L est l'ensemble des individus locaux.

- nous associons une estampille pour chaque point de vue, par exemple vp1 indique un point de vue estampillé par vp1.

- **Concept global** : un concept global est écrit avec un nom plus le symbole « $\hat{\circ}$ » par exemple le concept global Appartement s'écrit $\text{Appartement}^{\hat{\circ}}$.

Nous représentons un concept $C^{\hat{\circ}}$ par ses attribut comme suit : $C \equiv (\forall vp1 \dots vpn \ a1.codomaine1) \cap ((\forall vp1 \dots vpn \ a2 .codomaine2) \cap (\geq vp1, \dots vpn, \ 1a1) \cap (\leq vp1, \dots vpn, \ 1a1) \cap (\geq vp1, \dots vpn, \ 1a2) \cap (\leq vp1, \dots vpn, \ 1 \ a2) \cap (\forall vp1 \ \geq 0 \ a3.codomaine3)$.

Les attributs a1, et a2 sont vue dans tous les points de vue, l'attribut a3 est vu dans le point de vue local vp1. L'ensemble des attributs a1, a2 constitue la clé pour le concept C et l'attribut a3 n'est pas attribut clé donc il suffit l'existence sur la page des attributs



Chapitre 3 : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

a1, et a2 pour identifier le concept C, et l'absence de l'attribut a3 n'a pas d'influence sur l'identification du concept C.

- **Concept local:** nous représentons le concept local comme suit : « l'estampille du point de vue où ce concept est défini : le nom de concept » par exemple le concept local D selon le point de vue vp1 est représenté comme suit : vp1 : D

- Rôle global: nous le représentons comme suit :

Nom concept^ô (concept local, concept local). Par exemple : « habite^ô (vp2: Locataire_Riche, vp3: Appartement_CentreVille) »

Rôle local: nous le représentons comme suit :

Estampille : nom rôle local (nom concept local, nom concept local)

Par exemple : « vp2: habite_par (Appartement_Cher, Locataire_Riche) ».

Les passerelles : nous les représenté sous forme :

Une passerelle bidirectionnelle concept local ↔ concept local

Une passerelle unidirectionnelle concept local → concept local

3. L'approche proposée

Dans notre approche nous gardons le même principe et les mêmes étapes de l'approche classes d'annotation pour l'annotation sémantique, proposée dans [Sylvain 07], mais nous construirons des modèles de structure pour chaque point de vue. Par ailleurs, l'ontologie de domaine utilisée est une ontologie multi-points de vue. De ce fait nous construirons des classes d'annotation pour chaque point de vue.

En premier temps nous clarifierons la notion de la régularité de structure, sur la quelle se base notre approche, ensuite nous présenterons un schéma général illustrant les différentes étapes et activité de l'approche proposée. Cette dernière, sera détaillée dans les sous-sections qui suivent.



3.1. Notion de la régularité de structure

Dans le cadre de notre approche, nous nous concentrons sur l'organisation (ordre et présence) des éléments de la page et sur les relations entre ces éléments et non pas sur le style. Le code XML propose des balises personnalisées permettant de régler ce problème. Ainsi, à partir du code XML et à cause de la régularité de la structure, nous pouvons construire des modèles de structure et leur associer une sémantique.

Les pages portant sur un domaine donné partagent généralement le même type d'information [Sylvain 07]. Nous prenons comme exemple la page web dans le domaine de l'immobilier (agence immobilière, partie location) qui présente une liste de locataires Cette caractéristique permet de distinguer les pages web pertinentes pour un domaine. Dans le cadre de notre travail, nous supposons que la structure des pages à annoter est faite de manière régulière (i.e. toutes les pages d'immobilier suivent la même structure). Ainsi, toute page d'immobilier a une structure correspond à la description de structure (DTD) suivante :

```
<!DOCTYPE Racine [  
<!ELEMENT Racine (Location) +>  
<!ELEMENT Location (Personne, (Appartement) +)>  
<!ELEMENT Personne (Nom, Salaire)>  
<!ELEMENT Appartement (Adresse, Nbrpiece, Loyer)>  
<!ELEMENT Nom (#PCDATA)>  
<!ELEMENT Salaire (#PCDATA)>  
<!ELEMENT Adresse (#PCDATA)>  
<!ELEMENT Nbrpiece (#PCDATA)>  
<!ELEMENT Loyer (#PCDATA)>]>
```

3.2. Schéma illustre les différentes étapes et activités de l'approche proposée

Avant de détailler les différentes étapes et activités à suivre pour annoter une page web à partir d'une ontologie multi-points de vue d'un domaine donné, nous proposons d'illustrer l'enchaînement des étapes de notre approche, et l'enchaînement des activités de chaque étape, ainsi que les différentes ressources qui sont utilisées et produites.

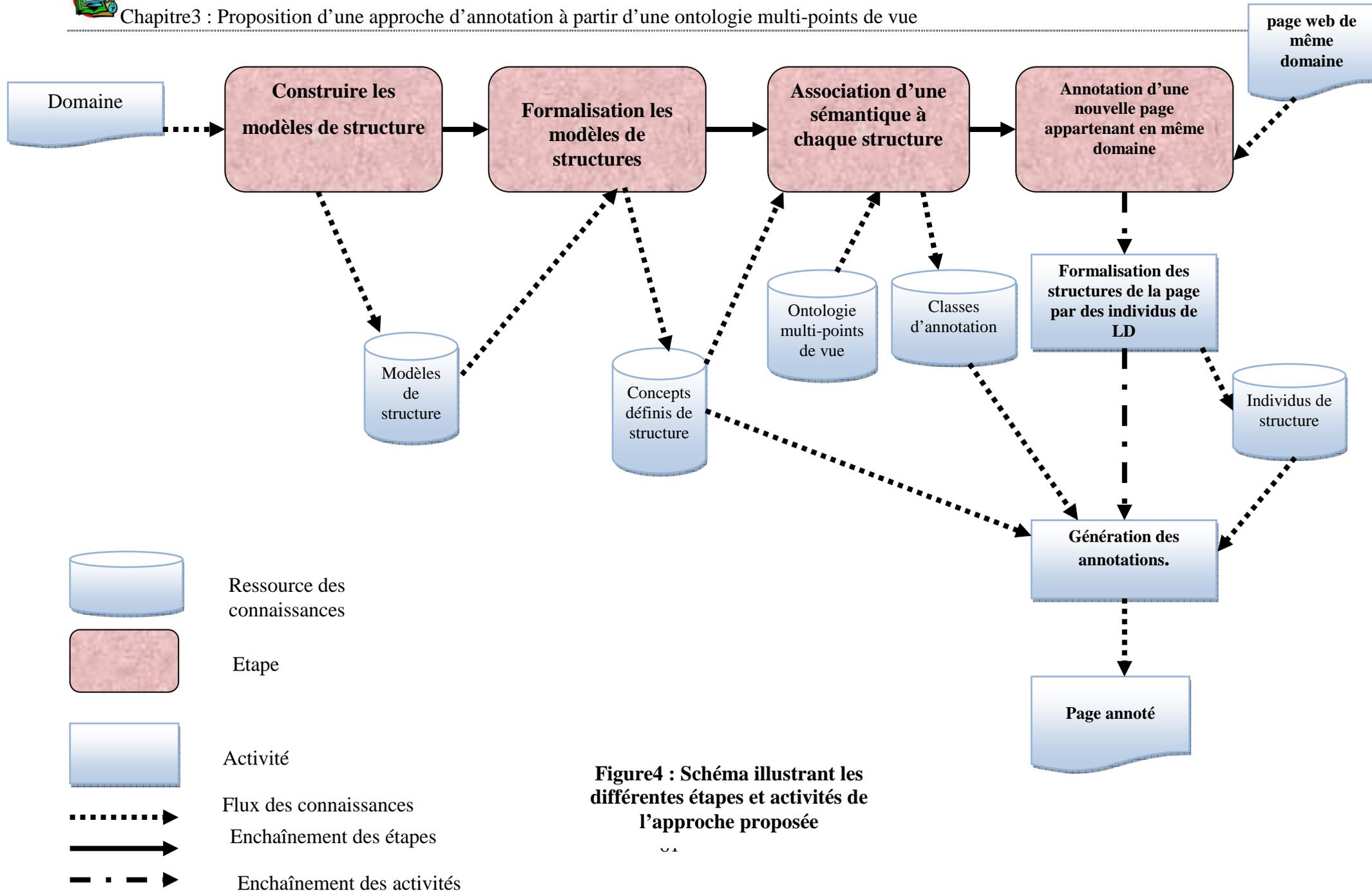


Figure4 : Schéma illustrant les différentes étapes et activités de l'approche proposée



3.3. Description de l'approche proposée

Pour annoter une page nous passons par les étapes suivantes :

3.3.1. Construire les modèles de structure

Cette première étape, consiste à faire sortir des modèles de structure pour une page appartenant à un domaine donné, pour chaque point de vue afin d'illustrer les relations qui existent entre les éléments de la page. Ceci, dans le but de construire des concepts de structure pour la page après la formalisation de ces modèles en logique de descriptions.

Si toutes les pages d'un domaine donné ont la même structure (régularité de structure), on peut dire que l'annotation pour la page d'un domaine donné doit définir que la partie de la page entre une balise ouvrante (`<balise>`) et une balise fermante (`</balise>`) décrit un concept de ce domaine. Prenons l'exemple de l'extrait de code XML de la page d'immobilier (Figure 2). Un modèle d'annotation pour cette page doit définir que la partie de la page entre la balise `< Location >` et `</ Location>` décrit un locataire. Le modèle doit également définir comment chaque élément définissant Locataire est modélisé, à son tour, à l'intérieur de la structure. Récursivement, des modèles sont définis pour l'identification de sous-structures identifiant l'appartement et la personne. Donc toutes les pages de ce domaine vont être annotées de la même façon.

Cette modélisation peut être effectuée grâce à la structure arborescente des pages web définie par le Document Object Model du XML (DOMXML) [Apparo 98].

Il est à noter que pour un point de vue donné, la page ne va pas être annotée entièrement, car, certains éléments sont pertinents dans ce point de vue et d'autres non. Pour cela, nous proposons de filtrer la page de telle manière qu'il ne reste que les éléments qui concernent ce point de vue.

On obtient alors, une page propre à chaque point de vue. Le nombre des pages obtenus égal le nombre de points de vue impliqués, plus une pour le point de vue global. Selon ce dernier, uniquement les éléments de la page originale, qui sont visibles et partagés



Chapitre3 : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

par tous les points de vue, sont considérés. Le problème est que nous ne pouvons pas modifier la page originale.

Ainsi, DOM dispose de plusieurs méthodes permettant de manipuler les fichiers XML, tel que la méthode *removeChild* qui permet de supprimer un noeud cible. Le noeud ne sera pas effacé dans le fichier lui-même (i.e. page originale), mais uniquement dans l'arbre XML présent en mémoire [Apparo 98].

De ce fait, à partir de l'arbre DOMXML original et grâce à la méthode «*removeChild*» qui nous permet d'extraire les noeuds présentant les informations qui ne sont pas visible dans un point de vue particulier, nous générons un sous arbre pour chaque point de vue. Nous obtenons alors, une structure propre à chaque point de vue au lieu d'une page pour chaque point de vue. Notons que les annotateurs sont les spécialistes du domaine. Chaque spécialiste connaît tous les concepts concernant sa spécialité (i.e. son point de vue). La tâche d'extraction des noeuds de l'arbre se fera donc par leur intermédiaire.

Une fois les sous-arbres DOM sont obtenus, ces derniers sont alors transformés en des arbres binaires car, chaque arbre est ordonné, chaque arête représente une relation parent/fils entre deux noeuds de même arbre. Par exemple supposant qu'on a l'arbre suivant défini par les noeuds éléments A, B, C, et D :

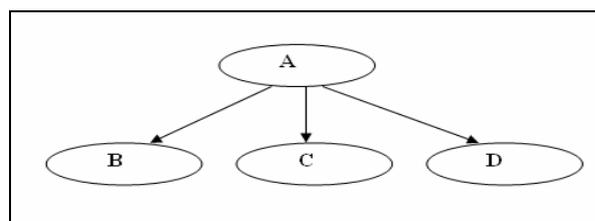


Figure4 : exemple d'un arbre DOM

Selon la figure 4 le noeud A est parent du B, il est aussi parent du C, ainsi que D. B est le fils de A, C est le fils de A, et D est le fils de A.

La transformation en arbre binaire est se faite à travers l'usage des relations suivantes entre les noeuds éléments qui sont typés par les noms des balises :



Chapitre3 : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

• firstChild, écrit fC , identifie le premier fils d'un nœud. C'est une relation binaire, puisque un nœud ne peut avoir qu'un seul premier fils. Par exemple (dans l'arbre figure 4) : $fC(A, B)$.

• nextSibling, écrit nS , identifie le prochain frère d'un nœud. C'est une relation binaire, puisque un nœud ne peut avoir qu'un seul prochain frère. Par exemple : $nS(B, C)$.

• noChild, écrit noC , identifie un nœud sans fils. C'est une relation binaire, puisque cette relation représente la négation d'une relation binaire qui est firstChild. : Non du $(\exists x fC(\text{nœud}, x)) \leftrightarrow \forall x \neg fC(\text{nœud}, x)$. Pour réduire l'écriture nous supposons que $(\forall x \neg fC(\text{nœud}, x)) \leftrightarrow noC(\text{nœud})$. Par exemple : $noC(B)$.

• noSibling, écrit noS , identifie un nœud sans frère. C'est une relation binaire, puisque cette relation représente la négation d'une relation binaire qui est nextSibling. : Non du $(\exists x nS(\text{nœud}, x)) \leftrightarrow \forall x \neg nS(\text{nœud}, x)$. Pour réduire l'écriture nous supposons que $(\forall x \neg nS(\text{nœud}, x)) \leftrightarrow noS(\text{nœud})$. Par exemple : $noS(D)$.

Donc l'arbre obtenu est un arbre binaire ordonné, cet arbre contient deux types des arêtes l'un représente une relation parent/fils entre deux nœuds de même arbre et l'autre représente une relation de frère entre deux nœuds de même arbre.

Par exemple la transformation de l'arbre figure 4 en arbre binaire se déroule comme suit :

Le premier fils de A est B donc on a seulement $fC(A, B)$, tel que $fC(A, C)$, et $fC(A, D)$ n'existe pas. Puis le prochain frère de B c'est C donc on a $nS(B, C)$, et $nS(C, D)$. En suite, on a $noC(B)$, $noC(C)$, $noC(D)$, et enfin $noS(D)$.

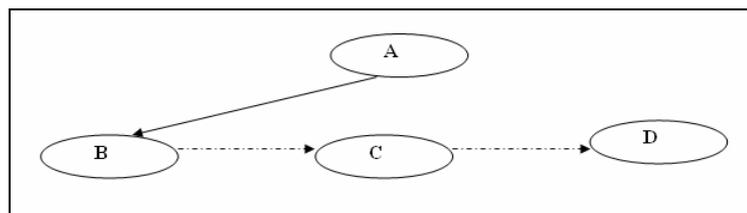


Figure5 : Transformation de l'arbre DOM figure4 en arbre binaire



Il est à noter que les sous arbres binaires obtenus sont appelés des modèles de structure.

3.3.2. Formaliser les modèles de structures

Cette deuxième étape, consiste à formaliser les différents modèles, obtenus dans la première activité en utilisant le formalisme de la logique de descriptions. Pour ce faire, chaque balise XML est représentée par un concept primitif nommé par le nom de la balise. Donc, il existe d'une part, des concepts primitifs communs entre les différents points de vue (les balises qui sont présentes dans tous les sous arbres), et d'autre part, des concepts primitifs propres à chaque point de vue (les balise présentes dans un sous arbre et absentes dans les autres). Par ailleurs, une relation entre deux balises A et B est formalisée par un rôle selon l'une des quatre formes suivantes :

- La relation parent/fils entre deux balises A et B est formalisée par le rôle de LD : fC, tel que $A \sqsubseteq \exists fC.B$,
- La relation " le prochain frère de A est B " est formalisée par le rôle de LD : nS tel que $A \sqsubseteq \exists nS.B$
- noC tel que $A \sqsubseteq noC$
- noS tel que $A \sqsubseteq noS$.

Les relations noC et noS représentent respectivement l'absence de prochain fils et de prochain frère.

Par exemple la formalisation du l'arbre figure 5 est se construit comme suit :

- A, B, C, et D sont des concepts primitifs.
- Le sous arbre avec la racine D (ce sous arbre est un concept défini) un concept défini) est formalisé comme suit : $(D \cap noC \cap noS)$.
- Le sous arbre avec la racine C (ce sous arbre est un concept défini) est formalisé comme suit : $(C \cap \exists nS .(sous\ arbre\ avec\ la\ racine\ D) \cap noC)$.

Donc le sous arbre avec la racine C est formalisé comme suit :

$$(C \cap \exists nS.(D \cap noC \cap noS) \cap noC)$$



Même raisonnement pour que le sous arbre avec la racine B (ce sous arbre est un concept défini) est formalisé comme suit :

$$- (B \cap \exists nS.(C \cap \exists nS.(D \cap noC \cap noS) \cap noC) \cap noC)$$

Et en fin l'arbre figure 5 (cet arbre est un concept défini) est formalisé comme suit :

$$(A \cap fC .(B \cap \exists nS .(C \cap \exists nS .(D \cap noC \cap noS) \cap noC) \cap noC) \cap noS)$$

Remarque \cap signifie « et »

Si par exemple, on ne contraint pas le fait que A ait un frère ou non. Alors l'arbre figure 5 sera formalisé comme suit : $(A \cap fC .(B \cap \exists nS .(C \cap \exists nS .(D \cap noC \cap noS) \cap noC) \cap noC))$. *Cette possibilité de ne pas spécifier certaines contraintes rend cette formalisation adaptée à la formalisation des structures régulières.*

La formalisation des modèles de structure permet d'obtenir, comme résultat, ce que nous appelons des Concepts de Structure (CS). Par exemple la formule $(A \cap fC .(B \cap \exists nS .(C \cap \exists nS .(D \cap noC \cap noS) \cap noC) \cap noC))$ est un concept défini de structure. Donc, on peut écrire par exemple : $S \equiv (A \cap fC .(B \cap \exists nS .(C \cap \exists nS .(D \cap noC \cap noS) \cap noC) \cap noC))$

Il est à noter, que le concept de structure obtenu à partir du point de vue global est plus général que tous les concepts de structure obtenu à partir des points de vue locaux. Ainsi si une page est instance d'un concept de structure obtenu à partir d'un point de vue local, elle est forcément instance de d'un concept de structure obtenu à partir d'un point de vue global.

Les concepts de structure obtenus sont stockés dans l'ontologie de domaine, pour préparer l'étape suivante.

3.3.3. Associer une sémantique à chaque structure

Cette étape, consiste à associer une sémantique prise de domaine à chaque modèle de structure et cela à travers l'usage de l'ontologie multi-points de vue (notre ontologie de domaine). Ainsi, Pour chaque point de vue (global et local), nous définissons des



Chapitre3 : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

classes d'annotation² afin de relier la structure de la page selon le point de vue concerné avec la sémantique des concepts et rôles définis dans l'ontologie O. Une classe d'annotation (CA) pour un point de vue particulier formalise l'annotation d'une page web sous la forme d'un modèle de structure (i.e. concept de structure) CS concernant ce point de vue relié par le rôle « **annotépar** » à un concept défini de domaine C en fonction de l'ontologie de domaine.

Une classe d'annotation (CA) annotant un élément d'une page par une instance d'un concept C de l'ontologie de domaine est définie par l'expression $CA \equiv CS \cap \exists \text{annotépar} .C$. Ainsi, une classe d'annotation (CA_{PVi}) pour un concept de structure CS_{PVi} , défini pour le point de vue PVi , est alors définie comme $CA_{PVi} \equiv CS_{PVi} \cap \exists \text{annotépar} .C$, tel que CS_{PVi} est un concept de structure et C est un concept global ou bien un concept local décrit dans l'ontologie multi-points de vue.

Afin de construire une classe d'annotation pour un point de vue, nous choisissons dans l'ontologie de domaine (et plus précisément dans la représentation concerne ce point de vue) le concept défini³ qui ses contraintes sémantique⁴ correspondent aux contraintes sémantique de concept défini de structure concernant ce point de vue.

Par exemple, si on prend le concept S (vu dans la section précédente), rappelant que :

$S \equiv (A \cap fC .(B \cap \exists nS .(C \cap \exists nS .(D \cap noC \cap noS) \cap noC)) \cap noC)$, alors il faut chercher tout concept défini E dans l'ontologie de la forme : $E \equiv F \cap \exists r1.G \cap \exists r2.H \cap \exists r3.I$ pour obtenir une classe d'annotation⁵ de telle sorte :

Les concepts primitifs de structure **A**, **B**, **C**, et **D** sont reliés respectivement par le rôle « **annotéPar** » aux concepts du domaine **F**, **G**, **H**, et **I** c'est-à-dire :

A \cap **annotépar** **F** existe si et seulement si **B** \cap **annotépar** **G** existe, et **C** \cap **annotépar** **H** existe et **D** \cap **annotépar** **I** existe.

² La construction des classes d'annotation s'effectue manuellement par les annotateurs qui sont spécialistes du domaine.

³ Si on trouve plus de un alors dans ce point de vue nous allons construire plusieurs classes d'annotation

⁴ Contrainte sémantique : Il s'agit de l'ensemble des concepts de l'ontologie appartenant à la définition de ce concept défini. Par exemple, Chercheur est un concept défini car chercheur est une personne qui a au moins un thème et un projet. Donc ces contraintes sémantiques sont : Personne, Thème et Projet.

⁵ On peut normaliser les concepts de domaine de tel sorte tous les concepts est formalisé sous forme des conjonctions des concepts.



Donc :

$S \cap \text{annotépar } .E \equiv (A \cap \text{annotépar } F \cap fC .(B \cap \text{annotépar } G \cap \exists nS .(C \cap \text{annotépar } H \cap \exists nS .(D \cap \text{annotépar } I \cap noC \cap noS) \cap noC)) \cap noC))$

A partir de cette classe :

fC entre A et le sous arbre de Racine B sera annoté par r1.

nS entre B et le sous arbre de la racine C sera annoté par r2.

nS entre C et le sous arbre de la racine D sera annoté par r3.

Donc le choix d'un concept approprié à une structure se fait comme suit :

- Nous faisons une comparaison entre l'arbre DOM de la page et la structure arborescente des concepts de domaine de telle façon :

- Les nœuds parents des nœuds texte vont être reliés à des attributs des concepts de domaine.
- .les nœuds parents des nœuds parents des nœuds texte vont être reliés à des concepts définis par des attributs.
- les nœuds parents des nœuds parents des nœuds parents des nœuds texte vont être reliés à des concepts définis par d'autres concepts. Ainsi de suite jusqu'à la racine.

A la fin, pour un seul point de vue, on peut trouver plusieurs concepts définis du domaine, où tous ces concepts sont appropriés à une seule structure, dans ce cas on obtient alors plusieurs classes d'annotation pour un point de vue.

La différence entre ces classes est les contraintes terminologiques. C'est-à-dire : nous avons vu que les nœuds parents des nœuds texte vont être reliés à des attributs de concepts de domaine, et nous savons que chaque attribut possède une contrainte terminologique, qui se représente par le type des valeurs que cet attribut peut les prendre (par exemple l'attribut nom d'une personne doit être une chaîne de caractère), cette contrainte va être projetée sur le nœud texte.

Pour un point de vue, certaines classes sont plus spécifiques que d'autres classes. Les classes du point de vue global sont plus générales que toutes les classes des points de vue locaux.



3.3.4. Annotation d'une nouvelle page appartenant en même domaine

Annoter une page web appartenant au même domaine traité selon un point de vue donné signifie créer des individus quiinstancient une classe d'annotation et sont reliés par des instances de rôles selon ce point de vue. Étant donnée une nouvelle page à annoter, le processus consiste en un ensemble des activités suivantes :

Activité 1 : Formalisation des structures de la page par des individus de LD :

Dans un premier temps, la page est codée sous la forme d'un arbre binaire à partir de sa représentation DOM. Puis, à partir de ce sous arbre, nous générons un sous arbre pour chaque point de vue.

Afin de classifier une nouvelle page à annoter par rapport aux concepts définis de structure (vus dans l'étape 2), la page est formalisée par des individus de la LD (au lieu des concepts cette fois ci) : Dans chaque sous arbre obtenu, chaque noeud élément (i.e. balise XML) x , est représenté par un individu i_x . Si i_x a un frère (resp. un fils), l'instance du rôle $nS(i_x, i_y)$ (resp. $fC(i_x, i_y)$) est générée. S'il n'a pas de frère (resp. de fils) une instance de noS (resp. noC) lui est associée. C'est-à-dire les balises sont formalisées par des individus des concepts primitifs de structures, et les arbres sont formalisés par des individus des concepts définis de structure. Nous utilisons un raisonneur classifie ces individus comme instances de concepts de structure.

Activité2 : Génération des annotations.

Pour chaque point de vue, nous créons un individu qui instancie une des classes d'annotation concernant ce point de vue. Par exemple, CA est une classe d'annotation dans un point de vue donné, CS le concept défini de structure dans ce point de vue, C un concept de domaine vu dans ce point de vue, et ics est un individu de CS . Rappelant que $CA \equiv CS \cap \text{annotépar} .C$, $(ics \cap \text{annotépar} .C)$ est un individu qui instancie CA si et seulement si l'individu ics peut être annoté par le concept C selon ce point de vue.

Par un raisonneur, nous vérifions que $(ics \cap \text{annotépar} .C)$ est un individu (instance) de CA , si le résultat retourne vrai alors toute la page va être annoté, si non la page ne va pas être annoté. Parmi les classes possibles, le raisonneur va choisir la classe d'annotation la plus spécifique appropriée à cette nouvelle page à annoter.



Chapitre3 : Proposition d'une approche d'annotation à partir d'une ontologie multi-points de vue

Enfin la page est annotée comme suit :

- Seulement les éléments de la page visibles dans le point de vue spécifié sont annotés, les autres non.
- La page possède une annotation globale et plusieurs annotations locales.
- Il peut y avoir des annotations locales identiques à l'annotation globale. Ce cas se produit lorsque on n'arrive pas à annoter par des concepts vus seulement dans le point de vue local.

4. Conclusion

Nous avons présenté une approche adaptée de l'approche présentée dans [Sylvain 07]. Nous avons construit des modèles de la page à partir du DOMXML de la page dans chaque point de vue afin de construire des classes d'annotation. Etant donné une nouvelle page à annoter, nous avons déterminé pour chaque point de vue les éléments de la page concernant ce point de vue afin d'annoter ces éléments par des instances de concepts connus dans ce point de vue. Chaque partie de cette page va être classifiée par rapport à une classe d'annotation dans chaque point de vue puis, les éléments de la page qui sont visibles dans un point de vue vont être annotés par des instances de concepts connus dans ce point de vue. L'intérêt de notre approche est de construire une structure propre pour chaque point de vue illustrant seulement les éléments concernant ce point de vue et les relations entre ces éléments. Ensuite, ces éléments sont annotés par des instances de concepts de domaine connus dans ce point de vue, et les relations entre ces éléments sont annotées par des instances de rôles.

Chapitre IV:

***Etude de cas : Application sur une
page web du domaine d'immobilier***



Chapitre4 : étude de cas : application sur une page web du domaine d'immobilier

1. Introduction

Afin de plus aisément comprendre la façon d'annoter une page web à travers l'approche proposée, le plus simple est d'en annoter une page *étape par étape*. Comme il fallait choisir un domaine, nous avons opté pour l'immobilier comme domaine applicatif. En premier temps, nous asseyons d'expliquer comment développer notre ontologie multi-points de vue concernant le domaine d'immobilier, pour ce la nous travaillant sur l'ontologie présenté dans [Hemam 08], [Hemam 09a], et [Hemam 09b] en ne considérant que deux point de vue à savoir : « taille » et « financier », et nous présentons seulement un sous ensemble de cette ontologie. Nous montrons ainsi comment l'ontologie a été conceptualisée puis formalisée et traduite dans le langage OWL en utilisant l'outil Protégé-2000.

Puis à partir d'un modèle de la page d'immobilier, nous construirons trois représentations : une pour le point de vue global, une deuxième pour le point de vue taille, et enfin une pour le point de vue financier. Ensuite nous générons pour chaque point de vue des classes d'annotation. Par la suite nous commencerons le processus d'annotation de la page web, en appliquant les différent étapes de l'approche proposée. Enfin nous présenterons quelques aspects d'implémentation de cette approche.

2. Construction de l'ontologie multi-points de vue

L'ontologie multi-points de vue est composée d'une représentation globale, et deux représentations locales : « taille » et financier ». Par ailleurs, cette ontologie est formalisée par la logique de description comme dans [Hemam 09a].

2.1. La spécification et la conceptualisation

On commence d'abord par la conceptualisation de la représentation globale. Ceci, en représentant les connaissances vues dans tous les points de vue. Puis à partir de la conceptualisation globale, nous générons un modèle conceptuel, pour chaque point de



vue, et cela par un raffinement de la conceptualisation globale et aussi par l'ajout d'autres connaissances concernant le point de vue en question.

Pour la représentation globale :

Glossaire global de termes :

Les connaissances ontologiques représentées à ce niveau sont dites globales (les concepts globaux, les rôles globaux, et les individus globaux) et sont supposés être visible dans tous les points de vue. A titre d'exemple, nous pouvons citer dans le point de vue global les concepts suivants :

- Un appartement est défini par l'attribut Adresse. Cet attribut constitue la clé du concept appartement car il est visible dans tous les points de vue. La valeur de l'attribut Adresse est une chaîne de caractère.
- Un locataire est une personne qui loue au moins un appartement.

Les rôles globaux sont :

Loue (Locataire, Appartement)

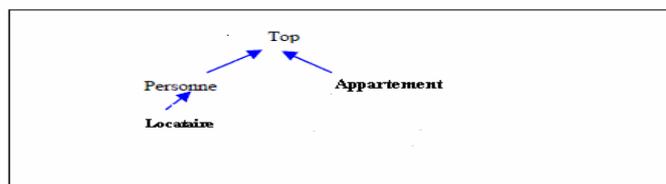
Loué_par (Appartement, Locataire)

ensemble_de (Locataire, Locataire)

Hierarchie globale des concepts :

Par ailleurs, un concept universel « Top », qui généralise tous les concepts racines, est utilisé pour former une seule hiérarchie, afin d'éviter d'avoir des concepts isolés. Une hiérarchie des concepts selon la représentation globale est la suivante :

Hiérarchie globale de concepts selon le point de vue « global »



Pour les représentations locales :

Les connaissances représentées selon un point de vue particulière sont dites locales (comme exemple les concepts locaux, les rôles locaux, et les individus locaux) qui sont visible dans ce point de vue. Parmi ces connaissances nous trouvons les connaissances globales.



Selon le point de vue « Taille » :

Glossaire local de termes selon le point de vue « Taille » :

Nous citons dans le point de vue « taille » les concepts suivants :

- Un appartement est défini par les attributs : Adresse, et Nbr_pièce. La valeur de l'attribut Nbr_pièce est un nombre. Donc l'attribut Nbr_pièce est visible uniquement dans le point de vue « taille ».
- Une Personne est défini par l'attribut Nom.
- Un locataire c'est une personne qui loue au moins un appartement.
- une Liste_locataire est un ensemble des locataires.
- Une Liste_locataire est représentée une agence.

Ainsi les concepts locaux concernent uniquement le point de vue « taille » : Petit_Appartement, Grand_Appartement, Studio, F1, F2, F3....

- Un petit appartement est un appartement dont le nombre de pièce appartient à l'ensemble {1,2}.
- Un grand appartement est un appartement dont le nombre de pièce est supérieurs ou égale à 3.
- Un studio est un appartement F1 (synonyme de F1).

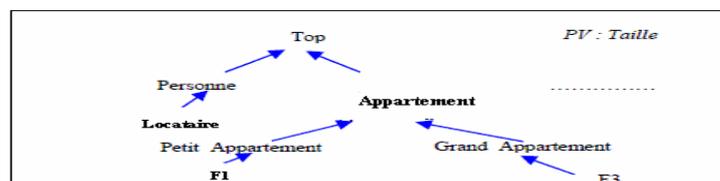
Les rôles locaux sont :

Loue (Locataire, Grand_Appartement)

Loue (Locataire, Petit_Appartement)

Hierarchie globale des concepts selon le point de vue « Taille » :

Hierarchie locale de concepts selon le point de vue « Taille »



Selon le point de vue « Financier » :

Glossaire local de termes selon le point de vue « Financier » :

Nous pouvons recueillir les concepts suivants :

- Un appartement est défini par ses attributs : Adresse, et Loyer.



La valeur de l'attribut Loyer est de type monétaire. Ce dernier est un attribut visible dans le point de vue « financier ».

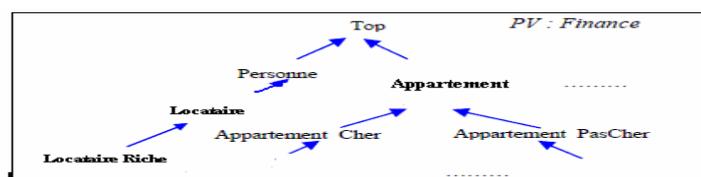
- Une Personne est défini par l'attribut *Nom*, sa valeur est une chaîne de caractère et par l'attribut *Salaire* dont valeur est de type monétaire. Ce dernier est un attribut visible dans le point de vue « financier ».
- Un appartement cher est un appartement dont le loyer est supérieur ou égale à 2100DA
- Un appartement pas cher est un appartement dont loyer est inférieur ou égale à 2099DA
- Locataire Riche est un locataire dont son salaire est ≥ 25000 DA
- Locataire pas Riche est un locataire dont son salaire est < 25000 DA
- une Liste_locataire est un ensemble des locataires.
- Une Liste_locataire est représente une agence.

Les rôles locaux sont :

- Loue (Locataire_riche, Appartement_cher)
- Loue (Locataire_riche, Appartement_pascher)
- Loue (Locataire_pasriche, Appartement_cher)
- Loue (Locataire_pasriche, Appartement_pascher)

Hierarchie globale des concepts selon le point de vue « Financier » :

Hiérarchie locale de concepts selon le point de vue « Finance »



Donc, on peut citer les rôles globaux qui lient deux concepts définis dans deux points de vue locaux différents comme suivant :

- Loue (Locataire_riche, Grand_Appartement)
- Loue (Locataire_pasriche, Grand_Appartement)
- Loue (Locataire_riche, Petit_Appartement)
- Loue (Locataire_pasriche, Petit_Appartement)
- Loué_par est le rôle inversé du rôle loue



Les passerelles :

Les passerelles sont les relations entre deux concepts définis dans deux points de vue différent. Dans notre exemple du domaine de l'immobilier on peut avoir les passerelles d'inclusions unidirectionnelles seulement qui sont:

- tous les Grand_Appartement sont des Appartement_cher
- tous les Petit_Appartement sont des Appartement_pascher

2.2. La formalisation

Soit $Vp = \{vp1, vp2\}$ ensemble des noms des points de vues respectivement « Taille », et « Financier ». L'ontologie $O, O = \langle C^G, R^G, Vp, M \rangle$

C^G est l'ensemble des concepts globaux, tel que : $C^G = \{ Appartement^{\hat{}} Personne^{\hat{}} Locataire^{\hat{}} Liste_locataire^{\hat{}} \}$

R^G est l'ensemble des rôles globaux tel que : $R^G = \{ Loue^{\hat{}} Ensemble-de^{\hat{}} Loué_par^{\hat{}} \}$

Les concepts sont formalisés comme suit :

$Appartement^{\hat{}} \equiv (\forall vp1, vp2 \text{ adresse .chaîne de caractère}) \cap (= vp1, vp2 \text{ 1 adresse}) \cap (\forall vp1, \geq 0 \text{ nbr_pièce .Nombre}) \cap (\forall vp2, \geq 0 \text{ loyer .Monétaire})$

De tel façon $(\forall vp1, \geq 0 \text{ nbr_pièce .Nombre}) \cap (\forall vp2, \geq 0 \text{ loyer .Monétaire})$ dans le point de vue global sont toujours valides car ces attribut (nbr_pièce et loyer) ne sont pas vus dans le point de vue global, donc les cardinalités des ces attribut toujours 0, alors :

$Appartement^{\hat{}} \equiv (\forall vp1, vp2 \text{ adresse .chaîne de caractère}) \cap (= vp1, vp2 \text{ 1 adresse}) \cap true \cap true \equiv (\forall vp1, vp2 \text{ adresse .chaîne de caractère}) \cap (= vp1, vp2 \text{ 1 adresse})$

Donc $Appartement^{\hat{}}$ selon le point de vue global s'écrit : $Appartement^{\hat{}} \equiv (\forall vp1, vp2 = 1 \text{ adresse .chaîne de caractère})$

Même raisonnement pour dire que : $Appartement^{\hat{}}$ selon le point de vue Taille s'écrit : $Appartement^{\hat{}} \equiv (\forall vp1, vp2 = 1 \text{ adresse .chaîne de caractère}) \cap (\forall vp1, \geq 0 \text{ nbr_pièce .Nombre})$.

Selon le point de vue Financier s'écrit : $Appartement^{\hat{}} \equiv (\forall vp1, vp2 = 1 \text{ adresse .chaîne de caractère}) \cap (\forall vp2, \geq 0 \text{ loyer .Monétaire})$



$Personne^{\hat{o}} \equiv (\forall vp1, vp2 \text{ nom .chaîne de caractère}) \cap (= vp1, vp2 \neq 1 \text{ nom}) \cap (\forall vp2, \geq 0 \text{ salaire .Monétaire})$

$Personne^{\hat{o}}$ selon le point de vue global s'écrit : $Personne^{\hat{o}} \equiv (\forall vp1, vp2 = 1 \text{ nom .chaîne de caractère})$

Selon le point de vue Taille s'écrit : $Personne^{\hat{o}} \equiv (\forall vp1, vp2 = 1 \text{ nom .chaîne de caractère})$

Selon le point de vue Financier s'écrit : $Personne^{\hat{o}} \equiv (\forall vp1, vp2 = 1 \text{ nom .chaîne de caractère}) \cap (\forall vp2, \geq 0 \text{ salaire .Monétaire})$

$Locataire^{\hat{o}} \equiv Personne^{\hat{o}} \cap \exists \text{ loue .Appartement}^{\hat{o}}$

$Liste_Locataire^{\hat{o}} \equiv (\forall \text{ensemble_de .Locataire}^{\hat{o}})$

$Liste_Locataire^{\hat{o}} \equiv \text{Agence.}$

$vp1 : \text{Petit_Appartement} \equiv \text{Appartement}^{\hat{o}} \cap \text{nbr_pièce} .\{1,2\}$

$vp1 : \text{Grand_Appartement} \equiv \text{Appartement}^{\hat{o}} \cap \text{nbr_pièce} .\{3 \infty\}$

$vp1 : F1 \equiv \text{Appartement}^{\hat{o}} \cap \text{nbr_pièce} .\{1\}$

$vp1 : F2 \equiv \text{Appartement}^{\hat{o}} \cap \text{nbr_pièce} .\{2\}$

$vp1 : F3 \equiv \text{Appartement}^{\hat{o}} \cap \text{nbr_pièce} .\{3\}$

$vp1 : F4 \equiv \text{Appartement}^{\hat{o}} \cap \text{nbr_pièce} .\{4\}$

$vp1 : F5 \equiv \text{Appartement}^{\hat{o}} \cap \text{nbr_pièce} .\{5\}$

...

$vp2 : \text{Appartement_cher} \equiv \text{Appartement}^{\hat{o}} \cap \text{loyer} .[- \infty 115DA]$

$vp2 : \text{Appartement_pascher} \equiv \text{Appartement}^{\hat{o}} \cap \text{loyer} .[116DA \infty]$

$vp2 : \text{Locataire_riche} \equiv \text{Locataire}^{\hat{o}} \cap \text{salaire} .[25000 DA \infty]$

$vp2 : \text{Locataire_pasriche} \equiv \text{Locataire}^{\hat{o}} \cap \text{salaire} . [-\infty 25000 DA]$

Les rôles sont formalisés comme suit :

$\text{Loue}^{\hat{o}} (\text{Personne}^{\hat{o}}, \text{Appartement}^{\hat{o}})$

$\text{Loué_par}^{\hat{o}} (\text{Appartement}^{\hat{o}}, \text{Personne}^{\hat{o}})$

$\text{ensemble_de}^{\hat{o}} (\text{Locataire}^{\hat{o}}, \text{Locataire}^{\hat{o}})$

$vp1 : \text{loué_par} (\text{Petit_Appartement}, \text{Locataire})$

$vp1 : \text{loué_par} (\text{Grand_Appartement}, \text{Locataire})$

...



vp2 : loué_par (Appartement_pascher , Locataire_pasriche)

vp2 : loué_par (Appartement_cher , Locataire_riche)

vp2 : loué_par (Appartement_cher , Locataire_pasriche)

vp2 : loué_par (Appartement_pascher , Locataire_riche)

...

Loué_par^ô (vp1 :Grand_Appartement, vp2 :Locataire_riche)

Loué_par^ô (vp1 :Grand_Appartement, vp2 :Locataire_pasriche)

Loué_par^ô (vp1 :Petit_Appartement, vp2 :Locataire_riche)

Loué_par^ô (vp1 :Petit_Appartement ,vp2 :Locataire_pasriche)

...

Les passerelles sont formalisées comme suit :

M est l'ensemble des passerelles tel que : $M = \{$

$vp1: Grand_Appartement \xrightarrow{\subseteq} vp2 : Appartement_cher$

$vp1: Petit_Appartement \xrightarrow{\subseteq} vp2 : Appartement_pascher \}$

L'instanciation de l'ontologie se fait comme suit :

$vp1: Petit_Appartement (chez-ali)$

$vp2: Appartement_pascher (chez-ali)$

L'individu *chez-ali* est une instance de *petit_Appartement* dans $vp1$ et elle est une instance de *Appartement_pascher* dans $vp2$.

...

2.3. Codification :

Pour implémenter l'ontologie, il faut la formuler dans un langage de représentation de connaissance tel que OWL. Cette implémentation est réalisée par l'outil Protégé et validé par le raisonneur RACER.

Le concept est défini si on ajoute des axiomes dans ces conditions nécessaires et suffisantes. Ces axiomes sont utilisés comme des règles d'inférence lors du raisonnement.

Les points de vue sont représentés par des préfixes par exemple le concept local F1 défini dans le point de vue « taille » est représenté dans protégé comme suit : « taille : F1 ».



Les connaissances définies dans deux préfixes et qui sont équivalents sont des connaissances globales. Par exemple le concept Personne est défini comme suit : « taille : Personne » et « financier : Personne » les concepts « taille : Personne \equiv financier : Personne ».

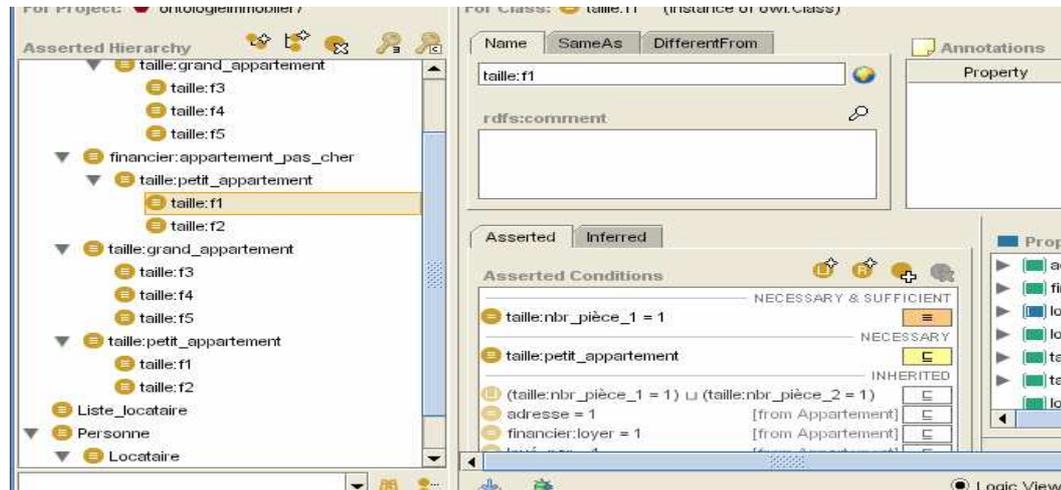


Figure1 Représentation de la notion de point de vue

La hiérarchie de classes (ou taxinomie) se présente dans Protégé sous la forme d'un arbre, ce qui correspond globalement à ce que nous avons spécifié en Logique de descriptions au niveau terminologique.

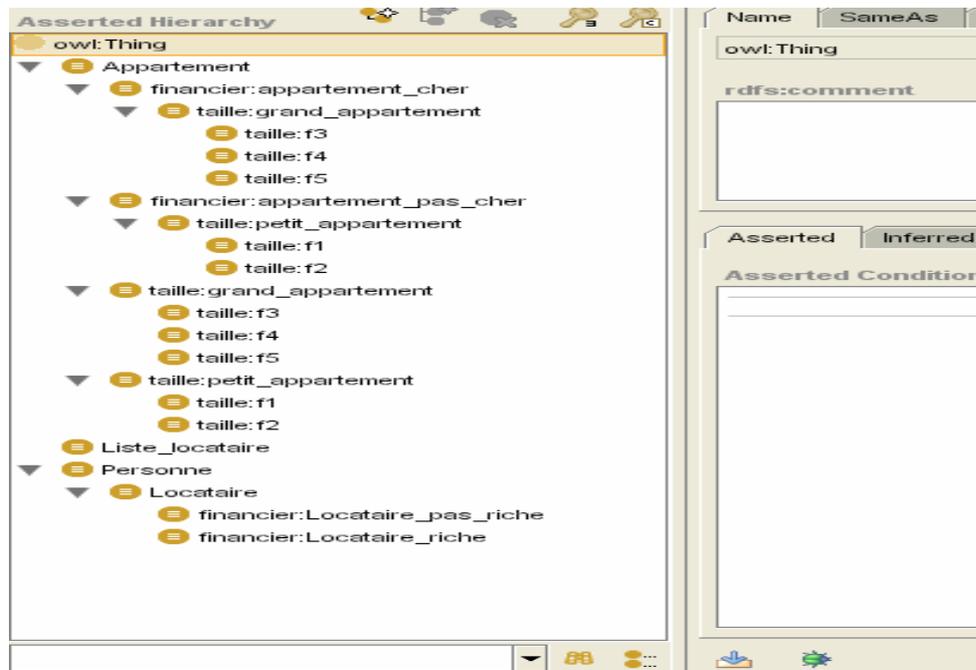


Figure2 hiérarchie de classes



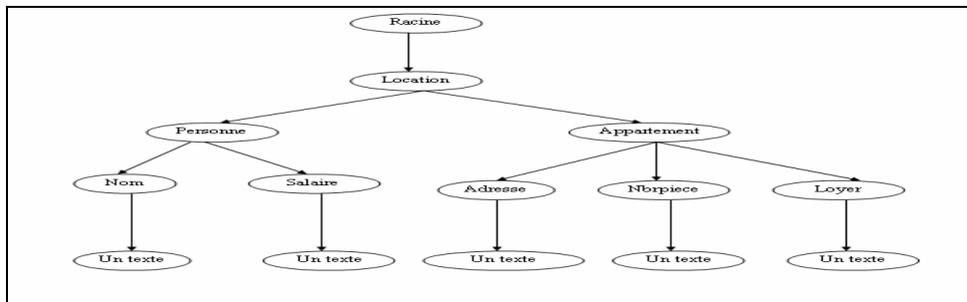
3. Application de l'approche sur le domaine d'immobilier

3.1. Construire les modèles de structure

Toutes les pages d'immobilier ont la même structure, et même informations. Toute page d'immobilier a une structure correspond à la description de structure (DTD) suivante :

```
<!DOCTYPE Racine [  
<!ELEMENT Racine (Location) +>  
<!ELEMENT Location (Personne, (Appartement) +)>  
<!ELEMENT Personne (Nom, Salaire)>  
<!ELEMENT Appartement (Adresse, Nbrpiece, Loyer)>  
<!ELEMENT Nom (#PCDATA)>  
<!ELEMENT Salaire (#PCDATA)>  
<!ELEMENT Adresse (#PCDATA)>  
<!ELEMENT Nbrpiece (#PCDATA)>  
<!ELEMENT Loyer (#PCDATA)>]>
```

Donc, nous pouvons définir un modèle de structure. Ce modèle est représenté par l'arbre DOMXML suivant (figure 4) :



**Figure4 : représentation de la structure de la page « Immobilier » en
Arbre DOMXML**

Nous considérons deux points de vue : *Taille et Financier*. Ainsi, nous obtenons les arbres DOMXML suivants pour une page de ce domaine d'immobilier :

Dans le point de vue Taille, un appartement est identifié par une adresse, et peut avoir un nombre de pièces. Une personne est identifiée par un nom. À partir de l'arbre original figure4, nous appliquons la méthode *removeChild* sur les nœuds : appartement pour extraire les fils *Loyer*, et sur les nœuds personne pour extraire les fils *salaire*. On obtient alors le sous arbre suivant:

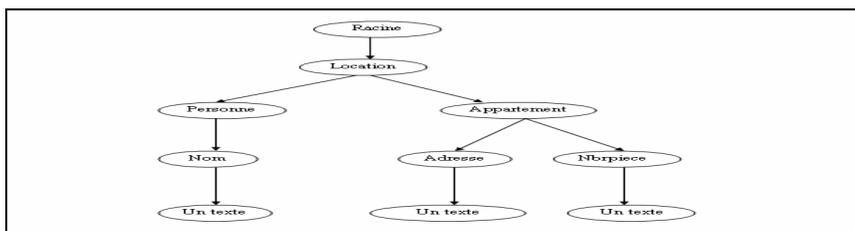


Figure5: présentation de la structure de la page « Immobilier » en Arbre DOMXML selon le point de vue Taille

De la même façon, nous enlèverons les connaissances inutiles dans le point de vue Financier : nous appliquons la méthode *removeChild* sur les nœuds : appartement pour extraire les fils *Nbrpiece*. Et on obtient alors le sous arbre suivant:

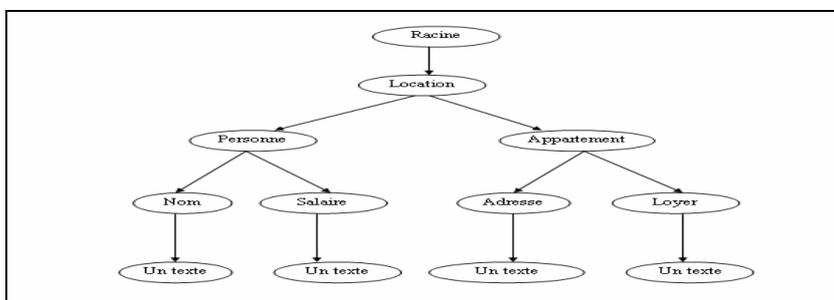


Figure6: présentation de la structure de la page « Immobilier » en Arbre DOMXML selon le point de vue Financier

Par ailleurs, dans le point de vue global, nous gardons uniquement les éléments qui sont visibles et partagés par les deux points de vue : nous appliquons la méthode *removeChild* sur les nœuds : appartement pour extraire les fils *Nbrpiece*, et *Loyer*, et sur les nœuds personne pour extraire les fils *salaire*. On obtient alors le sous arbre suivant :

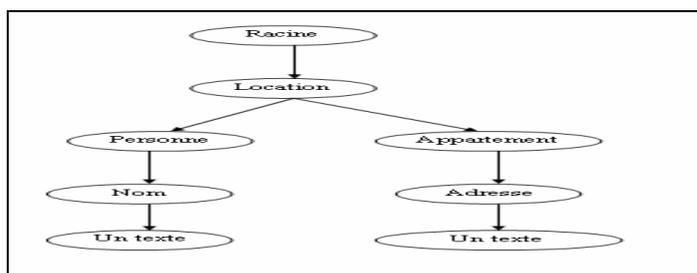


Figure7: présentation de la structure de la page « Immobilier » en Arbre DOMXML selon le point de vue Global

Ainsi, le sous-arbre de l'exemple figure 5 transformé en arbre binaire comme suit :

fC(Racine, Location) noS(Racine) noS(Location) fC(Location, Personne) nS(Personne, Appartement) noS(Appartement) fC(Personne, Nom) noC(Nom) noS(Nom)



fC (Appartement, Adresse) nS(Adresse, Nbrpiece) noC(Nbrpiece) noS(Nbrpiece)

Ainsi, le sous-arbre de l'exemple figure6 transformé en arbre binaire comme suit :

fC(Racine, Location) noS(Racine) noS(Location) fC(Location, Personne) nS(Personne, Appartement) noS(Appartement)
fC(Personne, Nom) nS(Nom, Salaire) noC(Salaire) noS(Salaire)
fC (Appartement, Adresse) nS(Adresse, Loyer) noC(Loyer) noS(Loyer)

Ainsi, le sous-arbre de l'exemple figure7 transformé en arbre binaire comme suit :

fC(Racine, Location) noS(Racine) noS(Location) fC(Location, Personne) nS(Personne, Appartement) noS(Appartement)
fC(Personne, Nom) noC(Nom) noS(Nom)
fC (Appartement, Adresse) noC(Adresse) noS(Adresse)

3.2. Formaliser les modèles de structures

Nous savons que XML impose que le document ne contient qu'une seule racine, donc la contrainte **noS(Racine)** peut être supprimée. Aussi d'après le DTD de la page d'immobilier, une Racine contient un ou plusieurs fils Location, donc la contrainte **noS(Location)** peut être supprimée. Ainsi que Location peut avoir un seul fils Personne et un ou plusieurs fils Appartement, donc la contrainte **noS(Appartement)** peut être supprimé.

De ce fait, pour le domaine d'immobilier (partie location) les concepts définis de la structure sont :

Selon le point de vue taille :

SPVTaille = Racine \cap \exists fC(Location \cap \exists fC(Personne \cap \exists fC (Nom \cap \exists noC \cap \exists noS) \cap \exists nS(Appartement \cap \exists fC (Adresse \cap \exists nS (Nbrpiece \cap \exists noC \cap \exists noS) \cap \exists noC)))))

Selon le point de vue financier :

SPVFinancier = Racine \cap \exists fC(Location \cap \exists fC(Personne \cap \exists fC (Nom \cap \exists nS (Salaire \cap \exists noC \cap \exists noS) \cap \exists noC) \cap \exists nS(Appartement \cap \exists fC (Adresse \cap \exists nS (Loyer \cap \exists noC \cap \exists noS) \cap \exists noC)))))

Selon le point de vue global :

SPVGlobal = Racine \cap \exists fC(Location \cap \exists fC(Personne \cap \exists fC (Nom \cap \exists noC \cap \exists noS) \cap \exists nS(Appartement \cap \exists fC (Adresse \cap \exists noC \cap \exists noS)))))



3.3. Associer une sémantique à chaque structure

Nous allons définir des classes d'annotation pour chaque point de vue.

Définition de classes d'annotation pour le point de vue global :

Dans ce point de vue, nous avons deux attributs qui sont : l'attribut ($\forall vp1, vp2 \text{ nom .chaîne de caractère}$) du concept \hat{O} Personne, et l'attribut ($\forall vp1, vp2 \text{ adresse .chaîne de caractère}$) du concept \hat{O} Appartement, et nous avons d'après l'arbre DOM figure 7 deux nœuds parents des nœuds textes qui sont Nom, et adresse. Donc :

-Le concept de structure Nom (la balise nom) peut être relié par le rôle « annotépar » à l'attribut ($\forall vp1, vp2 \text{ nom .chaîne de caractère}$) et aussi à l'attribut ($\forall vp1, vp2 \text{ adresse .chaîne de caractère}$).

-Le concept de structure Adresse (la balise adresse) peut être relié par le rôle « annotépar » à l'attribut ($\forall vp1, vp2 \text{ nom .chaîne de caractère}$) et aussi à l'attribut ($\forall vp1, vp2 \text{ adresse .chaîne de caractère}$).

Par conséquence :

-Le concept de structure Personne (la balise Personne) peut être relié par le rôle « annotépar » au concept \hat{O} Personne, ou \hat{O} Appartement.

-Le concept de structure Appartement (la balise Appartement) peut être relié par le rôle « annotépar » au concept \hat{O} Personne, ou \hat{O} Appartement.

Le concept de structure Location (balise Location) est relié par le rôle « annotépar » au concept \hat{O} Locataire si et seulement si le concept de structure Personne (la balise Personne) est relié par le rôle « annotépar » au concept \hat{O} Personne, **et** Le concept de structure Appartement (la balise Appartement) peut être relié par le rôle « annotépar » au concept \hat{O} Appartement.

Donc nous gardons seulement :

- le concept de structure Personne (la balise Personne) est relié par le rôle « annotépar » au concept \hat{O} Personne par conséquence : Le concept de structure Nom (la balise nom) est relié par le rôle « annotépar » à l'attribut ($\forall vp1, vp2 \text{ nom .chaîne de caractère}$)

-Le concept de structure Appartement (la balise Appartement) est relié par le rôle « annotépar » au concept \hat{O} Appartement par conséquence : Le concept de structure Adresse (la balise adresse) est relié par le rôle « annotépar » à l'attribut($\forall vp1, vp2 \text{ adresse .chaîne de caractère}$).



- Le concept de structure Location (balise Location) est relié par le rôle « annotépar » au concept Locataire ^o

En fin Le concept de structure Racine (balise Racine) relié par le rôle « annotépar » au concept Liste_Locataire ^o

Une Classe d'annotation CA_{PVG} pour le point de vue global crée par le concept défini de structure **SPVGlobal** avec le concept global Liste_locataire ^o, cette classe intègre la sémantique de l'ontologie comme suit :

$$CA_{PVG} \equiv \text{Racine} \cap \exists \text{annotépar} . \text{Liste_Locataire}^{\circ} \cap \exists fC (\text{Location} \cap \exists \text{annotépar} . \text{Locataire}^{\circ} \cap \exists fC (\text{Personne} \cap \exists \text{annotépar} . \text{Personne}^{\circ} \cap \exists fC (\text{Nom} \cap \exists \text{annotépar} . (\forall vp1, vp2 \text{ nom} . \text{Chaîne de caractère}) \cap \exists noC \cap \exists noS) \exists nS \text{ Appartement} \cap \exists \text{annotépar} . \text{Appartemant}^{\circ} \cap \exists fC (\text{Adresse} \cap \exists \text{annotépar} . (\forall vp1, vp2 \text{ adresse} . \text{Chaîne de caractère}) \cap \exists noC \cap \exists noS)))$$

Cette classe exige que les nœuds textes fils des nœuds Nom et Adresse doivent être une chaîne de caractère.

La relation entre $\exists nS$ entre les concepts de structure Personne, et Appartement est annotée par loue.

La relation $\exists fC$ entre Racine et Location est annotée par ensemble_de.

Définition de classes d'annotation pour le point de vue local « taille » :

De la même façon Une Classe d'annotation CA_{PVTaille} pour le point de vue taille crée par le concept défini de structure **SPVTaille** avec le concept global Liste_locataire ^o, cette classe intègre la sémantique de l'ontologie comme suit :

$$CA_{PVTaille} \equiv \text{Racine} \cap \exists \text{annotépar} . \text{Liste_locataire}^{\circ} \cap \exists fC (\text{Location} \cap \exists \text{annotépar} . \text{Locataire}^{\circ} \cap \exists fC (\text{Personne} \cap \exists \text{annotépar} . \text{Personne}^{\circ} \cap \exists fC (\text{Nom} \cap \exists \text{annotépar} . (\forall vp1, vp2 \text{ nom} . \text{Chaîne de caractère}) \cap \exists noC \cap \exists noS) \exists nS \text{ Appartement} \cap \exists \text{annotépar} . \text{Appartemant}^{\circ} \cap \exists fC (\text{Adresse} \cap \exists \text{annotépar} . (\forall vp1, vp2 \text{ adresse} . \text{Chaîne de caractère}) \cap \exists nS (\text{Nbrpiece} \cap \exists \text{annotépar} . (\forall vp1, \geq 0 \text{ nbr_pièce} . \text{Nombre}) \cap \exists noC \cap \exists noS))))$$

Cette classe exige que les nœuds textes fils des nœuds Nom, et Adresse doive être chaîne de caractère.

Ici pour que Le concept de structure Appartement (la balise Appartement) peut être relié par le rôle « annotépar » au concept Appartement ^o, il suffit que Le concept de structure



Adresse (la balise adresse) est relié par le rôle « annotépar » à l'attribut ($\forall vp1, vp2$ adresse .chaîne de caractère).mais si le concept de structure Nbrpiece est relié par le rôle « annotépar » au attribut .($\forall vp1, \geq 0$ nbr_piéce .Nombre), alors cette classe exige aussi que le noud texte fils du nœud Nbrpiece doit être un entier positif.

Car Le concept de structure Appartement (la balise Appartement) peut être relié par le rôle « annotépar » aussi aux concepts Taille :Grand-Appartement, Petit_Appartemant, F1, F2, F3, F4, et F5, on a aussi les classes d'annotation :

(Appartement \cap \exists annotépar .Grand_Appartemant) \equiv Appartement \cap \exists annotépar .Grand_Appartemant \cap \exists fC (Adresse \cap \exists annotépar.($\forall vp1, vp2$ adresse. Chaîne de caractère) \cap \exists nS (Nbrpiece \cap \exists annotépar.($\forall vp1, \geq 0$ nbr_piéce .[3 ∞]) \cap \exists noC \cap \exists noS))

Cette classe exige aussi que le nœud texte fils du nœud Nbrpiece doive être un entier supérieur à 3.

(Appartement \cap \exists annotépar .Petit_Appartemant) \equiv Appartement \cap \exists annotépar .Petit_Appartemant \cap \exists fC (Adresse \cap \exists annotépar.($\forall vp1, vp2$ adresse. Chaîne de caractère) \cap \exists nS (Nbrpiece \cap \exists annotépar.($\forall vp1, \geq 0$ nbr_piéce .{1,2}) \cap \exists noC \cap \exists noS))

Cette classe exige aussi que le nœud texte fils du nœud Nbrpiece doive être un entier inférieur à 3.

(Appartement \cap \exists annotépar .F1) \equiv Appartement \cap \exists annotépar .F1 \cap \exists fC (Adresse \cap \exists annotépar.($\forall vp1, vp2$ adresse. Chaîne de caractère) \cap \exists nS (Nbrpiece \cap \exists annotépar.($\forall vp1, \geq 0$ nbr_piéce .{1}) \cap \exists noC \cap \exists noS))

Cette classe exige aussi que le nœud texte fils du nœud Nbrpiece doive être un entier égal à 1.

(Appartement \cap \exists annotépar .F2) \equiv Appartement \cap \exists annotépar .F2 \cap \exists fC (Adresse \cap \exists annotépar.($\forall vp1, vp2$ adresse. Chaîne de caractère) \cap \exists nS (Nbrpiece \cap \exists annotépar.($\forall vp1, \geq 0$ nbr_piéce .{2}) \cap \exists noC \cap \exists noS))



Cette classe exige aussi que le noud texte fils du nœud Nbrpiece doive être un entier égal à 2.

$(\text{Appartement} \cap \exists \text{annotepar} .F3) \equiv \text{Appartement} \cap \exists \text{annotepar} .F3 \cap \exists fC$
 $(\text{Adresse} \cap \exists \text{annotepar} .(\forall vp1, vp2 \text{adresse} . \text{Chaîne de caractère}) \cap \exists nS (\text{Nbrpiece} \cap$
 $\exists \text{annotepar} .(\forall vp1, \geq 0 \text{ nbr_pièce} .\{3\}) \cap \exists noC \cap \exists noS))$

Cette classe exige aussi que le noud texte fils du nœud Nbrpiece doive être un entier égal à 3.

$(\text{Appartement} \cap \exists \text{annotepar} .F4) \equiv \text{Appartement} \cap \exists \text{annotepar} .F4 \cap \exists fC$
 $(\text{Adresse} \cap \exists \text{annotepar} .(\forall vp1, vp2 \text{adresse} . \text{Chaîne de caractère}) \cap \exists nS (\text{Nbrpiece} \cap$
 $\exists \text{annotepar} .(\forall vp1, \geq 0 \text{ nbr_pièce} .\{4\}) \cap \exists noC \cap \exists noS))$

Cette classe exige aussi que le noud texte fils du nœud Nbrpiece doive être un entier égal à 4.

$(\text{Appartement} \cap \exists \text{annotepar} .F5) \equiv \text{Appartement} \cap \exists \text{annotepar} .F5 \cap \exists fC$
 $(\text{Adresse} \cap \exists \text{annotepar} .(\forall vp1, vp2 \text{adresse} . \text{Chaîne de caractère}) \cap \exists nS (\text{Nbrpiece} \cap$
 $\exists \text{annotepar} .(\forall vp1, \geq 0 \text{ nbr_pièce} .\{5\}) \cap \exists noC \cap \exists noS))$

Cette classe exige aussi que le noud texte fils du nœud Nbrpiece doive être un entier égal à 5.

Définition de classes d'annotation pour le point de vue local « financier » :

Une Classe d'annotation $CA_{PvFinancier}$ pour le point de vue Financier crée par le concept défini de structure **SPVFinancier** avec le concept global Liste_locataire^ô, cette classe intègre la sémantique de l'ontologie comme suit :

$CA_{PvFinancier} \equiv \text{Racine} \cap \exists \text{annotepar} .\text{Liste_locataire}^{\hat{o}} \cap \exists fC (\text{Location} \cap \exists \text{annotepar}$
 $.\text{Locataire}^{\hat{o}} \cap \exists fC (\text{Personne} \cap \exists \text{annotepar} .\text{Personne}^{\hat{o}} \cap \exists fC (\text{Nom} \cap \exists \text{annotepar} .(\forall vp1, vp2$
 $\text{nom} .\text{Chaîne de caractère}) \cap \exists nS (\text{Salaire} \cap \exists \text{annotepar} .(\forall vp2, \geq 0 \text{ salaire} .\text{Monétaire}) \cap$
 $\exists noC \cap \exists noS)) \cap \exists nS (\text{Appartement} \cap \exists \text{annotepar} .\text{Appartement}^{\hat{o}} \cap \exists fC (\text{Adresse} \cap$
 $\exists \text{annotepar} .(\forall vp1, vp2 \text{adresse} . \text{Chaîne de caractère}) \cap \exists nS (\text{Loyer} \cap \exists \text{annotepar} .(\forall vp2, \geq$
 $0 \text{ loyer} .\text{Monétaire}) \cap \exists noC \cap \exists noS))))$



Cette classe exige que les nœuds textes fils des nœuds Nom, et Adresse doivent être chaîne de caractère. Et si le concept de structure Salaire est relié par le rôle « annotépar » à l'attribut $(\forall vp2, \geq 0 \text{ salaire } .Monétaire)$, alors cette classe exige aussi que le nœud texte fils du nœud Salaire doit être un entier réel suivi par un symbole monétaire (dans le cadre de notre implémentation ce symbole doit être DA) . Et si le concept de structure Loyer est relié par le rôle « annotépar » au attribut $(\forall vp2, \geq 0 \text{ loyer } .Monétaire)$, alors cette classe exige aussi que le nœud texte fils du nœud Loyer doit être un entier réel suivi par un symbole monétaire (dans le cadre de notre implémentation ce symbole doit être DA).

On a aussi les classes d'annotation

$(\text{Appartement} \cap \exists \text{annotépar } .\text{Appartemant_cher}) \equiv \text{Appartement} \cap \exists \text{annotépar } .\text{Appartemant_cher} \cap \exists fC (\text{Adresse} \cap \exists \text{annotépar} .(\forall vp1, vp2 \text{ adresse } .\text{Chaîne de caractère}) \cap \exists nS(\text{Loyer} \cap \exists \text{annotépar} .(\forall vp2, \geq 0 \text{ loyer } .[2100DA \infty]) \cap \exists noC \cap \exists noS)))$

Cette classe exige aussi que le nœud texte fils du nœud Loyer doit être supérieur à 2100DA.

$(\text{Appartement} \cap \exists \text{annotépar } .\text{Appartemant_pascher}) \equiv \text{Appartement} \cap \exists \text{annotépar } .\text{Appartemant_pascher} \cap \exists fC (\text{Adresse} \cap \exists \text{annotépar} .(\forall vp1, vp2 \text{ adresse } .\text{Chaîne de caractère}) \cap \exists nS(\text{Loyer} \cap \exists \text{annotépar} .(\forall vp2, \geq 0 \text{ loyer } .[0DA 2099DA]) \cap \exists noC \cap \exists noS)))$

Cette classe exige aussi que le nœud texte fils du nœud Loyer doit être inférieur à 2099DA.

$(\text{Location} \cap \exists \text{annotépar } .\text{Locataire_pasriche}) \equiv \text{Location} \cap \exists \text{annotépar } .\text{Locataire_pasriche} \cap \exists fC(\text{Personne} \cap \exists \text{annotépar } .\text{Personne}^{\hat{o}} \cap \exists fC(\text{Nom} \cap \exists \text{annotépar} .(\forall vp1, vp2 \text{ nom } .\text{Chaîne de caractère}) \cap \exists nS(\text{Salaire} \cap \exists \text{annotépar} .(\forall vp2, \geq 0 \text{ salaire } .[0DA 24999DA]) \cap \exists noC \cap \exists noS)) \cap \exists nS(\text{Appartement} \cap \exists \text{annotépar } .\text{Appartement}^{\hat{o}}))$

Cette classe exige aussi que le nœud texte fils du nœud Salaire doit être inférieur à 24999DA.



$$\begin{aligned}
 &(\text{Location} \cap \exists \text{annotepar} .\text{Locataire_riche}) \equiv \text{Location} \cap \exists \text{annotepar} \\
 &.\text{Locataire_riche} \cap \exists fC(\text{Personne} \cap \exists \text{annotepar} .\text{Personne} \hat{\circ} \cap \\
 &\exists fC(\text{Nom} \cap \exists \text{annotepar} . (\forall vp1, vp2 \text{ nom} .\text{Chaîne de caractère}) \cap \exists nS(\text{Salaire} \cap \\
 &\exists \text{annotepar} . (\forall vp2 , \geq 0 \text{ salaire} . [25000\text{DA} \infty]) \cap \exists noC \cap \exists noS)) \exists nS \\
 &(\text{Appartement} \cap \exists \text{annotepar} .\text{Appartement} \hat{\circ}
 \end{aligned}$$

Cette classe exige aussi que le nœud texte fils du nœud Salaire doive être supérieur à 25000DA.

3.4. Annotation d'une nouvelle page appartenant en même domaine

Supposant que la nouvelle page du domaine d'immobilier que nous allons annoter est la page de la figure 8 avec le code XML correspondant représenté dans la figure 9 et dont l'arbre DOMXML est représenté dans la figure10, ses sous arbres selon les points de vues Taille, Financier, et Global sont représentés respectivement dans les figures 11, 12, et 13.



Figure8 : la page web

```

<?xml version="1.0" encoding="UTF-8" ?>
<Racine>
<Location>
<Personne>
<Nom>Hemam</Nom>
<Salaire>40000DA</Salaire>
</Personne>
<Appartement>
<Adresse>appartement numero 112, etage4, Grue benbaiz25000</Adresse>
<Nbrpieces>1</Nbrpieces>
<Loyer>2000</Loyer>
</Appartement>
<Appartement>
<Adresse>appartement numero 10, etage1, constantine25000</Adresse>
<Nbrpieces>3</Nbrpieces>
<Loyer>8000DA</Loyer>
</Appartement>
</Location>
<Location>
<Personne>
<Nom>Mohamed</Nom>
<Salaire>14000DA</Salaire>
</Personne>
<Appartement>
<Adresse>appartement numero 11, etage4, Ain S'mara 25000</Adresse>
<Nbrpieces>2</Nbrpieces>
<Loyer>4000</Loyer>
</Appartement>
</Location>
</Racine>

```

Figure9 : l'extrait de code XML

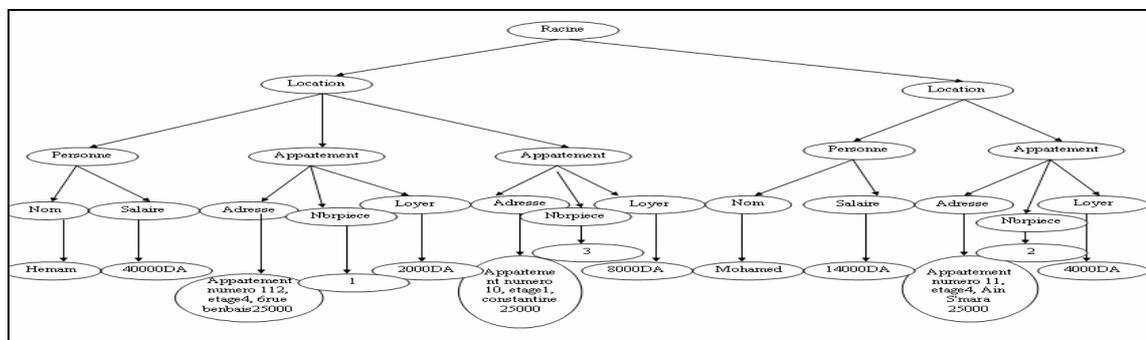


Figure10 : arbre DOM du page « Immobilier »

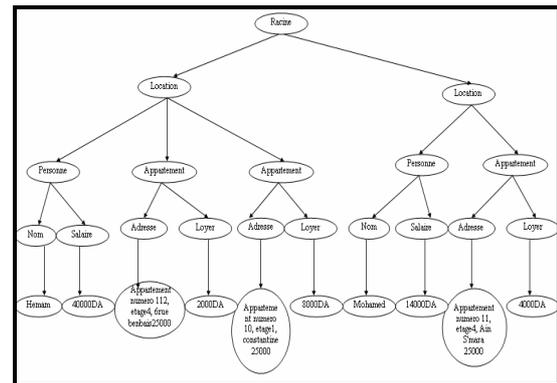
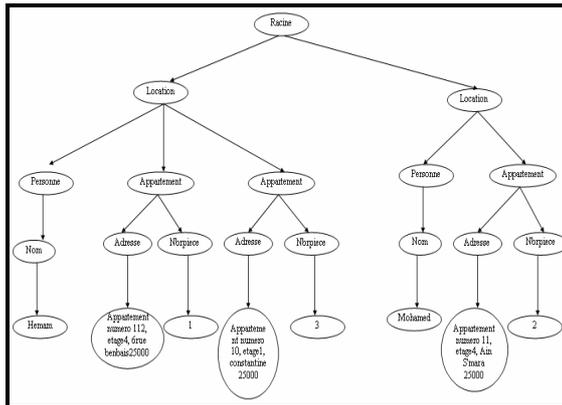


Figure11: Un modèle DOM de la page « Immobilier » selon le point de vue Taille

Figure12 : Le modèle DOM de la page « Immobilier » selon le point de vue

Finance

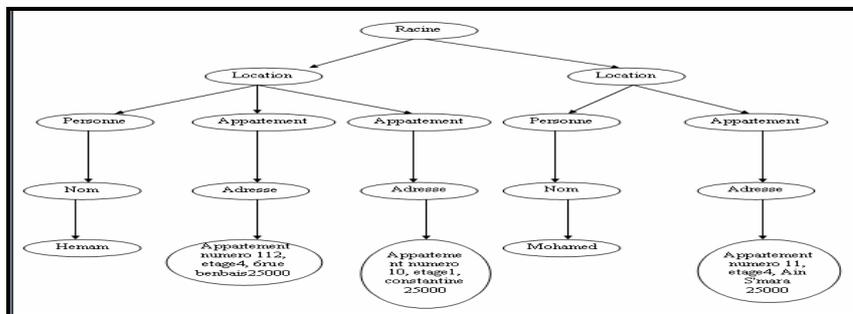


Figure13 : Le modèle DOM de la page « Immobilier » selon le point de vue global

Activité 1 : Formalisation des structures de la page par des individus de LD :

Afin de formaliser la page, le noeud élément **Racine** est formalisé par l'individu de la structure en LD i_0 , le premier noeud élément **Location** est formalisé par l'individu i_1 , le premier **Personne** par i_2 , le premier **Appartement** par i_3 , le premier **Nom** par i_4 , le premier **Salaire** par i_5 , le premier **Adresse** par i_6 , le premier **Nbrpiece** par i_7 , et le premier **Loyer** par i_8 , le deuxième **Adresse** par i_{61} , le deuxième **Nbrpiece** par i_{71} , et le deuxième **Loyer** par i_{81} , le deuxième noeud élément **Location** est formalisé par l'individu i_{12} , le deuxième **Personne** par i_{22} le troisième **Appartement** par i_{32} , le deuxième **Nom** par i_{42} , le deuxième **Salaire** par i_{52} , le troisième **Adresse** par i_{62} , le troisième **Nbrpiece** par i_{72} , et le troisième **loyer** par i_{82} ,

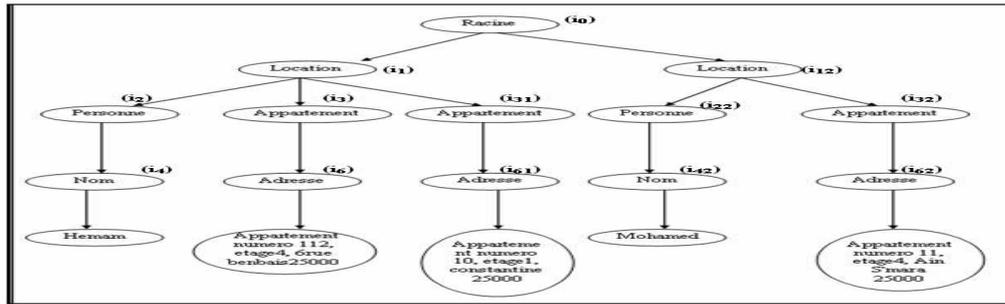


Figure14: DOM (et individus de structure) de la page à annoter selon le point de vue global

Ainsi, la formalisation de la page figure 14 selon le point de vue global est la suivante :

Racine (i₀), Location (i₁), Personne(i₂), Appartement (i₃), Appartement (i₃₁) Nom (i₄), Adresse (i₆) Adresse (i₆₁) Location (i₁₂), Personne(i₂₂), Appartement (i₃₂), Nom (i₄₂), Adresse (i₆₂).
fC(i₀,i₁) fC(i₁,i₂) nS(i₁, i₂) noS(i₁₂) nS(i₂,i₃) nS(i₃,i₃₁)noS(i₃₁) fC(i₂,i₄) noC(i₄) noS(i₄)
fC (i₃,i₆) noC(i₆) noS(i₆) fC (i₃₁,i₆₁) noC(i₆₁) noS(i₆₁)
fC(i₁₂,i₂₂)nS(i₂₂,i₃₂) noS(i₃₂) fC(i₂₂,i₄₂) noC(i₄₂) noS(i₄₂) fC (i₃₂,i₆₂) noC(i₆₂) noS(i₆₂)

De la même façon, nous formalisons la page figure 15 dans le point de vue Taille.

Racine (i₀), Location (i₁), Personne(i₂), Appartement (i₃), Nom (i₄), Adresse (i₆), Nbrpiece (i₇), Location (i₁₂), Personne(i₂₂), Appartement (i₃₂), Nom (i₄₂), Adresse (i₆₂), Nbrpiece (i₇₂), Appartement (i₃₁), Adresse (i₆₁), Nbrpiece (i₇₁) fC(i₀,i₁) fC(i₁,i₂) nS(i₁, i₂) noS(i₁₂) nS(i₂,i₃) nS(i₃,i₃₁) noS(i₃₁) fC(i₂,i₄) noC(i₄) noS(i₄) fC (i₃,i₆) nS (i₆, i₇) noC(i₇) noS(i₇) fC (i₃₁,i₆₁) nS (i₆₁, i₇₁) noC(i₇₁) noS(i₇₁) fC(i₁₂,i₂₂)nS(i₂₂,i₃₂) noS(i₃₂) fC(i₂₂,i₄₂) noC(i₄₂) noS(i₄₂) fC (i₃₂,i₆₂) nS (i₆₂, i₇₂) noC(i₇₂) noS(i₇₂)

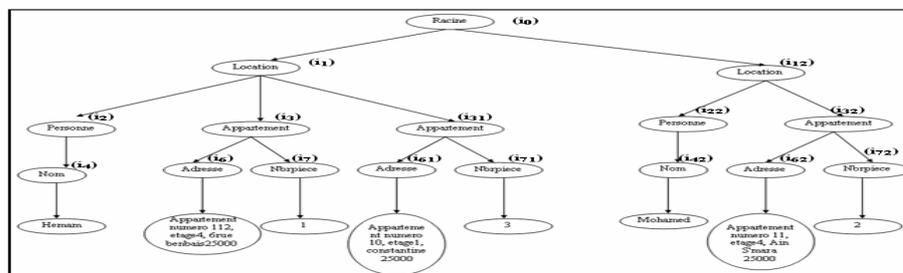


Figure15: DOM (et individus de structure) de la page à annoter selon le point de vue taille

Et dans le point de vue Financier (figure16).



Racine (i_0), Location (i_1), Personne(i_2), Appartement (i_3), Nom (i_4), Adresse (i_6), Salaire (i_5)Loyer (i_8) Location (i_{12}), Personne(i_{22}), Appartement (i_{32}), Nom (i_{42}), Salaire (i_{52}), Adresse (i_{62}), Loyer(i_{82}), Appartement (i_{31}), Adresse (i_{61}), Loyer (i_{81})
 $fC(i_0, i_1)$ $fC(i_1, i_2)$ $nS(i_1, i_{12})$ $noS(i_{12})$ $nS(i_2, i_3)$ $nS(i_3, i_{31})$ $noS(i_3)$ $fC(i_2, i_4)$ $nS(i_4, i_5)$ $noC(i_5)$
 $noS(i_5)$ $fC(i_3, i_6)$ $nS(i_6, i_8)$ $noC(i_8)$ $noS(i_8)$ $fC(i_{31}, i_{61})$ $nS(i_{61}, i_{81})$ $noC(i_{81})$ $noS(i_{81})$
 $fC(i_{12}, i_{22})$ $nS(i_{22}, i_{32})$ $noS(i_{32})$ $fC(i_{22}, i_{42})$ $nS(i_{42}, i_{52})$ $noC(i_{52})$ $noS(i_{52})$ $fC(i_{32}, i_{62})$ $nS(i_{62}, i_{82})$
 $noC(i_{82})$ $noS(i_{82})$

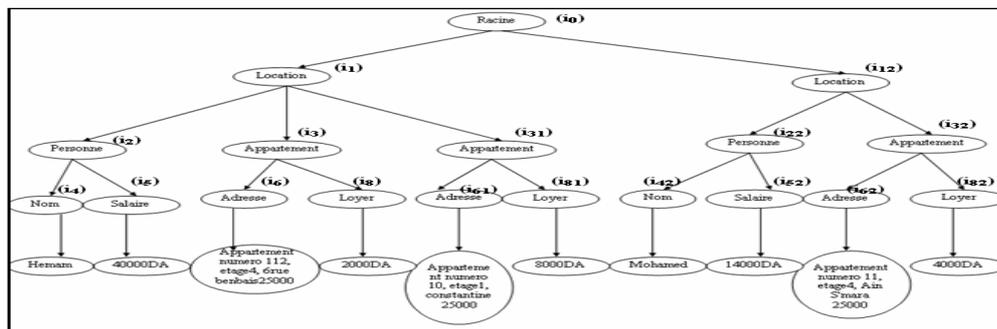


Figure16: DOM (et individus de structure) de la page à annoter selon le point de vue financier

Selon le raisonneur nous obtenons :

Selon le point de vue global :

- Les individus (i_4), (i_{42}) sont des instances du concept primitif de structure **Nom (balise nom)** de la structure.
- Les individus (i_6), (i_{62}) (i_{61}) sont des instances du concept primitif de structure **Adresse (balise adresse)** de la structure.
- Les individus (i_2), (i_{22}) sont des instances du concept primitif de structure **Personne**.
- Les individus (i_3), (i_{31}), **et** (i_{32}) sont des instances du **Appartement**.
- Les individus (i_1), (i_{12}) sont des instances du concept **Location**
- **l'individu (i_0)** est une instance du concept **Racine**

Donc la page est instance du concept **SPVGlobal**

Selon le point de vue taille :

- Les individus (i_7), (i_{71}), (i_{72}) sont des instances du concept primitif **Nbrpiece(balise Nbrpiece)** de la structure.
- La page est instance du concept **SPVTaille**.



Selon le point de vue financier :

- Les individus (i_5), (i_{52}) sont des instances du concept primitif **Salaire (balise Salaire)** de la structure.
- Les individus (i_8), (i_{81}), (i_{82}) sont des instances du concept primitif **Loyer (balise Loyer)** de la structure.
- La page est instance du concept **SPVFinancier**

Activité2 : Génération des annotations

Selon le raisonneur la page est instance de la classe CA_{PVG} alors

Selon le point de vue global :

i_0 annoté par Liste_locataire $\hat{\circ}$, i_1 annoté par Locataire $\hat{\circ}$, i_{12} annoté par Locataire $\hat{\circ}$, i_2 annoté par Personne $\hat{\circ}$, i_{22} annoté par Personne $\hat{\circ}$, i_3 annoté par Appartement $\hat{\circ}$, i_{31} annoté par Appartement $\hat{\circ}$, i_{32} annoté par Appartement $\hat{\circ}$, i_4 annoté par ($\forall vp1, vp2 \text{ nom } .Cha\^{\i}ne \text{ de caract\`e}re$), i_{42} annoté par ($\forall vp1, vp2 \text{ nom } .Cha\^{\i}ne \text{ de caract\`e}re$), i_6 annoté par ($\forall vp1, vp2 \text{ adresse. } Cha\^{\i}ne \text{ de caract\`e}re$), i_{62} annoté par ($\forall vp1, vp2 \text{ adresse. } Cha\^{\i}ne \text{ de caract\`e}re$), i_{61} annoté par ($\forall vp1, vp2 \text{ adresse. } Cha\^{\i}ne \text{ de caract\`e}re$)

Selon le point de vue taille :

i_0 annoté par Liste_locataire $\hat{\circ}$, i_1 annoté par Locataire $\hat{\circ}$, i_{12} annoté par Locataire $\hat{\circ}$, i_2 annoté par Personne $\hat{\circ}$, i_{22} annoté par Personne $\hat{\circ}$, i_3 annoté par F1, i_{31} annoté par F3, i_{32} annoté par F2, i_4 annoté par ($\forall vp1, vp2 \text{ nom } .Cha\^{\i}ne \text{ de caract\`e}re$), i_{42} annoté par ($\forall vp1, vp2 \text{ nom } .Cha\^{\i}ne \text{ de caract\`e}re$), i_{61} annoté par ($\forall vp1, vp2 \text{ adresse. } Cha\^{\i}ne \text{ de caract\`e}re$), i_{62} annoté par ($\forall vp1, vp2 \text{ adresse. } Cha\^{\i}ne \text{ de caract\`e}re$), i_{61} annoté par ($\forall vp1, vp2 \text{ adresse. } Cha\^{\i}ne \text{ de caract\`e}re$), i_7 annoté par. ($\forall vp1, \geq 0 \text{ nbr_pi\`e}ce \{1\}$), i_{71} annoté par. ($\forall vp1, \geq 0 \text{ nbr_pi\`e}ce \{3\}$), i_{72} annoté par. ($\forall vp1, \geq 0 \text{ nbr_pi\`e}ce \{2\}$)

Selon le point de vue financier :

i_0 annoté par Liste_locataire $\hat{\circ}$, i_1 annoté par Locataire_riche, i_{12} annoté par Locataire_pasriche, i_2 annoté par Personne $\hat{\circ}$, i_{22} annoté par Personne $\hat{\circ}$, i_3 annoté par Appartement_pascher, i_{31} annoté par Appartement_cher, i_{32} annoté par Appartement_cher, i_4 annoté par ($\forall vp1, vp2 \text{ nom } Cha\^{\i}ne \text{ de caract\`e}re$), i_{42} annoté par ($\forall vp1, vp2 \text{ nom } Cha\^{\i}ne \text{ de caract\`e}re$), i_5 annoté par ($\forall vp2, \geq 0 \text{ salaire. } [25000DA \infty]$), i_{52} annoté par ($\forall vp2, \geq 0 \text{ salaire. } [0DA \text{ } 24999DA]$), i_6 annoté par ($\forall vp1, vp2 \text{ adresse. } Cha\^{\i}ne \text{ de$



caractère), i_{62} annoté par $(\forall vp1, vp2 \text{ adresse. Chaîne de caractère})$, i_{61} annoté par $(\forall vp1, vp2 \text{ adresse. Chaîne de caractère})$, i_8 annoté par $(\forall v2, \geq 0 \text{ loyer. [0DA 2099DA]})$, i_{81} annoté par $(\forall vp2, \geq 0 \text{ loyer. [2100DA } \infty])$, i_{82} annoté par $(\forall vp2, \geq 0 \text{ loyer. [2100DA } \infty])$

D'après les résultats de l'annotation de notre page selon les points de vue: global, Taille, et Financier, les éléments de la page visibles dans le point de vue spécifié sont annotés, les autres non. Aussi un élément peut être annoté selon un point de vue spécifié différemment à s'annotation selon un autre point de vue.

4. Implémentation

Afin de montrer que notre approche proposée est réalisable, nous présentons quelques aspects d'implémentation pour notre approche sur une page appartenant au domaine d'immobilier.

4.1. Implémentation de l'ontologie en PHP5:

Nous avons vu dans une section précédente comment implémenter l'ontologie de domaine en OWL. Ainsi, pour pouvoir manipuler notre ontologie à travers le langage PHP, nous avons transformé le fichier OWL sous forme d'une base de connaissances orienté objet avec PHP, de telle façon les classes(concepts) en OWL sont implémentées par des classes avec PHP, les relations de subsomption entre les classes en OWL sont implémentées par l'utilisation de la notion d'héritage en PHP, les axiomes (contraintes) en OWL sont implémentés par des méthodes(fonctions) des classes en PHP, et enfin les individus sont implémentés par des instances des classes(c-à-d des objets) en PHP. Par exemple selon le point de vue global, l'implémentation du concept **Personne** ainsi que ces contraintes comme suit :

```
class Personne {
    var $nom;
    function defPersonne ($nom) {
        $this->nom = $nom;
    }

    function verPersonne ($nom) {
        if (ereg("[a-zA-Z0-9_.,-|`|è|é|â|ç|à|ô|\",|û]*$", $nom, $regs))
            return(true);
        else return(false);
    }
}
```

4.2. Implémentation des concepts de structure

En premier temps nous implémentons les concepts de structure de chaque point de vue en OWL grâce à l'outil protégé 2000, donc nous allons construire une ontologie de



structure pour chaque point de vue (global, taille, et financier). Puis à partir de OWL nous implémentons chaque ontologie grâce aux classes et objets en PHP dans chaque point de vue de la même façon que nous avons implémenté l'ontologie de domaine. .

4.3. Implémentation des classes d'annotation

Pour construire les classes d'annotation pour chaque point de vue, nous avons créé 3 copies de l'ontologie de domaine. Puis, nous avons fusionné (dans le sens de regrouper les concepts des deux ontologies dans une seule ontologie) chaque ontologie de structure avec l'ontologie de domaine c à d intégrer les concepts de structure concernant un point de vue dans la copie de l'ontologie de domaine (manuellement avec protégé 2000). Le but de fusionner une ontologie de structure avec l'ontologie de domaine et pour relier des concepts appartenant à l'ontologie de structure avec des concepts de domaine concernant le point de vue qui représente cette ontologie de structure par le rôle « annotepar » pour implémenter des classes d'annotation. Puis à partir de OWL nous implémentons les classes d'annotation de chaque point de vue grâce aux classes et objets en PHP. Par exemple l'implémentation de la classe $Personne \cap \text{annotepar CAPPVG}$ comme suit :

```
class CAPPVG {
var $Personnestructure; var $nomstructure; var $Personne; var $nom; var $valnom; var $existe;
function annoteparconcept ($nomstructure) {
if($nomstructure==true)
return(true);
else
return(false);}
function annoteparconceptattribut($nom,$valnom){
if(ereg("[a-zA-Z0-9_\.|-|\|è|é|â|ç|à|ô|\",|û]*$", $valnom,$regs)&&$valnom!="")
return(true);
else return(false);}
function annoteparpersonne ($Personnestructure,$Personne)
{return ('le concept «Personne»);}
function annoteparnom ($nomstructure,$nom)
{return ('attribut «nom» du concept "Personne");}
function existe($existe){
if($existe==true)
return(true);
else
return (false);}}
```

4.4. Annotation d'une nouvelle page appartenant en même domaine

Pour ce faire, nous passons par les activités suivantes :

Création des individus de structure :

Nous avons généré l'arbre DOMXML de la page à annoter grâce au script PHP5. Nous avons utilisé PHP parce que c'est le bon scripte qui nous permet de traiter une page web



notamment avec une page XML. En premier temps, on a créé un objet DOMDocument puis on récupère le fichier XML grâce à l'ensemble des instructions suivantes :

```
$Document = new DOMDocument;  
$Document->load("http://127.0.0.1/immobilier2.xml");  
$racine = $Document->documentElement;
```

Pour parcourir l'arbre nous utilisons l'ensemble des instructions suivantes :

```
$elements = $Document->getElementsByTagName($racine->nodeName);  
$element = $elements->item(0); // On obtient le noeud racine.  
$enfants = $element->childNodes; // On récupère les nœuds enfants de racine avec childNodes.  
i=0 ;  
foreach ($enfants as $enfant) // On prend chaque noeud enfant séparément.  
{  
$location[i] = $enfant ;  
}
```

Pour générer un sous arbre pour chaque point de vue, nous supprimons les nœuds ne concernant pas ce point de vue grâce à l'instruction :

```
$element->removeChild($nom-noeud);
```

Nous avons créé les individus de structure avec PHP sous forme des objets ayant des valeurs obtenus à partir de la page. Par exemple si on a un individu instance du concept de structure Personne, nous créons un objet de la classe Personne ayant comme nom la valeur du nœud texte fils.

Vérification de l'instanciation :

Chaque classe contient une méthode permet de vérifier si les conditions exigé par cette classe sont vérifiées sur l'objet créé, si ces méthode retourne faux, alors l'objet créé va être détruit.

Génération des annotations :

Pour chaque point de vue, nous avons créé des individus des classes d'annotation par création des objets en PHP, ces derniers sont des instances des classes d'annotation. Ces individus (objets) représentent des candidats à l'annotation. Par exemple selon le point de vue global, nous construirons un objet \$annoterparpersonne qui est une instance de la classe CAPPVG, puis nous vérifions les attributs \$nomnom, \$nomvaleur de l'objet \$annoterparpersonne par la méthode annoterparconceptattribut de la classe CAPPVG.

```
$annoterparpersonne= new CAPPVG;  
if ($annoterparpersonne->annoterparconceptattribut($nomnom,$nomvaleur) == true)  
$existeannoterparpersonne=true;  
else $existeannoterparpersonne=false;
```



Les interfaces de l'annotation de notre page sont présentées dans les figures en dessous :

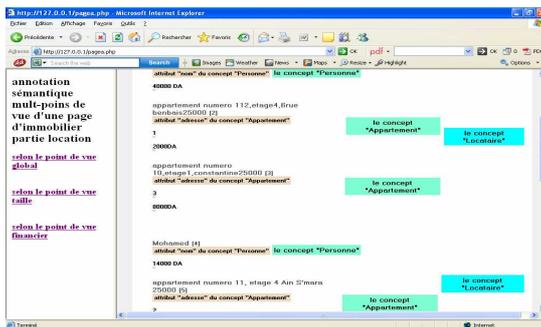


Figure17 : l'annotation de la page selon point de vue global

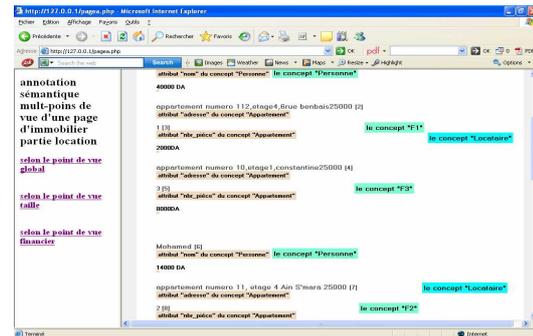


Figure18 : l'annotation de la page selon le point de vue taille

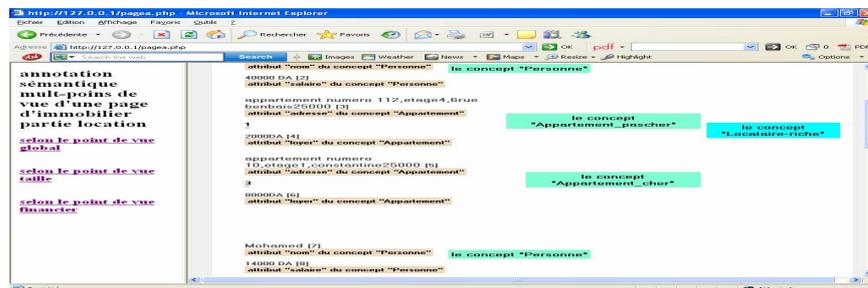


Figure20: l'annotation de la page selon le point de vue financier

4.5. Test

Nous avons effectué un test sur notre logiciel, sur un ensemble de pages appartenant au domaine d'immobilier. Par exemple nous avons utilisé des pages vides, des pages qui contiennent un ensemble de locataires.

5. Conclusion :

Nous avons montré comment l'ontologie multipoint de vue du domaine d'immobilier a été conceptualisée et formalisée, puis traduit dans le langage OWL sous l'éditeur Protégé 2000, ce dernier qui fourni une interface basée sur le modèle de la logique de descriptions. Puis nous avons appliqué les différentes étapes et activités de notre approche sur un exemple d'une page d'immobilier (partie location). A la fin, nous avons présenté quelques aspects pour implémenter notre approche.

Le chapitre suivant conclut notre travail mais permet surtout de dresser des perspectives.

Conclusion générale

Conclusion générale et perspectives

Conclusion générale :

L'annotation sémantique d'une page web à partir d'une ontologie multi-points de vue s'effectue de plusieurs façons, ce qui nécessite de déterminer les éléments concernant un point de vue particulier afin d'annoter ces éléments par des instances de concepts connus dans ce point de vue. L'intérêt de notre approche est de construire une structure propre pour chaque point de vue illustrant seulement les éléments concernant ce point de vue et les relations entre ces éléments. Ensuite, ces éléments sont formalisés comme des individus de structure qui sont annotés par des instances de concepts de domaine connus dans ce point de vue, et les relations entre ces éléments sont formalisées par des instances de rôles.

Limites et difficultés :

- L'absence des langages pour implémenter l'ontologie multi-points de vue rendre la tâche d'annotation difficile.
- Nous avons représenté le point de vue dans OWL par un préfixe par exemple un concept F1 vue dans le point de vue taille sera représenté comme suit : « taille : F1 ». Le problème est que cette représentation ne représente pas complètement la notion de point de vue, car le préfixe représente une méta-donnée. Donc nous sommes obligés de programmer dans un langage de programmation les notions concernant le point de vue afin d'exploiter notre ontologie dans le cadre de point de vue, ou bien transformer l'ontologie OWL en base des connaissances orienté objet pour introduire les notions de point de vue.
- Les raisonneurs existants ne supportent pas la notion de point de vue. Ces raisonneurs testent notre ontologie comme une ontologie simple qui utilise des métas-

donnée. De ce fait ils ne test pas la notion de point de vue, donc nous sommes obligés de faire un programme qui complète la tâche de vérification et de raisonnement sur la notion de point de vue.

- Il est possible d'écrire une classe d'annotation pour toutes les pages du corpus. Ceci devient toutefois coûteux si la structure n'est pas régulière.
- De plus, les logiques de descriptions fournissent l'explication de l'annotation des pages. En outre, la maintenance des annotations devient un problème de satisfaction : si la structure change, l'instanciation échoue.
- Une page va avoir plusieurs annotations, il se peut se produire une contradiction, c'est-à-dire : il se peut qu'un élément de la page annoté par des concepts ne soit pas compatible, par exemple : dans le point de vue taille un élément annoté par le concept `Petit_appartement` tandis que dans le point de vue financier cet élément est annoté par le concept `Appartement_cher` : où il n'est pas logique de considérer qu'un petit appartement soit cher. En effet, ce problème se produit car, nous n'avons pas pris en considération l'annotation par des instances de passerelles.
- En terme d'implémentation, il n'existe pas un code universel pour tous les domaines, si on change le domaine (la page modèle) il faut réécrire le code du programme totalement. Ce problème est lié à la manipulation de l'arbre DOMXML.
- Dans le cadre de notre travail, nous allons à annoter seulement un seul type de ressource, qui est une page web.

Perspectives :

- Une perspective naturelle de ce travail consiste à valider et à évaluer l'approche dans un domaine réel.
- Par ailleurs, nous envisageons d'automatiser la construction des classes d'annotation. L'objectif est de permettre à un expert du domaine d'instancier l'ontologie multi-points de vue sans avoir à construire des classes à la main.

- Nous envisagions d'automatiser la tâche d'extraction des noeuds de l'arbre DOMXML afin de construire une structure propre pour un point de vue particulier : Construire des outils qui sont capable de savoir quels sont les éléments de la page qui ne sont pas pertinents à un point de vue particulier afin de les extraire de l'arbre DOM.
- Développer des langages de représentation d'annotation supportant la notion multi-points de vue.
- Intégrer des outils qui devraient permettre la maintenance de l'ontologie de domaine, la recherche de pages web pertinentes, la génération automatique des classes d'annotation, le peuplement de l'ontologie, la maintenance des annotations et la génération de requêtes.
- Traiter le problème de la contradiction entre les annotations obtenues.
- Annoter par des instances des passerelles.
- Proposer une méthode d'annotation sémantique à partir d'une ontologie multi-points de vue pour annoter d'autres types de ressource qui ne sont pas des pages web, comme les textes, les ouvrages, les images, le CV

Bibliographie

Bibliographie

- [**Acacia 06**] Projet Acacia, “Corese: COncceptual REsource Search Engine”, INRIA, (2006).
- [**Apparo 98**] V. Apparo, T. Pixley, “The NGLayout Document Object Model (DOM)” Roadmap. Mozilla Organization, (1998). <http://www.mozilla.org/newlayout/dom-roadmap.html>.
- [**Bach 06**] T. L. Bach, “Construction d’un Web sémantique multi-points de vue”. Thèse de doctorat en sciences, École des Mines de Paris, Sophia Antipolis. (2006).
- [**Bachimont 00**] B. Bachimont, “Engagement sémantique et engagement ontologique : conception et réalisation d’ontologies en ingénierie des connaissances”. Eyrolles, (2000).
- [**Bechhofer 01**] S. Bechhofer, I. Horrocks, and C. Goble, “OilEd: a Reason-able Ontology Editor for the Semantic Web”. In Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence. (2001).
- [**Benchikha 07**] F. Benchikha et M. Boufaïda, “The Viewpoint Mechanism for Object-oriented Databases Modelling, Distribution and Evolution” In Journal of Computing and Information Technology. Vol 15, p. 95-110 (2007).
- [**Benerecetti 01**] M. Benerecetti, P. Bouquet, C. Ghidini, “On the Dimensions of Context Dependence: Partiality, Approximation, and Perspective”. Dans Proc. of CONTEXT: 59-72 (2001).
- [**Benjamins 99**] V.R. Benjamins, D. Fenel, S. Decker et A. GÓMEZ-PÉREZ, (KA) 2: “building ontologies for the internet: a mid term report”, in International Journal of Human Computer Studies, Volume 51, pp.687-712. (1999).
- [**Benslimen 03**] D. Benslimen, C. Vangenot, C. Roussy et A. Arara «The Multi Representation Ontologies: A contextual Description Logics Approach » In Proc of the 15th conference on advanced information systems engineering, LNCS 2798, Springer-Verlag, Heidelberg 2003.
- [**Blazquez 98**] M. Blazquez, M. Fernandez, J. Garcia-Pinar. et A. Gomez-Perez, “Building Ontologies at the Knowledge Level using the Ontology Design Environment”, in Proceedings of the Banff Workshop on Knowledge Acquisition for Knowledge-based Systems, (1998).

- [**Bobrow 77**] D. G. Bobrow, et T. Winograd, “An overview of KRL, a knowledge representation language”. *Cognitive Science*, 1(1), 3--46. (1977).
- [**Borgida 03**] A. Borgida, et L. Serafini, “Distributed description logics: Assimilating information from peer sources”. *Journal of Data Semantics*, 1:153–184, (2003).
- [**Bouquet 03**] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, et H. Stuckenschmidt, “C-OWL: Contextualizing Ontologies”. *International Semantic Web Conference*: 164-179 (2003).
- [**Bouquet 04**] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, et H. Stuckenschmidt, “Contextualizing Ontologies”. Dans *Journal of Web Semantics*, (2004).
- [**Burgun 01**] A. Burgun and O. Bodenreider, “Mapping the UMLS Semantic Network into general ontologies. *Proceedings of the AMIA Annual Symposium*”. pp 81-85. (2001).
- [**Carme 07**] J. Carne, R. Gilleron, A. Lemay et J. Niehren, “Interactive learning of node selecting tree transducers, machine learning”. *Machine Learning*, 66(1), 33–67. (2007).
- [**Charlet 03**] J. Charlet, “*L'ingénierie des connaissances : développements, résultats et perspectives pour la gestion des connaissances médicales*”. Mémoire d'habilitation à diriger des recherches, présenté à l'université Pierre et Marie Curie. 2003.
- [**Chandrasekaran 99**] B. Chandrasekaran, J. R. Josephson and V. R. Benjamins, “What are ontologies and why do we need them?” *IEEE Intelligent Systems*. 14(1):20-26. (1999).
- [**Cimiano 04**] P. Cimiano, S. Handschuh, et S. Staab, “Towards the Self-Annotating Web”, in *Proceedings of the 13th International World Wide Web Conference (WWW'04)*, ACM Press, New-York, USA, pp. 462-471. (2004).
- [**Ciravegna 04a**] F. Ciravegna et Y. Wilks, “Designing Adaptive Information Extraction for the Semantic Web in Amilcare in Annotation for the Semantic Web.” ACM Press, New-York, USA, pp. 1-32 (2004).
- [**Ciravegna 04b**] F. Ciravegna, S. Chapman, A. Dingli, et Y. Wilks, “Learning to harvest information for the semantic web”. In *ESWS*, pp. 312–326. (2004).
- [**D'Aquin 05a**] M. D'Aquin, Thèse “Un portail sémantique pour la gestion des connaissances en cancérologie”. (2005).

- [**D'Aquin 04**] M. D'Aquin, J. Lieber, et A. Napoli, "Représentation de points de vue pour le raisonnement à partir de cas". In *Langages et Modèles à objets (LMO'04)*, pp. 245-258, (2004).
- [**D'Aquin 05b**] M. D'Aquin, J. Lieber, et A. Napoli, "Decentralized Case-Based Reasoning for the Semantic Web". In *Proceedings of the 4th International Semantic Web Conference (ISWC'05)*, Springer, (2005).
- [**David 87**] H. David, "Views Multiple perspectives and structured objects in a knowledge representation language". Master's thesis, MIT. (1987).
- [**Desmontils 02**] E. Desmontils et C. Jacquin, "Indexing a Web Site with a Terminology Oriented Ontology", In *The Emerging Semantic Web*, p. 181–197. IOS Press, i.f. cruz and s. decker and j. euzenat and d. l. mcguinness edition. (2002).
- [**Drews 93**] O. M. Drews, "Raisonnement classificatoire dans une représentation à objets multipoints de vue". Thèse de doctorat, Université Joseph Fourier, Grenoble. pp 30-51. Mars (1993).
- [**Falquet 01**] G. Falquet, J.C.L Mottaz, "Navigation Hypertexte dans une Ontologie Multi-Points de Vue". NîmesTIC'2001, Nîmes, France (2001).
- [**Falquet 02**] G. Falquet, C.L. Mottaz Jiang, "A Model for the Collaborative Design of Multi-Point-of-View Terminological Knowledge Bases". Dans R. Dieng and N. Matta (Eds) *Knowledge Management and Organizational Memories*, Kluwer, (2002).
- [**François 03**] J. François, "Les langages du Web Sémantique", INRIA, (Mars 2003).
- [**Furst 02**] F. Furst, "L'ingénierie ontologique". Rapport de recherche N°02-07. (2002).
- [**Fürst 03**] F. Fürst, M. Leclère et F. Trichet, "Construction d'une ontologie opérationnelle : un retour d'expérience", 1re soumission à EGC'2002. (2003).
- [**Gerbé 01**] O. Gerbé, W.G. Mineau et R. K. Keller, "Conceptual graphs and Metamodeling". 9th International Conference on Conceptual Structures. (2001).
- [**Ghidini 01**] C. Ghidini, et F. Giunchiglia, "Local Models Semantics, or Contextual Reasoning = Locality + Compatibility". *Artificial Intelligence*, 127(2):221-259. (2001).
- [**Giunchiglia 94**] F. Giunchiglia, et L. Serafini, "Multilanguage hierarchical logics, or: How we can do without modal logics". Dans *Artificial intelligence*, Vol. 65, No. 1. pp. 29-70. (1994).
- [**Gruber 91**] T. Gruber, "The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases". In *Proceedings of the Second International Conference Principles of Knowledge Representation and Reasoning*, San Mateo, CA, (1991).

- [Gruber 93] T.R. Gruber, "A translation approach to portable ontologies." Knowledge Acquisition, 5(2):199-220, (1993).
- [Gruber 95] T.R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing". International Journal of Human Computer Studies. (1995).
- [Guarino 95a] N. Guarino, et P. Giaretta, "Ontologies and knowledge bases- towards a terminological clarification". Dans N.J. Mars, editor, Towards Very Large Knowledge Bases - Knowledge Building and Knowledge Sharing 1995, pages 25--32. IOS Press, Amsterdam, (1995).
- [Guarino 95b] N. Guarino and P. Giaretta, "Towards very large knowledge bases: knowledge building and knowledge sharing. Chapter: Ontologies and knowledge bases: Towards a terminological clarification". pp 25-32. IOS Press, Amsterdam. (1995).
- [Guarino 97] N. Guarino, "Understanding, building and using ontologies". International J. Human-Computer Studies, pp 293-310. (1997).
- [Haarslev 01] V. Haarslev, R. Möller, "Racer user's guide and reference manual version 1.6". Technical report, University of Hamburg, Computer Science Department, (2001).
- [Handschuh 01] S. Handschuh, "Ontomat. " (2001).
<http://km.aifb.uni-karlsruhe.de/annotation/index.html>.
- [Handschuh 02] S. Handschuh, S. Staab et F. Ciravegna, "S-cream-semi-automatic creation of metadata". Proc. of the European Conference on Knowledge Acquisition and Management. LNCS 2473, Springer-Verlag, Madrid, Espagne, pp. 379-387. (2002).
- [Handschuh 03] S. Handschuh et S. Staab (Eds.), "Frontiers in Artificial Intelligence and Applications", Volume 96, IOS Press, Springer-Verlag, pp. 112-127. (2003).
- [Heflin 01] J. Heflin et J. A. Hendler, "A Portrait of the Semantic Web in Action", in IEEE Intelligent Systems, 16(2), IEEE, pp. 54-59. (2001).
- [Heflin 02] J. Heflin, J. Hendler, S. Luke, C. Gasarch, Q. Zhendong, L. Spector et D. Rager, "The shoe knowledge annotator". (2002).
<http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html>.
- IMARKUP SOLUTIONS (2002). Imarkup. <http://www.imarkup.com/>.
- [Heflin 03] J. Heflin, J. A. Hendler et S. Luke, "Shoe: A blueprint for the semantic web". In Spinning the Semantic Web, p. 29–63. (2003).
- [Hemam 08] M. Hemam, Z. Boufaïda, "Prise en compte des points de vue dans la construction des ontologies en logique de descriptions". Colloque International sur l'Optimisation et les Systèmes d'Information COSI'08, Tizi-Ouzou, Algérie (2008).

- [**Hemam 09a**] M. Hemam, Z. Boufaïda, “Représentation d’ontologies multi-points de vue: une approche basée sur la logique de descriptions”. Papier court dans 20^{es} Journées Francophones d’Ingénierie des Connaissances (IC’09), Tunisie (2009).
- [**Hemam 09b**] M. Hemam, Z. Boufaïda, “Raisonnement par classification sur une ontologie multi-points de vue”. Dans Journées Francophones sur les Ontologies (JFO’09), Poitiers, France, ACM édition, (2009).
- [**Horrocks 02**] I. Horrocks, “DAML+OIL: A Description Logic for the Semantic Web”. IEEE Data Engineering Bulletin Num 1, vol.25, pp 4-9, (2002).
- [**Howard 03**] B. Howard et S.P, Helena, “Overview of Approach, Methodologies, Standards, and Tools for Ontologies”. (2003).
- [**Kahan 01**] J. Kahan, M.R. Koivunen, E. Prud’hommeaux et R. Swick, “Annotea: An Open RDF Infrastructure for Shared Web Annotations”, in Proceedings of the 10th International World Wide Web Conference (WWW’01), ACM Press, Hong-Kong, pp. 623-632. (2001).
- [**Kalyanpur 03**] A. Kalyanpur, B. Parsia, J. Hendler, et J. Golbeck, “SMORE - semantic markup, ontology, and RDF editor.” (2003).
- [**Kushmerick 97**] N. Kushmerick, “Wrapper induction for information extraction”. PhD thesis. Chairperson-Daniel S. Weld. (1997).
- [**López 99**] M. F. López, A. Gómez-Pérez et A. Pazos-Sierra; “Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment”. IEEE Intelligent Systems et their applications. (1999).
- [**Maedche 01**] A. Maedche, B. Motik et R. Volz, “The karlsruhe ontology and semantic web tool suite”. (2001).
<http://kaon.semanticweb.org/>.
- [**Mariño 93**] O. Mariño, “Raisonnement classificatoire dans une représentation à objets multi-points de vue”. Thèse d’informatique, université Joseph Fourier, Grenoble, (1993). <ftp://ftp.inrialpes.fr/pub/sherpa/theses/marino.ps.gz>
- [**Michard 99**] A. Michard, “XML : langage et applications” Eyrolles ISBN 2-212-09052-8 (1999).
- [**Miller 98**] E. Miller, “An Introduction to the Resource Description Framework”. (1998).
- [**Napoli 97**] A. Napoli, “Une introduction aux logiques de descriptions”. Rapport de recherche N° 3314. (1997).

- [**Nguyen 92**] G. T. Nguyen, D. Rieu, et J. Escamilla, “An Object Model for Engineering Design.” Dans Proc. of European Conference on Object-Oriented Programming. ECOOP’92. Lecture Notes in Computer Science vol. 615. Springer-Verlag, (1992).
- [**Noy 01**] N. Noy et D. McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology”, Stanford Medical Informatics Report, SMI-2001-0880. (2001).
- [**Paolo 03**] B. Paolo, G. Fausto, V.H. Frank, S. Luciano, et S. Heiner, “C-OWL: Contextualizing Ontologies”. International Semantic Web Conference: 164-179 (2003).
- [**Popov 03**] B. Popov, A. Kiryakov, D. Manov, A. Kirilov, D. Ognyanoff et M. Goranov, “Towards Semantic Web Information Extraction”, in Proceedings of the Human Language Technologies Workshop (ISWC’03), Sanibel, Floride, pp. 1-22. (2003).
- [**RDFS 00**] RDFS, “Resource Description Framework Schema Specification”, (2000). <http://www.w3.org/TR/2000/CR-rdf-schema20000327/>
- [**Rivière 02**] M. Rivière, R. Dieng, “A Viewpoint Model for Cooperative Building of Ontology”. ICCS 2002: 220-234. (2002).
- [**Rivière 97**] M. Rivière, et R. Dieng, “Introduction of viewpoints in conceptual graph formalism.”In 5th International Conference on Conceptual Structures”, ICCS’97, Washington, USA, LNAI 1257, Springer Verlag, pp. 168-182, (1997).
- [**Rivière 98**] M. Rivière, “Using Viewpoints and CG for the Representation and Management of a corporate memory in Concurrent Engineering.” Proc. of the 6th Int. Conference on Conceptual Structures (ICCS’98), Montpellier, August 10-12, 1998, Springer-Verlag, LNAI 1453. (1998).
- [**Rivière 99**] M. Rivière, “Représentation et gestion de multiples points de vue dans le formalisme des graphes conceptuels”. Thèse de doctorat en informatique, Nice-Sophia Antipolis, (1999).
- [**Sandra 04**] B. Sandra, B. Catherine, C. Jean, “Les documents et les annotations du dossier patient hospitalier, Information” - Interaction – Intelligence, (2004).
- [**Smith 03**] B. Smith, J. Williams and S. Schulze-Kremer, “The Ontology of the Gene Ontology, in Biomedical and Health Informatics: From Foundations to Applications, Proceedings of the Annual Symposium of the American Medical Informatics Association.”Washington DC, 609–613. (November 2003).
- [**Sowa 84**] J. Sowa, “Conceptual Structures: Information Processing In Mind and Machine”. Addison-Wesley, (1984).

- [**Staab 01**] S. Staab, A. Maedche et S. Handschuh, “An Annotation Framework for the Semantic Web”, in Proceedings of the 1st International Workshop on MultiMedia Annotation, Tokyo, Japon, (2001).
- [**Sylvain 07**] T. Sylvain, T. Yannick, “Classes d’annotation pour l’annotation sémantique”. inria-00196064, version 1 - 12 (2007).
- [**Sylvain 07 a**] T. Sylvain, N. Amedeo, P. Xavier, T. Yannick, “Annotation sémantique de pages web” inria-00196048, version 1 – (12 décembre 2007).
- [**Sylvain 07 b**] T. Sylvain, T. Yannick, “Annotation sémantique par classification”. inria-00196058, version 1 – (12 décembre 2007).
- [**Uren 06**] V. Uren, P. Cimiano, S. Handschuh, M. Vargaz-vera, E. Motta et F. Ciravegna, “Semantic annotation for knowledge management: requirements and a survey of the state of the art”, in Journal of Web Semantics, Science, Services and Agents on the World Wide Web, 4(1), Elsevier, pp.14- 26. (2006).
- [**Uschold 95**] M. Uschold et M. King, “Towards a Methodology for Bbuilding Ontologies”, in Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI’95. (1995).
- [**Uschold 96**] M. Uschold et M. Grüninger, “ONTOLOGIES: Principles, Methods and Applications”. Knowledge Engineering Review. (1996).
- [**Vargaz-vera 02 a**] M. Vargaz-vera, E. Motta, J. Domingue, M. Lanzoni, A. Stutt et F. Ciravegna, “MnM: Ontology Driven Tool for Semantic Markup”, in Proceedings of the Workshop on Semantic Authoring, Annotation et Knowledge Markup (SAAKM’02), Lyon, France, (2002).
- [**Vargaz-vera 02 b**] M. Vargaz-vera, E. Motta, J. Domingue, M. Lanzoni, A. Stutt et F. Ciravegna, “MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup”, in Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW’02), LNCS 2473, Springer-Verlag, Madrid, Espagne, , pp. 379-391. (2002).
- [**W3C 03**] **Web** Ontology Language Reference Version 1.0. W3C Working Draft (Mars 2003). <http://www.w3.org/TR/owl-ref/>.
- [**W3C 99**] “Resource Description Framework (RDF) Model and Syntax Specification”. W3C Recommendation 22 February (1999).

Glossaire

Ce glossaire regroupe et donne la définition des principaux termes utilisés dans ce mémoire

DL : Description Logic

OWL: Ontology Web Langage

RACER: Renamed Abox and Concept Expression Reasoner

HTML: Hypertext Markup Langage

XML : eXtensible Markup Langage

DOM Document Object Model

DTD : Document Type Definition

RDF: Resource Description Framework

URI : Uniform Resource Identifiers

URL: Uniform Resource Locator

W3C: World Wide Web Consortium

WWW: World Wide Web

PV: Point de Vue

MVP-OWL: Multi View Point Ontology Web Langage

PHP: Hypertext Preprocessor