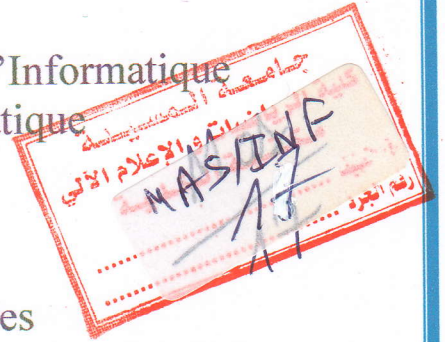


Mustapha

Ministère de l'enseignement supérieur
et de la recherche scientifique
Université de M'sila
Faculté des Mathématiques et de l'Informatique
Département de l'informatique



Mémoire de fin d'études
Pour l'obtention de Diplôme de **Master** en Informatique

Domaine : Mathématiques et Informatique
Filière : Informatique
Spécialité : Système d'informations avancées

Thème

Vérification des ontologies avec Prolog

Réalisé par :
AISSAT Aicha

Dirigé par :
Dr. BOURAHLA Mustapha

2010 /2011

Table des matières

Introduction générale

Chapitre 01 : Les ontologies et logique de description

Les ontologies	6
Introduction	6
1 L'Ontologie, qu'est ce que c'est ?	6
1.1 Définitions	6
1.2 Types de composantes	7
2 Ingénierie ontologique (IO)	7
2.1 Langages de construction d'ontologies	7
2.1.1 Les langages traditionnels	8
2.1.2 Les langages du Web (mark up)	8
3 Types d'ontologies	9
4 Grands modèles informatiques d'ontologies	9
La logique de description	10
1 Connaissances et représentation de connaissances	10
1.1 Approche logique de la représentation de connaissances	10
2 Logique de description	10
2.1 Langage de base AL	11
2.2 Langages de la famille AL	11
2.2.1 Les constructeurs de la famille AL	11
2.3 Syntaxe et sémantique	12
2.3.1 Description de concept	13
2.3.2 Terminologie	14
2.3.3 Description de monde	15
2.3.4 Inférence	16
2.4 Relations entre la logique de description et d'autres formalismes	16
2.4.1 Graphe conceptuel	16
2.4.2 Modèle de données orienté-objet	17
OWL (Web Ontology Langage)	18
1 Le langage OWL	18
2 Objectifs et motivation	18
3 Structure d'une ontologie OWL	18
3.1 Espaces de nommage	19
3.2 En-têtes d'une ontologie	20
4 Eléments du langage	20
4.1 Les classe	20
4.2 Héritage	22
4.3 Définition d'un individu	22
4.4 Les propriétés	23
Conclusion	25

Chapitre 02: Langage de la programmation logique(Prolog)

Introduction	28
1 La programmation logique	28
1.1 Un mode de programmation à part	29
1.1.1 Les autres modes de programmation	29
1.1.1.1 La programmation impérative	29

1.1.1.2	La programmation fonctionnelle	29
1.2	La logique du premier ordre classique et sa sémantique	29
2	Prolog	30
2.1	Un survol de Prolog	31
2.2	Notion de programme Prolog	31
2.3	Constitution d'un programme prolog	32
2.3.1	Les faits	32
2.3.2	Les règles	33
2.3.3	Les questions	34
2.3.4	Les variables (inconnues)	35
2.3.5	Les conjonctions	36
2.4	Création de programmes	36
3	SWI-Prolog	37
3.1	Les conventions de SWI-PROLOG	38
4	Exécution d'un programme PROLOG	39
4.1	L'interactivité par les requêtes	39
4.1.1	Des questions et des réponses	39
4.1.2	Un programme évolutif	40
4.2	La recherche de preuves par l'unification	42
4.2.1	Fonctionnement général	42
4.2.1.1	La logique	42
4.2.1.2	L'implémentation:	42
	Conclusion	43
Chapitre 03: les règles de transformation et d'inférence		
	Introduction	46
	Partie 01 : Les règles de transformation	46
	Partie 02 : Les règles de vérification	54
	Conclusion	64
Chapitre 04: Test et vérification avec un exemple		
	Introduction	67
	Exemple d'ontologie OWL (description d'une population)	67
2	Présentation de l'ontologie en UML	69
3	Présentation de l'ontologie en OWL	70
4	Le code OWL	73
5	Transformation le code OWL en Prolog	82
6	quelques tests	91
	Conclusion	93
	Conclusion générale	94
	Bibliographie	
	Annexes	

Introduction générale

L'utilisation de connaissances dans les systèmes informatiques a pour but d'améliorer les mécanismes existants en accédant à la sémantique associée à l'information traitée.

Une ontologie est vue comme un moyen de « décrire de façon explicite la conceptualisation des connaissances représentées dans une base de connaissances » [BENJAM99]. Cette définition est une extension de celle de Gruber (1993), décrivant une ontologie comme « une spécification explicite d'une conceptualisation ».

Ontologie est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations, que ce soit par les métadonnées d'un espace de noms, ou les éléments d'un domaine de connaissances. L'ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts. Elle est employée pour raisonner à propos des objets du domaine concerné. Les concepts sont organisés dans un graphe dont les relations peuvent être :

- Des relations sémantiques,
- Des relations de subsomption (inclusion).

Une ontologie est une représentation structurée des connaissances d'un domaine, sous forme d'un réseau de concepts reliés entre eux par des liens sémantiques. Les recherches sur les ontologies sont essentielles pour la réalisation du Web sémantique. En effet une fois construite et acceptée par une communauté particulière, une ontologie doit traduire un certain consensus explicite et un certain niveau de partage qui sont essentiels pour permettre l'exploitation des ressources Web par différentes applications ou agents logiciels.

Pour représenter les ontologies, le W3C propose un standard qui est le (OWL *Ontology Web Language*). OWL est tout comme RDF un langage XML profitant de l'universalité syntaxique du XML. OWL offre la possibilité d'écrire des ontologies Web.

Dans une représentation logique, la base de connaissances consiste en un ensemble d'axiomes décrivant une situation, un état de choses, sur lesquels des règles d'inférence

opèrent et fournissent de nouvelles formules que l'on peut considérer comme valides. Celles-ci constituent alors de nouveaux états de choses dans la base. Le langage de programmation Prolog est fondé directement sur ces principes.

Il existe plusieurs travaux dans ce domaine. Comme transformation des ontologies vers UML, développement des ontologies avec protégé, et transformation des ontologies vers Prolog par le processeur (XSLTs) mais notre projet se situe sur la transformation des ontologies vers Prolog manuellement à partir d'un ensemble de règles qui permet la transformation. Le but de notre travail est de concevoir et vérifier la consistance des ontologies avec Prolog.

Donc l'objectif de notre travail est la proposition des règles de transformation et de vérification. Ces règles seront testées sur un exemple.

Ce mémoire est organisé en quatre chapitres:

➤ **Le premier chapitre: les ontologies et logique de description**

Nous allons faire une partie de l'état de l'art d'ontologie et logique de description, effectivement, les notions de base, et langage de représentation des ontologies qui sont basées sur la logique de description avec des exemples de codes sources pour définir les classes, les liens sémantiques (les propriétés), les hiérarchies.

➤ **Le deuxième chapitre: programmation logique (Prolog):**

Tout d'abord insistons-nous sur la logique du premier ordre et sur le fait que Prolog est un langage de programmation logique, langage déclaratif, particulièrement pour résoudre tous les problèmes ayant une relation avec la logique. Comme nous l'avons vu l'exécution d'un programme Prolog.

➤ **Le troisième chapitre: les règles de transformation :**

Nous proposons des règles correspondants à certains codes OWL pour transformer les ontologies en OWL vers Prolog, et des règles Prolog pour vérifier les ontologies.

➤ **Le dernier chapitre: Test et vérification avec un exemple :**

Permet d'envisager la vérification des ontologies avec Prolog. Pour réaliser l'objectif à partir des étapes suivantes :

Chapitre 01 - Les ontologies et logique de description -

- 1- représentation du code avec protégé en graphe et écrire le code source en OWL et en UML (diagramme de class)
- 2- transformé le code en OWL vers Prolog à partir les règles de transformation présentées dans le chapitre 03
- 3- vérifier la consistance d'ontologie

Ce mémoire est conclu avec un bilan expliquant le travail réalisé et donnant quelques perspectives.

*Chapitre 01
Les ontologies et
Logique de Description*

Conclusion générale

Conclusion

Dans ce chapitre nous avons fait la partie des tests avec générer le code OWL de l'exemple, transformation le code OWL de l'exemple qui présenté notre étude et faire des testes sur les règles de vérification avec un exemple (vérification l'ontologie).

Cette transformation est manuelle et elle est basée sur un ensemble de règles de transformation que nous avons proposées.

Pour tester la consistence des ontologies avec Prolog, nous avons développé un programme Prolog que nous appelons le raisonneur. Ce raisonneur nous permet de former des règles très utiles pour la vérification et ainsi la validation des ontologies.

Dans ce mémoire, nous avons proposé un exemple (étude de cas) pour tester notre approche qui est basé sur un ensemble de règles de transformation et de vérification.

Comme perspective nous souhaitons l'automatisation de cette transformation, ainsi la proposition des règles pour des cas non traités.

Bibliographie

Conclusion générale

Dans ce mémoire, nous avons présenté un travail pour la transformation des ontologies décrites avec le langage OWL vers des ontologies correspondantes avec Prolog. Cette transformation est manuelle et elle est basée sur un ensemble de règles de transformation que nous avons proposées.

Pour tester la consistance des ontologies avec Prolog, nous avons développé un programme Prolog que nous appelons le raisonneur. Ce raisonneur nous permet de formuler des requêtes très utiles pour la vérification et ainsi la validation des ontologies.

Dans ce mémoire, nous avons proposé un exemple (étude de cas) pour tester notre approche qui est basée sur un ensemble de règles de transformation et de vérification.

Comme perspective nous souhaitons l'automatisation de cette transformation, ainsi la proposition des règles pour des cas non traités.

[6] Michel Gagnon, Logique descriptive et OWL, page 3813
http://www.cours.polymtl.ca/info410/Documents/logique_descriptive.pdf

[7] Programmation fonctionnelle et logique, licence informatique, 2001/2002
<http://perso.crans.org/martin/Doe/CoursPFL/Cours%20PROLOG.pdf>

[8] Programmation logique (1/2) 'se trouve en'
http://lamp.epfl.ch/teaching/programmation_avancee/documents/programmation_avancee_13-2x2.pdf

[9] Introduction à la programmation logique
<http://members.unine.ch/jacques.savoy/lectures/SemCL/Prolog.pdf>

[10] Sowa, J. F. (1984). Conceptual structures: Information Processing in Mind and Machine. Addison Wesley.

[11] Hopcroft, J.E., Ullman, J.D. (1979). Introduction to Automata Theory, Languages And Computation. Addison Wesley.

Les sites internet

<http://www.swi-prolog.org>

Bibliographie

- [1] Julien Dallier, Thèse professionnel master-base de données progiciel L5Get web sémantique, page 23, 2005 <http://www.gnu.org/copyleft/fdl.html>.
- [2] Bernard Epinasse, professeur a l'université d'Aix-Marseille, Introduction aux ontologies, pages 14 à 21 -2010 <http://www.lsis.org/epinasseb/Supports/ONTOWS-2010/Onto-2010-4p.pdf>
- [3] Valéry Psyché, L'ontologie, cours le cadre du cours DIC9340 systèmes a base de connaissances <http://www.info2.uqam.ca/~nkambou/DIC9340/seances/seance2-3-4-6/seance6.pdf>
- [4] Xavier Lacot, Introduction aux OWL- langage XML d'ontologie web, juin 2005 http://lacot.org/public/introduction_a_owl.pdf
- [5] C.Leduc, Thèse, Transformation des ontologies basées sur logique de description, application dans le commerce électronique, partie1, pages 13, 14, 17,25, http://www.i3s.unice.fr/~nlt/telechargement/theses/040706_CLeduc.pdf
- [6] Michel Gagnon, Logique descriptive et OWL, page 3à13 http://www.cours.polymtl.ca/inf6410/Documents/logique_descriptive.pdf
- [7] Programmation fonctionnelle et logique, licence informatique, 2001/2002 <http://perso.crans.org/martin/Doc/Cours/PFL/Cours%20PROLOG.pdf>
- [8] Programmation logique (1/2) 'se trouve en ' http://lamp.epfl.ch/teaching/programmation_avancee/documents/programmation_avancee_13-2x2.pdf
- [9] Introduction à la programmation logique <http://members.unine.ch/jacques.savoy/lectures/SemCL/Prolog.pdf>
- [10] Sowa, J. F. (1984). Conceptual structures: Information Processing in Mind and Machine. Addison Wesley.
- [11] Hopcroft, J.E., Ullman, J.D. (1979). Introduction to Automata Theory, Languages And Computation. Addison Wesley.

Les sites internet

<http://www.swi-prolog.org>

المخلص:

لكي نسهل عملية تحويل الأنتولوجيا و التحقق منها نفرض حل لهذا المشكل عن طريق مجموعة قواعد للتحويل و مجموعة قواعد أخرى عامة للتحقق من ثبات الأنتولوجيا

Ontologie, OWL, Les règles de transformation, Les règles de vérification, Prolog **كلمات-المفتاح:**

RESUME:

Afin de faciliter la transformation et la vérification des ontologies avec Prolog, nous proposons une solution à ce problème par la proposition d'un ensemble de règles de transformation et de raisonnement (vérification) pour vérifier la consistance d'ontologie

MOT-CLES: Ontologie, OWL, Les règles de transformation, Les règles de vérification, Prolog.

ABSTRACT:

In order to ease the processing and verification of an ontology with Prolog, we propose a solution to this problem by proposing a set of rules for transformation and verification to check the consistency of ontologies.

KEY-WORD: Ontology, OWL, Transformation rules, Verification rules, Prolog.