



**HAL**  
open science

## OWLet la Description de Ressources Pédagogiques

Anfana Traoré, Danièle Hérin

► **To cite this version:**

Anfana Traoré, Danièle Hérin. OWLet la Description de Ressources Pédagogiques. [Rapport de recherche] 04055, LIRMM (UM, CNRS). 2004, pp.10. lirmm-00109193

**HAL Id: lirmm-00109193**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00109193>**

Submitted on 24 Oct 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# OWL et la description de ressources pédagogiques

Anfana Traoré, Danièle Hérim

Laboratoire d'Informatique Robotique Micro-électronique de Montpellier  
Université de Montpellier II  
161 rue ada 34392 Montpellier Cedex 5, France  
{traore, dh}@lirmm.fr  
WorkShop 2 AS-WebLearn - Montpellier, 22-23 mars 2004

## 1 Introduction

OWL est destiné à être utilisé quand l'information contenue dans les documents doit être traitée par des applications, par opposition aux situations où le contenu doit seulement être présenté (i.e. la forme). OWL est employé pour représenter explicitement la signification (i.e. le fond) des termes dans les vocabulaires et les relations entre ces termes. Cette représentation des termes et de leurs corrélations s'appelle une ontologie. OWL offre plus de facilités pour exprimer la signification et la sémantique que XML, RDF et RDF-S. OWL va ainsi au-delà de ces langages dans sa capacité de représenter le contenu compréhensible par une machine sur le Web. OWL est une révision du langage d'ontologie du Web DAML+OIL intégrant les leçons apprises de la conception et de l'application de DAML+OIL. [4]

L'objet de ce papier est de présenter les fonctionnalités offertes par OWL (i.e. ce que l'on peut représenter avec OWL) et de les illustrer dans le contexte de la description de ressources pédagogiques.

### 1.1 Pourquoi OWL ?

Le Web sémantique est une vision futuriste du Web dans laquelle l'information est donnée avec une signification explicite afin de faciliter le traitement automatique et l'intégration de l'information disponible sur le Web. Le Web sémantique repose à la fois sur la capacité de XML à définir des schémas tagués adaptés et sur la flexibilité de l'approche RDF à représenter les données. Le premier niveau au-dessus de RDF requis pour le Web sémantique est un langage d'ontologie qui peut décrire formellement la signification de la terminologie utilisée dans les documents Web. Si on veut que les machines accomplissent des tâches de raisonnement utiles sur ces documents, le langage doit aller au-delà de la sémantique de base de RDF Schéma. OWL a été conçu pour satisfaire le besoin d'un langage d'ontologie du Web.

OWL ajoute plus de vocabulaire pour décrire les propriétés et les classes. On peut citer entre autres : les relations entre classes (par exemple la disjonction), les cardinalités (par exemple exactement un), l'égalité, typage plus riche des propriétés, caractéristiques des propriétés (par exemple la symétrie) et les classes énumérées.

## 1.2 Principales caractéristiques de OWL/Usage des trois niveaux

OWL fournit trois sous langages de plus en plus expressifs conçus pour l'usage des communautés spécifiques des utilisateurs et des développeurs.

OWL Lite convient aux utilisateurs qui ont principalement besoin d'une hiérarchie de classification et de contraintes simples. Par exemple, alors qu'il supporte des contraintes de cardinalité, il autorise seulement des valeurs de cardinalité 0 ou 1. Il devrait être plus simple de fournir un outil d'aide pour OWL Lite que ses parents plus expressifs. OWL Lite fournit un chemin rapide de migration pour les thésaurus et d'autres taxonomies.

OWL DL convient aux utilisateurs qui veulent le maximum d'expressivité tout en maintenant la complétude (toutes les conclusions sont calculées) et la décidabilité (tous les calculs s'effectuent dans un temps fini). OWL DL inclut tous les constructeurs du langage OWL, mais ils peuvent être utilisés seulement sous certaines restrictions (par exemple, lorsqu'une classe est sous classe de plusieurs classes, elle ne peut pas être instance d'une autre classe). OWL DL est appelé ainsi en raison de sa correspondance avec les logiques de description.

OWL Full convient aux utilisateurs qui veulent le maximum d'expressivité et la liberté syntaxique de RDF sans aucune garantie sur la conclusion. Par exemple en OWL Full, une classe peut être traitée simultanément comme une collection d'individus et comme un individu "in its own right". OWL Full permet à une ontologie<sup>1</sup> d'augmenter la signification de son vocabulaire (RDF ou OWL) prédéfini. Il est peu probable que n'importe quel logiciel de raisonnement pourra approuver le raisonnement complet pour chaque caractéristique de OWL Full.

Chacun de ces sous langages est une extension de son prédécesseur, à la fois dans ce qui peut être légalement exprimé et dans ce qui peut être bien conclu.

Dans la suite, nous décrirons d'abord les caractéristiques de OWL Lite<sup>2</sup>, suivi d'une description des caractéristiques qui sont ajoutées dans OWL DL et OWL Full. OWL DL et OWL Full contiennent les mêmes caractéristiques mais OWL Full est plus souple sur la façon dont ces caractéristiques peuvent être combinées.

## 2 OWL Lite

Les préfixes `rdf:` ou `rdfs:` sont employés lorsque les termes sont déjà présents dans RDF ou RDF Schéma. Sinon les termes sont introduits par OWL. Ainsi le terme

---

<sup>1</sup> Une ontologie décrit un ensemble de concepts et les liens entre ces concepts.

<sup>2</sup> OWL Lite uses only some of the OWL language features and has more limitations on the use of the features than OWL DL or OWL Full. ... are also only allowed between named classes, ... Similarly, restrictions in OWL Lite classes can only be defined in terms of named superclasses (superclasses cannot be arbitrary expressions), and only certain kinds of class restrictions can be used. Equivalence between classes and subclass relationships between classes are also only allowed between named classes, and not between arbitrary class expressions. Similarly, restrictions in OWL Lite use only named classes. OWL Lite also has a limited notion of cardinality - the only cardinalities allowed to be explicitly stated are 0 or 1.

`rdfs:subPropertyOf` indique que `subPropertyOf` est déjà dans le vocabulaire de RDF-S. En outre, le terme `class`, plus précisément énoncé comme `owl:class` est un terme introduit par OWL.

## 2.1 Classes

Les classes fournissent un mécanisme d'abstraction pour regrouper des ressources ayant des caractéristiques communes. Comme les classes RDF, toute classe OWL est associée à un ensemble d'individus appelé l'extension de la classe.

### 2.1.1 Restrictions de propriété

Une restriction de propriété est une sorte de description de classe. Elle définit une classe anonyme, une classe dont tous les individus satisfont la restriction. OWL distingue deux sortes de restrictions : les contraintes de valeur et les contraintes de cardinalité.

La contrainte de valeur est une propriété qui lie une classe de restriction à une classe de description ou à un range de donnée. OWL Lite inclut une forme limitée de restrictions de cardinalité. **Les restrictions de cardinalité de OWL (et OWL Lite) sont des restrictions locales** puisqu'elles sont énoncées sur des propriétés en ce qui concerne une classe particulière.

### 2.1.2 Intersection

L'intersection est un constructeur de classe employé dans la logique de description. Il peut être vu comme l'opérateur AND sur les classes.

## 2.2 Propriétés

OWL distingue deux types de propriétés :

??Les propriétés d'objet ont un "range" de valeur d'individus de classe et lie ainsi les individus aux individus.

??Les propriétés de type de donnée ont un range de valeur de données, et lie ainsi les individus aux valeurs de données.

Souvent, les axiomes de propriété définissent les caractéristiques additionnelles de propriétés. OWL supporte les constructeurs suivants :

??Les constructeurs RDF-S : `rdfs:subPropertyOf`, `rdfs:domain`, et `rdfs:range`

??Les relations à d'autres propriétés : `owl:equivalentProperty` et `owl:inverseOf`

??Les contraintes globales de cardinalité : `owl:FunctionalProperty` et `owl:InverseFunctionalProperty`

??Les caractéristiques logiques de propriété : `owl:SymmetricProperty` et `owl:TransitiveProperty`

### 2.3 Individus

OWL fournit trois constructeurs concernant l'identité des individus :

?? `owl:sameAs` est utilisé pour énoncer que deux références se rapportent à un même individu. Le constructeur `owl:sameIndividualAs` est un synonyme de `owl:sameAs`.

?? `owl:differentFrom` est utilisé pour énoncer que deux références se rapportent à des individus différents.

?? `owl:AllDifferent` est utilisé pour énoncer qu'une liste d'individus sont tous différents.

### 2.4 Types de données

OWL utilise les mécanismes de RDF pour les valeurs de données [1]. Les valeurs de données sont des instances de la classe RDF-Schéma `rdfs:Literal`. Les littéraux peuvent être typés ou non. Les types de données sont des instances de la classe `rdfs:Datatype`. Les types de données XML Schéma recommandés pour les ontologies de OWL sont les suivants :

?? Le type de données primitif `xsd:string` et ses dérivés suivants : `xsd:normalizedString`, `xsd:token`, `xsd:language`, `xsd:NMTOKEN`, `xsd:Name`, et `xsd:NCName`.

?? Le type de donnée primitif `xsd:boolean`.

?? Le type de donnée numérique primitif `xsd:decimal`, `xsd:float`, et `xsd:double`, et les dérivés du type `xsd:decimal` (`xsd:integer`, `xsd:positiveInteger`, `xsd:nonPositiveInteger`, `xsd:negativeInteger`, `xsd:nonNegativeInteger`, `xsd:long`, `xsd:int`, `xsd:short`, `xsd:byte`, `xsd:unsignedLong`, `xsd:unsignedInt`, `xsd:unsignedShort`, `xsd:unsignedByte`).

?? Les types de données primitifs relatifs au temps : `xsd:dateTime`, `xsd:time`, `xsd:date`, `xsd:gYearMonth`, `xsd:gYear`, `xsd:gMonthDay`, `xsd:gDay`, et `xsd:gMonth`.

?? Les types de données primitifs `xsd:hexBinary`, `xsd:base64Binary`, et `xsd:anyURI`.

### 2.5 Information d'en-tête[2]

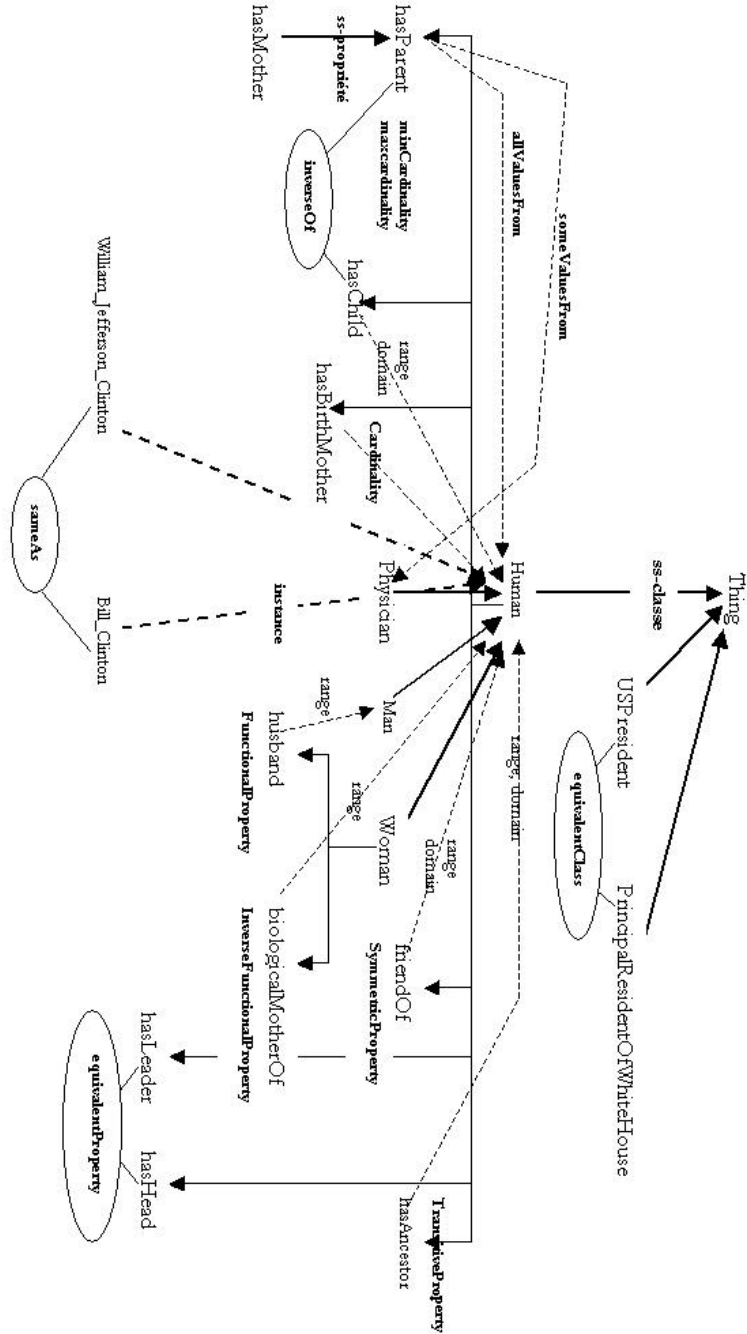
OWL Full ne pose aucune contrainte sur les annotations dans une ontologie. OWL DL permet les annotations sur les classes, propriétés, individus et en-têtes d'ontologie, mais seulement dans les conditions suivantes :

?? Les propriétés d'annotations doivent avoir un typage explicite de la forme :

?? `AnnotationPropertyID rdf:type owl:AnnotationProperty`

?? Les propriétés d'annotations ne doivent pas être employées dans les axiomes de propriétés (par exemple, les sous propriétés ne sont pas autorisées).

2.6 Synthèse de OWL Lite



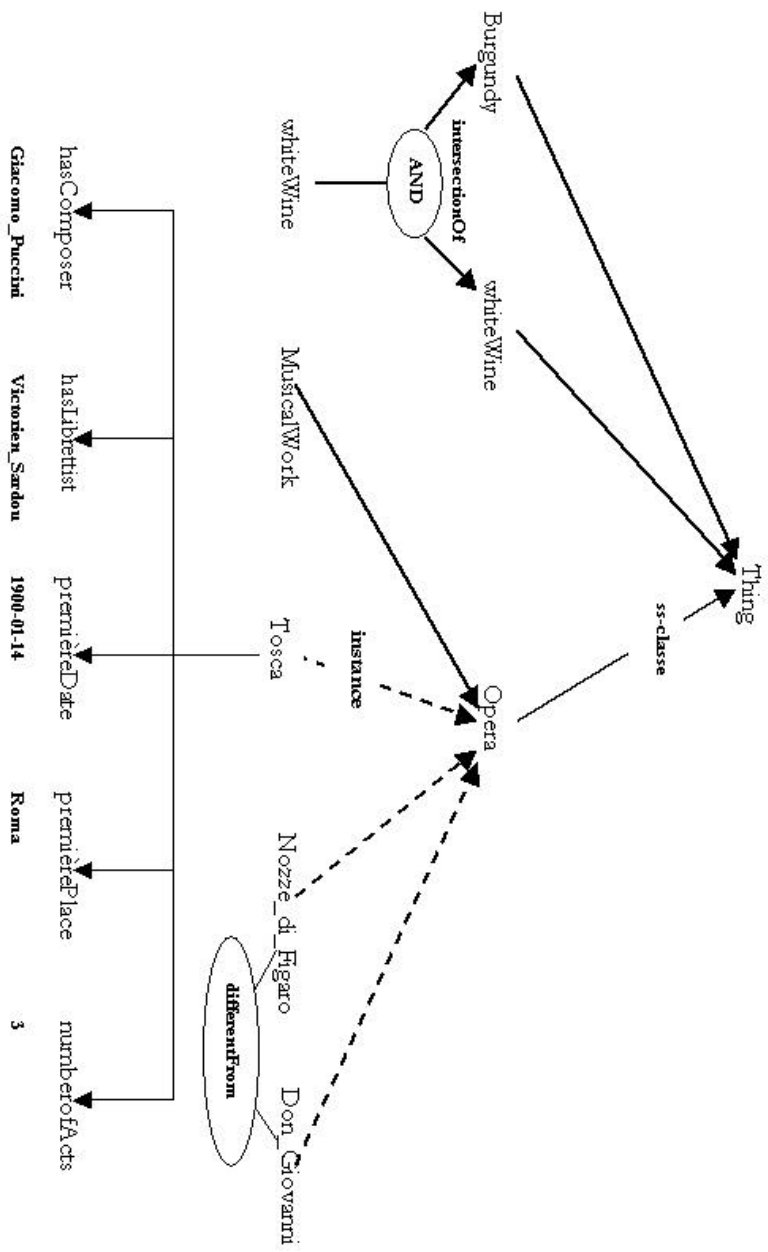


Fig. 1. Un exemple de description visuelle des concepts de OWL Lite

### 3 Extensions de Lite : DL et Full

En OWL Full, `owl:Thing` est équivalent à `rdf:ressource`, `owl:Class` est équivalent à `rdfs:Class`. OWL Full traite les classes comme des instances. `owl:ObjectProperty` est équivalent à `rdf:Property`. Ce qui entraîne que `DataProperty` est sous classe de `ObjectProperty`. `DataProperty` et `ObjectProperty` ne sont pas disjointes. OWL Full contient tous les constructeurs de OWL mais ne fournit aucune contrainte quant à leur utilisation.

Par contre dans OWL Lite et OWL DL, `owl:Class` est sous classe de `rdfs:Class`. OWL DL place un nombre de contraintes sur l'utilisation des constructeurs. `DataProperty` et `ObjectProperty` sont disjointes. Les propriétés suivantes ne peuvent pas être appliquées aux propriétés de type de donnée : `inverseOf`, `inverse` fonctionnelle, `symétrie` et la `transitivité`. Il n'y a pas de contraintes de cardinalité (locale ni globale) sur les propriétés transitives, ou leur inverse ou leur super propriété. Tous les axiomes doivent être bien formés. Les axiomes (faits) à propos de l'égalité et la différence doivent être sur des individus nommés.

OWL Lite respecte toutes les restrictions imposées par OWL DL mais interdit l'usage de : `owl:oneOf`, `owl:unionOf`, `owl:complementOf`, `owl:hasValue`, `owl:disjointWith` et `owl:DataRange`.

## 4 Application à la description des ressources pédagogiques

### 4.1 Les méta données standards et annotations [3][4]

OWL décrit une ressource à partir d'un nom unique (`rdf:ID`), d'un commentaire (`rdfs:comment`), de labels dans de multiples langues (`rdfs:label`), d'une liste de ressources qui sont identiques (`owl:sameAs`), et d'une liste de ressources qui ne sont pas identiques (`owl:differentFrom`).

Les annotations sont des commentaires sur la ressource. Elles n'ont pas de sémantique formelle, elles sont ignorées pendant le raisonnement. Il existe des annotations prédéfinies mais l'utilisateur peut définir ses propres annotations. Par exemple, les informations sur la version (OWL-VERSION-INFO), les informations sur l'auteur (OWL-IS-DEFINED-BY).



## 4.2 Propriétés

### 4.2.1 Relation entre propriétés : Inverse

Si la propriété P1 est l'inverse de la propriété P2 alors, si X est relié à Y par la propriété P1, alors Y est relié à X par la propriété P2. Dans la description du plan du cours, les propriétés nœud père et nœud fils gauche sont inverses, c'est-à-dire si X est fils gauche du nœud Y alors Y est nœud père de X.

### 4.2.2 Caractéristiques logiques des propriétés : Transitivité

Pour décrire la structure arborescente du plan du cours, on exploite la caractéristique de la transitivité du nœud père et du nœud fils gauche. Ainsi, si le paragraphe C2 contient le paragraphe 2.1 et le paragraphe 2.1 contient le paragraphe 2.1.1 alors le paragraphe C2 contient le paragraphe 2.1.1.

## 4.3 Description de classe

### 4.3.1 Contrainte de valeur : hasValue

Elle lie une classe à une restriction de valeur V. V peut être un individu ou une donnée. Stanford\_Cours\_Info décrit une classe dont tous les individus ont une valeur sémantiquement égale à STANFORD en ce qui concerne la propriété Université.

### 4.3.2 Combinaison booléennes : Union

Cette combinaison décrit une classe anonyme dont l'extension contient les individus qui sont dans au moins une des extensions des classes de la liste de range. Par exemple, la classe Ressource est l'union des supports de Cours, des Exercices, des Corrigés, des Examens et des Faqs.

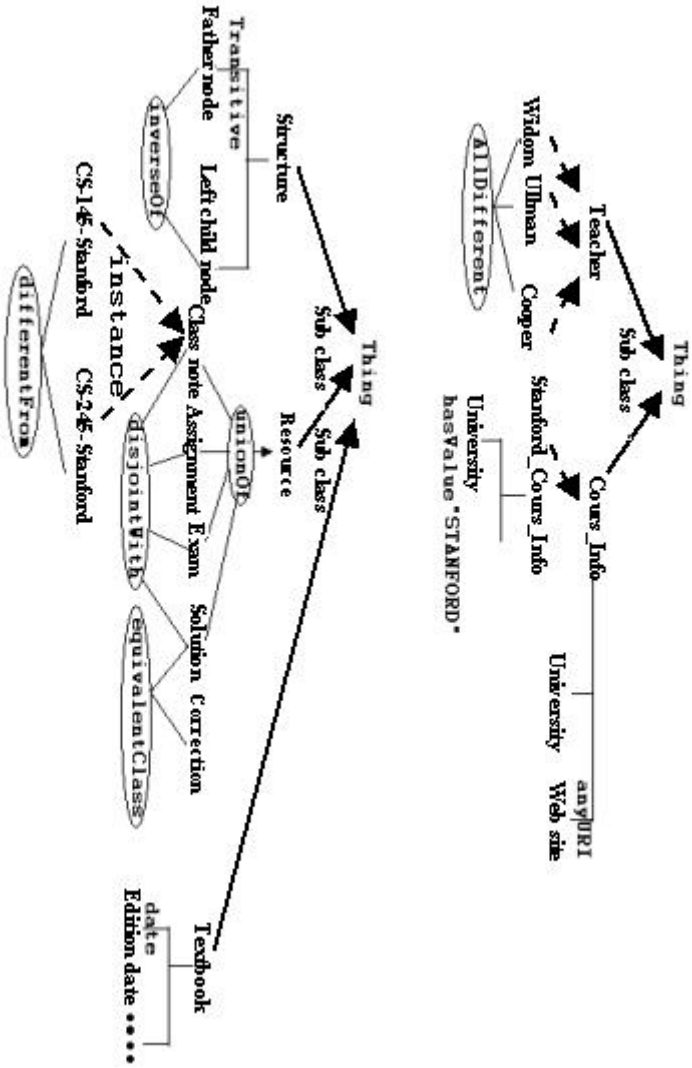
### 4.3.2 Axiome de classe : Disjonction

La disjonction exprime que les extensions de deux classes n'ont aucun individu en commun. Par exemple, les supports de Cours, les Exercices, les Corrigés, les Examens et les Faqs sont deux à deux disjoints.

## 4.4 Type de données

Pour faire le lien entre la méta donnée et la ressource, on dispose du type de donnée anyURI. Dans la mise œuvre de ce lien, on utilise plutôt le type "String" qui est ensuite interfacé par le type URL Widget.

4.5 Synthèse schématique



**Fig. 2.** Un exemple de représentation visuelle synthétique de description des ressources pédagogiques avec OWL

## 5 Conclusion

Nous avons donné une vue d'ensemble du langage d'ontologie du Web OWL en fournissant une brève introduction sur la nécessité d'un langage d'ontologie du Web. Nous avons aussi présenté les trois niveaux de OWL à savoir Lite (le noyau), DL (les logiques de description) et Full (le niveau méta).

L'application de OWL à la description des ressources pédagogiques montre l'étendue de son pouvoir expressif et son adéquation au domaine du e-learning.

## 6 Références

1. Sean Bechofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein, *OWL Web Ontology Language Reference*, Editors Mike Dean, Guus Schreiber, Editor's Draft document – no status at this time June 19 2003. <http://www.daml.org/2002/06/webont/owl-ref-proposed#datatype>
2. Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein, *OWL Web Ontology Language Reference*, Editors Mike Dean, Guus Schreiber, W3C Working Draft March 31, 2003. <http://www.w3.org/TR/owl-ref/#Header>
3. Holger Knublauch, Mark A. Musen, Natasha F. Noy, *Tutorial : Creating Semantic Web (OWL) Ontologies with Protégé*, <http://protege.stanford.edu/plugins/owl/download/Protege-OWL-Tutorial-ISWC03.pdf> 2<sup>nd</sup> International Semantic Web Conference (ISWC2003), Sanibel Island, Florida, USA, October 20-23<sup>th</sup>, 2003
4. Deborah L. McGuinness (Knowledge Systems Laboratory, Stanford University), Frank van Harmelen (Vrije Universiteit, Amsterdam), *OWL Web Ontology Language, Overview*, W3C Working Draft March 31, 2003. <http://www.w3.org/TR/owl-features/>