

الجمهورية الجزائرية الديمقراطية الشعبية
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جامعة فرحات عباس - سطيف 1
UNIVERSITÉ FERHAT ABBES –SÉTIF1-



Vice rectorat chargé de la post graduation نيابة الجامعة المكلفة بدراسات ما بعد التدرّج
مدرسة الدكتوراه في الإعلام الآلي تخصّص هندسة الأنظمة المعلوماتية

École doctorale d'informatique
Option
Ingénierie des Systèmes Informatiques

Mémoire

Présenté pour l'obtention du

Diplôme de Magister

Par HACINE GHERBI Ahcine

Thème:

Construction d'une Ontologie pour le WEB

Sémantique

Membres du Jury :

Président : Dr MOUSSAOUI.A	Professeur	U.F.A SETIF1
Rapporteur : Dr TOUAHRIA.M	Professeur	U.F.A SETIF1
Examineur : Dr KHABABA.A	Professeur	U.F.A SETIF1

Table des matières

Remerciements.....	i
Introduction.....	1

Chapitre 1: Le web sémantique

1.1	Introduction.....	4
1.2	Les principes du Web Sémantique.....	7
1.2.1	Principe 1: Identification par URI.....	7
1.2.2	Principe 2 : Typage des ressources et liens Web.....	7
1.2.3	Principe 3 : Tolérance de l'information partielle.....	8
1.2.4	Principe 4 : L'indisponibilité d'une vérité absolue.....	8
1.2.5	Principe 5 : Support de l'évolution.....	10
1.2.6	Principe 6 : Architecture minimaliste.....	10
1.3	Les principales composantes du web sémantique.....	11
1.3.1	Ontologie.....	12
1.3.2	Annotation sémantique.....	14
1.3.3	Langage de représentation de connaissance.....	15
1.4	Quelques outils pour le Web Sémantique.....	22
1.5	Conclusion.....	25

Chapitre 2: Sur la notion d'ontologie

2.1	Introduction.....	26
2.2	Apparition d'ontologie	26
2.3	La notion d'ontologie.....	27
2.3.1	L'interopérabilité.....	27
2.3.2	La réutilisation.....	28
2.4	Les constituants d'une ontologie.....	28
2.5	Discussion.....	30
2.5.1	Différence entre ontologie et base de connaissance.....	30
2.5.2	Différence entre ontologie et hiérarchie de classes.....	30
2.6	Classification d'ontologies.....	31
2.6.1	Classification selon l'objet de conceptualisation.....	31
2.6.2	Classification selon le niveau de complétude.....	32
2.6.3	Classification selon le niveau de détail.....	32
2.6.4	Classification selon le formalisme utilisé	33
2.7	Méthodes et outils de construction d'ontologie.....	33
2.7.1	METHONTOLOGY.....	33
2.7.2	L'approche TOVE.....	35
2.7.3	L'approche SENSUS.....	35
2.7.4	Outils d'ingénierie ontologique à partir du texte.....	36
2.7.5	Outils d'ingénierie ontologique à partir des modèles conceptuels ou opérationnels.....	38
2.8	Conclusion.....	39

Chapitre 3: Méthodes de construction d'ontologie à partir des modèles conceptuels UML

3.1	Introduction.....	40
3.2	Quelques solutions permettant d'ajouter de la sémantique aux modèles conceptuels.....	41
3.2.1	Les profils UML.....	41
3.2.2	Le standard OCL.....	43
3.2.3	Action Semantics.....	44
3.2.4	La spécification ODM.....	45
3.3	Les raisons qui empêchent l'utilisation d'UML pour modéliser directement une ontologie.....	46
3.4	Quelques travaux.....	48
3.5	Conclusion.....	49

Chapitre 4: Conception d'une ontologie

4.1	Introduction.....	50
4.2	Utilisation des ontologies en biologie.....	50
4.3	Conception de l'ontologie du domaine de l'écologie évolutive	50
4.3.1	Domaine de l'ontologie	50
4.3.2	Processus d'affinement de l'ontologie	52
4.4	Respect des principes de construction.....	52
4.5	Présentation de cette ontologie.....	53
4.5.1	Les axes primitifs de l'arbre ontologique construit	53
4.6	L'héritage multiple.....	56
4.7	Les propriétés des classes de cette ontologie.....	56
4.7.1	Propriété d'objet	56
4.7.2	Propriété de type de données.....	56
4.8	Représentation hiérarchique des concepts.....	57
4.9	Conclusion.....	59

Chapitre 5: Implémentation

5.1	Introduction.....	60
5.2	Environnement de développement.....	60
5.3	Implémentation	60
5.3.1	Normalisation des noms de l'ontologie.....	60
5.3.2	Création d'un nouveau projet.....	61
5.3.3	L'édition de l'ontologie	62
5.4	L'exportation et l'importation de cette ontologie.....	65
5.4.1	L'exportation et l'importation avec UMLBackend (Protégé3.4.4).....	65
5.4.2	Importation avec OWL 2.0 (sous protégé 4.2)	68
5.4.3	L'outil OWLGrtd.....	69
5.5	Le raisonnement	70

5.6	Conclusion.....	70
	Conclusion générale.....	71
	Références.....	73

Remerciements

Je tiens tout d'abord à remercier profondément mon Encadreur Pr. Touahria Mohammed pour la gentillesse et la patience qu'il a manifestées à mon égard. Je le remercie, aussi, pour m'avoir guidé, encouragé, conseillé, tout au long de la réalisation de ce mémoire.

Mes profonds remerciements vont également au Pr. Moussaoui Abdelouahab pour m'avoir fait l'honneur de présider le Jury de soutenance du présent mémoire, pour le temps qu'il a consacré pour lire ce manuscrit et pour ces valeureuses remarques.

Mes profonds remerciements vont également, aussi, au Pr. Khababa Abdallah pour m'avoir fait l'honneur de participer au Jury de soutenance du présent mémoire, pour le temps qu'il a consacré pour lire ce manuscrit et pour ces valeureuses remarques.

Enfin, mes remerciements ne seraient pas complets sans mentionner l'ensemble de mes enseignants qui ont participé à notre formation. Qu'ils trouvent ici, l'expression de mon profond respect et de ma haute considération.

Introduction générale

Introduction générale

Contexte de recherche

Le Web classique peut être vu comme un ensemble de ressources (textes, images, services, etc.) conçues et interconnectées, suivant une syntaxe bien définie, pour être traitées et comprises par des utilisateurs humains. Mais, avec le volume croissant de la quantité d'information existante sur le Web, les outils présumés accompagner l'utilisateur à y accéder, ont montré leurs limites.

Pour permettre une exploitation, qualitativement supérieure, aussi bien par les humains que par les machines de ces grands volumes d'informations et de services variés, il a été proposé d'étendre le Web Classique vers le Web Sémantique où, grâce à une représentation sémantique du contenu de l'information, les applications peuvent en raisonner directement dessus, en exploitant la sémantique sous-jacente et cela dans le but de soutenir l'utilisateur et lui faciliter ses différentes tâches (recherche, commerce électronique...).

Le Web Sémantique a pour but d'ajouter des informations caractéristiques aux informations existantes pour aider à l'automatisation des services, de découvrir, de relier et d'inférer des connaissances qui sont vraisemblables. Son architecture s'appuie sur une pyramide de langages pour représenter des connaissances sur le web en satisfaisant les critères de standardisation, d'interopérabilité et de flexibilité.

L'un des aspects les plus importants du Web Sémantique est de pouvoir utiliser des annotations sémantiques des documents Web. Ces annotations sont des informations

descriptives permettant l'accès, la recherche et l'utilisation d'une ressource. Le Web sémantique fait recours à l'ingénierie des connaissances pour disposer d'un outil de représentation des connaissances adéquat et s'articule particulièrement sur la notion d'ontologie. Cette dernière est vue comme un ensemble de connaissances, y compris le vocabulaire et les relations sémantiques, avec quelques règles simples d'inférence et de logiques relatives à des sujets particuliers.

Se baser seulement sur une ontologie déjà définie permet d'enrichir l'annotation en lui attribuant une sémantique la rendant, ainsi, utilisable. Une ontologie doit, de plus, être conçue comme un modèle réutilisable et partagé des connaissances afin de constituer une base commune à l'ensemble du domaine pour l'échange et l'interprétation des informations.

Problématique

Les ontologies sont souvent capturées dans des langages de représentation de la connaissance (KR : Knowledge Representation) qui proviennent des théories de l'intelligence artificielle. Ces langages sont pour la plupart structurés suivant des formalismes de logique tels que la logique de prédicats. Cette structure a rendu la syntaxe de ces langages inhabituelle et peu commode à ceux qui ont l'habitude d'utiliser d'autres langages de modélisation, ce qui est le cas de la plupart des développeurs de logiciel. Ceci explique le manque d'utilisation large de ces langages et donc le développement d'ontologies, retardant ainsi l'adoption des technologies du Web Sémantique.

Sur un autre plan, UML est un langage de modélisation adopté par une large communauté dans le cadre de la modélisation conceptuelle. De plus, une large gamme d'outils UML est disponible ainsi que l'existence de plusieurs modèles de domaines et la similarité de ces modèles aux ontologies.

Les ontologies construites à partir des modèles conceptuels sont plus riches sémantiquement à cause de la définition explicite des associations entre les entités dans le schéma conceptuel.

Objectif

La familiarité des utilisateurs avec UML, la disponibilité des outils UML, l'existence de plusieurs modèles de domaines et la similarité de ces modèles aux ontologies, suggèrent qu' UML puisse être utilisé pour faire des développements d'ontologies. Ceci permettra aussi de créer un lien entre la communauté UML et celle du Web Sémantique.

Organisation du manuscrit

Après cette introduction générale de notre travail, le reste de ce mémoire est structuré comme suit :

Chapitre 1: présente les notions, principes, composantes principales et quelques outils du Web sémantique.

Chapitre 2: introduit la notion de l'ontologie. Nous y décrivons les principes de sa construction à travers une représentation de ses constituants, les différentes classifications, méthodologies et outils nécessaires pour son développement.

Chapitre 3: présente quelques solutions permettant d'ajouter de la sémantique aux modèles conceptuels

Chapitre 4: Dans ce chapitre nous présentons une ontologie dédiée à la description du domaine de l'écologie évolutive. La conception a été faite dans une perspective de construction d'un vocabulaire partageable pour l'objectif recherché, de couvrir et de représenter l'ensemble des mots et expressions utilisés dans ce domaine.

Chapitre 5: Dans ce chapitre nous décrivons une implémentation de l'ontologie présentée dans le chapitre précédent. La création de l'ontologie est faite en utilisant Protégé et l'exportation est réalisée avec les extensions UMLBackend et OWLGrtd.

Nous terminons ce manuscrit par une conclusion générale suivie de quelques suggestions que nous jugeons importantes pour une meilleure adoption du Web Sémantique.

Chapitre 1

Le web sémantique

Au Sommaire de ce chapitre

1.1 Introduction

1.2 Les principes du Web Sémantique

1.2.1 Principe 1: Identification par URI

1.2.2 Principe 2 : Typage des ressources et liens Web

1.2.3 Principe 3 : Tolérance de l'information partielle

1.2.4 Principe 4 : L'indisponibilité d'une vérité absolue

1.2.5 Principe 5 : Support de l'évolution

1.2.6 Principe 6 : Architecture minimaliste

1.3 Les principales composantes du Web Sémantique

1.3.1 Ontologie

1.3.2 Annotation sémantique

1.3.3 Langage de représentation de connaissance

1.4 Quelques outils pour le Web Sémantique

1.5 Conclusion

1.1 Introduction

Nous pouvons considérer le Web classique comme un ensemble de documents multimédias (textes, images, sons, vidéos, etc.) reliés entre eux par des liens hypertextes. Ce réseau de documents repose essentiellement sur le protocole de communication http (HyperText Transfer Protocol) qui permet d'accéder à des documents identifiés et localisés grâce à leurs URL (Unified Resource Locator). HTML (HyperText Markup Language) est le langage de présentation généralement utilisé et qui permet une mise en forme graphique des documents présents sur le Web. Ainsi, le Web actuel peut être vu comme un ensemble de ressources interconnectées (textes, images, services, etc.) et conçues pour être traitées et comprises par des utilisateurs humains. Mais, avec le volume croissant de la quantité d'informations existantes sur le Web, les outils présumés accompagner l'utilisateur à y accéder ont montrés leurs limites. Obligés de les traiter à un niveau totalement syntaxique (mots clés, co-occurrence de termes, etc.) ils ne peuvent fournir des résultats adéquats vis-à-vis du contenu de ces ressources. Les tâches difficiles de tri, de sélection et de combinaison d'informations, sont cédées à l'utilisateur, qui est le seul à pouvoir les expliquer et donc de les comprendre. C'est ce fait qui est à la source du Web Sémantique.

En effet, le web classique ne dispose pas de mécaniques pour décrire et organiser ses ressources de manière acceptable afin de permettre un accès adéquat à l'information. C'est pour remédier ces défauts qu'il a été proposé [1] d'étendre le web classique vers un web où l'information aura un sens bien défini permettant ainsi aux applications d'exploiter directement la sémantique des ressources et de soutenir l'utilisateur afin de lui faciliter ses tâches (recherche, commerce électronique...).

Pour comprendre de quelle manière le Web intègre la notion de sémantique, considérons l'exemple d'une recette de gaspacho [26]. Un gaspacho est une soupe froide, d'origine espagnole, généralement à base de tomates, et servie le plus souvent en entrée. Comme tout document décrivant une recette, une page Web dédiée à la recette du gaspacho (voir figure 1) contiendra une liste d'ingrédients, une suite d'étapes de préparation, une photographie, etc. Il est assez pratique pour un utilisateur connaissant la langue française de savoir qu'il s'agit d'une recette, que cette recette est celle du gaspacho et que la photographie représente le résultat attendu de la recette. Cependant, le langage HTML utilisé ne s'intéresse

qu'à la mise en forme graphique du texte et à l'inclusion d'éléments multimédias. Le contenu de ce document n'est ainsi compris par un programme informatique que comme un ensemble de données, une séquence de mots.



Figure 1 – Page web (HTML) décrivant une recette de gaspacho [26].

Une première étape dans la description sémantique de cette recette serait d'en délimiter et d'en reconnaître les distinctes parties, c'est-à-dire d'en représenter la structure. Il s'agit pour cela d'associer à la description du document des informations concernant le contenu même des informations ou, en d'autres mots, d'y joindre des métadonnées. Le langage le plus utilisé pour structurer les informations dans cette vision est XML (eXtensible Markup Language), qui définit des balises permettant à repérer les éléments de contenu des documents. La figure 2 montre un exemple de représentation réalisable de notre recette de gaspacho dans ce langage. Sur la base de cette structure XML, il sera par exemple facile de rechercher parmi un ensemble de recettes celles qui comportent une grande quantité de tomates, peu d'étapes de préparation ou un temps de cuisson minime, et de retrouver alors la recette du gaspacho. Les métadonnées sont ainsi utilisées comme base pour l'accès aux informations, ainsi que pour leurs échanges, au sein de programmes informatiques.

```

<recette>
  <titre>Gaspacho</titre>
  <tempsprep unite="minutes">5</tempsprep>
  <niveau>facile</niveau>
  <tempscuisson>0</tempscuisson>
  <cout>faible</cout>
  <nombre-personne>4</nombre-personne>
  <photo>http://www.loria.fr/daquin/gaspacho/photo.jpg</photo>
  <listingredients>
    <ingredient quantite="1" unite="grosses boites ">tomates</ingredient>
    <ingredient quantite="1/2" >poivron vert</ingredient>
    <ingredient quantite="1/2" >concombre</ingredient>
    <ingredient quantite="1" >échalote grise</ingredient>
    <ingredient quantite="1" >gousse d'ail entière</ingredient>
    <ingredient quantite="2" unite=" cuillères à soupe ">huile d'olive</ingredient>
    <ingredient quantite="1/2" unite=" cuillères à café ">vinaigre balsamique<
  /ingredient>
    <ingredient >tabasco ou cayenne (au goût )</ingredient>
    <ingredient quantite="1/2" unite="cuillères à café">sucre</ingredient>
    <ingredient >sel et poivre</ingredient>
    <ingredient quantite="15" unite="cl">eau</ingredient>
  </listingredients>
  <preparation>
    <etape ordre="1">Mettre tous les ingrédients coupés grossièrement dans un mixer,
démarrer l'appareil à petite vitesse et terminer à grande vitesse</etape>
    <etape ordre="2">Laisser refroidir au réfrigérateur</etape>
    <etape ordre="3">Ajouter des glaçons au moment de servir</etape>
  </preparation>
</recette>

```

Figure 2 - Description XML d'une recette de gaspacho.

Néanmoins, XML ne s'intéresse qu'à la structure des informations, à la syntaxe selon laquelle elles doivent être représentées. Il ne permet pas d'en exploiter la sémantique. En effet, chercher une recette dont la liste des ingrédients contient le terme « tomate » ne suppose pas de comprendre le sens des mots « Ingrédients » et « tomate ». Un programme informatique s'appuyant sur cette structure ne sera ainsi pas capable de reconnaître la recette du gaspacho comme une recette ne contenant aucune viande et ne demandant que peu de matériel. Il ne saura pas non plus admettre que le plat obtenu sera froid et peu calorique. Ainsi, dans l'objectif de permettre aux programmes informatiques de comprendre et de manipuler les informations selon la sémantique qu'elles portent, il est indispensable qu'ils

aient des connaissances, expliquant de manière formelle les concepts mis en œuvre et les relations entre ces concepts. C'est le rôle des ontologies au sein du Web sémantique.

1.2 Les principes du Web Sémantique

En associant des méta-données aux documents, cela nous permet de chercher, par exemple, des documents ayant dans les méta données "UFAS" comme université. Nous pouvons aussi faire des requêtes du genre : "afficher tous les documents ayant comme catégorie : document de recherche". Le Web Sémantique ne fournit pas des URIs seulement aux documents, comme le fait le Web actuel, mais il le fait aussi pour les concepts et les relations qui définissent et mettent en relation les ressources du Web. Dans l'exemple ci-dessus en donnant un identifiant unique aux universités et en définissant les concepts "université" et "document de recherche", on fournit de façon claire, une manière d'identifier l'université et quelle est la relation entre ce nom et un document particulier. Ces différentes informations descriptives peuvent être arrangées et incorporés dans le but d'avoir une connaissance plus globale d'un domaine donné.

Le Web Sémantique a pour but d'ajouter une information caractéristique au Web pour aider à l'automatisation des services, de découvrir, relier et inférer des connaissances qui sont vraisemblables. Nous allons maintenant voir les principes qui régissent le Web Sémantique.

1.2.1 Principe 1: Identification par URI

Les personnes, les places et les choses du monde réel peuvent être indexées dans le Web Sémantique en utilisant un ensemble d'identifiants. Dans le Web Sémantique les identifiants sont des URIs (Uniform Resource Identifier)

1.2.2 Principe 2 : Typage des ressources et liens Web

Le Web actuel est composé de ressources et de liens. Les ressources sont des documents web conçus pour la compréhension humaine et ne possèdent pas de descriptions expliquant en quoi elles sont utilisées et quelles relations entretiennent elles avec d'autres documents. Quand une personne peut à directement connaître si un tel document est une facture, un roman ou encore un article de recherche, une machine ignore fréquemment ce type d'information. De la même manière, une personne peut connaître le lien entre deux ressources en lisant le libellé du lien hyper-texte qu'il y a entre elles. Alors que pour une

machine il est impossible de procéder de la même manière. L'un des objectifs du Web Sémantique est de typer les ressources et les liens du web.

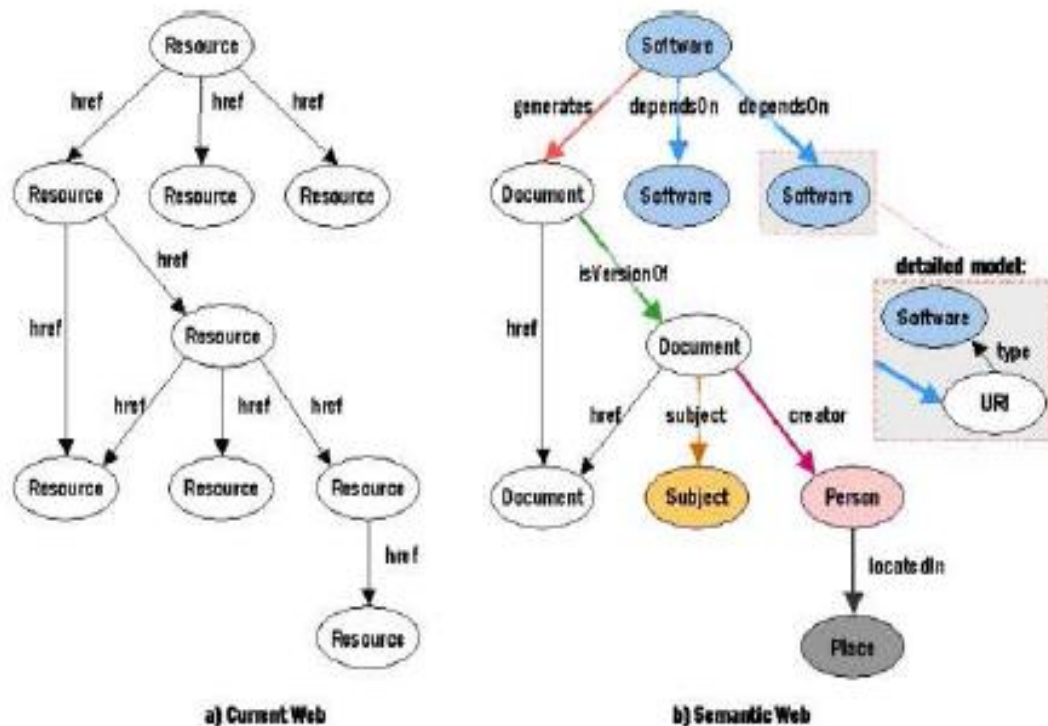


Figure 3 - Typage des ressources et liens du Web [45]

1.2.3 Principe 3 : Tolérance de l'information partielle

Lorsqu'une ressource indexée n'est plus disponible, le fonctionnement doit continuer. Le web actuel est évolutif (scalable) car une information incomplète est déjà tolérée comme le montre la Figure 5. Le Web Sémantique doit continuer de tolérer une information incomplète.

1.2.4 Principe 4 : L'indisponibilité d'une vérité absolue

Un des problèmes du Web est que n'importe qui peut dire n'importe quoi sur n'importe quoi posant ainsi la question : jusqu' où et comment peut-on croire à quelque chose? Dans ce sens, les signatures numériques apportent un élément de réponse en permettant de savoir qui a dit quoi, quand et dans quel but.

La confiance dans le Web est la même que chez l'homme : on croit plus à nos amis, un peu moins à leurs amis et moins aux amis de ces derniers.

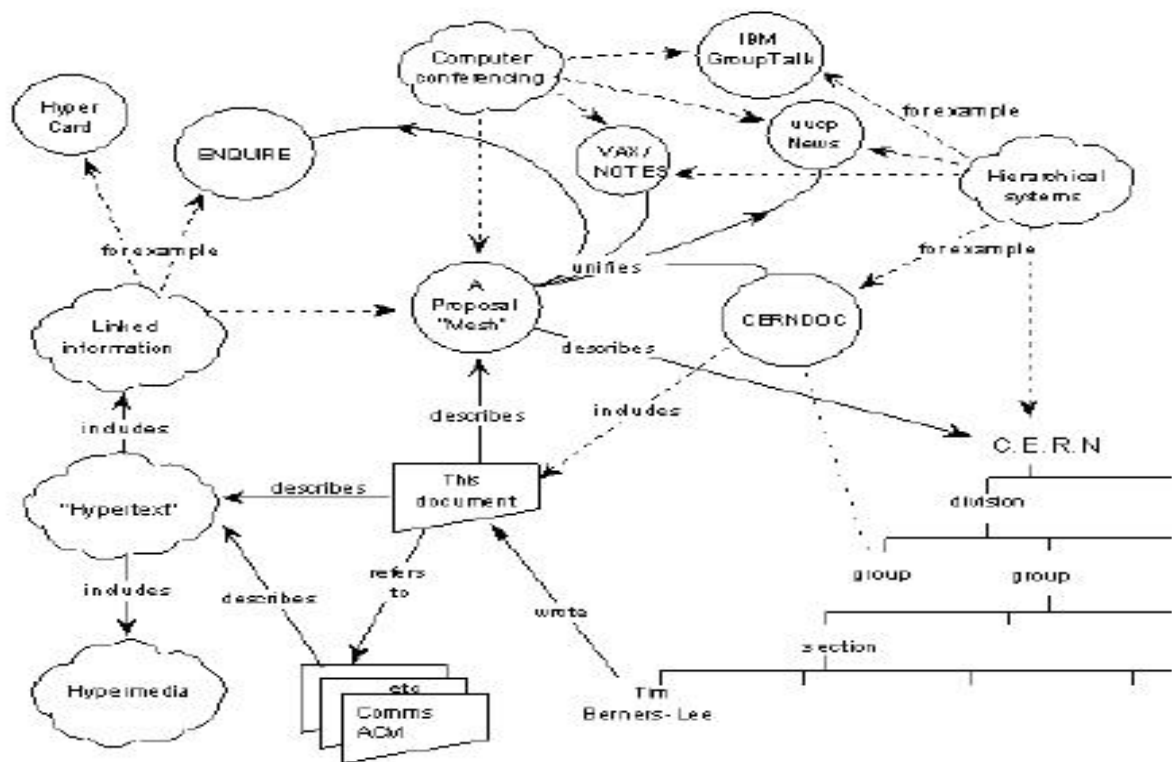


Figure 4 - La vision du web de Tim B. Lee 1989

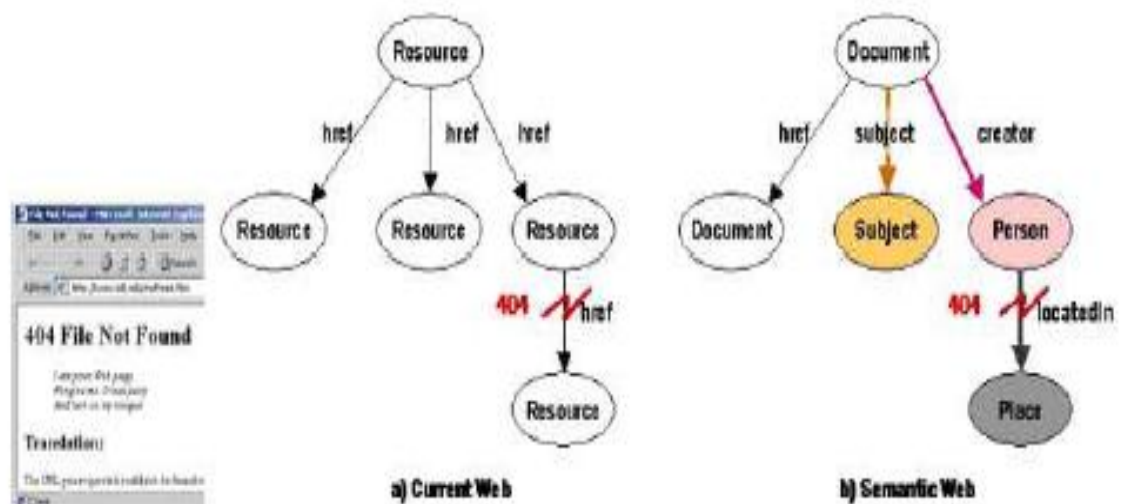


Figure 5 - Information partielle tolérée [45]

La véracité ou fiabilité du contenu du web doit être validée par l'application qui utilise l'information issue du Web. Ces applications doivent décider en quoi croire en se basant sur le contexte des déclarations : qui a dit quoi, quand et en qualité de quoi

1.2.5 Principe 5 : Support de l'évolution

Il est habituel de voir des concepts semblables définis par différents groupes de personnes à des endroits différents ou même par le même groupe à des moments différents. Généralement, il convient de combiner des données du Web utilisant ces concepts. On doit pouvoir combiner une nouvelle information et une ancienne sans modifier cette dernière comme le montre la Figure 6. Pour ce faire, le Web Sémantique utilise des conventions descriptives qui peuvent être étendues comme c'est le cas de la compréhension humaine. En plus, ces conventions facilitent la combinaison réelle de différents travaux de divers groupes, même si ils utilisent des vocabulaires distincts. Le Web Sémantique fournit, ainsi, des outils communautaires qui peuvent être utilisés pour lever des ambiguïtés et aider à clarifier des inconsistances.

1.2.6 Principe 6 : Architecture minimaliste

Le Web Sémantique ne doit pas être compliqué et cela dans le but d'une adaptation simple et rapide.

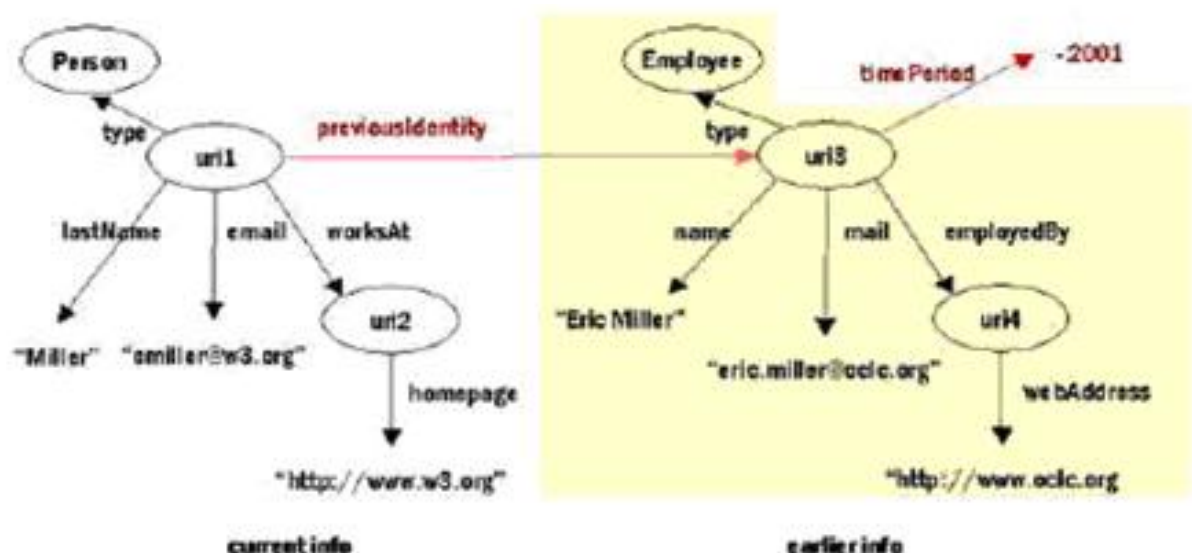


Figure 6 - Support de l'évolution

1.3 Les principales composantes du web sémantique:

L'architecture du web sémantique s'appuie sur une pyramide de langages pour représenter des connaissances sur le web en satisfaisant les critères de standardisation, d'interopérabilité et de flexibilité. Cette architecture en couches (Figure 7) peut permettre une approche progressive dans les processus de standardisation et d'acceptation par les utilisateurs. Un langage de la couche haute doit être une extension du langage de la couche au-dessous.

La liste suivante introduit la fonction principale de chaque couche dans l'architecture du web sémantique [27]:

- XML est utilisé comme couche de base syntaxique du web sémantique. Le langage XML est actuellement considéré comme un standard pour le transport de données sur le web.
- La couche RDF représente les métadonnées pour les ressources web.
- La couche « ontologie », fondée sur une formalisation commune, spécifie la sémantique de métadonnées fournies dans le web sémantique.
- La couche « logique » s'appuie sur des règles d'inférence qui permettent le raisonnement intelligent exécuté par des agents logiciels.

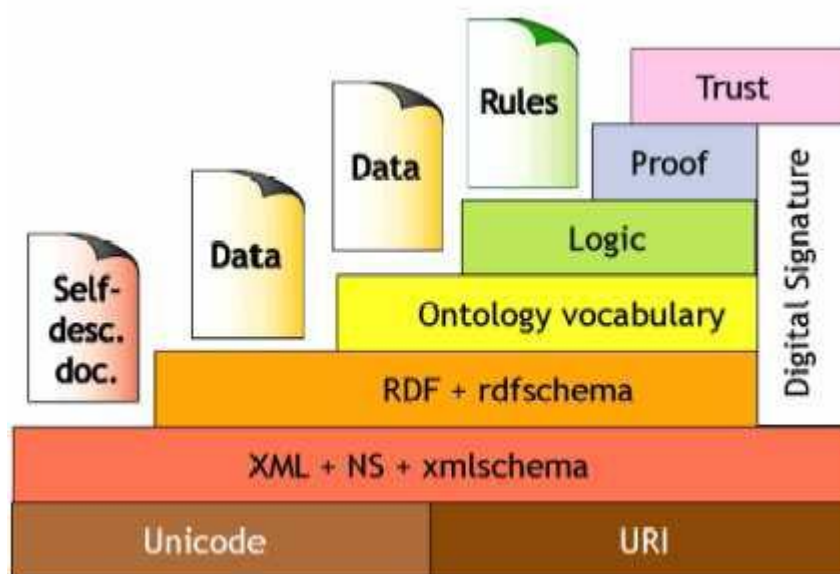


Figure 7 - Les couches du Web Sémantique

1.3.1 Ontologie

Une ontologie, en philosophie, est une théorie à propos de la nature de l'existant, des types de choses qui existent et des types de leurs liens. Dans le Web sémantique, une ontologie est vue comme un ensemble de connaissances, y compris le vocabulaire et les relations sémantiques, avec quelques règles simples d'inférence et de logiques relatives à des sujets particuliers.

Dans [28], une ontologie est définie comme suit : « une ontologie est une spécification explicite d'une conceptualisation ». Plus concrètement, il est commun de considérer une ontologie comme la définition formelle des concepts relatifs à un domaine et des relations entre ces concepts. Elle doit de plus être conçue comme un modèle réutilisable et partagé des connaissances afin de constituer une base commune à l'ensemble du domaine pour l'échange et l'interprétation d'informations. La forme la plus courante selon laquelle les ontologies sont présentées consiste en une hiérarchie de concepts, organisés des plus généraux aux plus spécifiques. Une ontologie relative aux recettes de cuisine contiendra par exemple les concepts d'aliments et de plats. La figure 8 présente un extrait de la hiérarchie de concepts d'une telle ontologie. Chaque concept doit être associé à une définition formelle. Ici, le concept de recette pourra être défini comme une façon permettant de réaliser un plat. Un plat froid pourra quant à lui être défini comme un plat dont la recette a un temps de cuisson nul. L'expression de ces définitions, c'est-à-dire la syntaxe utilisée et la sémantique associée, dépend du langage de représentations dans lequel l'ontologie est formalisée.

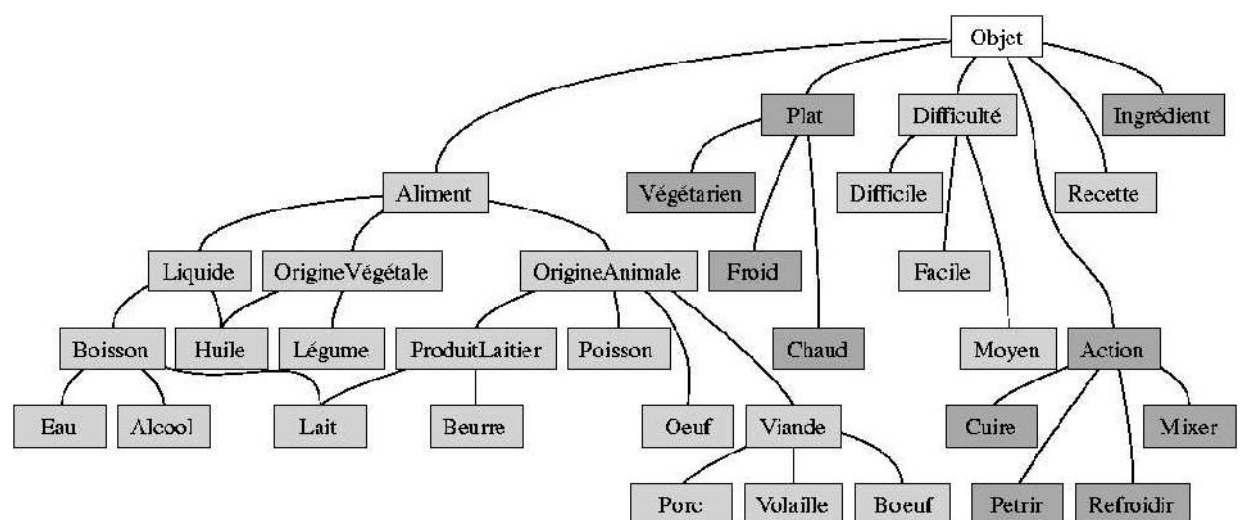


Figure 8 – Hiérarchie de concepts d'une ontologie pour les recettes de cuisine [26].

Une ontologie correspond à un vocabulaire contrôlé et organisé et à la formalisation explicite des relations créées entre les différents termes du vocabulaire. Une ontologie formelle peut être considérée comme la différenciation entre les entités du monde réel et les classes utilisées pour décrire des objets (concepts, propriétés, relations). Une ontologie formelle est une évolution systématique, formelle et axiomatique de la logique de toutes les formes et modes d'existence.

Les ontologies ont pour rôle principal la représentation d'un domaine.

Elles sont utilisées dans divers domaines y compris:

- la description de l'information dans différentes spécialités (biologie, médecine..).
- la recherche d'informations;
- le commerce électronique afin de permettre, par exemple, la communication entre les fournisseurs et les acheteurs;
- les environnements de formation à distance.

Les différents types d'ontologies:

On peut définir deux classes d'ontologies : (i) une catégorie basée sur la structure de la conceptualisation et (ii) une deuxième catégorie basée sur le sujet de la conceptualisation [29].

Dans la première catégorie, on distingue trois sous catégories :

- les ontologies terminologiques (lexiques, glossaires...),
- les ontologies d'information (schéma d'une BD),
- les ontologies des modèles de connaissances.

Dans la deuxième catégorie, on distingue quatre sous catégories:

- les ontologies d'application : elles contiennent toutes les informations nécessaires pour modéliser les connaissances pour une application particulière.
- les ontologies de domaine : elles fournissent un ensemble de concepts et de relations décrivant les connaissances d'un domaine spécifique.

- les ontologies génériques (dites aussi de haut niveau) : elles sont similaires aux ontologies de domaine, mais les concepts qui y sont définis sont plus génériques et décrivent des connaissances tels que l'état, l'action, l'espace et les composants. Généralement, les concepts d'une ontologie de domaine sont des spécialisations des concepts d'une ontologie de haut niveau.
- Les ontologies de représentation (méta-ontologies) : elles fournissent des primitives de formalisation pour la représentation des connaissances. Elles sont généralement utilisées pour écrire les ontologies de domaine et les ontologies de haut niveau. Exemples : Frame Ontology et RDF Schema Ontology .

1.3.2 Annotation sémantique:

L'un des aspects les plus importants du Web Sémantique est de pouvoir utiliser des annotations sémantiques de documents Web. Une annotation est une information descriptive permettant l'accès, la recherche et l'utilisation d'une ressource. Se baser sur une ontologie déjà définie permet d'enrichir l'annotation en lui attribuant une sémantique la rendant, ainsi, utilisable sans modification par une application. Ainsi, les annotations sémantiques décrivent le contenu des documents, en joignant une sémantique à ces descriptions. On peut les considérer comme des métadonnées de documents, ressources du Web.

Intérêt des métadonnées:

Les métadonnées présentent des aspects spécifiques pour :

- Faciliter la recherche d'information
 - ✓ décrire le contenu et les relations entre les fichiers d'un site,
 - ✓ classer le contenu suivant un degré de difficulté ou un public cible,
 - ✓ mieux référencer un site ou une page sur Internet.
- Faciliter l'interopérabilité
 - ✓ partager et échanger des informations.

- Faciliter la gestion et l'archivage
 - ✓ informer sur le cycle de vie des documents,
 - ✓ gérer des collections de ressources,
 - ✓ gérer des archives électroniques.
- Gérer et protéger les droits
 - ✓ les droits de propriété intellectuelle,
 - ✓ les droits d'accès à des pages web (restrictions de consultation).
- Authentifier un texte
 - ✓ encoder une signature électronique pour valider un texte sur Internet.

Plusieurs outils ont été proposés pour permettre la génération automatique ou manuelle de ces annotations sémantiques. Nous présentons ci dessous quelques exemples:

Annotea: [40]

C'est un système de génération (manuelle) d'annotations RDF pour les pages web développées au sein du W3C.

MnM: [46]

MnM fournit un environnement pour la génération semi-automatique d'annotations sémantiques associées aux documents Web. Il est basé essentiellement sur des techniques d'apprentissage et des méthodes de TALN (Traitement Automatique de la Langue Naturelle).

KIM: [47]

KIM fournit une plate-forme de génération automatique d'annotations sémantiques et de recherche documentaire basée sur ces annotations.

1.3.3 Langage de représentation de connaissance:

Les ontologies ont un rôle clé dans la construction du Web Sémantique en facilitant le partage d'information entre les communautés d'humains et les applications. Les schémas

XML et les DTDs sont utilisés pour échanger les données dans le Web. Cependant, l'absence de sémantique ne permet pas aux personnes de créer de nouveaux vocabulaires XML. RDF et *RDF Schema* ont essayé de résoudre ce problème en associant des sémantiques simples aux identificateurs. Le langage OWL (*Ontology Web Language*) a été conçu pour être utilisé par les applications qui traitent le contenu de l'information au lieu de la présenter seulement aux utilisateurs. OWL facilite l'interopérabilité au niveau machine du contenu du Web plus que ce qui est déjà supporté par les XML, RDF et *RDF Schema* (RDF-S) en fournissant du vocabulaire supplémentaire avec des sémantiques formelles.

Dans ce qui suit, nous allons présenter les langages XML, RDF, DAML+OIL et OWL puisqu'ils constituent des recommandations W3C.

1.3.3.1 XML:

Le langage eXtensible Markup Language (XML) est un méta-langage permettant de représenter un document texte de manière arborescente en utilisant un système de balisage. XML fournit ainsi un cadre de structuration des données qui peut être employé pour la définition rapide et sans ambiguïté syntaxique d'un format de document.

Ce langage a été élaboré pour faciliter l'échange, le partage et la publication des données à travers le web. Ainsi, la majorité des langages/modèles proposés pour le web sémantique sont exprimés en XML.

XML permet de structurer un document en définissant ses propres balises en fonction des besoins et sans tenir compte ni de la signification de cette structure ni des systèmes informatiques qui vont l'exploiter. Des standards comme XPath et XQuery ont été développés afin de parcourir et d'interroger l'arborescence XML des documents.

Structure d'un fichier XML:

Un document XML est structuré en 3 parties:

La première partie, appelée prologue permet d'indiquer la version de la norme XML utilisée pour créer le document. Le prologue se poursuit avec des informations facultatives sur des instructions de traitement à destination d'applications particulières. La syntaxe est la suivante:

<? Instruction de traitement?>

La deuxième partie est une déclaration de type du document (à l'aide d'un fichier annexe de type DTD ou schémas XML). Les DTD (Document Type Definition) et les schémas XML permettent de forcer l'utilisation des balises et de leurs imbrications pour la représentation des informations. De cette façon, XML permet de déclarer, et donc de partager, une syntaxe propre à la représentation d'informations utiles à un domaine ou à une application.

Et enfin, la dernière partie d'un fichier XML est l'arbre des éléments.

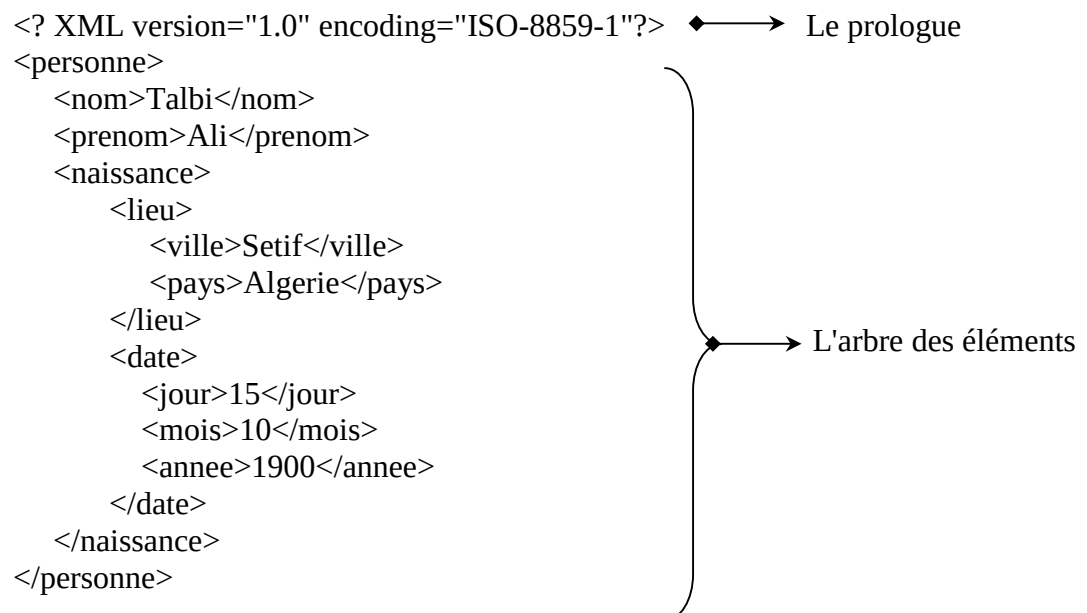


Figure 5- Exemple de fichier XML simple

La définition DTD du fichier ci-dessus est la suivante :

```

<!ELEMENT personne (nom, prenom, naissance) >
<!ELEMENT naissance (lieu, date) >
<!ELEMENT lieu (ville, pays) >
<!ELEMENT date (jour, mois, année) >
<!ELEMENT nom (#PCDATA) >
<!ELEMENT prenom (#PCDATA) >
<!ELEMENT ville (#PCDATA) >
<!ELEMENT pays (#PCDATA) >
<!ELEMENT jour (#PCDATA) >
<!ELEMENT mois (#PCDATA) >
<!ELEMENT annee (#PCDATA) >
  
```

On distingue 2 types de DTD : les internes et les externes.

- Une DTD interne est une DTD qui est écrite dans le même fichier que le document XML. Elle est généralement spécifique au document XML dans lequel elle est écrite.
- Une DTD externe est une DTD qui est écrite dans un autre document que le document XML. Si elle est écrite dans un autre document, c'est que souvent, elle est commune à plusieurs documents XML qui l'exploitent.

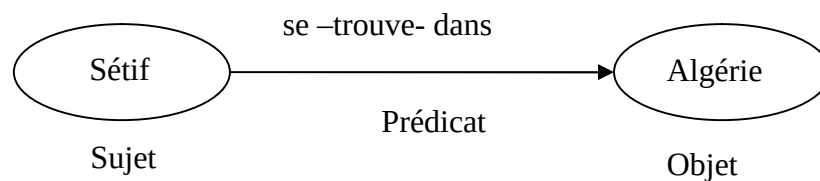
1.3.3.2 RDF/RDFS:

RDF (Resource Description Framework) est un modèle de données pour les objets (ressources) et les relations entre eux, fournissant des sémantiques simples pour ce modèle de données qui peuvent être représentés en XML.

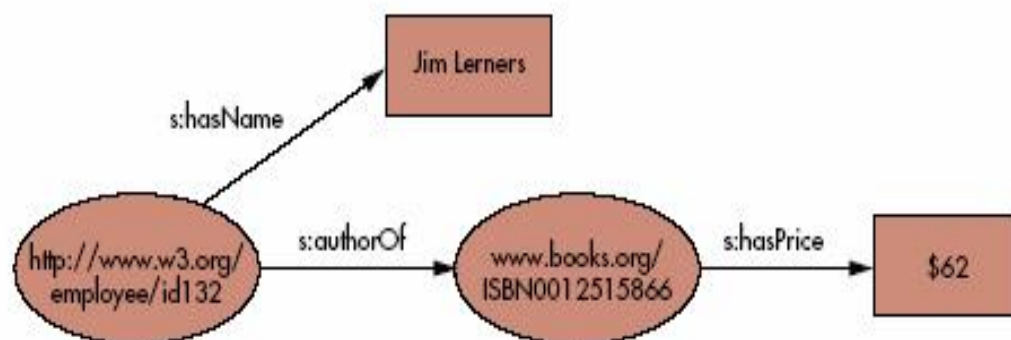
RDF est, aussi, un langage pour les métadonnées sur le web. La syntaxe de RDF est basée sur celle de XML. Le modèle de base de ce langage est conçu pour permettre d'associer des attributs aux ressources du Web en utilisant la description de métadonnées sémantiques. Il est la spécification d'un système d'expression d'assertions sémantiques simples. Ainsi, RDF permet de voir le Web comme un ensemble de ressources reliées par les liens étiquetés «sémantiquement», et permet, alors d'accroître la facilité de traitement automatique des ressources Web.

La structure fondamentale de toute expression en RDF est une collection de triplets, chacun composé d'un sujet, un prédicat et un objet.

- Ressource (Sujet) : une entité d'informations pouvant être référencée par un identificateur. Cet identificateur doit être une URI (Universal Resource Identifiers). Les sujet ou les ressources sont variés : une page Web, une partie d'une page Web, un site Web complet, un objet non accessible par le Web comme un livre...
- Propriété (prédicat): un aspect spécifique, un attribut, une caractéristique ou relation utilisée pour décrire une ressource...
- Valeur : Un littéral (simple chaîne de caractères) ou une ressource.

**Figure 9 - triplet**

Un document RDF est un ensemble de triplets de la forme < sujet, prédicat, objet >. Les éléments de ces triplets peuvent être des URIs (Universal Resource Identifiers), des littéraux ou des variables. Cet ensemble de triplet peut être représenté par un graphe. Plus exactement, un multigraphe orienté étiqueté où les éléments apparaissant comme sujet où les objet sont des sommets, et chaque triplet est représenté par un arc dont l'origine est son sujet et la destination est son objet. Ce document sera traduit par un document RDF/XML, et est souvent représenté sous une forme graphique (voir la figure 10).

**Figure 10 - Exemple de graphe RDF**

RDFS est un méta modèle recommandé par le W3C permettant la définition de schéma/modèle décrivant l'univers sémantique des déclarations RDF. RDFS est le langage de définition de vocabulaires pour RDF, et fournit ainsi un système de typage pour les déclarations RDF. Il permet la définition des classes et des sous-classes (`rdfs:Class`, `rdfs:subClassOf`) décrivant les ressources à annoter et donnant un sens aux propriétés associées aux ressources. Il permet aussi la formulation de contraintes sur les valeurs associées à une propriété afin de lui assurer une signification (`rdfs:domain`, `rdfs:range`).

Voici un exemple qui définit une classe << etudiant>>, et qui instancie de cette classe une ressource <<http:www.ontoknowledge.org/Definedclass>>, et la définit comme sous classe de la classe <<humain>>:

```
<rdfs:Class rdf:ID="etudiant">  
<rdf:type rdf:resource="http:www.ontoknowledge.org/#DefinedClass"/>  
<rdfs:subClassOf rdf:resource="#humain"/>  
</rdfs:Class>
```

1.3.3.3 DAML+OIL:

DAML+OIL est un langage de balises pour la représentation et l'échange d'ontologies. C'est le résultat de la fusion des langages OIL et DAML-ONT (DARPA Agent Markup Language) et il s'appuie sur les langages RDF et RDF Schéma en les enrichissant avec de nouvelles primitives.

D'un point de vue formel, DAML + OIL peut être vu comme une logique de description très expressive à l'aide des types de constructeurs supportés à savoir : les constructeurs qui permettent de définir des classes et des propriétés ainsi que des axiomes.

1.3.3.4 OWL:

OWL (Ontology Web Language) est une extension du RDF pour palier aux insuffisances de ce dernier. En effet, RDF souffre de certaines limites, qui sont, entre autres :

- ✓ rdfs:range définit le domaine de valeurs d'une propriété quelle que soit la classe concernée. Par exemple, il ne permet pas d'exprimer que les nourrissons ne boivent que du lait alors que d'autres niveaux d'ages peuvent boire du Soda.
- ✓ RDFS ne permet pas d'exprimer que deux classes sont disjointes. Par exemple, les classes des hommes et des femmes sont disjointes.
- ✓ RDFS ne permet pas de créer des classes par combinaison ensembliste d'autres classes (intersection, union, complément). Par exemple, on veut construire la classe **Personne** comme l'union disjointe des classes des hommes et des femmes.
- ✓ RDFS ne permet pas de définir de restriction sur le nombre d'occurrences de valeurs que peut prendre une propriété. Par exemple, on ne peut pas dire qu'une personne a exactement deux parents.

- ✓ RDFS ne permet pas de définir certaines caractéristiques des propriétés: transitivité (par exemple: estPlusGrand-Que), unicité (par exemple: estLePèreDe), propriété inverse (par exemple: mange est la propriété inverse de estMangéPar).

Le langage OWL (Ontology Web Language) devrait être construit sur RDFS tout en disposant d'une syntaxe XML. OWL permet de définir des classes de manière plus complexe en utilisant les opérateurs ensembliste (et / ou logique) tels que : l'intersection, l'union, la restriction, etc.), des propriétés inverses ou transitives ou bien encore des restrictions de cardinalité sur les propriétés, et cela en se basant sur une logique de description. OWL facilite, aussi, l'interopérabilité au niveau machine du contenu du Web plus que ce qui est déjà supporté par les XML, RDF et *RDF* Schema en fournissant du vocabulaire supplémentaire avec des sémantiques formelles.

En plus, OWL est doté de trois sous langages offrant des capacités d'expression croissantes et c'est selon les besoins, on choisit le langage qui convient.

- **OWL Lite** est le sous langage de OWL le plus simple. Il est destiné aux utilisateurs qui ont besoin d'une hiérarchie de concepts simple. OWL Lite est adapté, par exemple, aux migrations rapides depuis d'anciens thésaurus.
- **OWL DL** est plus complexe que OWL Lite, permettant une expressivité bien plus importante. OWL DL est fondé sur la logique descriptive (d'où son nom, OWL Description Logics), un domaine de recherche étudiant la logique, et conférant donc à OWL DL son adaptation au raisonnement automatisé. Malgré sa complexité relative face à OWL Lite, OWL-DL garantit la complétude des raisonnements (toutes les inférences sont calculables) et leur décidabilité (leur calcul se fait en une durée finie).
- **OWL Full** est la version la plus complexe d' OWL, mais également celle qui permet le plus haut niveau d'expressivité. OWL Full est destiné aux situations où il est plus important d'avoir un haut niveau de capacité de description, quitte à ne pas pouvoir garantir la complétude et la décidabilité des calculs liés à l'ontologie. OWL Full offre cependant des mécanismes intéressants, comme par exemple la possibilité d'étendre le vocabulaire par défaut de OWL.

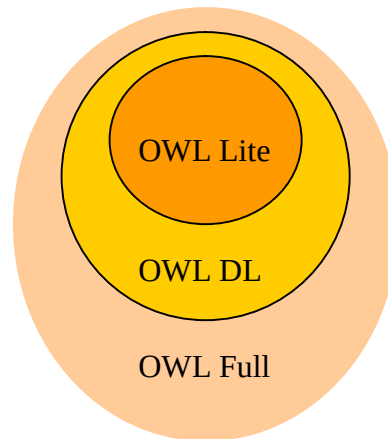


Figure 11- Les couches de L'OWL

Il existe entre ces trois sous langage une dépendance de nature hiérarchique : toute ontologie OWL Lite valide est également une ontologie OWL DL valide, et toute ontologie OWL DL valide est également une ontologie OWL Full valide.

1.4 Quelques outils pour le Web Sémantique

Le projet Annotea

Le projet Annotea [40] est un projet du W3C pour faire de l'annotation web. Les annotations sont des commentaires, des notes, des explications ou d'autres types de remarques externes qui peuvent être associés à tout document web ou une partie sans que cela ne soit nécessaire pour modifier le document.

Une fois qu'on a un document on peut charger les annotations le concernant depuis un serveur afin de les exploiter.

On utilise un schéma basé sur RDF pour décrire les annotations comme des méta-données.

Xpointer est utilisé pour localiser des annotations dans un document annoté.

Les annotations sont stockées dans des serveurs qui peuvent être interrogés [76]. IL existe des clients d'annotation (editor/browser) comme Amaya.

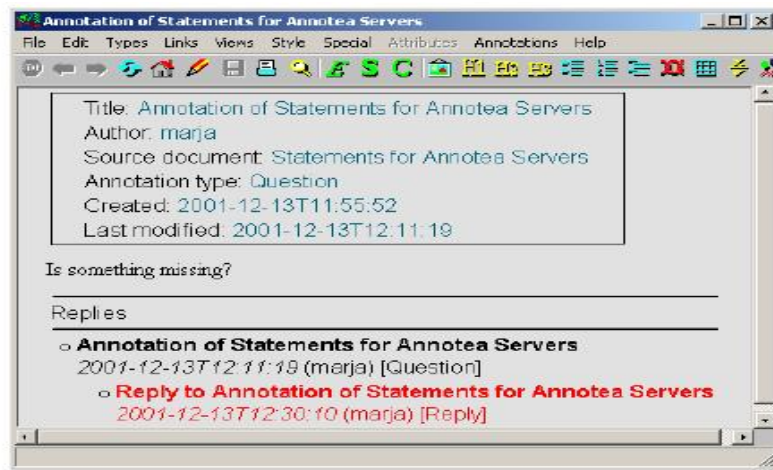


Figure 12 - l'éditeur d'annotation Annotea

JENA

Jena [42] est un framework en Java qui permet de construire des applications Web Sémantique. Il fournit un environnement de programmation pour RDF, RDFS et OWL (voir la Figure). Il inclut également un moteur de règles d'inférence.

Le framework contient :

- ✓ Un api RDF.
- ✓ Un api pour la lecture et l'écriture de RDF en RDF/XML, N3, et N-Triples.
- ✓ Un api OWL.
- ✓ Un système de persistance.
- ✓ RDQL, un langage de requête pour RDF.

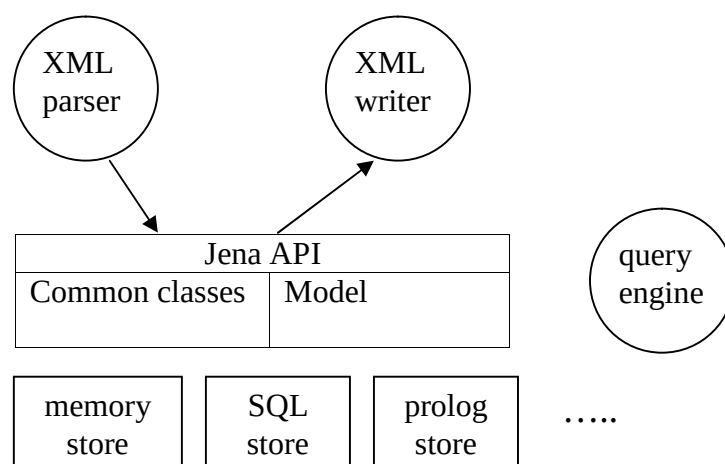


Figure 13 - Architecture de Jena

Music Brainz

Music Brainz est une méta-base de données de musique qui offre des informations concernant un CD ou une piste mp3. Un lecteur de CD peut utiliser Music Brainz pour identifier un CD. On peut aussi l'utiliser pour "tagger" des mp3. La Figure 14 montre le schéma RDF de Music Brainz. Music Brainz qui s'appelait avant CDIndex, est l'un des premiers services Web Sémantique. Le principe n'est pas nouveau et depuis longtemps, a été proposé par d'autre système comme Internet Compact Disc DataBase (CDDDB).

La nouveauté qu'apporte Music Brainz est que c'est un projet ouvert (open source) qui est alimenté par les utilisateurs et se base entièrement sur RDF. Music Brainz fournit également une bibliothèque (API) qui peut être utilisée par d'autres programmes.

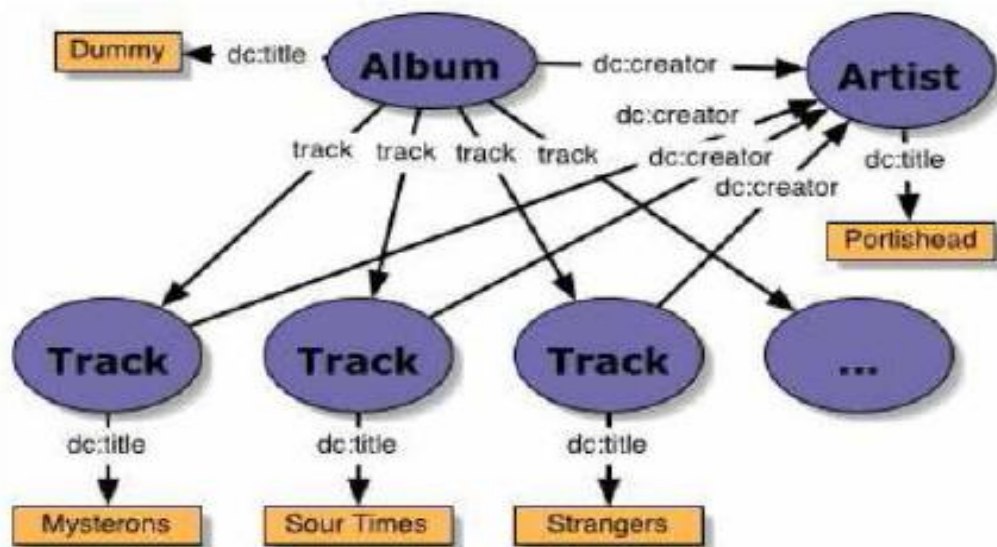


Figure 14 - Schéma RDF de Music Brainz

L'Editeur d'Ontologies Protégé

Protégé [48] est un éditeur d'ontologies distribué en open source par l'université en informatique médicale de Stanford. Il permet de construire une ontologie pour un domaine donné, de définir des formulaires d'entrée de données, et d'acquérir des données à l'aide de ces formulaires sous forme d'instances de cette ontologie.

Protégé est aussi une plate-forme extensible, grâce au système de plug-ins, qui permet de gérer des contenus multimédias, interroger, évaluer et fusionner des ontologies, etc. L'outil

Protégé possède une interface utilisateur graphique (GUI) lui permettant de manipuler aisément tous les éléments d'une ontologie : classe, propriété, instance,...etc. Protégé peut être utilisé dans n'importe quel domaine où les concepts peuvent être modélisés en une hiérarchie des classes.

Protégé permet aussi de créer ou d'importer des ontologies écrites dans les différents langages d'ontologies tel que : RDF-Schéma, OWL, DAML, OIL, ...etc. Cela est rendu possible grâce à l'utilisation de plugins qui sont disponibles en téléchargement pour la plupart de ces langages.

Conclusion

Dans ce chapitre nous avons présenté le Web sémantique, qui permet l'exploitation des ressources sur le web par les humains et les machines. Le principe du Web sémantique est de mettre en place une structure permettant de donner un sens au contenu du Web en typant non seulement les ressources mais aussi les liens entre elles dans le Web. Il repose sur une architecture en couche et est considéré comme une extension du Web actuel. Ainsi, le Web Sémantique permet, à travers les standards proposés, une adaptation rapide et une utilisation facile du contenu web existant, en y ajoutant de la sémantique.

Chapitre 2

Sur la notion d'ontologie

Au Sommaire de ce chapitre

2.1 Introduction

2.2 Apparition d'ontologie

2.3 La notion d'ontologie

2.3.1 L'interopérabilité

2.3.2 La réutilisation

2.4 Les constituants d'une ontologie

2.5 Discussion

2.5.1 Différence entre ontologie et base de connaissance

2.5.2 Différence entre ontologie et hiérarchie de classes

2.6 Classifications d'ontologies

2.6.1 Classification selon l'objet de conceptualisation

2.6.2 Classification selon le niveau de complétude

2.6.3 Classification selon le niveau de détail

2.6.4 Classification selon le formalisme utilisé

2.7 Méthodes et outils de construction d'ontologie

2.7.1 METHONTOLOGY

2.7.2 L'approche TOVE

2.7.3 L'approche SENSUS

2.7.4 Outils d'ingénierie ontologique à partir du texte

2.7.5 Outils d'ingénierie ontologique à partir des modèles conceptuels ou opérationnels

2.8 Conclusion

2.1 Introduction

Les données sur le Web constituent une grande source d'informations représentées sous des formes utiles pour l'être humain mais difficiles pour des traitements automatiques. Le contenu des ressources du Web peut être décrit en utilisant les méta-données formelles. Ces dernières doivent s'appuyer sur des ontologies afin de pouvoir les exploiter par des machines et les munir de sémantique.

L'objectif principal d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné. Elle jouent ainsi un rôle important dans le Web sémantique et sont employées comme une forme de représentation formelle de la connaissance sur un sujet du monde réel et traduire, ainsi, un consensus explicite. Dans le Web sémantique, les ontologies sont utilisées pour, d'une part, modéliser des ressources du Web à partir des représentations conceptuelles des domaines concernés et, d'autre part, dans l'objectif de permettre un traitement automatiquement dessus.

2.2 Apparition d'ontologie

L'ingénierie des connaissances (IC) a pour but la résolution automatique des problèmes, alors que les Systèmes à Base de Connaissances (SBC) devraient permettre le stockage et la consultation et la modification des connaissances, ainsi que le raisonnement automatique dessus [02].

Le partage de connaissances entre systèmes informatiques permettra, de plus, une interaction et une coopération entre ces derniers et l'utilisateur humain. Cela se manifeste, par exemple dans : les systèmes d'aide à la décision, les systèmes d'enseignement assisté par ordinateur, la recherche d'information sur le web, etc. Pour permettre un traitement automatique efficace, les représentations à utiliser par les machines (les modèles) doivent être chargées de sens et cela en reliant les informations ainsi collectées et représentées à d'autres types d'informations vouées essentiellement à la sémantique sous-jacente. Cela a donné naissance à l'ingénierie ontologique.

2.3 La notion d'ontologie

Le mot «ontologie» vient du mots grec: **ontos** pour être et **logos** pour univers et est employé dans des contextes très différents (la philosophie, la linguistique ou l'intelligence artificielle).

Dans [03], cette notion est définie comme suit : «une ontologie est une spécification formelle, explicite d'une conceptualisation partagée». Cette définition est enrichie dans [04] par l'explication suivante : "Une conceptualisation fait référence à un modèle abstrait de certains phénomènes du monde, modèle qui identifie les concepts pertinents de ce phénomène. Explicite, signifie que le type des concepts utilisés et les contraintes sur leur utilisation sont explicitement définis. Formelle, se réfère au fait que l'ontologie doit être compréhensible par les machines. Partagée, reflète la notion de connaissance consensuelle décrite par l'ontologie, c'est-à-dire qu'elle n'est pas restreinte au point de vue de certains individus seulement, mais reflète un point de vue plus général, partagé et accepté par un groupe".

Ainsi une ontologie représente un modèle conceptuel sur lequel il est possible de développer des systèmes de connaissances qui soient partageable et réutilisables permettant ainsi, l'interopérabilité entre plusieurs sources d'information et de connaissances.

2.3.1 L'interopérabilité

L'information échangée entres personnes est souvent perçue différemment par ces derniers ce qui est dû principalement aux ambiguïtés des langages utilisés (expressions, mots, etc.). Cela se traduit directement dans les idées et concepts transmis.

Les ontologies peuvent constituer une solution à ce problème en préservant la cohérence et en éliminant la confusion terminologique concernant les termes utilisés. Cela contribuera à résoudre le problème d'hétérogénéité sémantique en exprimant explicitement la sémantique inhérente permettant ainsi, plus d'interopérabilité entres les différents acteurs d'un système (machines ou humains). Des systèmes peuvent, dès lors, coopérer sans avoir aucune confusion concernant les objet, concepts, appels, méthodes, etc. provenant des systèmes différents [08].

2.3.2 La réutilisation

Les ontologies constituent un moyen efficace de réutilisation des concepts et méthodes déjà établis, réduisant ainsi, le temps et le coût de développement des systèmes. En effet, les ontologies permettent de caractériser les classes, les règles, les tâches, les relations, etc. d'un domaine donné. Les parties à réutiliser sont adaptées, dans un second temps, aux besoins de l'application finale. Cette adaptation se fait en "contextualisant" les connaissances décrites dans le système [10], en identifiant le scénario de réutilisation (des concepts et les relations d'un domaine).

2.4 Les constituants d'une ontologie

Les connaissances décrivant un domaine spécifié sont représentées par les éléments d'ontologie suivants [11]:

- ✓ Concept / classe.
- ✓ Individus.
- ✓ Propriétés.
- ✓ Relations de spécialisation.
- ✓ Restrictions / règles / axiomes.

2.4.1 Concept /classe

Un concept représente un ensemble d'objets et leurs propriétés communes. Les concepts peuvent être une idée, un principe, une notion profonde, ils sont appelés aussi termes ou classes de l'ontologie. Ces concepts peuvent être classifiés en plusieurs niveaux:

- ✓ Niveau d'abstraction (concret ou abstrait).
- ✓ Niveau d'atomicité (élémentaire ou composée).
- ✓ Niveau de réalité (réel ou fictif).

Les propriétés portant sur des concepts sont [12]:

L'abstraction: un concept est abstrait si toute instance de ce concept est aussi instance d'un de ses concepts. Par exemple dans une hiérarchie comportant les concepts *Homme* et *Femme*, fils du concept *Humain*, le concept *Humain* est abstrait.

La subsomption: un concept C1 subsume un concept C2 si toute propriété sémantique de C1 est aussi une propriété sémantique de C2, c'est-à-dire si C1 est plus spécifique que C2. Par exemple *Homme* subsume *Humain*

L'équivalence: deux concepts sont équivalents s'ils ont la même extension. Par exemple, *Homme* et *male adulte*.

La disjonction (on parle aussi d'incompatibilité): deux concepts sont disjoints si leurs extensions sont disjointes. Par exemple, *Homme* et *Femme*.

Si tous les concepts fils d'un concept abstrait sont disjoints deux à deux, ils forment une partition du concept abstrait.

2.4.2 Individus / Instances

L'individu est une instance de concept, autrement dit, c'est l'élément décrit par le concept, par exemple les individus Djamel et Samir sont des occurrences du concept *Personne*.

2.4.3 Relations

Les relations d'une ontologie désignent différentes interactions et corrélations entre les concepts de l'ontologie. Ces relations réunissent les associations suivantes: sous classe de (spécification ou généralisation), partie de (agrégation ou composition), associé a, instance de, est un ... etc.

Généralisation / spécification (généralisation est une relation inverse de spécialisation) est une relation binaire entre un concept général et un concept plus spécifique.

Composition/agrégation (relation composé-composant) est une relation qui permet de construire des concepts complexes à partir d'autres concepts appelés composants.

2.4.4 Restrictions / règles / axiomes

Pour exprimer toute chose autre que concept ou propriété:

Restriction: chose qui doit être vraie pour valider ce qui est exprimé.

Règle: affirmation sous la forme antécédent > conséquent décrivant des inférences possibles (ex: $\text{oncle}(x,y) \rightarrow \text{frère}(x,z) \wedge \text{père}(z,y)$).

Axiome: assertion générale sur les fondements de l'ontologie, les axiomes sont utilisés pour décrire les assertions de l'ontologie qui seront considérés comme vrais. Cette détermination a

pour but de définir les significations des composants d'ontologie, les contraintes sur les valeurs des attributs, et les arguments de relations.

2.4.5 Fonction:

Elles constituent des cas particuliers de relation dans laquelle un élément de la relation, le n ème, est défini en fonction des $n-1$ éléments précédents.

2.5 Discussion

2.5.1 Différence entre ontologie et base de connaissance

Une différenciation claire entre « ontologie » et « base de connaissances » devait se faire à partir de son rôle. Une ontologie fournit un système de concepts qui sont utilisées pour construire une base de connaissances par-dessus; par conséquent, une ontologie peut être une spécification de la conceptualisation du monde cible que se fait l'ingénieur qui construit la base de connaissances, donc un méta système d'une base de connaissances traditionnelle [13].

2.5.2 Différence entre ontologie et hiérarchie de classes

Au niveau supérieur, la méthode de développement d'une ontologie et celle d'une hiérarchie orientée objet sont similaires.

Cependant au niveau inférieur, l'ontologie se concentre sur les aspects déclaratifs tandis que la hiérarchie orientée objet se concentre sur les aspects reliés à la performance. Par conséquent, la différence essentielle entre les deux est que l'ontologie exploite la représentation déclarative, tandis que le paradigme orienté objet est intrinsèquement procédural, la signification d'une classe, d'une relation entre des classes, ainsi que les méthodes sont intégrées de façon procédurale.

Dans le paradigme ontologique, [13]. Les descriptions sont faites de façon déclarative, ce qui permet au système de modifier son comportement en modifiant la connaissance qu'il possède.

2.6 Classifications d'ontologies:

On peut distinguer quatre typologies selon plusieurs critères [14]:

- Classification selon le niveau de complétude.
- Classification selon l'objet de conceptualisation.
- Classification selon le niveau du formalisme.
- Classification selon le niveau de détail.

2.6.1 Classification selon l'objet de conceptualisation

Les ontologies peuvent être subdivisées en plusieurs niveaux qui sont, entre autres :

Les ontologies de type thesaurus : Sont aussi appelées taxonomie, elles servent à définir un vocabulaire de référence.

Les ontologies du domaine : Ces ontologies expriment des conceptualisations spécifiques à un domaine. Elles sont réutilisables pour plusieurs applications de ce domaine. L'ontologie du domaine caractérise la connaissance du domaine où la tâche est réalisée.

Les ontologies applicatives : Ces ontologies contiennent des connaissances du domaine nécessaire à une application donnée, elles sont spécifiques et non réutilisable.

Les ontologies génériques ou ontologies de haut niveau (top-ontologies) : Ces ontologies expriment des conceptualisations valables dans différents domaines. Son sujet est l'étude des catégories des choses qui existent dans le monde. Comme les concepts de haute abstraction tels que les entités, les événements, les états, les actions, le temps, l'espace, les relations, etc.

Les ontologies de représentation ou méta-ontologies : Ces ontologies conceptualisent les primitives des langages de représentation des connaissances.

Les ontologies géographiques : les ontologies de l'espace plus spécifiquement dédiées à la description des concepts qui caractérisent l'espace comme le point, la ligne, etc... Ces ontologies sont typiquement élaborées par de grands organismes de normalisation. Les ontologies spatialisées (ou spatio-temporelles), sont des ontologies dont les concepts sont localisés dans l'espace. Une composante temporelle est souvent nécessaire en complément pour la modélisation de l'information géographique.

2.6.2 Classification selon le niveau de complétude

On peut définir trois niveaux de complétude :

Niveau sémantique : Tous les concepts, caractérisés par un terme/libellé, doivent respecter les quatre principes différentiels :

- ✓ Communauté avec l'ancêtre ;
- ✓ Différence, spécification, par rapport à l'ancêtre ;
- ✓ Communauté avec les concepts frères, situés au même niveau ;
- ✓ Différence par rapport aux concepts frères.

Ces principes correspondent à l'engagement sémantique et assurent que chaque concept aura un sens univoque et non contextuel associé. Deux concepts sont identiques si l'interprétation du terme/libellé à travers les quatre principes différentiels aboutit à un sens équivalent.

Niveau-Référentiel : Les concepts référentiels ou formels, se caractérisent par un terme/libellé dont la sémantique est définie par une extension d'objets. L'engagement ontologique spécifie les objets du domaine qui peuvent être associés au concept, conformément à sa signification formelle. Deux concepts formels seront identiques s'ils possèdent la même extension.

Niveau-Opérationnel : Les concepts du niveau opérationnel ou computationnel sont caractérisés par les opérations qu'il est possible de leur appliquer pour générer des interfaces ou engagement computationnel.

2.6.3 Classification selon le niveau de détail

On peut différencier les ontologies selon le niveau de description utilisé :

Granularité fine : Ce niveau correspond à des ontologies très détaillées, possédant ainsi un vocabulaire plus riche capable d'assurer une description détaillée des concepts pertinents d'un domaine ou d'une tâche.

Granularité large : Ce niveau correspond à des vocabulaires moins détaillés. Par exemple les scénarios d'utilisation spécifiques ou les utilisateurs sont déjà préalablement d'accord à propos d'une conceptualisation sous-jacente. Les ontologies de haut niveau possèdent une granularité large, compte tenu que les concepts qu'elles traduisent sont normalement raffinés ultérieurement dans d'autres ontologies de domaine ou d'application.

2.6.4 Classification selon le formalisme utilisé

On peut distinguer les ontologies selon le formalisme utilisé pour les exprimer.

Informelle : l'ontologie est exprimée en langage naturelle. Cela peut permettre de rendre plus compréhensible l'ontologie pour l'utilisateur, mais cela peut rendre plus difficile la vérification de l'absence de redondances ou de contradiction ;

Semi-informelle : l'ontologie est exprimée dans une forme restreinte et structurée de la langue naturelle ; cela permet d'augmenter la clarté de l'ontologie tout en réduisant l'ambiguïté ;

Semi-formelle : l'ontologie est exprimée dans un langage artificiel définit formellement ;

Formelle : l'ontologie est exprimée dans un langage artificiel disposant d'une sémantique formelle, permettant de prouver des propriétés de cette ontologie. L'intérêt d'une ontologie formelle est la possibilité d'effectuer des vérifications sur l'ontologie : complétude, non redondance, consistance, cohérence, etc.

2.7 Méthodes et outils de construction d'ontologie

La définition d'une démarche de construction d'ontologies reste un problème ouvert qui continue à susciter beaucoup d'intérêt dans la communauté du Web sémantique. La majorité des approches existantes commence la construction des ontologies par l'identification, puis l'organisation et la structuration des concepts et des relations à représenter. Ainsi, les ontologies réalisées sont très différentes les unes des autres. Dans cette section nous citons quelques méthodologies et outils existants.

2.7.1 METHONTOLOGY

METHONTOLOGY [15] se veut être une approche intégrée s'inscrivant dans le cadre dans un processus de gestion de projet complet, allant la spécification des besoins jusqu'à la phase réalisation et maintenance. Ce processus est composé des étapes suivantes:

1. Spécification : déterminer l'utilisation future de l'ontologie;
2. Conceptualisation : obtenir un modèle du domaine au niveau des connaissances;
3. Formalisation : transformation du modèle conceptuel en modèle formel;
4. Intégration : réutilisation d'autres ontologies;
5. Implémentation : construction d'un modèle opératoire utilisable par un ordinateur;
6. Maintenance : mise à jour de l'ontologie en cas de besoin.

METHONTOLOGY permet de caractériser les ontologies au niveau des connaissances et insiste sur la nécessité de travailler à partir de représentations intermédiaires des connaissances lors de la phase de conceptualisation.

Premièrement, il faut créer un glossaire de termes divisé en concepts et verbes. Les concepts sont regroupés en arbres de classification de concepts et les verbes permettent de créer des diagrammes de relations binaires. De ces deux structures, un dictionnaire des concepts est créé pour regroupe toutes les informations concernant ces derniers (nom et synonymes, instances, attributs de la classe et de ses instances, relations rattachées au concept). D'autres structures sont aussi créées, à savoir : table des relations binaires, table des attributs d'instances, table des attributs de classes, table des axiomes logiques, table des constantes, table des formules (pour calculer des valeurs d'attributs), arbres de classification des attributs et table des instances.

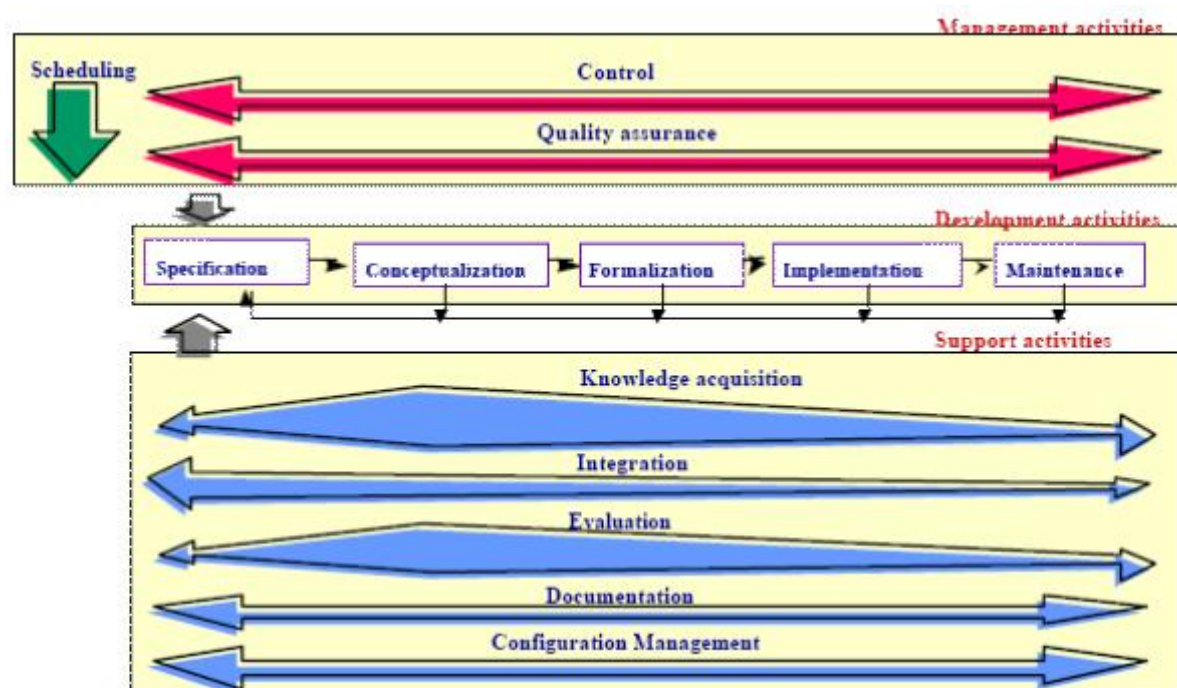


Figure 15 - processus de développement et le cycle de vie de METHONTOLOGY

2.7.2 L'approche TOVE

Cette méthode tente de construire un modèle logique de connaissance où les ontologies sont construites en respectant les étapes suivantes [17]:

- Identification de scénarii (problèmes) dépendants d'une application ;
- Formulation de questions informelles (basées sur les scénarii) auxquelles l'ontologie doit permettre de répondre ;
- Spécification d'une terminologie à partir des termes apparaissant dans les questions.
- Spécification formelle (en KIF) des axiomes et des définitions pour les termes de la terminologie.
- Evaluation de la complétude de l'ontologie.

2.7.3 L'approche SENSUS

Avec cette approche, les ontologies de domaine à développer sont dérivées d'une ontologie générale qui est l'ontologie SENSUS. Cela est fait en reliant les termes spécifiques du domaine à cette ontologie et d'enlever dans SENSUS, les termes qui ne sont pas pertinents dans la nouvelle ontologie. Ainsi, le squelette de cette nouvelle ontologie est générée automatiquement avec l'outil OntoSaurus.

La construction d'une ontologie dans un domaine spécifique suit les étapes suivantes :

- *Première étape* : identifier les termes clés du domaine.
- *Deuxième étape* : relier manuellement les termes clés à SENSUS
- *Troisième étape* : inclure tous les concepts qui se trouvent sur le chemin depuis le terme clé jusqu'à la racine de SENSUS.
- *Quatrième étape* : ajouter les nouveaux termes de domaine. Dans cette étape on ajoute manuellement tous les termes qui sont pertinentes pour le domaine et qui ne sont pas encore apparues. Ainsi les étapes 2 et 3 sont répétées pour inclure les concepts sur le chemin, des nouveaux concepts jusqu'à la racine de SENSUS.
- *Cinquième étape* : Ajouter le sous arbre entier. Dans cette étape, celui qui fait l'ontologie doit faire attention aux noeuds qui ont un grand nombre de chemins qui passe à travers le nouvel arbre généré.

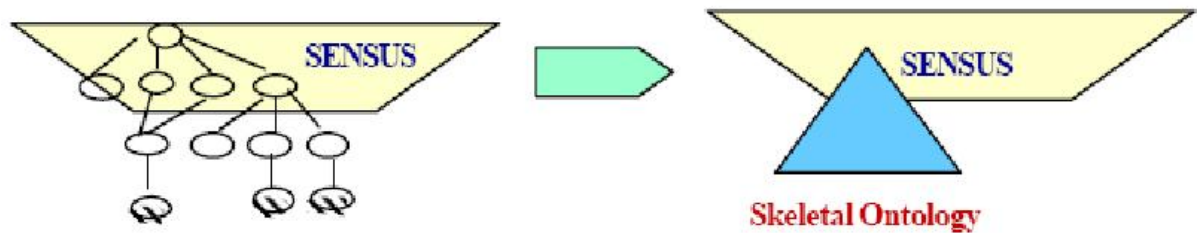


Figure 16 -approche SENSUS

2.7. 4 Outils d'ingénierie ontologique à partir du texte

2.7.4.1 TERMINAE

La méthodologie TERMINAE [20] propose une approche pour sélectionner les concepts, leurs propriétés, les relations et leur regroupement. Cette méthodologie repose sur l'utilisation d'outils de traitement automatique des langues analysant les termes de textes et les relations lexicales. Les termes sont regroupés suivant leur contexte et facilitent la création de concepts et de relations sémantiques. Les concepts et relations sont ensuite formalisés dans un modèle.

Cette méthodologie est composée de plusieurs étapes :

- La première consiste en la description des besoins (utilisation de l'ontologie, connaissance à représenter...).
- L'étape suivante conduit à construire un corpus sur lequel les outils de traitement automatique de langues se reposent.
- La troisième étape correspond à l'étude linguistique. Des outils sont appliqués au corpus afin d'extraire les termes et leurs relations lexicales et syntaxiques. Une application de la méthodologie est proposée par les auteurs à partir des outils LEXTER.
- La phase suivante, appelée phase de normalisation, vise à conceptualiser les résultats de l'étape précédente. Les termes à conserver sont sélectionnés en fonction de leur contexte et définis à partir d'une définition en langage naturel. Les concepts sont

ensuite identifiés ainsi que les relations sémantiques entre eux. Ils sont représentés sous forme d'un réseau sémantique.

- La dernière étape est celle de la formalisation. Le réseau sémantique précédemment obtenu est traduit et enrichi dans un langage formel.

TERMINAE a l'avantage de répondre à certaines questions et d'axer le choix des concepts et des relations de l'ontologie sur l'extraction de termes d'un corpus de référence

2.7.4.2 KAON

KAON [21] est un environnement de construction d'ontologie qui associe différents modules, correspondant à autant de phases identifiées de l'ingénierie ontologique : création, stockage, raffinage, exploitation, maintenance et application d'ontologies. Dans l'ensemble de ce processus, le module de « création ontologique » : TextToOnto applique des « stratégies de fouille de texte à des corpus textuels pour la création semi-automatique d'ontologies ».

2.7.4.3 Acquisition des termes à partir d'une analyse linguistique

Les avantages d'une analyse de corpus sont nombreux [19]:

- Les textes contiennent des connaissances explicites par opposition aux connaissances des experts souvent tacites, difficilement exprimables et difficilement accessibles.
- L'analyse automatisée de corpus est un gain de temps par rapport aux entretiens avec des experts du domaine.
- Les textes sont des ressources fiables et stables puisqu'ils fixent par écrit, à un instant t, un certain nombre de connaissances sur le domaine.
- Les modèles sont a priori plus facilement compréhensibles et donc maintenables car le retour aux textes est possible.

2.7.4.3.1 LEXTER & SYNTAX

Lexter extrait dans un premier temps des syntagmes nominaux maximaux, en s'arrêtant aux unités linguistiques constituant des frontières des syntagmes; ces frontières de syntagmes sont constituées par une ponctuation forte, un verbe conjugué, un pronom ou

déterminées selon des informations contextuelles complexes. Ensuite, « Le module acquiert par lui-même, à l'aide d'une méthode d'apprentissage endogène sur corpus, les informations de sous catégorisation des noms et des adjectifs, propres aux corpus, dont il a besoin pour résoudre les cas d'ambiguïté de rattachement prépositionnel.

Et enfin, un dernier module calcule les dépendances syntaxiques au sein des syntagmes nominaux maximaux et construit un réseau de têtes et d'expansions de ces candidats termes.

2.7.4.3.2 FASTR

FASTR, est un outil d'acquisition terminologique qui repère les variantes en corpus d'une même unité lexicale. Ce regroupement permet d'associer différents libellés à un seul terme. Le système calcule des variantes syntaxiques, morpho-syntaxiques et sémantico-syntaxiques.

2.7.5 Outils d'ingénierie ontologique à partir des modèles conceptuels ou opérationnels :

Il existe plusieurs travaux sur l'extraction automatique des termes à partir des schémas de bases de données conceptuel (E/A ou UML) ou relationnelle pour générer les ontologies. L'objectif principal de ces travaux est l'annotation des bases de données. Pour ce faire il n'est pas nécessaire d'aller construire l'ontologie à partir de zéro mais ils ont pensé de la développer soit à partir du modèle conceptuel ou à partir du schéma relationnel.

Z. Xu et al. [22] présente ER2WO qui est un outil d'extraction automatique basé sur une approche formelle de transformation du modèle conceptuel Entité/ Association vers une ontologie. ER2WO produit une ontologie codée en OWL DL à partir d'un fichier XML contenant le modèle E/A obtenu par l'outil PowerDesigner4, ER2WO valide ses résultats par l'OWL Ontology Validator 5.

Afin de réaliser cet outil, les auteurs ont défini un module de translation basé sur un ensemble de règles pour générer une ontologie dans la syntaxe abstraite du OWL DL. Cette ontologie constituera l'entrée d'un autre module de transformation pour générer l'ontologie finale dans la syntaxe RDF/XML.

Dans [23], [24] [25], ils proposent un mapping à partir d'un schéma relationnel de base de données à un langage de représentation d'ontologie (OWL lite, Relational.OWL, OWL, respectivement). Leur objectif est de faire la description d'un ensemble de BD hétérogènes par une ontologie pour lui permettre de bien communiquer ou pour faciliter la recherche d'informations dans ces diverses BDs.

L'utilisation d'un schéma relationnel pour créer l'ontologie donne des résultats acceptables mais avec une sémantique pauvre à cause de la grande quantité d'informations perdue lors du passage d'un MDC au MPD.

De plus, un simple changement au niveaux de la BD pose un problème au niveau d'ontologie. Dans [18] ils ont proposé une méthode nommé « Coévolution » dans laquelle les changements de la base de données sont appliqués directement à son ontologie d'annotation.

2.8 Conclusion

Nous avons présenté dans ce chapitre le concept de l'ontologie ainsi que les différentes taxonomies adoptées. Nous avons présenté, également, quelques méthodologies et outils automatiques pour construire des ontologies. La tendance actuelle cherche à définir de nouvelles méthodes pour l'acquisition automatique des connaissances dans l'objectif de réduire l'effort et le coût de la construction d'ontologie.

Chapitre 3

Méthodes de construction d'ontologie à partir des modèles conceptuels

UML

Au Sommaire de ce chapitre

3.1 Introduction

3.2 Quelques solutions permettant d'ajouter de la sémantique aux modèles conceptuels

3.2.1 Les profils UML

3.2.2 Le standard OCL

3.2.3 Action Semantics

3.2.4 La spécification ODM

3.3 Les raisons qui empêchent l'utilisation d'UML pour modéliser directement une ontologie

3.4 Quelques travaux

3.5 Conclusion

3.1 Introduction

Depuis quelques années il y a de plus en plus d'informations disponibles sur le Web d'où la nécessité d'avoir des outils automatiques pour exploiter cette masse de données judicieusement. Les ontologies représentent, en effet, une définition formelle des concepts et des relations qui existent au sein d'une collection de données permettant, par conséquent, l'automatisation de tâches par les machines.

Une ontologie doit être exprimée dans une syntaxe bien définie dans le but de représenter une connaissance et cela en définissant les termes et concepts où plusieurs aspects de l'information sont considérés (taxonomie des connaissances, thésaurus, lexique, modèles conceptuel, etc).

Les ontologies sont ainsi exprimée dans des langages de représentation de la connaissance (KR : Knowledge Representation) qui sont structurés suivant des formalismes de logique tels que la logique de prédicats. Cependant, ces langages rendent leur utilisation difficile aux développeurs de logiciel. Par conséquent, cela entrave leur diffusion et adoption par les technologies du Web Sémantique.

Sur un autre plan, UML [31] est un langage de modélisation adopté par une large communauté dans le cadre de modélisation conceptuelle. De plus, une large gamme d'outils UML est disponible ainsi que l'existence de plusieurs modèles de domaines et la similarité de ces modèles aux ontologies. Ainsi, il serait profitable de réunir les avantages d'UML et des ontologies et rapprocher la communauté UML et celle du Web Sémantique pour bénéficier de [32] :

- La génération de descriptions d'ontologies standard depuis des modèles UML existants ;
- La génération de modèles UML depuis une ontologie ;
- L'intégration d'ontologies standard dans les modèles UML [33].

3.2 Quelques solutions permettant d'ajouter de la sémantique aux modèles conceptuels

Dans ce que suit nous allons décrire les principes des profils UML, du langage de contrainte OCL et Action Semantics et La proposition ODM.

3.2.1 Les profils UML

UML est un outil de modélisation orienté objet. Il propose des concepts qui peuvent être utilisés dans plusieurs domaines à différents niveaux d'abstraction. Cela nous conduit, en regardant un modèle de savoir, à décider s'il s'agit de celui d'une application objet, d'un méta-modèle, d'une base de données ou tout autre concept [34]. D'où est né le concept de profil UML qui permet de spécialiser un modèle UML et l'adapter à un domaine d'application et cela d'une façon dynamique sans modifier la structure du méta-modèle.

Un profil UML est défini par le biais du concept de stéréotype. Ce dernier peut être considéré comme un label que l'on peut attribuer à n'importe quel élément d'un modèle UML permettant, ainsi à travers son nom, de définir une nouvelle signification de l'élément. Par exemple, coller un stéréotype nommé " Auxiliaire " sur une classe UML signifie que la classe en question n'est plus une simple classe UML mais qu'elle est une classe qui prend en charge une autre classe. Utiliser un profil consiste donc à coller sur un modèle UML un ensemble de stéréotypes. L'autre classe peut avoir le stéréotype « focus ».

Un stéréotype doit être adopté par tous les intervenants sur les mêmes modèles et se présente graphiquement comme est une chaîne de caractères qui doit apparaître au-dessus du nom de la classe comme le montre l'exemple de la figure 17.

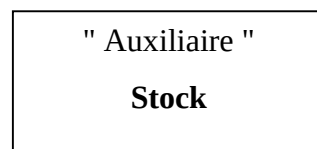


Figure 17- Exemple d'utilisation d'un stéréotype sur une classe

L'OMG a défini des profils standard qui définissent un ensemble de stéréotypes et expliquent leur signification en langage naturel. A ces profils est associé des traitements

spécifiques au domaine couvert et qui permettent de rendre les modèles UML profilés beaucoup plus productifs que les modèles UML simples puisqu'ils renferment des informations supplémentaires grâce aux stéréotypes.

UML2.0 Superstructure [31] facilite la création de nouveaux profils en simplifiant la définition des concepts de profils et de stéréotypes dans le méta-modèle. Ces concepts de profils sont représentés par des méta-classes. Un modèle instance de ces méta-classes correspond à la définition d'un nouveau profil et non à l'application d'un profil existant dans un modèle UML.

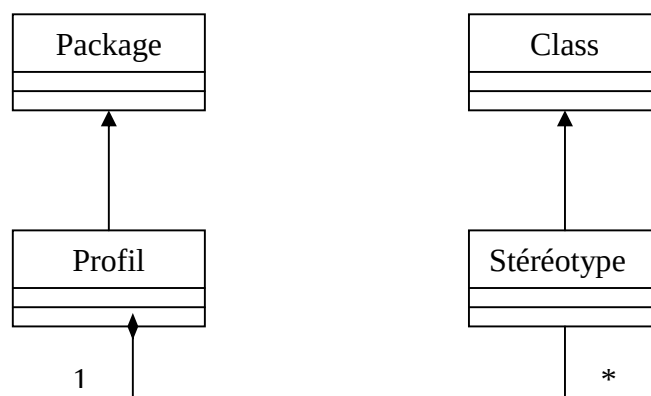


Figure 18 - Les profils dans UML2.0

La figure 18 illustre la partie du méta-modèle UML2.0 qui contient les méta-classes relatives aux profils. Dans ce méta-modèle la méta-classe Profil hérite de la méta-classe Package. Cela signifie que les définitions de nouveaux profils sont maintenant considérées comme étant des modèles UML (des packages). La méta-classe Stéréotype hérite de son côté de la méta-classe Class. Cela veut dire que les définitions de stéréotypes sont considérées comme étant des classes UML avec le nom de la classe correspondant au nom du stéréotype.

Avantages et inconvénients:

Les profils UML permettent de spécialiser un modèle UML. Ils peuvent être utilisés dans tous les modèles sans modifier leur structure. Et comme ces profils sont considérés comme un ensemble d'étiquettes, on peut traiter la sémantique dans des modèles en raisonnant sur leurs labels. Par ailleurs, il faut noter qu'un tel raisonnement, malgré qu'il est

possible, n'est pas exploitable par une machine car il n'est pas un raisonnement automatique puisque les profils UML ne permettent pas de définir une sémantique formelle.

3.2.2 Le standard OCL

Dans le méta-modèle UML, une opération est définie uniquement par son nom, ses paramètres et les exceptions qu'elle émet. Le langage OCL [35], Object Constraint Language, a été défini dans le but d'exprimer exactement le rôle d'une opération (l'ensemble des instructions) et permettre, donc, la modélisation du corps des méthodes (opérations) dans un modèle UML.

OCL peut exprimer des contraintes sur tous les éléments d'un modèle UML moyennant des pré et post-conditions sur les opérations. Par exemple l'opération `etablirBonDeLivraison` a comme pré-condition `verifierQuantitStock`, ce qui veut dire qu'avant de faire appel à l'opération qui permet la livraison de produits, il faut d'abord vérifier que les quantités en stock sont suffisantes.

Une contrainte OCL est une expression dont l'évaluation doit retourner vrai ou faux et son évaluation ne modifie pas l'état du modèle auquel elle est rattachée. Ainsi, on peut aisément vérifier si la contrainte est respectée ou non.

Il est à noter qu'une expression OCL traite un élément du modèle UML. Et pour l'évaluer, elle doit être rattachée à un contexte lié directement à un modèle UML. L'exemple suivant permet de spécifier, pour un bon de livraison, que la valeur de retour de l'opération `getStock` doit être positive avant toute invocation de l'opération `quantiteLivree` :

```
context etablirBonDeLivraison : : getStock () : quantiteLivree () :Integer  
pre : quantiteCommandee < quantiteDemandee
```

Le méta-modèle OCL permet de représenter différentes expressions textuelles OCL sous forme de modèle. Cette transformation se fait pour la plupart du temps d'une façon manuelle.

Avantages et inconvénients:

L'Object Constraint Language permet d'exprimer tous les types de contraintes sur des modèles UML. Par contre, OCL ne permet pas d'exprimer des opérations d'effets de bord. Aussi, il n'est pas possible de traiter automatiquement les contraintes OCL puisque la sémantique n'est pas formellement définie.

3.2.3 Action Semantics

Le langage OCL ne permet pas de spécifier des créations, des suppressions ou des modifications d'éléments de modèles, puisque celui-ci est sans effet de bord, ce qui a donné naissance au langage AS (Action Semantics) [34].

L'objectif de Action Semantics est de permettre de définir des actions. Avec ce langage, une action est une opération sur un modèle qui fait changer l'état du modèle. Grâce à ces actions, il est possible de modifier les valeurs des attributs, de créer ou de supprimer des objets, de créer de nouveaux liens entre les objets, etc. Ainsi le concept d'Action Semantics permet de spécifier tout le corps des opérations UML et est standardisé sous forme de méta-modèle . Cependant, aucune syntaxe n'est définie et c'est au concepteur d'outils de modélisation de fournir un format concret et de trouver les mécanismes pour le traduire en un format de modèles. Ce manque de format concret (textuel), rend l'utilisation de AS complexe.

Le package IntermediateActions de UML2.0 Superstructure [31] définit des méta-classes représentant des actions exécutables sur des éléments de modèle (contient une cinquantaine de méta-classes). Il existe, par exemple, des actions pour ajouter des liens entre les objets, émettre des messages, naviguer parmi les instances d'une classe, etc.

Afin de mieux comprendre l'importance d'un format concret, nous allons monter un exemple simple d'activité. Cette dernière contient trois actions : la création d'un article en stock, la réception d'une quantité donnée et la livraison de cet article avec une autre quantité. Nous allons exprimer cette activité en utilisant une syntaxe proche du langage Java de la façon suivante :

```
ArticleStock cd = new ArticleStock () ;  
cd.quantite = 100 ;  
cd.sortie(10) ;
```

La Figure 19 - donne une représentation de cette activité sous forme de modèle. Avec ce schéma, les actions sont sous forme de cercle et les flux de données sous forme de flèche. Les activités y sont définies, ces dernières définissent la séquence entre les actions. Nous constatons, rien avec cet exemple, combien il est difficile de passer d'un format concret à un format de modèle avec Action Semantic.

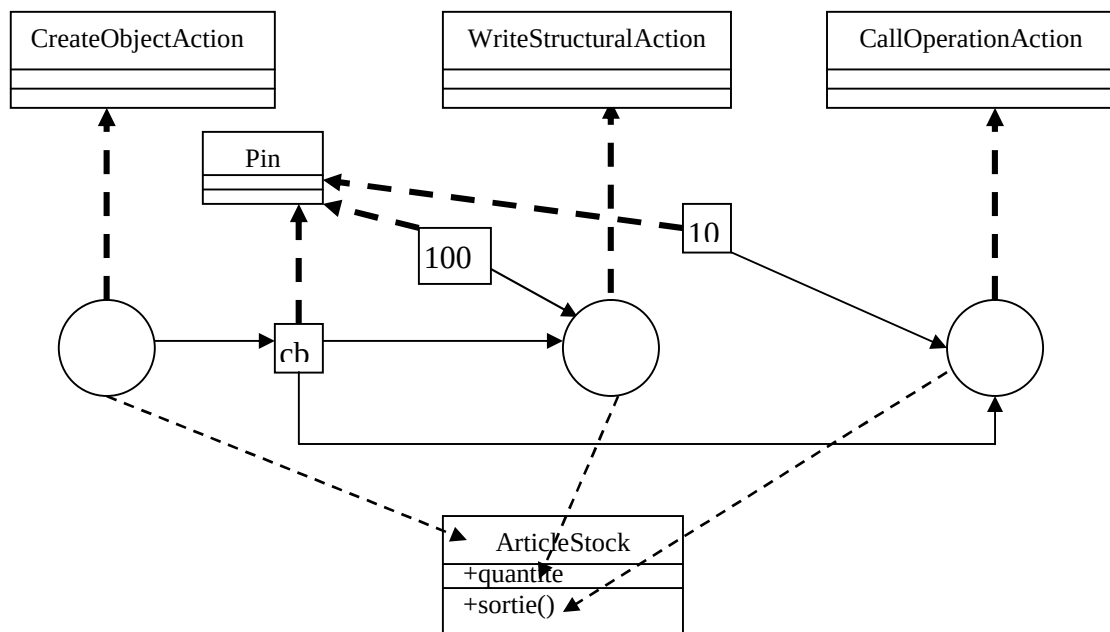


Figure 19 - Les actions sous forme de modèle dans Action Semantics

Avantages et inconvénients:

Action Semantics a été standardisé dans le but d'exprimer des opérations de création, de suppression et de modification d'éléments de modèles (en plus des opérations sans effet de bord). Ce qui permet de spécifier tout le corps des opérations dans un modèle UML. Cependant, AS existe seulement sous forme de méta-modèle et aucun format concret (textuel) n'est proposé ce qui rend son utilisation complexe. De plus, AS souffre du manque de formalisation de la sémantique (absence d'une possibilité d'automatisation)

3.2.4 La spécification ODM

La proposition ODM définit un ensemble de méta-modèles d'ontologies, de profils associés et de mise en correspondance et ce dans le but de permettre l'échange des

connaissances exprimées dans des formalismes différents, entre autres UML. En effet, ODM propose trois méta-modèles (RDFS, OWL et Topic Maps) pour les formalismes les plus utilisés dans le cadre du Web sémantique.

Il convient de noter qu'il existe plusieurs points communs entre UML et les langages de description des ontologies : classes, relations, propriétés, héritage, etc. Cependant, il existe aussi des différences significatives entre ces outils. Parmi les différences notoires, on peut citer le concept de propriété où dans UML, un attribut a une portée liée à la classe, alors que dans les ontologies, une propriété représente une notion de premier ordre qui peut exister séparément d'une classe. On peut remarquer, aussi, certaines limitations d'UML qui concerne, essentiellement la définition de propriétés sur les ensembles (disjonction, complément) ce qui ne permet pas l'utilisation d'outils d'inférence automatique sur des modèles UML.

Les profils UML qui sont des ensembles de stéréotypes et des labels peuvent s'appliquer sur la majorité des éléments d'un modèle tels que les classes, attributs et/ou les associations. Les profils UML existant dans ODM offrent une passerelle entre UML et les différents langages de représentation de connaissances.

3.3 Les raisons qui empêchent l'utilisation d'UML pour modéliser directement une ontologie

Il convient de noter que UML n'est pas assez puissante pour modéliser complètement une ontologie. En effet, il n'est pas possible de l'utiliser directement et cela pour les raisons suivantes [36] :

- UML ne permet aucune forme de traitement automatique de la sémantique ;
- En UML pour utiliser la notion de disjonction, il faut que les classes qui interviennent aient un super-type commun, ce qui n'est pas le cas en ontologie ;
- En UML une instance ne peut exister sans la classe qui définit sa structure alors qu'en ontologie, ceci est envisageable ;
- En UML une propriété, qui est un attribut ou une association, ne peut pas être un élément autonome au même titre qu'une classe (du fait de l'encapsulation) alors qu'en

modélisation d'ontologies une propriété existe en dehors de toute structure. En UML les propriétés sont de seconde classe alors que dans les ontologies elles sont de première classe [37].

- Méta-niveaux : Les langages d'ontologies n'ont pas une séparation rigide entre les méta-niveaux. Par exemple, en OWL Full, une instance d'une classe peut être une autre classe. Ce qui n'est pas le cas de UML.
- Spécialisation/généralisation : avec UML une relation de spécialisation/généralisation a pour but de permettre la réutilisation d'un comportement. Alors que dans les langages d'ontologies l'idée derrière une relation de spécialisation/généralisation est un ensemble théorique.
- Modularité : les langages d'ontologies n'ont pas de profils, de packages ou aucun autre mécanismes de modularité supportés par UML et les autres langages orientés objet.
- Conteneurs et listes : les packages UML ne sont pas au même méta-niveau que les conteneurs RDF (Bag, Seq et Alt). La manière de représenter les packages font qu'ils ne soient pas adaptés aux conteneurs RDF.
- Les langages d'ontologies peuvent construire des classes en utilisant des opérations booléennes (unions, intersection et complément) et des quantificateurs. En UML il n'y a pas de correspondant.
- Contraintes de cardinalité : cette incompatibilité vient de la nature première classe des propriétés d'ontologies. Dans les langages d'ontologies il est possible de spécifier une contrainte de cardinalité globale à un domaine d'une propriété, alors qu'avec UML la contrainte de cardinalité doit être spécifiée séparément pour chaque association vers une propriété.
- Espace de nommage (namespace) : les langages d'ontologies pour le Web Sémantique utilisent les URIs pour référencer leurs constructions. UML dispose de plusieurs manières de se connecter à ses constructions (par exemple chaque classe est un namespace de ses attributs, qui peuvent avoir une portée privée, protégée ou publique).

3.4 Quelques travaux

Il existe de nombreux travaux combinant UML et des langages de représentation d'ontologies. Parmi les travaux portant sur la génération d'ontologie dans une démarche d'ingénierie dirigée par les modèles [44]:

- Dans (Brockmans, 2006) est une démarche où l'objectif visé correspond à la visualisation d'ontologies OWL en utilisant la notation graphique d'UML,
- Il y a également le projet Eclipse EODM pour EMF Ontology Definition Metamodel qui se veut une implémentation du standard ODM en utilisant les technologies EMF (Eclipse Modeling Framework). EODM se propose d'offrir des éditeurs textuels et arborescents d'ontologies exprimées en RDFS et OWL. De plus, EDOM intègre une couche d'inférence et implémente les transformations UML vers RDFS et UML vers OWL par l'intermédiaire des APIs Java générées grâce à EMF.
- L'adaptation d'outils existants comme le module d'extension (plugin) UMLBackend [49] de l'éditeur d'ontologies Protégé. Ce module permet l'importation et l'exportation de modèles UML 1.4 au format XMI mais possède de nombreuses limitations : la transformation n'est pas configurable, plusieurs éléments importants d'UML comme les stéréotypes et les valeurs étiquetées ne sont pas prises en compte. L'outil OWLGrtd [50] se propose comme une solution à ce problème.

3.5 Conclusion

Les ontologies sont un paradigme puissant pour la représentation et l'échange des connaissances dans le domaine du Web Sémantique qui ne s'intéressent qu'au traitement automatique de la sémantique. Néanmoins, elles doivent être exprimées dans des formalismes qui devraient en tirer pleinement profit, tout en offrant des outils de modélisations simples et efficaces.

UML trouve beaucoup de succès au sein des concepteurs de modèles et a bénéficié, par conséquent, d'un cumul d'expérience très important ainsi qu'une large gamme d'outils dédiés. Cependant, ces outils (UML profile, OCL, AS) ne permettent pas la définition d'une sémantique formelle ce qui entrave l'inférence automatique sur des modèles UML.

Il serait profitable de tirer profit de ces deux puissants outils pour rajouter de la sémantique aux modèles conceptuels. Chacun de ces deux outils ayant ses propres mécanismes, il est donc nécessaire de trouver un point de rattachement qui est possible avec l'utilisation de ODM, où il est possible d'utiliser un même éditeur pour faire des modèles UML et des ontologies. ODM nous permettra de passer de UML au OWL et donc de pouvoir utiliser les mécanismes de traitement automatique.

Chapitre 4

Conception d'une ontologie

Au Sommaire de ce chapitre

4.1 Introduction

4.2 Utilisation des ontologies en biologie

4.3 Conception de l'ontologie du domaine de l'écologie évolutive

4.4 Respect des principes de construction

4.5 Présentation de cette ontologie

4.6 L'héritage multiple

4.7 Les propriétés des classes de cette ontologie

4.8 Représentation hiérarchique des concepts

4.9 Conclusion

4.1 Introduction

Dans ce chapitre nous présentons une ontologie dédiée à la description du domaine de l'écologie évolutive, branche de la biologie. La conception a été faite dans une perspective de construction d'un vocabulaire partageable pour l'objectif recherché, de couvrir et de représenter l'ensemble des mots et expressions utilisés dans ce domaine.

4.2 Utilisation des ontologies en biologie

Les informations issues du domaine de la science de la vie et de la nature sont de grande taille, chose qui est due principalement à la diversité, non encore cernée, de la nature et des êtres vivants. Structurer cette abondante quantité d'information est maintenant nécessaire pour pouvoir extraire les données pertinentes en un temps acceptable ce qui permettra de mieux connaître notre présent pour pouvoir anticiper et bien préparer notre futur. Les ontologies offre une solution pour organiser et partager ces connaissances, offrant ainsi un outil automatique capturant à la fois données et sémantique.

La liste des ontologies disponibles dans le domaine de la biologie :

- Ontologie de Gène (*Gene Ontology* – GO -)
- Ontologie des Séquences Biologiques (*Sequence Ontology* – SO -)
- Normes et Ontologies pour les Génomiques Fonctionnelles (*Standards and Ontologies for Functional Genomics* – SOFG -)
- *Microarray Gene Expression Data* (MGED)

4.3 Conception de l'ontologie du domaine de l'écologie évolutive

4.3.1 Domaine de l'ontologie

Dans ce qui suit, nous présentons ce domaine et sa portée.

A. Introduction à l'écologie évolutive

L'écologie permet de faire le lien entre trois disciplines de la biologie:

- *la génétique*: qui étudie les mécanismes de l'hérédité;

- *l'évolution*: qui s'intéresse aux changement de composition génétique des populations;
- *la biologie de développement*: qui s'intéresse à la transformation du matériel génétique en individus.

L'écologie évolutive; s'intéresse particulièrement aux formes et aux fonctionnements des organismes en tant qu'adaptation à leur environnement. C'est une branche de la biologie évolutive qui traite plus particulièrement de l'influence des contraintes de l'environnement sur l'évolution des comportements (éthologie) et des organismes vivants.

Les sciences de l'évolution se partagent entre la paléontologie qui étudie les formations et organismes du passé et en biologie ou écologie évolutive qui étudie les organismes actuels et leur variation. L'écologie évolutive met particulièrement l'accent sur les contraintes des milieux et sur la variation des stratégies des organismes qui y répondent. En s'intéressant aux stratégies sexuelles, à la sélection naturelle, à la sélection sexuelle et au conflit sexuel, l'écologie évolutive peut se définir comme une discipline appartenant aux sciences de l'évolution.

Ainsi, les sciences de l'évolution impliquent toutes les sciences du vivant (génétique et génomique, écologie, biologie du développement, etc.) et s'interfacent avec tout un ensemble de champs disciplinaires tels que les sciences de l'univers (paléontologie, paléo-environnements et paléoclimats), l'informatique (bioinformatique), les mathématiques (modélisation) ou les sciences humaines (biologie évolutive humaine) [38].

B. Les besoins examinés par cette ontologie

Présentation

Ecosystème est une unité écologique fonctionnelle qui regroupe une communauté animale et végétale (biocénose), et le milieu que cette communauté occupe (biotope). La biocénose et le biotope constituent deux éléments indissociables qui réagissent l'un sur l'autre pour former un système plus ou moins stable, l'écosystème. Ainsi, un étang ou une forêt constituent chacun un l'écosystème. L'ensemble des l'écosystèmes forme la biosphère, mince couche superficielle de la Terre occupée par les êtres vivants.

Les êtres vivants présentent une grande diversité de formes et de modes de vie, mais certains possèdent des caractères communs. On s'intéresse dans cette ontologie de mettre l'accent sur l'être vivant, son milieu de vie, l'écosystème qui le comprend, comment il est classé, la diversité des classifications appliquées, sur les taxons qui sont la base de certaine classification, ainsi que le type de reproduction adopté et la structure cellulaire de chaque espèce vivante.

4.3.2 Processus d'affinement de l'ontologie

On ne peut pratiquement dissocier les étapes de construction d'une ontologie, car il s'agit d'un processus non linéaire, plusieurs allers-retours sont souvent nécessaires lors du développement d'une ontologie, pour les raisons suivantes:

- il n'est pas possible de savoir dès le départ, que les termes collectés sont suffisant pour répondre à l'objectif pour lequel l'ontologie a été construite. Il est possible d'ajouté des nouveaux termes lorsque cela s'avère nécessaire, comme il est possible, aussi, de retirer des termes jugés inutiles.
- il n'est pas toujours facile de prédire qu'un terme va jouer le rôle d'une classe ou celui d'un attribut, plusieurs modifications seraient effectuées dans ce sens.

Pour représenter l'ontologie choisie, il est nécessaire de construire:

- ✓ Une liste de concepts.
- ✓ Une liste d'attributs.
- ✓ Une liste de relations
- ✓ Une représentation hiérarchique des concepts.

4.4 Respect des principes de construction

- *Clarté et objectivité*: pour répondre à ce principe, tous les termes utilisés dans cette ontologie ont été associés à des définitions.
- *Complétude*: pour répondre à ce principe, les définitions des concepts et des relations de cette ontologie ont été associés à des conditions nécessaires, d'autres ont été associés à des conditions nécessaires et suffisantes, dans la mesure de pouvoir définir ces conditions.

- *Extensibilité ontologique maximale*: la définition d'un terme n'explique que le terme lui-même, sa définition ne peut être la même que celle d'un terme plus général, ou d'un terme plus spécialisé.
- *Principe de distinction ontologique*: les concepts dans cette ontologie sont suffisamment disjoints.

4.5 Présentation de cette ontologie

Une ontologie est composée d'un vocabulaire partagé. Un premier travail consiste donc à construire un corpus, qui rassemble le vocabulaire employé dans le domaine. A partir du vocabulaire contenu dans le corpus, on a extrait les primitives constituant les concepts de cette ontologie.

4.5.1 Les axes primitifs de l'arbre ontologique construit

A. Classification scientifique des espèces vivantes [39]

i. Méthodes

La classification scientifique classe les êtres vivants selon leurs ressemblances, on réunit les organisme ayant au moins un caractère en commun (ex: animaux possédant une colonne vertébrale → groupe des vertébrés). En utilisant de nombreux caractères, on peut diviser le groupe en d'autres groupes plus petits.

Pour classer les être vivants, on utilise des *critères morphologiques*: nombre de pattes, forme des os ... l'originalité de la classification scientifique est que son but est de regrouper les individus dans l'objectif de comprendre l'évolution: les espèce d'un même groupe sont celles qui possèdent un ancêtre commun.

ii. Intérêts

Dans la classification scientifique, l'espèce est l'unité de base, le groupe élémentaire. On ne divise pas les membres d'une même espèce en plusieurs groupes.

B. L'arbre phylogénétique

Un arbre phylogénétique est un arbre schématique qui montre les relations de parentés entre des entités supposées avoir un ancêtre commun. Chacun des nœuds de l'arbre

représente *l'ancêtre commun* de ses descendants; le nom qu'il porte est celui du clade formé des groupes frères qui lui appartiennent, non celui de l'ancêtre qui reste impossible à déterminer. L'arbre peut être enraciné ou pas, selon qu'on est parvenu à identifier l'ancêtre commun à toutes les feuilles.

Les arbres phylogénétiques ne considèrent pas les transferts horizontaux, et un nouveau modèle se développe en phylogénie, celui de graphe ou réseau phylogénétique qui permet de les prendre en compte, ainsi que les recombinaisons. Il existe plusieurs techniques de construction des arbres phylogénétique, plus ou moins rapides et plus ou moins fiables.

Il est important de noter que la différence entre la classification classique des espèces vivantes et la classification phylogénétique c'est que la classification classique est une classification *fixe* qui suppose que toutes les espèces sont apparues en même temps et repose sur la technique « bottomup », contrairement à la classification phylogénétique qui est une classification variable représente une notion d'origine évolutive commune et repose sur la technique « top-down ».

C. Taxon

En taxinomie, un taxon est une entité conceptuelle qui est censée regrouper tous les organismes vivants possédant en commun certains caractères taxinomiques ou diagnostiques biens définis.

Dans la classification classique, ces caractères sont réputés homogènes en fonction de leur rang taxinomique. Leur poids (valeur taxinomique relative) étant laissée à l'appréciation des systématiciens. Dans les classifications les plus modernes, comme la classification phylogénétique, les rangs de taxons n'existent pas et les taxons sont organisés en clades qui s'emboîtent les uns à l'intérieur des autres.

La classification classique propose une hiérarchie codifiée en sept rangs principaux et cinq rangs secondaires, présentée, dans l'ordre décroissant, de la façon suivante:

Monde vivant:

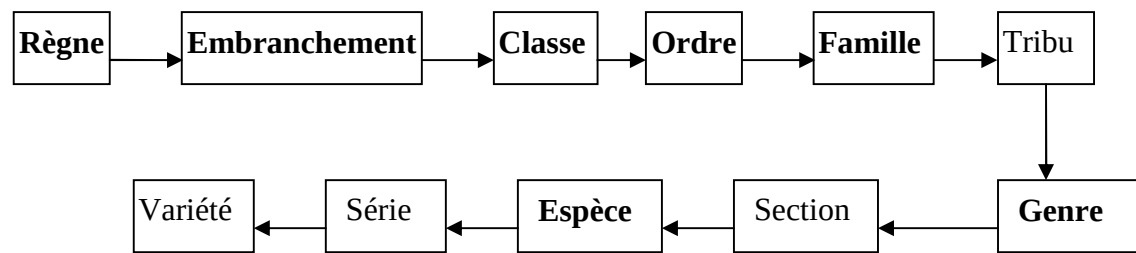


Figure 20 - L'ordre des Taxons selon la classification classique

Dans cette ontologie, on n'a présenté que les sept rangs principaux. En passant à la classification phylogénétique, au dessus du règne, on parle maintenant d'empire:

Monde vivant: **empire** → **règne** → **embranchement** →

D. La reproduction

Les êtres vivants peuvent se reproduire, c'est-à-dire donner naissance à d'autres êtres vivants semblables à eux-mêmes. Les mécanismes de cette reproduction sont très différents selon les espèces. C'est une des activités fondamentales, partagée par tous les êtres vivants (avec la nutrition et la croissance). En effet, toute espèce doit posséder un système de reproduction efficace, sans quoi elle est menacée d'extinction.

On distingue deux types de la reproduction:

- ✓ La reproduction sexuée
- ✓ La reproduction asexuée

E. La structure cellulaire

Cet axe permet de décrire la structure *physiologique* de certaine espèce si elle est unicellulaire, pluricellulaire ainsi que si elle a de noyau ou elle est dépourvue de ce dernier.

F. L'écosystème

Il est composé de:

i. biotope

Un biotope est le milieu physique et chimique dans lequel vivent les végétaux et les animaux. Ce milieu est l'élément non vivant, ou abiotique, de l'écosystème. Il renferme la totalité des ressources nécessaires à la vie.

ii. Biocénose

Le second élément de l'écosystème comprend l'ensemble des êtres vivants, végétaux, animaux, et micro-organismes, qui trouvent dans le milieu des conditions leur permettant de vivre et de se reproduire. L'ensemble de ces êtres vivants constitue une communauté (une biocénose).

Les trois catégories d'organismes d'une biocénose sont les producteurs (les végétaux chlorophylliens), les consommateurs (animaux herbivores et carnivores), les décomposeurs (champignons, bactéries et certains animaux).

4.6 L'héritage multiple

Au cours de la construction de la hiérarchie de classes, la notion de l'héritage multiple était apparue à certain niveau: une classe peut être une sous-classe de plusieurs classes. L'emplacement d'un être vivant dans chaque taxon-base se diffère d'une classification à une autre.

4.7 Les propriétés des classes de cette ontologie

Les classes seules ne fourniront pas assez d'informations pour assurer la compétence de l'étape 1. Après avoir défini l'ensemble des classes, on devait décrire la structure interne des concepts.

4.7.1 Propriété d'objet

Sont des propriétés qui relient les instances d'une classe aux instances d'une autre classe. Chaque propriété dans l'ontologie relie les classes entre elles. Ces dernières font partie d'un domaine.

4.7.2 Propriété de type de données

Relie une instance à des valeurs. Dans cette ontologie on s'y intéresse de représenter comment un être vivant est classé, et de représenter la classe à laquelle il appartient, comment elle s'adapte avec son environnement, et comme les espèces vivantes sont très nombreuses, il est impossible de citer les instances de chaque classe. Toutefois, pour montrer

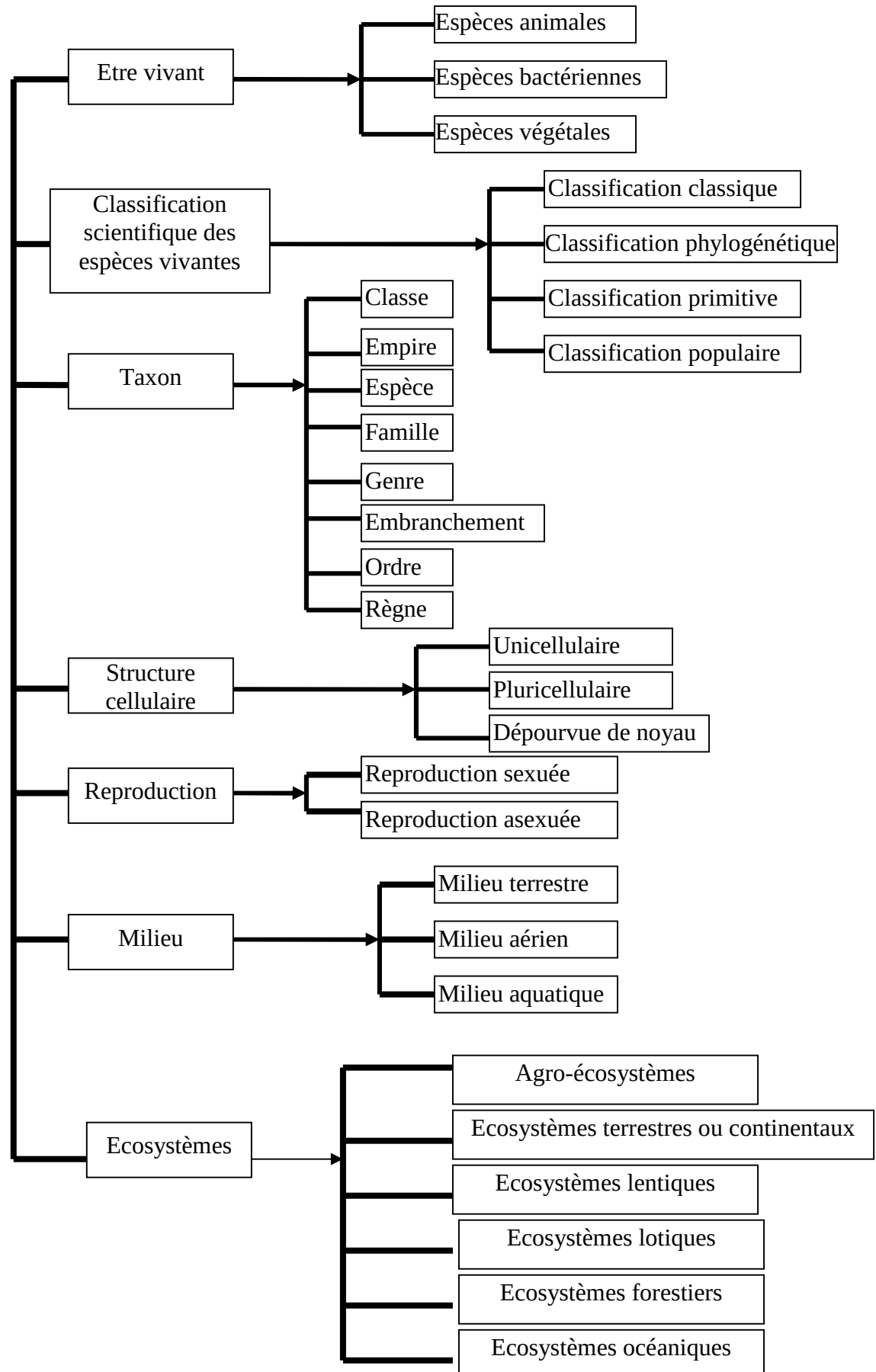
les notions «instance et valeur», on a pris 3 êtres vivants bien connus et sont classifiés selon la classification classique: L'homme, le chat et le merle.

Taxon	Homme	Merle	Chat
Règne	Animale	Animale	Animale
Embranchement	Vertébrés	Vertébrés	Vertébrés
Classe	Mammifères	Oiseaux	Mammifères
Ordre	Primates	Passériformes	Carnivores
Famille	Hominidés	Turdidés	Félidés
genre	Homo	Turdus	Felis
Espèce	Sapiens	Merulas	Silvetris

Tableau – Propriété de type de données

4.8 Représentation hiérarchique des concepts

L'ontologie choisit, couvre un bon nombre de termes, c'est pour cette raison que la hiérarchie de classe ci-dessous ne représente que les classes primitives.



4.9 Conclusion

Dans ce chapitre nous avons présenté les différentes phases pour concevoir une ontologie en suivant des étapes bien définies. Cela a permis d'avoir une liste de concepts, une liste de relations et une autre liste des attributs en plus d'une représentation hiérarchique des connaissances du domaine. La prochaine étape consiste à rendre cette ontologie opérationnelle et exploitable par un ordinateur.

Chapitre 5

Implémentation

Au Sommaire de ce chapitre

5.1 Introduction

5.2 Environnement de développement

5.3 Implémentation

5.4 L'exportation et l'importation de cette ontologie

5.5 Le raisonnement

5.6 Conclusion

5.1 Introduction

Dans ce chapitre nous décrivons une implémentation de l'ontologie présentée dans le chapitre précédent. Les outils utilisés sont Protégé2000 et les extension Pellet et UMLBackend et OWLGrtd.

5.2 Environnement de développement

L'ontologie peut être éditée avec l'éditeur d'ontologie Protégé et implémentée avec les techniques OWL DL. Avant l'édition de l'ontologie avec Protégé, on choisi le langage de spécification. Celui que nous avons choisi est le langage OWL DL, parce que, d'une part, l' OWL DL permet d'exprimer des cardinalités multiples et d'autre part, les autres langages sont insuffisants ou plus complexe.

RDFS paraissait insuffisant pour coder l'ontologie présentée, dans la mesure où il ne permet pas d'exprimer des contraintes de cardinalité. De même, OWL Lite ne permet d'exprimer que des contraintes simples de cardinalité 0 ou 1. Au contraire, OWL Full offre un plus haut niveau d'expressivité plus que nécessaire à l'ontologie présentée, tandis qu' OWL DL offre un niveau d'expressivité assez suffisant.

En plus, le codage d'une ontologie sous format OWL présente l'avantage de rendre cette ontologie réutilisable, grâce à l'utilisation des propriétés d'équivalence, de disjonction entre les concepts et les relations. Cette ontologie est vérifiée, finalement, à l'aide du raisonneur Pellet.

5.3 Implémentation

5.3.1 Normalisation des noms de l'ontologie

Les caractéristiques suivantes de l'éditeur d'ontologies affectent le choix de la convention de nomination:

- Le système a-t-il le même espace de nomination pour les classes, attributs et instances? C'est-à-dire, permet-il d'avoir une classe et un attribut ayant le même nom.

- Quels délimiteurs le système autorise-t-il pour les noms? C'est-à-dire, les noms peuvent-ils contenir des espaces, des virgules, etc. ?

L'éditeur d'ontologie Protégé avec lequel cette ontologie a été éditée, maintient un espace de nommage unique pour les classes, attributs et instances, et il ne permet pas l'apparition des espaces dans les noms, les délimiteurs autorisés sont bien "-" et "_" .

C'est pour ces considérations que les noms déjà présentés dans l'ontologie conceptuelle ont été légèrement modifiés, lors de leurs éditions sous Protégé.

5.3.2 Création d'un nouveau projet

Dans ce qui suit nous allons présenter comment nous avons édité cette ontologie à partir du lancement de Protégé jusqu'à la génération du code OWL.

Création d'un nouveau projet après Lancement de Protégé3.4.4 sous windows XP

Il est réalisé en précisant respectivement le type de projet, l'espace des noms, le langage avec lequel sera éditée l'ontologie.

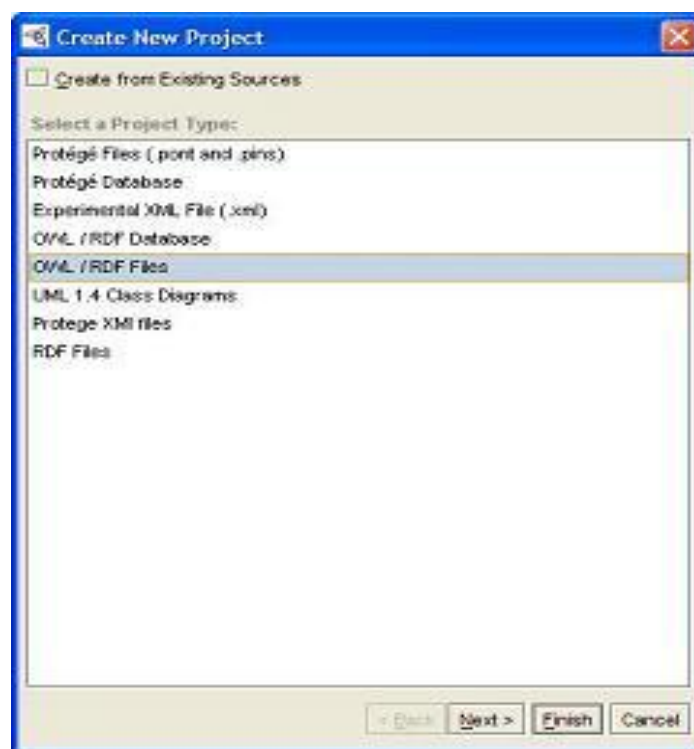


Figure 21 - Choix de type de projet



Figure 22 - Choix d'un espace des noms



Figure 23 - Choix d'un langage

5.3.3 L'édition de l'ontologie

Après avoir spécifié les propriétés du projet, sur la fenêtre suivante que nous allons éditer l'ontologie. Nous commençons par la création des classes de façons hiérarchique, et nous définissons les classes disjointes

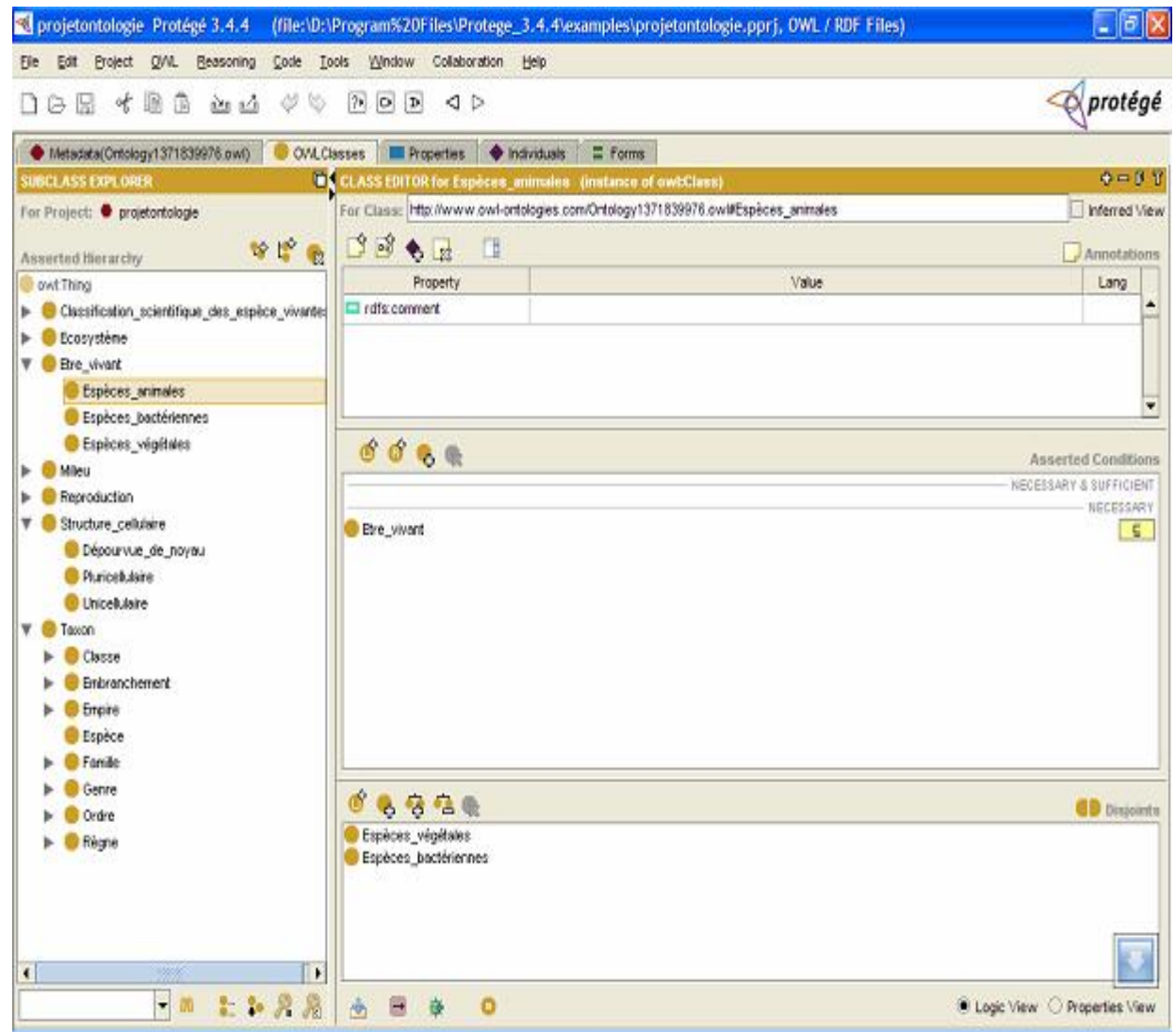


Figure 24 - Création des classes

Après la création des classes, nous pouvons éditer les propriétés, les relations et les relations inverses.

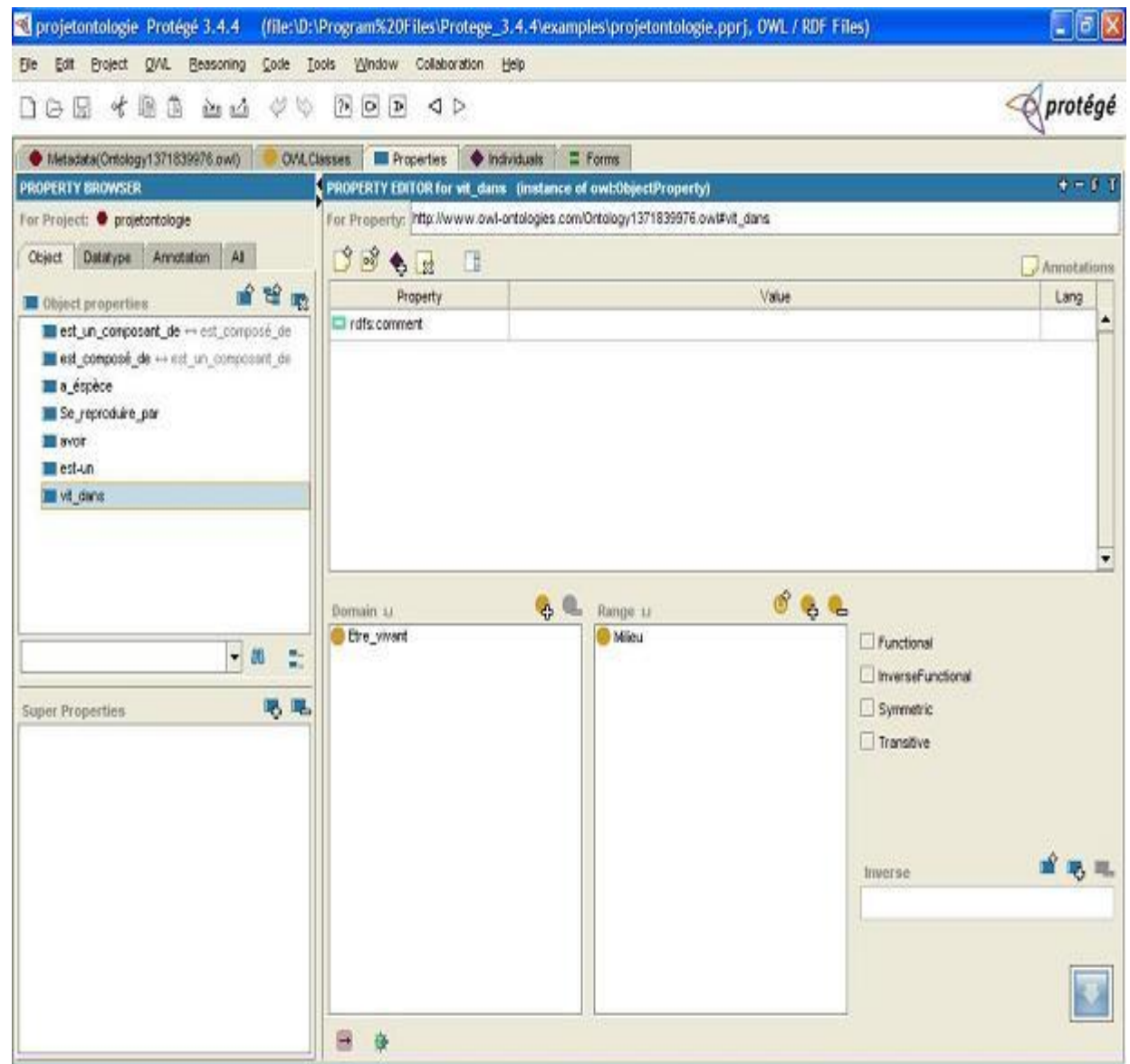


Figure 25 - Ajout les propriétés

Dans l'étape suivant et après l'édition des propriétés, nous spécifions les contraintes de cardinalité

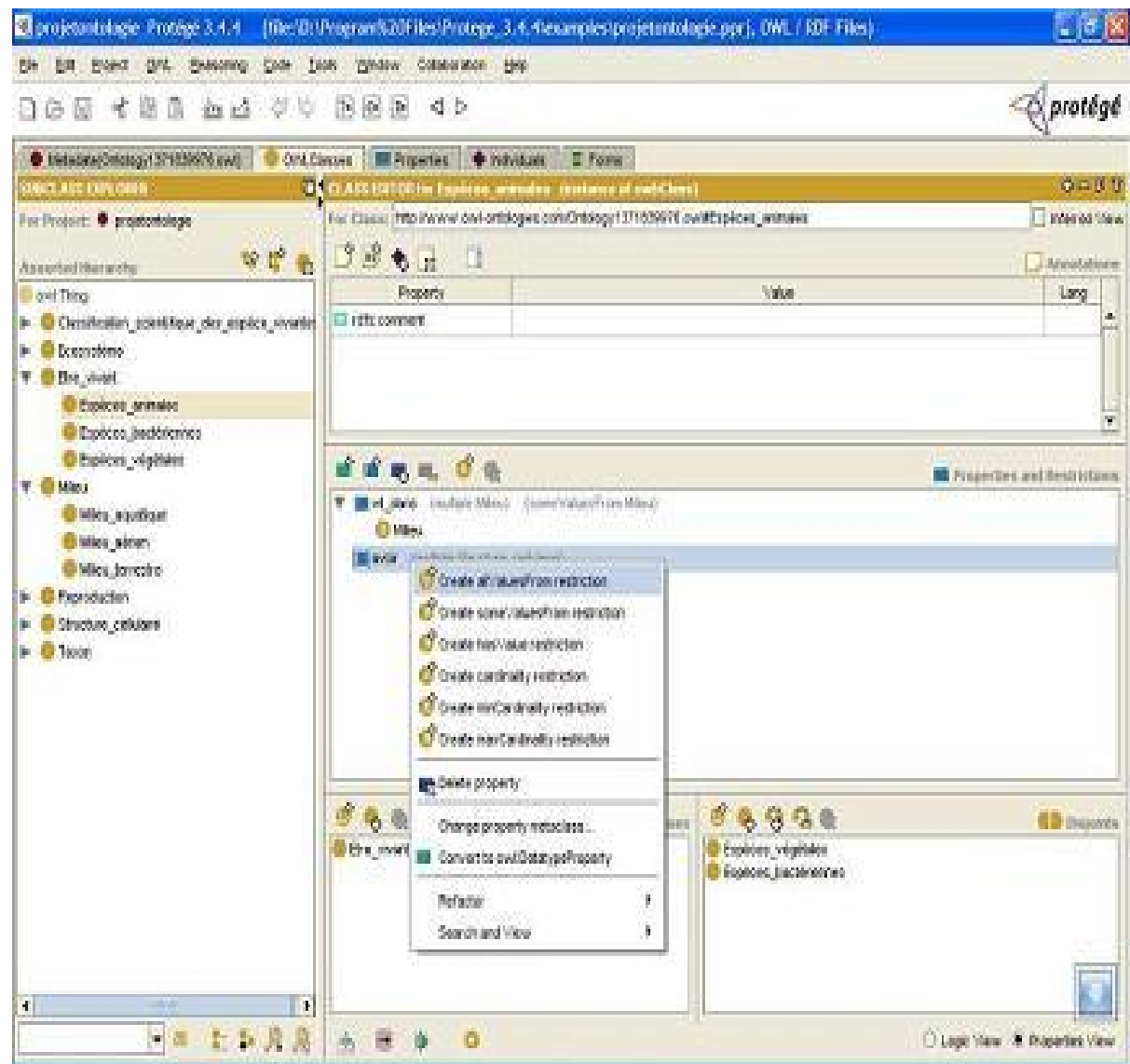


Figure 26 - Les contraintes de cardinalité

5.4 L'exportation et l'importation de cette ontologie

5.4.1 L'exportation et l'importation avec UMLBackend (Protégé3.4.4)

Parmi les modules d'extension (plugin) de l'éditeur d'ontologies Protégé3.4.4: on a UMLBackend [49]. Ce module permet l'importation et l'exportation de modèles UML 1.4 au format XMI 1.1 ou 1.2 mais possède de nombreuses limitations qui sont,

entre autres: la transformation n'est pas configurable, plusieurs éléments importants d'UML comme les stéréotypes et les valeurs étiquetées ne sont pas prises en compte.

Pour importer des fichiers XMI à partir de versions antérieures XMI ou UML dans Protégé, on peut utiliser Poseidon 1.6.

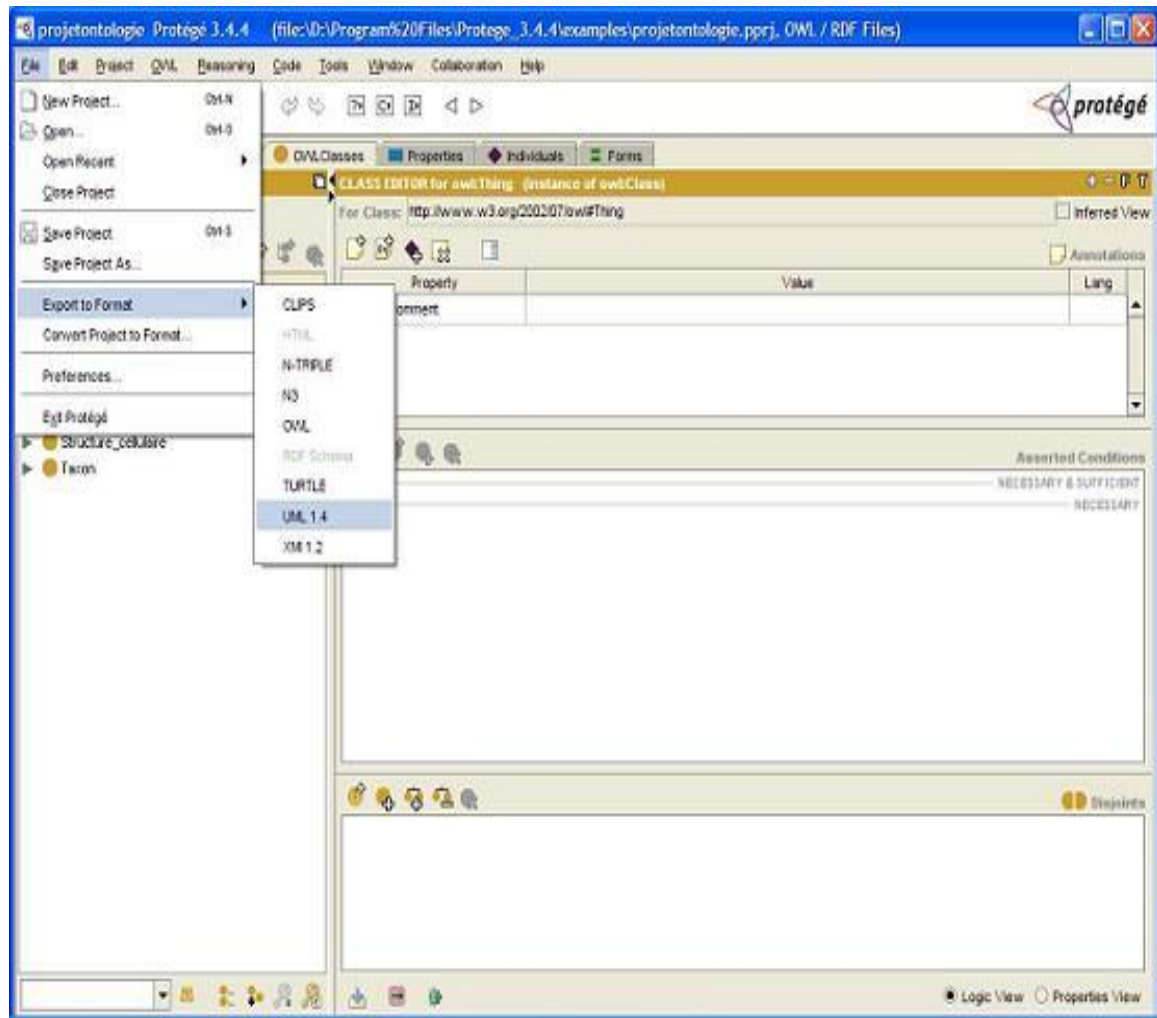


Figure 27 - Exportation de l'ontologie OWL1.1 à UML1.4

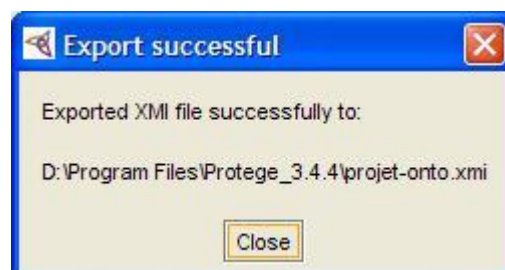


Figure 28 - Exporter l'ontologie au format XMI

Pour importer un fichier XMI représentant le modèle UML1.4, d'abord, nous créons un nouveau projet à partir d'un fichier existant et puis nous sélectionnons le modèle UML.

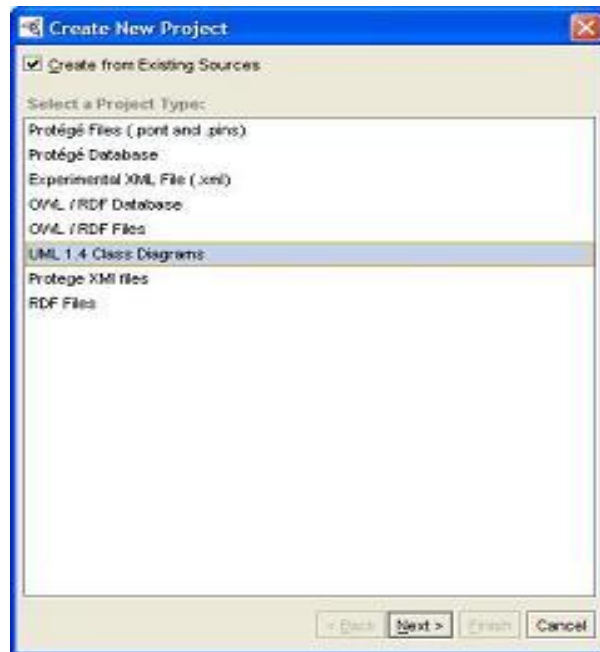


Figure 29 - Les modèles des fichiers pouvant être importés sous Protégé 3.4.4

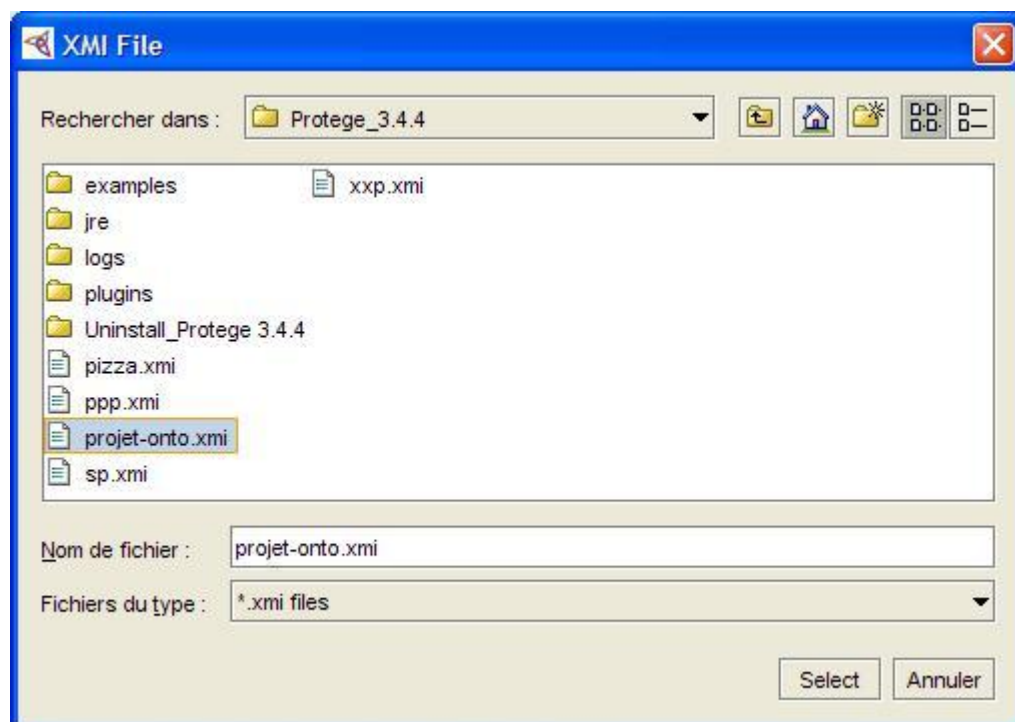


Figure 30 -Importation d'un fichier XMI

Nous constatons bien sur la figure suivante, la perte d'informations après avoir exporter l'ontologie vers le modèle UML1.4

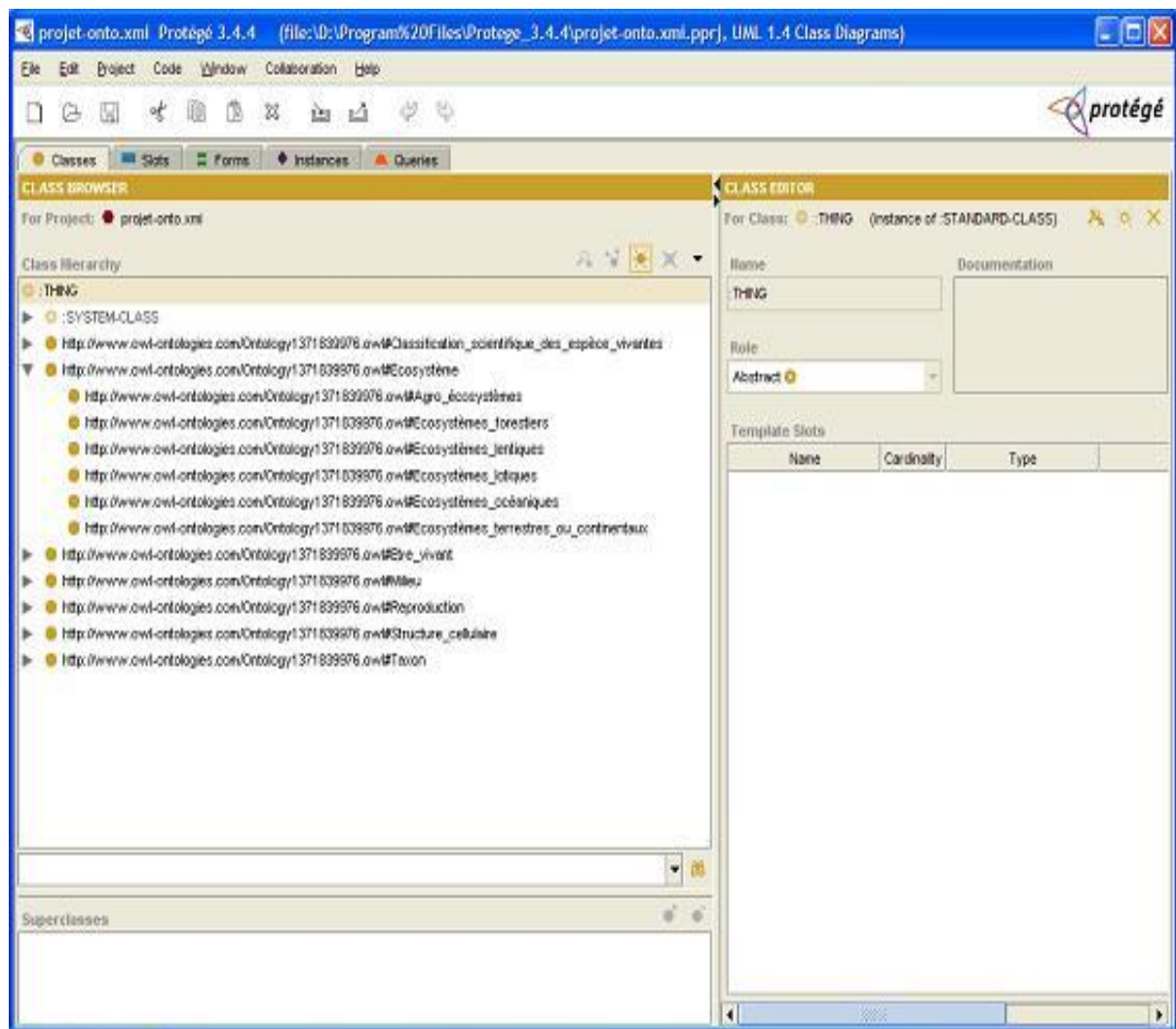


Figure 31 - Perte d'information de l'ontologie après exportation

5.4.2 Importation avec OWL 2.0 (sous protégé 4.2)

Dans OWL 2, les importations sont traitées différemment de ce qu'ils étaient en OWL 1.0. Où il est possible de prendre en charge les versions des ontologies en utilisant l'importation par son emplacement plutôt que l'importation par son nom.

Pour ce faire, OWL 2.0 permet à une ontologie d'avoir deux IRI en son nom. Le premier est l'IRI de l'ontologie. Le second nom est l'IRI de la version de l'ontologie. Dans de nombreux cas, la version IRI sera nulle. Mais lorsque la version

IRI ne l'est pas, cela signifie que l'ontologie est une version spécifique de l'ontologie mère.

5.4.3 L'outil OWLGrtd

Est un outil qui génère automatiquement le diagramme UML qui représente une ontologie OWL active en utilisant l'ontologie du métamodèle de définition de l'OMG.

OWLGrEd [50] peut interagir avec Protégé (4.2 ou ultérieure). Et pour activer cette fonction, on doit télécharger le plugin lv.lumii.tda.jar Protégé (version 2.0) et le placer dans le dossier plugins Protégé (par exemple, "C: \ Program Files \ Protege_4.2 \ plugins \").

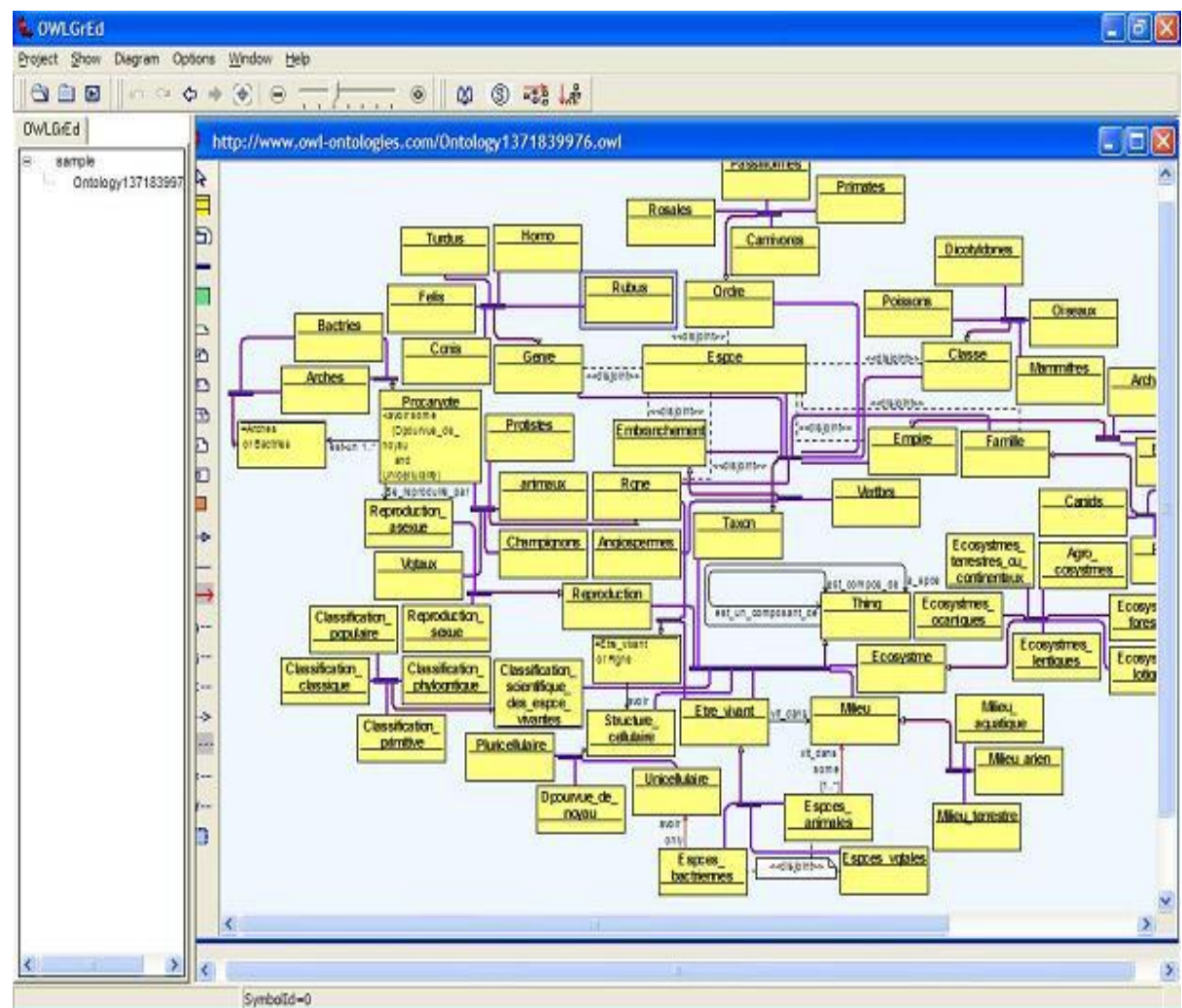


Figure 32 - Création d'un diagramme UML de l'ontologie

5.5 Le raisonnement

Pour exploiter convenablement et assurer la qualité d'une ontologie, il est important de pouvoir raisonner dessus et ce aux différentes étapes de son développement. En effet, étant une représentation formelle des connaissances du domaine, les solutions que propose une ontologie sont en relation étroite avec les données collectées ainsi qu'avec les contraintes posées. Pour assurer la validité des valeurs en entrée, il est impératif de procéder par un contrôle continu de la cohérence de la base de connaissance et signaler les anomalies détectées et cela tout au long du processus de modélisation. Pour ce faire, nous avons utilisé, dans notre cas, le moteur Pellet puisque ce dernier est libre et intégré dans Protégé.

5.6 Conclusion

Dans ce chapitre, nous avons présenté les détails d'implémentation de l'ontologie du domaine écologie évolutive présentée dans le chapitre précédent. Nous avons essayé d'exporter cette ontologie vers UML et en utilisant OWLGrtd et UMLBackend. Nous pouvons conclure que la transformation des ontologies entre les modèle OWL et UML1.4 n'est pas suffisante car il y a certains obstacles qui empêchent cette transformation, entre autres, la compatibilité des versions des outils. Cela a été résolu avec la version UML2.0 et OWL2.0.

Conclusion générale

Conclusion générale

Les données sur le Web constituent une grande source d'informations représentées sous des formes utiles pour l'être humain mais difficiles pour des traitements automatiques. Le contenu des ressources du Web peut être décrit en utilisant les méta-données formelles. Ces dernières doivent s'appuyer sur des ontologies afin de pouvoir les exploiter par des machines et les munir de sémantique fournissant ainsi, une extension plausible du Web Actuel vers le Web Sémantique.

L'objectif principal d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné. Elles jouent ainsi un rôle important dans le Web sémantique et sont employées comme une forme de représentation formelle de la connaissance sur un sujet du monde réel et traduire, de ce fait, un consensus explicite. Dans le Web sémantique, les ontologies sont utilisées pour, d'une part, modéliser des ressources du Web à partir des représentations conceptuelles des domaines concernés et, d'autre part, dans l'objectif de permettre un traitement automatiquement dessus.

Le partage de connaissances entre systèmes informatiques permettra, de plus, une interaction et une coopération entre ces derniers et l'utilisateur humain. Cela se manifeste, par exemple dans : les systèmes d'aide à la décision, les systèmes d'enseignement assisté par ordinateur, la recherche d'information sur le web, etc. Pour permettre un traitement automatique efficace, les représentations à utiliser par les machines (les modèles) doivent être chargées de sens et cela en reliant les informations ainsi collectées et représentées à d'autres types d'informations vouées essentiellement à la sémantique sous-jacente.

Les ontologies sont un paradigme puissant pour la représentation et l'échange des connaissances dans le domaine du Web Sémantique qui ne s'intéressent qu'au traitement automatique de la sémantique. Néanmoins, elles doivent être exprimées dans des

formalismes qui devraient en tirer pleinement profit, tout en offrant des outils de modélisations simples et efficaces.

UML trouve beaucoup de succès au sein des concepteurs de modèles et a bénéficié, par conséquent, d'un cumul d'expérience très important ainsi qu'une large gamme d'outils dédiés. Cependant, ces outils (UML profile, OCL, AS) ne permettent pas la définition d'une sémantique formelle ce qui entrave l'inférence automatique sur des modèles UML.

Il serait profitable de tirer profit de ces deux puissants outils pour rajouter de la sémantique aux modèles conceptuels. Chacun de ces deux outils ayant ses propres mécanismes, il est donc nécessaire de trouver un point de rattachement qui est possible avec l'utilisation de ODM, où il est possible d'utiliser un même éditeur pour faire des modèles UML et des ontologies. ODM nous permettra de passer de UML au OWL (et inversement) et donc de pouvoir utiliser les mécanismes de traitement automatique.

Suggestions

- ✓ Augmenter la définition des profils UML pour une meilleure interconnexion avec les outils qui prennent en charge la définition d'une sémantique formelle.
- ✓ Proposer des outils pour la validation de la cohérence des modèles UML transformés depuis des ontologies.
- ✓ Proposer un format standard et simple pour Action Semantics puisqu'il existe seulement sous forme de méta-modèle.
- ✓ Prise en charge des versions antérieures des outils de création et de modélisation des ontologies lors d'une éventuelle mise à niveau.

Références

Références

- [01] BERNERS-LEE Tim, HENDLER James & LASILLA Ora (2001), The Semantic Web, Scientific American.
- [02] Delia Codruta ROGOZAN (2008), Gestion de l'évolution d'une ontologie : méthodes et outils pour un référencement sémantique évolutif fondé sur une analyse des changements entre versions de l'ontologie proposition de recherche doctorale en informatique cognitive (DIC 9410) Télé-Université de Québec.
- [03] Gruber, T. R. (1993), A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220.
- [04] InterOp, I. (2006), State of the art and state of the practice including initial possible research orientations, d8.1. Rapport technique, InterOp, Interoperability Research for Networked Enterprises Applications and Software.
- [05] Welty, C. et Nancy, I. (1999), Using the right tools: enhancing retrieval from marked-up documents. J. Computers and the Humanities, 33(10):59-84.
- [06] Guarino, N. (1998a), Formal ontologies and information systems. In FOIS'98, Trento, Italy, IO Press.
- [07] John F. Sowa (2000), Knowledge Representation, Brooks Cole Publishing Co, Pacific Grove, CA.
- [08] R.Rifaih, N.Benharkat (2004), Studying Semantic Shanring Problem within Enterprise Information Systems, Americas Conference on Information Systems 2004, August 6-8, New Yourk City, NY, USA, < http://aisle.isworld.org/article_by_author.asp?Author_ID=5561 > ,
- [09] Erich Gamma, Richard Helm, Ralph Johnson, and John vlissides (1994), Design Patterns Elements of Reusable Object-Oriented Software, Addison Wesley Professional Computing Series , MA, USA, 1994, p.395, ISBN 0-201-63361-2.
- [10] Riichiro Mizoguchi, mitsuru Ikeda, and Katherine Sinita (1997), Roles of Shared Ontology in AI-ED Research: Intelligence, Conceptualization, Standardization, and Reusability, In Proceedings of the 8th World Conference On Artificial Intelligence In Education AIED-97, Kobe, Japan, August, 1997, 99.537-544, <<http://www.ei.sanken.osaka-u.ac.jp/ieee/Them.paper.html>> ,
- [11] Yannick Prié, Ingénierie ontologique, UFR Informatique, Université Claude Bernard Lyon 1, 2010.
-

-
- [12] Frédéric Furst, Contribution à l'ingénierie des ontologies: une méthode et un outil d'opérationnalisation, thèse de doctorat, Ecole Polytechnique de l'Université de Nantes (EPUN), 25 Novembre 2004.
- [13] Le rôle de l'ingénierie ontologique dans le domaine des EIAH. Sticef.org 2004.
- [14] Ouafek Naouel, Les ontologies spatiales : la génération automatique d'ontologies spatiales à partir du modèle conceptuel spatio-temporel MADS, thèse de Magister, Université Mentouri de Constantine, 28/04/2009.
- [15] Fernandez, M., Gomez-Pérez, A., Juristo, N. (1997) METHONTOLOGY: From Ontological Art Towards Ontological Engineering. Proceedings of the AAAI-97 Spring Symposium Series on Ontological Engineering, Stanford, CA, USA 1997, p 33-40.
- [16] Isaac A. Conception et utilisation d'ontologies pour l'indexation de documents audiovisuels. Thèse de doctorat, l'université paris iv-Sorbonne, page 131, décembre 2005.
- [17] Gruninger M., Fox. M, The logic of enterprise modeling. In Brown, J. & O'Sullivan, D., (Eds.), Reengineering the Enterprise, Chapman and Hall, pages 83–98, 1995.
- [18] Bachimont, B. (2000), Engagement sémantique et engagement ontologique : conception et réalisation d'ontologie en ingénierie des connaissances , In J. Charlet, M. Zacklad, G.Kassel et D. Bourigoult (éd.), Ingénierie des connaissances. Evolutions récentes et nouveaux défis, Eyrolles, 2000, pp. 305-323.
- [19] Baneyx A., Construire une ontologie de pneumologie. Thèse de doctorat, Université Paris6 page61 ; 85. Février 2007.
- [20] Nathalie H., Josiane M. TtoO: une méthodologie de construction d'ontologie de domaine à partir d'un thésaurus et d'un corpus de référence, IRIT, Toulouse- France 2006.
- [21] Malaisé V, Méthodologie linguistique et terminologique pour la structuration d'ontologies différentielles à partir de corpus textuels , PhD thesis, Université Paris 7 - Denis Diderot, 2005.
- [22] Xu Z., Cao X., Dong Y., and Su W., «Formal Approach and Automated Tool for Translating ER Schemata into OWL Ontologies", Advances in Knowledge Discovery and Data Mining, Springer-Verlag Berlin Heidelberg, 2004, pp. 464-475.
- [23] Kupfer A., Eckstein S., Neumann K., and Mathiak B., « Changes Handling of Database Schemas and Corresponding Ontologies» , Springer, 2006, pp.227-236.
- [24] Of Laborda C., Conrad S., «Relational.OWL - A Data and Schema Representation Format Based on OWL, » Conceptual Modelling2005 (APCCM05), Australian Computer Society, 2005, pp. 89-96.
- [25] Xu Z., Zhang S., Dong Y., «Mapping between Relational Database Schema and OWL
-

-
- Ontology for Deep Annotation », Proceedings of the International Conference 2006IEEE/WIC/ACM, 2006.
- [26] Henri Poincaré, Un portail sémantique pour la gestion des connaissances en cancérologie, Thèse de Doctorat, Université Nancy 1, 14 décembre 2005.
- [27] Tuan Dung CAO, Exploitation du web sémantique pour la veille Technologique, Thèse de doctorat, UNIVERSITE de Nice-Sophia Antipolis, 29 Novembre 2006.
- [28] <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html> 2010.
- [29] Van Heijst G., Schreiber A., et Wielinga B. (1997): Using explicit ontologies in KBS development. Int. J. of Human-Computer Studies, 46(2/3):183–292, 1997.
- [30] Yue Pan, Guotong Xie, Li Ma, Yang Yang, ZhaoMing Qiu, and Juhnyoung Lee. IBM Research Report : An MDA-based system for Ontology Engineering. Technical Report RC23795, IBM Research Division, Yorktown Heights, NY US, November 2005.
- [31] The Object Management Group. Unified Modeling Language : Superstructure. OMG Specification, February 2004.
- [32] The Object Management Group OMG. Request For Proposal for Ontology Definition Metamodel. OMG Request For Proposal, March 2003.
- [33] The Object Management Group OMG, IBM, and Sandpiper Software. Ontology Definition Metamodel. OMG Specification, June 2006.
- [34] Xavier Blanc. MDA en action. Eyrolles, France, 2005.
- [35] The Object Management Group OMG. UML 2.0 OCL Specification. OMG Specification, October 2003.
- [36] The Object Management Group OMG, IBM, and Sandpiper Software. Ontology Definition Metamodel. OMG Specification, June 2006.
- [37] Dragan Djuric, Dragan Gasevic, and Vladan Devedzic. Ontology Modeling and MDA. Journal of Object Technology, 4(1) :109_128, 2005.
- [38] Ricklefs Miller, Ecologie: traduction de la 4^{ème} édition américaine par Michel et Virgine Baguette, Frank d'Amico et Ggégoty Mahy avec la collaboration de Max Bugnicourt, 2005.
- [39] <http://www.ecosociosystemes.fr/classification.html>, 2010.
- [40] José Kahan, Marja-Riitta Koivunen, Eric Prud'hommeaux, and Ralph R. Swick. Annotea: an open RDF infrastructure for shared Web annotations. Computer Networks, 39(5):589-608, 2002.
- [41] Vincent Quint and Irène Vatton (2005). Towards active web clients. In DocEng '05: Proceedings of the 2005 ACM symposium on Document engineering, pages 168-176,
-

New York, NY, USA, 2005. ACM Press.

[42] Hewlett-Packard Development Company. Jena: A Semantic Web Framework for Java. Framework, 2000.

[43] David Huynh, Stefano Mazzocchi and David R. Karger (2005). Piggy Bank: Experience the Semantic Web Inside Your Web Browser. In International Semantic Web Conference, pages 413-430, 2005.

[44] Cyril Faucher, Frédéric Bertrand and Jean-Yves Lafaye : Génération d'ontologie à partir d'un modèle métier UML annoté, Laboratoire L3i – Université de La Rochelle, 2008.

[45] MOUHAMED DIOUF, Spécification et mise en Œuvre d'un formalisme de règle métier, thèse de doctorat, l'Université de Bordeaux I, 2007.

[46] Vargas-Vera M, Motta E, Domingue J, Lanzoni M, Stutt A. et Ciravegna F. (2002) : MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup, In Proc. of the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002, Springer Verlag LNAI 2473, 379-391.

[47] Popov B, Kiryakov A, Ognyanoff D, Manov D. et A. Kirilov (2004) : KIM – a semantic annotation platform for information extraction and retrieval. Natural Language Engineering, 10, Issues 3-4, 375-392.

[48] <http://protege.stanford.edu/support.php>. 2010

[49] <http://protege.cim3.net/cgi-bin/wiki.pl?UMLBackend>. 2010

[50] http://owlgred.lumii.lv/get_started. 2011