République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed KHIDER - BISKRA
Faculté des Sciences exactes et de Sciences de la nature et de la vie
Département d'Informatique

N° d'ordre : ……...
Série : .....

Mémoire

Présenté en vue de l'obtention du diplôme de
Magister en Informatique

Option: Intelligence Artificielle et Systèmes Distribués

Thème

Une approche basée agent pour le processus génération d'ontologie de domaine

Présenté par : M[r]. BARKAT Abdelbasset

Sous la direction de

M. KAZAR Okba

Soutenu le : / /2011

Devant le jury composé de :

Pr. BOUFAIDA Mahmoud	Président	Professeur	Université de Constantine.
Dr. KAZAR Okba	Rapporteur	M.C.A	Université de Biskra.
Pr. ZAAROUR Nacereddine	Examinateur	Professeur	Université de Constantine.
Dr.BABAHENINI Mohamed Chaouki	Examinateur	M.C.A	Université de Biskra.

Remerciements

Je remercie sincèrement mon directeur de mémoire **Okba KAZAR** Maître de conférences au département d'informatique à l'université de Biskra, et je remercie aussi tous les membres de mon jury de mémoire qui ont pris de leurs temps pour lire et juger ce travail ainsi que pour leur déplacement le jour de la soutenance.

Pour terminer, je remercie ma grande famille pour leur soutien et pour leur aide dans la préparation de ce mémoire et pour leurs encouragements.

Dédicace

Nous savons qu'un jour l'existence se termine pour chacun d'entre nous. Je dédie ce travail à mon frère BARKAT Aissa, qui reste toujours présente dans nos coeurs.

ملخص

الانتولوجي هي مواصفات واضحة لمفهوم معين, هذا المصطلح يكون غالبا مرتبط مع الويب الدلالي الذي هو نسخة حديثة من الويب الحالي, يهدف الويب الدلالي إلى جعل الموارد الخاصة بالويب متاحة إلى مجموعة من البرامج لتسهيل استخدام هذه الموارد. الانتولوجي تستعمل كتمثيل للمعارف لتمييز موارد الويب و أيضا في التواصل بين الأعوان.

دور الانتولوجي مهم جدا ولكن للأسف عملية بناءها مكلفة جدا, ولهذا السبب فكرنا في كمية المعارف الكثيرة المتوفرة على شكل نصوص لاستعمالها لتأليه عملية بناء الانتولوجي , ولأننا نتعامل مع النصوص وجب علينا المرور عبر مرحلة تعتبر مرحلة أساسية, هذه المرحلة تستعمل تقنيات المعالجة الآلية للنصوص لاستخراج معلومات نحوية و تحويلها إلى معلومات ذات دلالة.

لكي نصل إلى هذا الهدف نحن نقترح طريقة تعتمد على الأعوان لبناء الانتولوجي من النصوص, وكذلك طورنا نظام متعدد الأعوان الذي يتبع هذه الطريقة والذي يعطينا انتولوجي انطلاقا من مجموعة من النصوص.

الكلمات المفاتيح : البناء الآلي الانتولوجي. انتولوجي, المفهوم. تحليل المفاهيم, تحليل العلاقات.

Résumé

Une ontologie est une spécification explicite d'une conceptualisation, ce terme est souvent lié avec celui de Web Sémantique qui est une nouvelle version de web actuel. Il vise à rendre les ressources du web accessibles par des agents logiciels, afin de faciliter leurs utilisations. Les ontologies sont utilisées comme des représentations de connaissances formalisées pour annoter ces ressources web et elles sont aussi utilisées dans la communication entre les systèmes informatique.

Le rôle des ontologies est très important, mais malheureusement leur construction est coûteuse, pour cela nous avons pensé à l'énorme quantité d'informations disponible sous forme textuelle pour l'utiliser pour automatiser le processus de construction d'ontologie. Puisque on traite des textes on est obligé de passer par une étape considérée comme une étape de base pour le domaine de construction d'ontologie à partir du texte, cette étape consiste a faire appel au domaine du traitement automatique de langages naturelles pour extraire des informations syntaxique de texte, puis les transformer en informations sémantiques

Afin d'accomplir cet objectif on a proposé une approche basée agent pour construire une ontologie à partir du texte, et on a implémenté un système multi agents guidé par cette approche qui à partir d'un ensemble de documents textuels nous donne une ontologie formelle sous forme d'OWL.

Mots-clés : La construction automatique d'ontologies, ontologie, Concept formel, Formal Concept Analysis , Relationnal Concept Analysis.

Abstract

An ontology is an explicit specification of a conceptualization, the term is often linked with the Semantic Web which is a new version of the current web aims to make web resources accessible by software agents to facilitate their use, ontologies are used as representations of formalized knowledge to annotate web resources and are also used in communication between the computer system.

The role of ontologies is very important, but unfortunately their construction is expensive, this is why we think of the enormous amount of information available in text format to use it for automate the process of building ontology, and since we deal with texts, we should think about the NLP(), which considered as the basic on ontology construction from text.

To accomplish this goal we proposed an agent-based approach to build ontology from text, and we implemented a multi-agent system guided by this approach, which start from a set of textual documents gives us a formal ontology in form of OWL.

We chose here to use the two formals techniques Formal Concept Analysis FCA and Relational Concept Analysis RCA to move from the syntactic level to the semantic level.

Keywords: Automatic ontology construction, ontology, concept formel, Formal Concept Analysis , Relationnal Concept Analysis.

Sommaire

Chapitre I: Etat de l'art sur les ontologies

Chapitre II: Les processus de développement d'ontologie

Chapitre III : Une approche basée agent pour le processus génération d'ontologie de domaine

Chapitre IV : Implémentation et Evaluation

Liste des figures :

Introduction Générale

Toutes les personnes qui ont essayé de chercher des informations sur le web, en utilisant les moteurs de recherche classiques, ont tombé dans la situation tel que les résultat fournit par ces moteurs ne sont pas qu'est ce que nous volons chercher au début, et on a l'esprit que ce moteur est totalement hors champ. Mais au contraire à qu'est ce que on pense ce moteur fonctionne correctement mais selon les principes de web actuel qui base sur les approches syntaxiques. Cela a poussé les chercheurs de penser a une autre version de web s'appelle le web sémantique.

Malgré que le web sémantique soit une solution à l'échec de web actuel mais ce dernier est entièrement fondé sur le web et ne le remet pas en cause. Et parmi les autres piles que le web sémantique base sur eux, les ontologies sont considérés comme un moyen pour annoter les ressources web pour les rendes accessible par des agents logiciels.

Selon Thomas R Gruber, *Une ontologie est une spécification explicite d'une conceptualisation* [13], et elle est utilisée dans différents domaines (les systèmes multi agents, le web sémantique…), ces domaines ont besoin à des ontologies puissantes et bien définies, c'est pour cela plusieurs méthodes et méthodologies de construction des ontologies sont apparaît dans le domaine de recherche *(Ontological* Engeneering), METHONTOLOGIE et On-To-Knowledge sont des exemples de ces méthodologies.

Un seul obstacle peut limiter l'évolution des ontologies, cet obstacle est que le processus de construction de ces derniers est coûteux, et cela se réfère à plusieurs critères parmi lesquels on cite :

- Puisque l'ontologie est une entité partagée par plusieurs personnes ou une communauté, sa construction est un peu compliquée parce que toutes les différentes parties doivent trouver un point de vue commun entre eux.
- l'ontologie doit couvrir le domaine entier, et offrir un model adéquat par la détermination d'une généralisation significative et logique en même temps, et la difficulté réside dans que l'ontologie doit modéliser une grande quantité d'information et garde la signification de ce modèle.

Pour affronter ce problème, on a essayé d'automatiser le processus de construction d'ontologie assez possible qu'on peut, et cela nous a crée le domaine de recherche

Ontology Learning, en fonction de support qu'a partir de lui on construit l'ontologie, l'ontologie learning peut prendre plusieurs façons :

- *Ontologie learning à partir des textes* : un ensemble de corpus textuel est analysé afin d'extraire les informations relèvent à notre domaine, et les utiliser pour la construction d'ontologie selon plusieurs techniques.
- *Ontologie learning à partir des instances* : particulières instances peuvent être utiliser pour construire des ontologies.
- *Ontologie learning à partir de schémas :* bases de données relationnelles, les modèles entité/relation et les schémas XML peuvent être utiliser pour générer une ontologie par un processus de re-ingénierie.
- *Ontologie learning for interoperability*. Ici on travaille avec deux ontologies et on essaie de trouver des liens sémantiques entre les éléments de ces ontologies. Les liens sémantiques sont utilisés pour fusionner les deux ontologies.

Dans le cadre d'Ontologie learning à partir des textes, la méthode d'*Aussenac-Gilles* et ses collègues et la méthode *Maedche* et ses collègues, sont les deux méthodes les plus célèbres dans ce domaine. On se basant sur cette base théorique plusieurs tentatives pour créer des outils sont apparaît pour construire d'ontologie à partir du texte, parmi lesquels on cite : *Text20nto* et *OntoGen* [23].

Alors dans ce but on va élaborer une approche basée agent pour construire une ontologie à partir du texte et implémenter un système multi agents pour tester cette approche.

Structure de mémoire :

Le présent mémoire est scindé en trois chapitres :

Chapitre 1 : ce chapitre est comme une introduction, il couvre l'état d'art sur les ontologies.

Chapitre 2 : dans ce chapitre on présente les processus de développement d'ontologie pour comprendre comment on construit une ontologie manuellement, puis on va présenter les processus de construction d'ontologie à partir du texte, afin de rendre le processus de construction automatique. A la fin de ce chapitre on va voir les anciens travaux à base d'agent pour le développement d'ontologie, pour les étudier et voir les résultats de ces travaux.

Chapitre 3 : Le troisième chapitre est consacré à proposer une approche pour construire une ontologie à partir de texte et le nouveau système à base d'agent qui modélise cette approche.

Chapitre 4 : Dans le dernier chapitre on va dresse un bilan des possibilités et limites de notre système.

Chapitre I

Etat de l'art sur les ontologies

I.1. Introduction

L'apparition de l'Internet comme une globale infrastructure de distribution de l'information, ainsi que le faible coût de stockage et la communication par unité d'information a encouragé la production de l'information. La quantité de l'information est devenu plus en plus grande, mais cette grandeur n'est pas toujours un avantage, elle peut être un inconvénient, parce que l'information est devenu inutile ou bien difficile à utiliser, en plus de cela les techniques de recherche sur le web sont des techniques syntaxiques (recherche plein texte ou recherche de texte libre) ou le moteur de recherche examine tous les mots dans chaque document enregistré lorsqu'il essaye de faire correspondre avec les mots clés fournis par l'utilisateur, cette technique ne donnent pas des résultats satisfaisants a cause de l'ambiguïté inhérente au langage naturel, par exemple, le mot *football* peut se référer soit au soccer, au football américain, ou au football canadien, au football gaélique ou aux règles du football australien, etc., alors que la personne qui cherche est probablement intéressée par l'un de ces sports. Un autre exemple, le mot *avocat* peut se référer soit au fruit, soit à la profession.

Il est claire que si nous avons un moteur de recherche qui connaît que:
Avocat: réfère au concept à une personne dont la profession est de plaider en justice.
Le résultat de la recherche sera parfait.

Dans le domaine d'intelligence artificielle nous savons que la capture de connaissance est la clé pour construire des grands et puissants IA systèmes, mais la représentation de connaissance est aussi difficile et prend assez de temps. Malgré que nous avons développé des outils pour nous aidez à l'acquisition de connaissance, la construction de base de connaissance reste parmi les épines dans le développement des IA systèmes.

Presque pour chaque système nous devons construire une nouvelle base de connaissance, comme conséquence la plus part des systèmes reste petits ou moyens. Le coût de cet effort dupliqué est devenu grand et se sera un obstacle quand nous voulons développer de grands systèmes.

Pour traverser cette barrière nous devons trouver une manière pour préserver les bases de connaissance existantes et les partager, réutiliser et construire sur eux.

Cette idée a été analyser et discuter par *Neches* et ses collègues, et ils sont arrivé à une conclusion que : "*Au lieu du processus de construction des systèmes à base de connaissance on va commencer à assembler des composants réutilisables."*[10]

Ces composants réutilisables qu'ils les appellent ultérieurement des ontologies.

I.2. C'est quoi une Ontologie

Le terme ontologie est emprunté de la philosophie, qui signifier « *l'étude de l'être en tant qu'être, c'est-à-dire l'étude des propriétés générales de ce qui existe* », dans l'informatique et spécialement dans l'intelligence artificielle ce qui est existe, est ce qui peut être représenter pour manipuler par des machines.

Parmi les premières Définitions qui sont données à l'ontologie on trouve, la définition de Neches :

> *Une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles pour combiner les termes et les relations pour définir des extensions pour le vocabulaire.* [10]

Deux ans plus tard Thomas R Gruber a donné une définition qui reste la plus utilisée dans la littérature

> *Une ontologie est une spécification explicite d'une conceptualisation.* [13]

Borst a augmenté la définition de Gruber par l'ajout du terme "formelle" :

> *Les ontologies sont définies comme une spécification formelle d'une conceptualisation* [03].

Les deux dernières définitions (Gruber et Borst) ont été fusionnées et expliquées par Studer comme suite:

> *Une ontologie est une spécification formelle et explicite d'une conceptualisation commune.*
>
> *Conceptualisation se réfère à un modèle abstrait de certains phénomènes dans le monde en avoir identifié les concepts pertinents de ce phénomène. Explicite signifie que le type de concepts utilisés, et les contraintes sur leur utilisation sont explicitement définis.*
>
> *Formelle se réfère au fait que l'ontologie doit être lisible par la machine. Commune reflète l'idée que l'ontologie capture une consensuelle connaissance, qui n'est pas privée de certains individus, mais accepté par un groupe.* [23]

I.3. Les principaux composants d'une ontologie

D'après ces définitions il est claire que depuis le début d'apparition des ontologies, ces dernières sont considérées composées d'un ensemble de concepts relient entre eux par des relations. Qu'on peut les voir comme des graphes, ces sommets sont les concepts et ces arcs sont les relations. Et parce que les termes utilisés pour désigner les concepts et les relations sont déférents (on parle ici du points de vues d'une personne à l'autre) d'un domaine à l'autre, l'ontologie ne peut être construire que dans un domaine précis.

I.3.1. Concepts

Un concept peut représenter un objet, une notion, une idée, il est désigné par un terme (ou plusieurs) lexical, expliqué par une notion et instancier par un ensemble d'objets [09]:

- **Le terme :** est un élément lexical qui permet d'exprimer le concept en langue naturelle, il peut admettre des synonymes. [04]
- **La notion (intension du concept) :** contient la sémantique du concept, exprimée en termes de propriétés et attributs, et de contraintes. [04]
- **Les objets (extension du concept) :** regroupe les objets manipulés à travers le concept ; ces objets sont appelés instances du concept. [04]

Les trois points derniers composent le triangle de signification (Meaning triangle) proposé par *Sowa* [11].

Il reste a noter que le concept a un ensemble de propriétés et sont devisées en : des propriétés portant sur un seul concept et des propriétés portant sur deux concepts.

Les propriétés portant sur un concept sont :

- **la généricité :** Un concept peut avoir une extension vide (par exemple le concept *Intention*).
- **l'identité :** (propriété proposée par N. GUARINO) un concept porte une propriété d'identité si cette propriété permet d'identifier un instance du concept d'une manière unique, par exemple le concept *livre* une propriété d'identité avec son code isbn.
- **la rigidité :** (propriété proposée par N. GUARINO) un concept est rigide si toutes les instances de ce concept sont les instances possibles, par exemple *Animal* est un concept rigide, *Mammifère* est un concept non rigide ;
- **l'anti-rigidité :** (propriété proposée par N. GUARINO) un concept est anti-rigide si toutes les instances possibles de ce concept sont des instances d'un autre concept. *e.g: Mammifère* est un concept anti-rigide car toute mammifère est un *Animal* ;
- **l'unité :** (propriété proposée par N. GUARINO) un concept est un concept unité si, pour chacune de ses instances, les différentes parties de l'instance sont liées par une relation qui ne lie pas d'autres instances de concepts. *e.g:* les deux parties d'un couteau, manche et lame sont liées par une relation « emmanché » qui ne lie que cette lame et ce manche.

Les propriétés portant sur deux concepts sont :

- **l'équivalence :** deux concepts sont équivalents s'ils ont même extension.
- **la disjonction :** (Incompatibilité) deux concepts sont disjoints si leurs extensions sont disjointes. *e.g: homme* et *femme*.
- **la dépendance :** (propriété proposée par GUARINO) un concept C1 est dépendant d'un concept C2 si pour toute instance de C1 il existe une instance de C2 qui ne soit ni partie ni constituant de l'instance de C2. *e.g: parent* est un concept dépendant de *enfant* (et vice-versa).

I.3.2. Relations

Une relation est un lien sémantique entre des instances des concepts, la relation est désignée aussi comme le concept par un terme, et pour que la définition de la relation soit complète on doit déterminer le nombre des instances que la relation lie (signature de la relation), leur type et l'ordre des concepts, c'est–à–dire le sens pour la lire. La relation a aussi un ensemble de propriétés qui sont devisées en trois groupes :

Les propriétés intrinsèques à une relation sont [04]

- **les propriétés algébriques :** symétrie, réflexivité, transitivité ;
- **la cardinalité :** nombre de concepts ou instances possibles participe aux relations, par exemple une mère a au moins un enfant, un gouvernement a au plus un chef.

Les propriétés liant deux relations sont [04]

- **l'incompatibilité :** deux relations sont incompatibles si elles ne peuvent lier les mêmes instances de concepts. *e.g:* les relations « être rouge » et « être vert » sont incompatibles ;
- **l'inverse :** nous avons dit qu'une relation a un sens de lecture, alors deux relations binaires sont à l'inverse, si la deuxième se lit a l'inverse de la première, par exemple les deux relations *est conduit par* et *conduit* sont inverses parce que : le véhicule *est conduit par* un chauffeur, mais le chauffeur *conduit* un véhicule.
- **l'exclusivité :** deux relations sont exclusives, si elles lient les même deux concepts, mais si une lie des instances de concepts, l'autre ne doit pas lier ces instances. Eg les deux relations mathématiques d'appartenance et la non appartenance sont exclusives.

Les propriétés liant une relation et des concepts sont [18]**:**

- **le lien relationnel :** il existe un lien relationnel entre une relation R et deux concepts C1 et C2 si, pour tout couple d'instances des concepts C1 et C2, il existe une relation de type R qui lie les deux instances de C1 et C2. Un lien relationnel peut en outre être contraint par une propriété de cardinalité, ou porter directement sur une instance de concept. *e.g:* il existe un lien relationnel entre les concepts « texte » et « auteur » d'une part et la relation « a pour auteur » d'autre part ;

- **la restriction de relation :** Si une relation R lie un concept C1 avec des autres concepts, ces derniers doivent être de type spécifique selon le type de C1, par exemple la relation *est conduit par* lie les deux concepts : *moyen de transport* et *chauffeur*, si le moyen de transport est un avion, l'autre concept doit être forcément un pilote.

Après avoir aborder les deux principaux constituants d'une ontologie (concepts et relations), on peut donner une définition rigoureusement formelle d'une ontologie, et on commence par un exemple d'une ontologie qui représente le domaine d'organisation d'un séminaire :

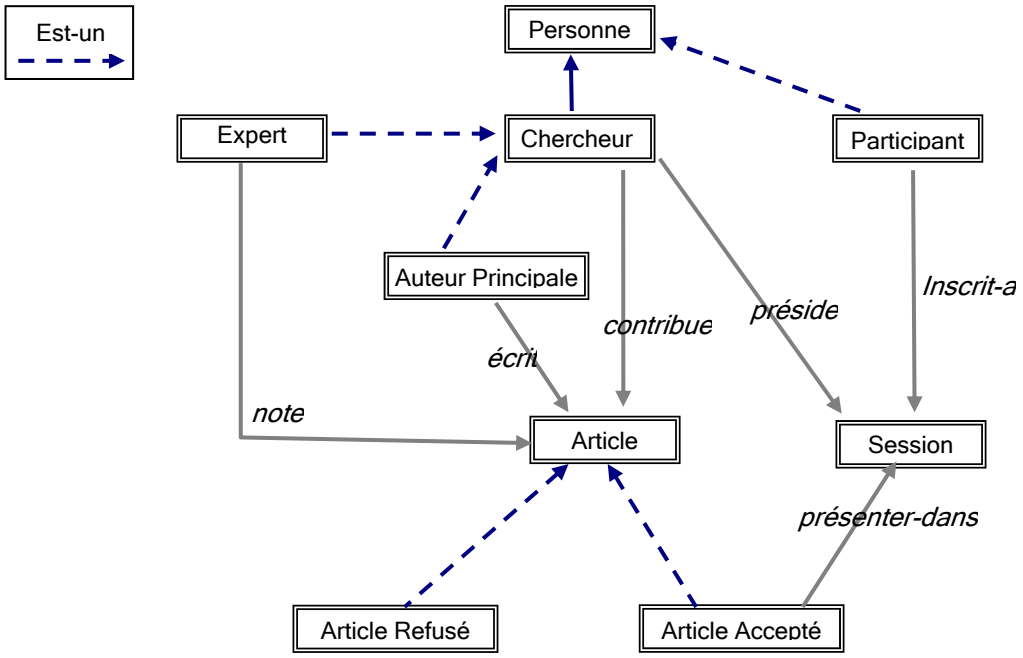


Figure I.1 : Exemple d'ontologie.

I.4. Ontologie Formelle

I.4.1. Définition : une ontologie est une structure [05] :

$$O := (C, \leq_C, R, \sigma_R, \leq_R, A, \sigma_A, T)$$

Composé de :

– Quatre ensembles disjoints C, R, A et T qui sont respectivement l'ensemble des concepts, l'ensemble des relations, l'ensemble des attributs et l'ensemble de type de donnés.

- $\leq_C$ sur C
- Une fonction $\sigma_R : R \to C^+$, appelée la signature de relation, qui donne l'ensemble des concepts (instances) participe à la relation.
- Un ordre partiel $\leq_R$ sur R, appelé hiérarchie de relations, avec r_1 et r_2 deux relations et si $r_1 \leq r_2$ et si et seulement si

$$|\sigma_R(r_1)| = |\sigma_R(r_2)| \quad et \quad \pi_i(\sigma_R(r_1)) \leq_C \pi_i(\sigma_R(r_2)) pour \quad chaque \quad 1 \leq i \leq \sigma_R(r_1)$$

avec $\pi_i(t)$ représente l'élément numéro i de t.

- Une fonction $\sigma_A \to C \times T$, appelée signature des attributs.
- Un ensemble T de type de donnés comme integer, string…ect.

I.4.2. Domaine et Range d'une relation

Le domaine d'une relation binaire est le concept (instance) de départ et le range est le concept (instance) d'arrivé, si r est une relation $dom(r) := \pi_1(\sigma(r))$ et $range(r) := \pi_2(\sigma(r))$

Si $C_1 \leq_C C_2$ alors C_1 est un *sous concept* de C_2, et C_2 est un *super concept* de C_1, et si n'existe aucun C_3 tel que $C_1 \leq_C C_3 \leq_C C_2$ alors on dit que alors C_1 est un *directe sous concept* de C_2, et C_2 est un *directe super concept* de C_1 et on note la relation de directe sous/super concept par <.

Exemple: on considère l'ontologie O avec :

C = {Personne, Chercheur, Expert, Auteur principale, Participant, Article, Article accepté, Article refusé, Session} est l'ensemble des concepts.

R = {est un, note, préside, présenté dans, inscrit à, contribue} [05].

Selon la relation directe super concept nous avons : Chercheur < Personne, Participant < Personne, Expert < Chercheur, Auteur principale < Chercheur, Article accepté < Article, Article refusé < Article.

Pour les relations et les attributs nous avons :

- σ (note) = (Expert, Article)
- σ (écrit) = (Auteur principal, Article)
- σ (préside) = (Chercheur, Session)
- σ (présenté dans) = (Article, Session)
- σ (inscrit a) = (Participant, Session)
- σ (contribue) = (Chercheur, Article)

I.5. Types des Ontologies

Il n'existe aucune classification standard des ontologies, mais il y a plusieurs classifications selon le point de vue de la personne ou bien le groupe qui a fait :

I.5.1. Classification 1 (Mizoguchi et ses collègues 1995)

Ils ont proposés quatre types d'ontologie [24] :

- Ontologies de contenu : pour la réutilisation de connaissance, peut décomposer en : ontologie de tache, de domaine et ontologie générale ou commun.
- Ontologies de commination : des ontologies pour le partage de connaissance (de type Tell & Ask).
- Ontologies d'indexation : pour la recherche et la récupération des informations.
- Meta ontologies : sont les ontologies de représentation de connaissance.

I.5.2. Classification 2 (Uschold et Michael Gruninger 1996)

Ils ont proposés une classification selon le degré de formalisation de l'ontologie, et ils les décomposent en quatre catégories [08] :

- Fortement informelles : ils sont exprimées en langage naturel.
- Semi informelle : ils sont exprimées en langage naturel selon une forme en respectant un ensemble de règles.
- Semi formelle : ils sont exprimées dans un langage formel et artificiel.
- Rigoureusement formelles : aussi exprimées dans un langage artificiel doter d'un sémantique formel.

I.5.3. Classification 3 (Heijst et ces collègues 1997)

Ils ont classifiés les ontologies selon deux dimensions :

1) le niveau et le type de structuration d'une conceptualisation : regroupe trois sous groupes :

Ontologies terminologiques : comme un dictionnaire.

Ontologies d'information : comme une base de donnés.

Ontologies de modélisation de connaissance : qui représente une spécification d'une conceptualisation.

2) Le sujet de conceptualisation : qui regroupe quatre catégories : des ontologies de représentation, générique, de domaine et d'application.

1.5.4. Classification 4 (Guarino 1998)

Il classifie l'ontologie selon son dépendance et il distingue entre : ontologie d'haut niveau, de domaine, d'application et de taches.

I.5.5. Classification 5 (Lassila et McGuinness 2001)

Ils ont classifié les ontologies selon les informations qui doivent être exprimées par eux et la richesse de ces internes structures, et ils ont fait une distinction entre : vocabulaires contrôlés, glossaires, les thesaurus, des hiérarchies informels *est-un*, des hiérarchies formelles *est-un*, instances formels, frames, restriction de valeur et générales contraintes logiques [25].

I.5.6. Classification 6 (Gḿez-Pérez et ces collègues 2004)

C'est une classification basée sur les travaux présentés au dessus, ils ont distingués entre deux classifications, selon la richesse de la structure interne de l'ontologie et le sujet de la conceptualisation [02].

Type des ontologies en basant sur la richesse de la structure interne :

- Vocabulaires contrôlé**s :** une liste fini de termes.
- Glossaire : une liste de termes avec leurs significations.
- Thesaurus : fournit des sémantiques additionnelles entre les termes comme par exemple la relation de synonyme.
- Hiérarchies informelles *est-un* : hiérarchie de termes par la relation *est un* exprimé informellement (cette hiérarchie n'est pas une structure strictement *est un*).
- Hiérarchies formelles *est-un* : est une hiérarchie strictement de type *est un* exprimée formellement.
- Hiérarchies formelles *est-un* avec des instances : c'est la même que la précédente mais avec des instances de domaine.
- Frames : l'ontologie contient des classes et ces propriétés, les classes peuvent être structurer en utilisant la relation *est un* pour exprimer l'héritage entre les classes.

- Restriction de valeur : ce sont des ontologies qui peuvent mettre des restrictions sur les valeurs possibles pour une propriétés.
- Ontologie qui exprime générales contraintes logiques : ce sont les ontologies les plus riches, parce qu'ils sont augmentés par des contraintes entre les termes en utilisant la logique de premier ordre.

Type des ontologies en basant sur le sujet de la conceptualisation :

- Les ontologies de représentation de connaissances (Van Heijst 1997): capture les primitives essentiels pour formaliser la connaissance sous un paradigme de représentation de connaissance.
- Les ontologies générales : sont les mêmes ontologies générales de *Van Heijst* ou bien les ontologies communes de *Mizoguchi*, ils sont utilisés pour représenter les connaissances communes entre plusieurs domaines afin de les utiliser sur ces domaines, l'exemple le plus connu de ce type est l'ontologie des unités standards définis par *T Grubrt* qui contient les définitions des unités de mesure.
- Les ontologies d'haut niveau : sont des ontologies qui décrivent des concepts très générales, un exemple sur ce type d'ontologie est le Cyc ontologie.
- Les ontologies de domaines (Van Heijst et Mizoguchi) : sont des ontologies spécifique a un domaine précis (Médecine, Architecture, Physique…).
- Les ontologies de tâches (Mizoguchi et Guarino) : décrire le vocabulaire relative à une tâche ou activité générale comme le diagnostique et la planification.
- Les ontologies de taches relative a un domaine : sont des ontologies de tâches utilisables sur un domaine précis, par exemple une ontologie de planification d'un voyage.
- Les ontologies de méthode : donne la définition des concepts et relations appliquées dans un processus pour accomplir une tâche.
- Les ontologies d'application (Van Heijst) : sont des ontologies qui modélisent le connaissances nécessaires pour une application particulière.

I.6. L'utilisation des ontologies

Il existe plusieurs utilisations de l'ontologie, mais *Fikes* a identifié quatre cas principaux d'utilisation: la collaboration, l'interopérabilité, l'éducation, et modélisation [12].

I.6.1 La collaboration

Des différentes personnes travaillent ensemble, peuvent avoir des points de vue différents sur le même problème, et cela, lui-même est un problème parce que la collaboration ne sera pas parfaite. Pour arriver à un point de vue commun entre ces personnes, des ontologies peuvent être utilisé pour unifier les connaissances de domaine.

Les ontologies sont aussi utilisées dans la collaboration entre les agents intelligents, quand un agent envoie un message a un autre agent, ce dernier doit avoir le même modèle sur l'environnement (la même ontologie) que l'autre agent

I.6.2 L'interopérabilité

Les applications peuvent avoir besoin d'accéder à plusieurs sources pour rassembler des informations, ces sources peuvent contenir des informations différentes en format et niveau de détail. L'existence de la même ontologie dans ces sources peut faciliter cette opération.

I.6.3 L'éducation

Parce que les ontologies représentent les connaissance d'un domaine quelconque, ils peuvent être considérés comme une source des informations pour les personnes qui veulent apprendre plus sur le domaine (on parle ici sur les ontologies informelles c'est-à-dire exprimés en langage naturel).

I.6.4 La modélisation

Les ontologies sont considérées comme des entités réutilisables dans la modalisation des applications à base de connaissances.

Un autre cas d'utilisation des ontologies, est dans le domaine de recherche des informations sur le web, cette combinaison a donné naissance à un nouveau domaine *le web sémantique.*

I.6.5. Web sémantique

Selon Tim Berners-Lee (1999), le web sémantique est une prolongation du web actuel, dans lequel l'information est donnée sémantiquement bien formée. Les ontologies se trouvent dans le fond de web sémantique.

I.7. Formalismes de représentation des connaissances

La représentation de connaissance a ces racines dans le domaine de développement des systèmes à base de connaissances, qui sont des programmes capables de raisonner sur un domaine d'application pour résoudre un certain problème, et pour permettre à ces systèmes de raisonner, les connaissances du domaine sont représentées par des entités qui ont une description syntaxique à laquelle est associée une sémantique.
Parmi les formalismes utilisés dans les systèmes a base de connaissances, on trouve trois qui sont plus utilisés dans l'ingénierie des ontologies : Langages de frame, les graphes conceptuels, les logiques de descriptions.

I.7.1. Les langages de frame (Minsky 1970)

Les frames sont utilisés comme un mécanisme de base pour la représentation de connaissances dans le domaine de l'Intelligence Artificielle (IA), une frame est associé à chaque phénomène du monde réel, pour capturer ce phénomène de façon manipulable par machine.
Les frames sont considéré comme une structure a trois niveaux :

- Un frame : représente un objet ou bien un concept.
- Attributs (slots) : propriétés du concept.
- facettes : décrit les attributs.

Les frames admet la relation de généralisation et spécification entre les frames qui permettent de produire une sorte d'hiérarchie (héritage entre frames).

FRAME : Chercheur
ATTRIBUT

(Nom	§Une chaîne
Prénom	§Une chaîne
Date de naissance	§Une date
)	

Les frames ont deux types de raisonnement : le raisonnement globale qui travaille sur tous les frames comme l'héritage, le filtrage et la classification et le raisonnement local qui est un raisonnement dans le frame lui-même sur ses propriétés et facettes comme les réflexes de contrôle et les réflexes de calcule.

I.7.2. Les graphes conceptuels (J. Sowa 1984)

Les graphes conceptuels sont des outils de représentation de connaissances graphiques (visuel), selon *J Sowa* ces graphes bénéficent de trois avantages essentiels: la possibilité de transformer en des bases de données relationnelles, ils peuvent servir comme des bases sémantiques pour les langages naturels et finalement ils supportent des inférences automatique pour calculer des relations qui ne sont pas explicitement mentionnées [11] [26].

Un graphe conceptuel est un *graphe fini, connecté, bipartie et non orienté, dont les noeuds sont de deux types : les concepts et les relations sémantiques* [11].

Si nous voulons présenter une partie de notre ontologie par les graphes conceptuels, elle sera comme suit :

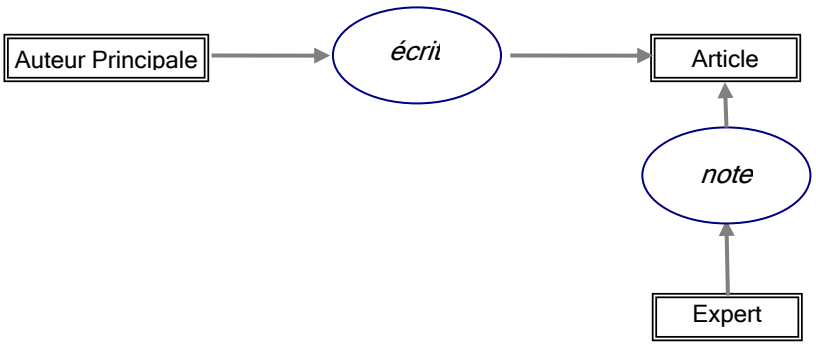
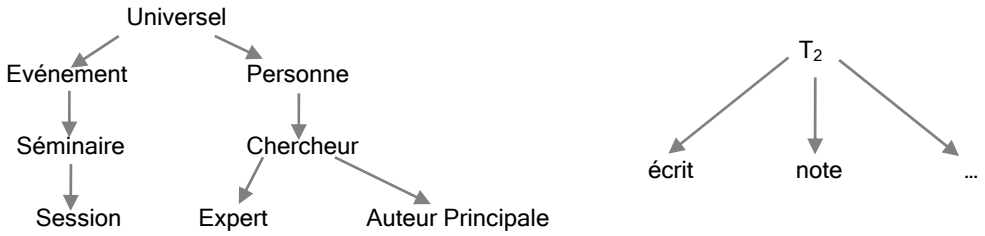


Figure I.2 : Exemple de graphe conceptuel.

Un graphe conceptuel est défini sur un support qui est composé de deux hiérarchies de type, l'une pour les concepts dans laquelle, les concepts sont ordonnés partiellement par une relation « sorte de » (notée ≤) possédant un plus grand élément (noté T) appelé type universel. Et l'autre pour ordonner partiellement les relations, tels que ces relations sont partitionnées en sous-ensembles de types de relations de même nombre d'arrêts possibles.

Par exemple le support pour notre graphe d'exemple est :



Les graphes conceptuels sont basés sur la logique des prédicats du premier ordre, c'est pour cela ils bénéficient d'une base théorique mathématique très solide, d'après le créateur *J Sowa,* il est possible de transformer des graphes conceptuels (notation linéaire) en formules de calcul prédicatives du premier ordre. Lors de la transformation d'un graphe en formule prédicative, les concepts individuels sont transformés en constantes, les concepts génériques sont transformés en variables.

I.7.3. Logiques de descriptions (Brachman 1979)

Comme tous autres familles des logiques, les logiques de descriptions sont basées sur la logique de premier ordre et ils sont appelés parfois les logiques terminologiques [01].

La logique de description divise la base de connaissance en deux parties, la première appelée la partie terminologique TBox qui contient une terminologie (le vocabulaire de domaine)**,** et la deuxième appelée la partie des assertions ABox qui contient les individus.

Le vocabulaire est constitué de concepts et rôles, les concepts dénotent et regroupent un ensemble d'individus (équivalant à la classe dans l'orienté objet), et les rôles sont des relations binaires entre les individus. Les concepts et les rôles peuvent être des entités atomiques (primitives) ou bien complexes (définis), si sont atomiques, ils sont désignés par un simple nom, par exemple le concept *Chercheur* et le rôle *note* dans notre exemple, mais si ils ne sont pas atomiques, ils sont construits à l'aide d'un langage de logiques de descriptions, par exemple si nous voulons ajouter un autre concept qui est *Présidant* (président d'une session) a notre exemple, nous devons le construire comme suite :

$$\text{Président} := \text{Chercheur} \cap \text{préside.Session}$$

La logique de description est définie par le Tbox et le Abox dans la figure: I.3.

Une base de connaissances représentée par la logique de description permet à un système intelligent de raisonner sur ces connaissance, il peut détermine si :

- Un concept C est *satisfaisable* quand il n'est pas vide.
- Un concept C1 subsume un autre C2 (C2 ⊆ C1) quand tout individus de C2 est un individu de C1 (Chercheur ⊆ Personne).
- Un concept C1 est *équivalent* à un concept C2 quand (C2 ⊆ C1) et (C1 ⊆ C2)
- Un individu est une instance d'un concept donné C.
- Deux concepts sont incompatibles si ils n'ont aucun individu en commun.

- d'un ensemble Pc de concepts primitifs,
- un ensemble Pr de rôles primitifs,
- des constantes T et ^,
- des règles de syntaxe suivantes :

C, D →	T	plus général
	^	absurde (le plus spécifique)
	P	concept primitif
	C ∩ D	conjonction de concepts
	C ∪ D	disjonction de concepts
	¬C	négation
	∀r.C	restriction universelle (définit le co-domaine du rôle r)
	∃r.C	restriction existentielle (il existe au moins un objet relié par le rôle r au concept C)
	≤n r.C	cardinalité maximum
	≥n r.C	cardinalité minimum

r →	q	rôle primitif
	r1 ∩ r2	conjonction de rôles
	r1 ∪ r2	disjonction de rôles

Figure I.3 : La logique de description ABox+TBox. [32]

Avant de terminer il est à noter qu'il existe plusieurs systèmes utilisent les logiques de descriptions. Parmi les plus connus, on peut citer *CLASSIC, LOOM,* et *RACER*.

I.8. Le cycle de vie d'une ontologie :

L'ontologie est comme toute entité informatique possède un cycle de vie, ce dernier est inspiré de cycle de vie définis dans le génie logiciel. Avant tout, nous devons étudier nos besoins pour savoir à quoi sert cette ontologie, puis on va la construire, par la suite la diffusion de l'ontologie est une étape nécessaire avant son utilisation.

Après l'utilisation de l'ontologie, les besoins et l'ontologie à leurs tour peuvent être réévaluer et un processus de reconstruction et déclancher si nécessaire, c'est pour cela on parle d'un cycle de vie comme le montre la figure I.4.

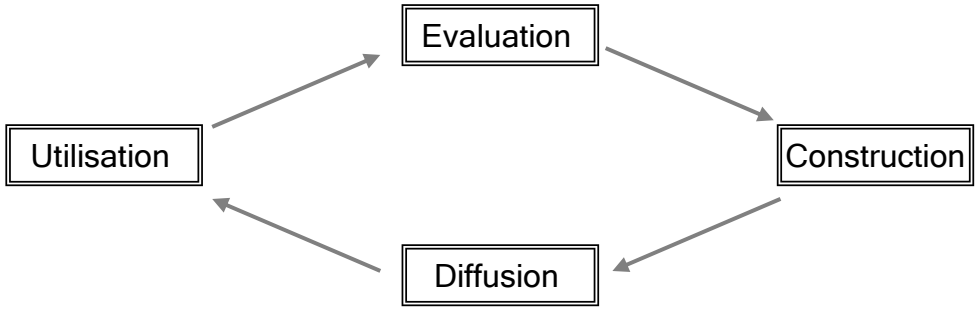


Figure I.4 : Cycle de vie de l'ontologie.

L'étape la plus importante dans ce cycle est la construction de l'ontologie, parce qu'elle répond à la question délicate, comment construire une ontologie? Pour cela elle est décomposée en sous étapes qui sont: la conceptualisation, l'ontologisation et l'opérationnalisation, ces trois étapes peuvent être compléter par une quatrième étape d'intégration, s'il est nécessaire d'intégrer des autres ontologies déjà construites dans notre ontologie en cour de construction.

I.8.1. La construction de l'ontologie

I.8.1.1. L'évaluation des besoins

C'est la première étape dans le processus de construction, elle sert comme une porte d'entrer dont laquelle on va identifier les objectifs de construction et les futurs utilisateurs de l'ontologie ainsi que nous devons bien préciser le domaine d'application.

I.8.1.2. Conceptualisation

Il existe plusieurs ontologies possibles pour le même domaine d'application, à cause des points de vue déférents des personnes au domaine, cette diversité est produite en cours de la conceptualisation de l'ontologie.

Dans cette étape de conceptualisation, en découvert le connaissance de domaine à travers deux techniques :

- **Les interviews avec les experts :** c'est la meilleur façon pour capturer une conceptualisation sur le domaine, mais il est très coûteux en terme de temps et d'effort de réalisation, et parce que les connaissances humaines sont subjectives et ne sont pas objectives, une étape de normalisation sémantique est nécessaire pour aboutir une unique sémantique claire et non ambiguë.
- **L'analyse de documents relatifs à notre domaine :** il est fait soit manuellement ou bien automatiquement pour réduire l'effort de la première technique. Quand nous voulons analyser des documents, il faut faire attention car ces documents peuvent contenir des connaissances qui ne sont pas du domaine, mais elles sont utilisées seulement pour exprimer les connaissances de domaine, même les connaissances relatives au domaine risquent de ne pas être compréhensibles seulement et seulement si elles sont lues par des experts.

Le résultat de cette étape est un modèle conceptuel informel exprimé en langage naturel, et constitué d'un ensemble de concepts et relations, ainsi que leurs instances communes et leurs attributs.

Une autre influence doit être prendre en compte c'est que l'ontologie cible est limitée par ces cas d'utilisation (une ontologie pour le domaine de vendre des voitures n'est pas la même que pour la construction des voitures), donc nous devons identifier ces cas d'utilisation, et pour le faire on peut utiliser une technique appelé les questions de compétences, qui sont des questions exprimées en langage naturel, l'ontologie sera capable de les répondre, après la construction de l'ontologie. Ces questions peuvent être formalisées et utilisées pour valider l'ontologie construite.

Le point le plus délicat dans cette étape, c'est qu'il existe des connaissances relatives au domaine ne sont pas identifier par les experts ni par l'analyse des documents, mais elles sont identifiées lors d'une phase d'utilisation ou un test, et cela implique un retour à l'étape de conceptualisation (le processus de construction n'est pas séquentiel).

I.8.1.3. Ontologisation

Dans cette étape on va construire une ontologie semi formelle, en basant sur le modèle conceptuel construit lors de l'étape précédente et exprimé en langage naturel. Pour diriger et contrôler cette transformation, *T Gruber* propose cinq critères afin de

guider le processus d'ontologisation : la clarté, la cohérence, l'extensibilité, un encodage minimal et un engagement ontologique minimal (ces critères sont détaillés dans les lignes suivants).

Deux types de concepts sont émergés au milieu de l'ontologisation, les concepts sémantiques qui sont des concepts exprimés de manière claires et non ambiguës, à l'aide des principes différentiels (engagement sémantique), et des concepts formels qui sont définis par leur extension (engagement ontologique). Ces engagements sémantiques et ontologiques sont assurés par une hiérarchie sémantique des connaissances, construites en basant sur les liens de subsumption, ces hiérarchies peuvent être construire par l'un des techniques expliqués par *Mike Uschold* et *Michael Gruninger* : *Bottom-up, Top-down* et *Midlle-out* (ces techniques sont détaillées dans le chapitre suivant).

Après la construction de l'hiérarchie conceptuel, il faut la construire dans un langage semi formel, parmi les quels on site le langages de frame, les graphes conceptuels et les logiques de descriptions vu précédemment.

I.8.1.4. Opérationnalisation :

Le but de cette étape est de rendre l'ontologie manipulable par les machines, donc nous devons coder l'ontologie dans un langage approprié (langage opérationnel), souvent cette étape n'est pas nécessaire car le langage d'ontologisation choisi est déjà opérationnel. A la fin de cette étape nous avons une ontologie prête à intégrée dans les machines. Mais avant de le faire, des tests par rapport aux cas d'utilisation identifier au début de construction par les questions de compétences sont appliquées.

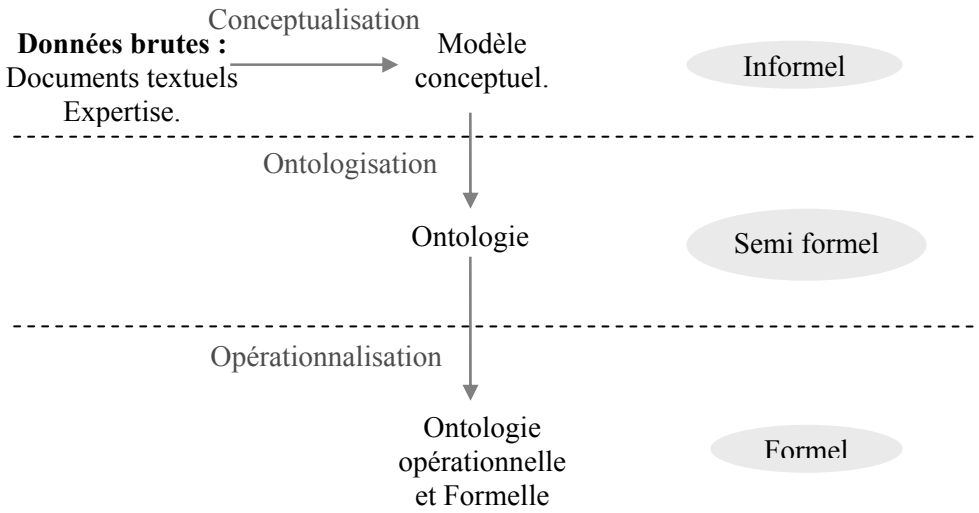


Figure I.5 : Les étapes de construction de l'ontologie.

I.9. Critères pour la construction des ontologies

Il existe quelques critères pour guider le processus de construction des ontologie, nous présentons ici des critères définis et expliquer par Thomas R Gruber dans [Gruber 1993] qui sont :

I.9.1. Clarté

Une ontologie devrait transmettre efficacement la signification de termes définis. Les définitions devraient être objectives. Tandis que la motivation pour définir un concept pourrait être des situations sociales ou des conditions informatiques [13].

I.9.2. Cohérence

Une ontologie devrait être cohérence, c'est-à-dire elle doit être sanctionner les inférences qui sont compatibles avec la définition de l'ontologie. Au moins les termes définis doivent être logiquement cohérents [13].

I.9.3. Extensibilité

Une ontologie devrait être conçue pour prévoir les utilisations du vocabulaire partagé. Elle devrait offrir une base conceptuelle pour une gamme des tâches prévues, et la représentation doit permettre d'étendre et spécialiser l'ontologie En d'autres termes, on devrait pouvoir définir des nouvelles termes pour utilisations spéciales on basant sur le vocabulaire existant, sans que cela exige la révision des définitions existantes [13].

I.9.4. Encodage minimal

La conceptualisation de L'ontologie doit être indépendamment du langage d'implémentation. Pour le partage (l'ontologie) entre différentes applications utilisant des langages différents [13].

I.9.5. Engagement ontologique minimal

Une ontologie devrait exiger le minimal engagement ontologique suffisamment pour capturer et partager la connaissance. C'est-à-dire capturer un maximum de connaissance par un minimum de termes [13].

I.10. Web sémantique

Le Web sémantique désigne un ensemble de technologies visant à rendre le contenu des ressources du World Wide Web accessible et utilisable par les programmes et

agents logiciels, grâce à un système de métadonnées formelles, utilisant notamment la famille de langages développés par le W3C [http://fr.wikipedia.org/wiki/Web_sémantique].

I.10.1 Langages pour le web sémantique

Le web sémantique doit pouvoir être manipulé par les machines. Dans l'état actuel de la technologie, il est alors nécessaire de disposer de langages pour: exprimer les données et les métadonnées, exprimer les ontologies et décrire les services. Parmi ces langages on site RDF et OWL.

I.10.1.1 RDF (*Resource Description Framework*)

RDF est un langage formel qui permet d'affirmer des relations entre des ressources. Il sera utilisé pour annoter des documents écrits dans des langages non structurés, ou comme une interface pour des documents écrits dans des langages ayant une sémantique équivalente (des bases de données, par exemple). RDF est muni d'une syntaxe, et d'une sémantique. Aucun mécanisme d'inférence n'est cependant proposé dans la recommandation.

Un document RDF est un ensemble de triplets de la forme <sujet, prédicat, objet>. Ce document sera codé en machine par un document RDF/XML ou N3, mais est souvent représenté sous une forme graphique.

Les éléments de ces triplets peuvent être des URIs (Universal Resource Identifiers), des littéraux ou des variables. Un tel ensemble peut être représenté de façon naturelle par un graphe (plus précisément un multi-graphe orienté étiqueté) [31].

I.10.1.2 Web Ontology Language

Le langage OWL est basé sur la recherche effectuée dans le domaine de la logique de description. Il peut être vu en quelque sorte comme un format de fichier pour certaines logiques de description. Il permet de décrire des ontologies, c'est-à-dire qu'il permet de définir des terminologies pour décrire des domaines concrets. Une terminologie se constitue de concepts et de propriétés (aussi appelés « rôles » en logiques de description). Un domaine se compose d'instance de concepts.

OWL permet, grâce à sa sémantique formelle basée sur une fondation logique largement étudiée, de définir des associations plus complexes des ressources ainsi que les propriétés de leurs classes respectives. OWL définit trois sous-langages, du moins expressif au plus expressif : OWL-Lite, OWL-DL et OWL-Full.

I.11. Conclusion

Une ontologie est une spécification explicite d'une conceptualisation. Formellement on peut la voir comme un réseau sémantique entre un ensemble de concepts relie entre eux par un ensemble de relations, L'utilisation la plus connue des ontologies est dans le web sémantique dont lequel, les ontologies sont considérées comme la deuxième pile après les métadonnés. Et elles sert a modélisé les connaissances nécessaires à la description - et au traitement - d'un ensemble de ressources.

Dans ce chapitre, nous avons essayé de construire une idée sur le domaine des ontologies et tous ce qui est en relation avec ce domaine, au début on a commencé par expliquer le mot ontologie selon plusieurs acceptation, puis déterminer le principaux composants, ainsi que les différents types d'ontologie.

Après que nous avons expliqué à quoi sert les ontologies, et les différents formalismes de représentation de connaissance pour les formaliser, et comme toute entité informatique l'ontologie à un cycle de vie et nous avons l'expliquer que chaque ontologie passe par les étapes de construction, Diffusion, Utilisation, Evaluation puis encore la construction si la décision est dans l'étape d'évaluation.

Enfin, on a jeté un coup d'œil sur le domaine de web sémantique, en tant que le domaine qui utilise plus les ontologies, ainsi que les langages formels qui sert à formaliser ces ontologies.

Puisque les ontologies sont largement utilisées dans des différents domaines, et souvent dans des domaines critiques comme le web sémantique, elles doivent êtres construites selon un processus (méthode) bien étudier, et pour cela, Dans le chapitre suivant on va détailler les méthodes et les méthodologies qui sert comme un guide au processus de construction des ontologies

Chapitre II

Les processus de développement d'ontologie

II.1. Introduction

Jusqu'au milieu des années 1990, le processus de construction d'ontologie était un art plutôt que comme une ingénierie a ces principes et règles, et puisque aucune méthodologie générale n'ait pour l'instant réussi à s'imposer, de nombreux principes et critères de construction d'ontologies ont été proposés. Ces méthodologies peuvent porter sur l'ensemble du processus et guider l'ontologiste à toutes les étapes de la construction. C'est le cas de METHONTOLOGY, élaborée en 1998 par A. GOMEZ-PEREZ, qui couvre tout le cycle de vie d'une ontologie.

M. USCHOLD et M. KING ont également proposé une méthodologie générale, inspirée de leur expérience de construction d'ontologies dans le domaine de la gestion des entreprises. La méthodologie présentée par M. GRUNINGER et M. S. FOX dans [06] est elle est aussi issue d'une expérience de construction d'ontologie sur ce domaine.

D'autres méthodologies se focalisent sur l'étape de conceptualisation afin d'essayer d'automatiser la construction d'ontologie, parmi les quelles, on va voir, la méthode de Maedche et ses collègues qui repose sur l'utilisation d'une ontologie de base, et la

méthode de Aussenac-Gilles et ses collègues qui utilise le traitement automatique de langages naturels comme un outil d'extraction de connaissance.

Les méthodes et les méthodologies que nous avons cité partagent presque les mêmes étapes de processus globale de construction d'une ontologie, et chaque méthode a une étape dans laquelle le performance est mieux que la même étape dans les autres méthodes, et nous allons essayer de découvrir cette étape dans chaque méthode afin d'essayer de former une méthodologie globale.

II.2. Uschold et King's méthode

Elle est aussi reconnue sous le nom "la méthode ENTERPRISE", c'est le résultat d'une expérience de Mike Uschold et Martin King dans le domaine de construction des ontologies d'entreprise. Et c'est la première méthode proposée pour guider le processus de construction des ontologies [09].

La méthode ENTERPRISE devise le processus en quatre étapes, qui sont :

1. Identifier les objectifs de l'ontologie.
2. Construction de l'ontologie : la construction aussi est devisée en trois points:
    - 2.1 Capturer la connaissance.
    - 2.2 Coder la connaissance.
    - 2.3 Intégrer des autres ontologies dans l'ontologie en cours de construction.
3. Evaluer l'ontologie.
4. documenter l'ontologie.

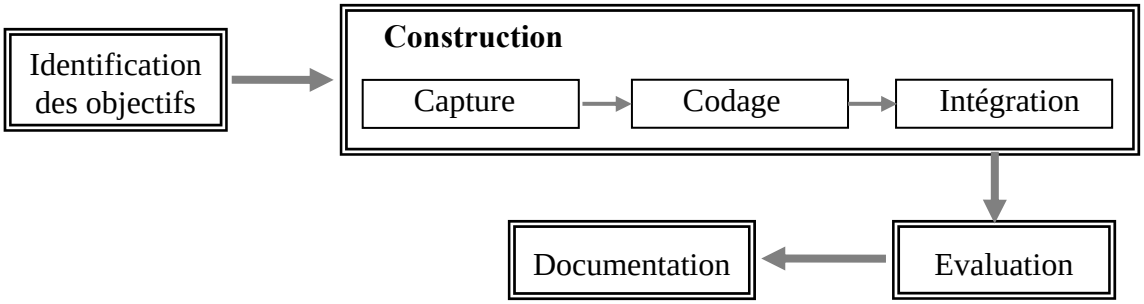


Figure II.1 : La méthode Uschold et King

II.2.1. Etape 1 : Identifier les objectifs de l'ontologie

Dans cette étape en repoudrer aux questions suivantes :

- Pourquoi cette ontologie sera construite.

- Quels sont les futurs utilisateurs de l'ontologie et en quoi elle sert.
- Quels sont les termes utilisés dans le domaine d'application.

II.2.2. Etape 2 : Construction de l'ontologie

A la fin de cette étape, l'ontologie est réellement (physiquement) construite, et comme nous avons dit elle est constitué de trois sous étapes :

2.1 La capture de connaissance : pour capturer la connaissance nous devons **:**

- Identifier les concepts clés et les relations entre eux.
- Produire des définitions textuelles précises et non ambiguës pour chaque concept et relation.
- Identifier les termes qui réfèrent a ces concepts et relations.

Les définitions textuelles ne sont pas des textes littératures, mais des textes qui peuvent se contenir des réfères a des autres termes et inclus des notions comme : Classe, Relation…etc. Ces définitions déterminent la connaissance de l'ontologie.

Mike Uschold et Michael Gruninger ont proposé une extension à la méthode (1996), dans laquelle ils sont expliqués comment identifier les concepts selon trois stratégies :

Bottom-up, *Top-down* et *Midlle-out*.

La stratégie *Bottm-up* : Au début les concepts les plus spécifiques sont identifiés puis ils sont généralisés dans des concepts plus abstraits. On remarque que le résultat aura un niveau très élevé de détails et on risque de :

- Augmenter l'effort global.
- La détection des relations simples entre concepts est difficile.
- Augmenter le risque de l'inconsistance qui conduit a plus de travail c'est-à-dire plus d'effort.

La stratégie *Top-Down* : Au début les concepts les plus abstraits sont identifiés puis ils sont spécifiés en des concepts plus spécifiques.

C'est vrai que cette stratégie permet un mieux de contrôle de niveau de détail, mais si on commence de l'haut on risque d'avoir un niveau très élevé de catégorisation.

La stratégie *Midlle-out* : Au début les concepts les plus importants (concepts de base, concepts au centre de domaine) sont identifiés puis ils sont spécifiés ou bien généralisés selon nos besoins.

Il est clair qu'un meilleur contrôle de niveau de détail et de catégorisation.

2.2 Le codage de connaissance : Transférer le résultat de l'étape précédente en code selon le langage choisis (génération de code)

2.3 Intégration des autres ontologie : cette étape n'est pas obligatoire parce qu'on ne peut pas avoir aucune ontologie à intégrer. L'intégration d'une autre ontologie peut se fait en parallèle avec l'étape de codage ou bien après la fin de ce dernier.

II.2.3.Etape 3 : Evaluation de l'ontologie

L'évaluation de l'ontologie est fait par rapport à une référence, cette référence peut être une spécification, des questions de compétences et/ou le monde réel.

II.2.4. Etape 4 : Documenter l'ontologie

Des documents sont produits dans chaque étape du processus de construction de l'ontologie.

II.3. la méthodologie METHONTOLOGIE

II.3.1. Le processus de développement d'ontologie

Le processus de développement d'ontologie défini dans METHONTOLOGIE est basé sur les standards de développement des logiciels identifiés par IEEE, ce processus réfère aux activités exécuter durant le développement d'ontologie. Ces activités sont groupées dans trois catégories [02] :

II.3.1.1. Les activités de management d'ontologie

Est devisé en :

- **Planification :** identifier les tâches à exécuter, et identifier pour chaque tâche son arrangement, le temps et les ressources nécessaires que se complètent
- **Contrôle :** assure que les tâches déjà planifiés sont exécutés et terminés de la manière attendue.
- **Assurance de qualité :** assure que chaque sortie de ce processus (ontologie, software et documentation) est satisfaisable.

II.3.1.2. Les activités orientées au développement d'ontologie

Il est devisé aussi en trois groupes d'activités :

- **Les activités de pré développement :** contient les activités suivantes :
  - **Etude de l'environnement:** pour savoir ou l'ontologie sera utilisée (plateforme), et les applications ou l'ontologie sera intégrée.

  - **Etude de faisabilité :** pour savoir, est ce que il est possible de construire une tel ontologie, et est-il utile de la construire.

- **Les activités de développement :** contient les activités suivantes :
  - **Spécification :** définir pourquoi l'ontologie doit être construire, quels sont ces cas d'utilisations dans le futur et quels sont leurs futurs utilisateurs.
  - **Conceptualisation :** construire un modèle conceptuel sur la connaissance de domaine.
  - **Formalisation :** transformer le modèle conceptuel en modèle formel.
  - **Implémentation :** coder l'ontologie avec un langage approprié.
- **Les activités post-développement :** contient les activités suivantes :

Maintenance : assure la mis a jour et la correction de l'ontologie.

Utilisation : dans cette activité l'ontologie peut être utilisée par des autres ontologies.

II.3.1.3. Les activités de support d'ontologie

Contient aussi un ensemble d'activités qui doivent être exécutées pour affiner la construction de l'ontologie [02]**:**

- **Acquisition de connaissance :** acquérir le connaissance de domaine.
- **Evaluation :** évaluer l'ontologie, son environnement et la documentation, et prendre des jugements par rapport à une référence.
- **Intégration :** si nous avons besoin aux autres ontologies pour les utiliser dans l'ontologie en cour de construction.
- **Merging :** union de deux ontologies de même domaine par la fusion de ces derniers dans une seule super ontologie.
- **Alignement :** union de deux ontologies de même domaine par l'établissement des différents liens entre ces ontologies, cette activité préserve les deux ontologies originaux.
- **Documentation :** produire des documents claires pour chaque étape et chaque produit.
- **Gestion de configuration :** enregistrer tous les documents et les codes générés pour permettre un contrôle de changement.

Les activités du processus de développement d'ontologie et leur organisation dans des différents groupes sont illustrées par la figure II.2. Ce processus ne spécifie aucun ordre dans laquelle les activités sont exécutés, cela est discuté dans la section suivante.

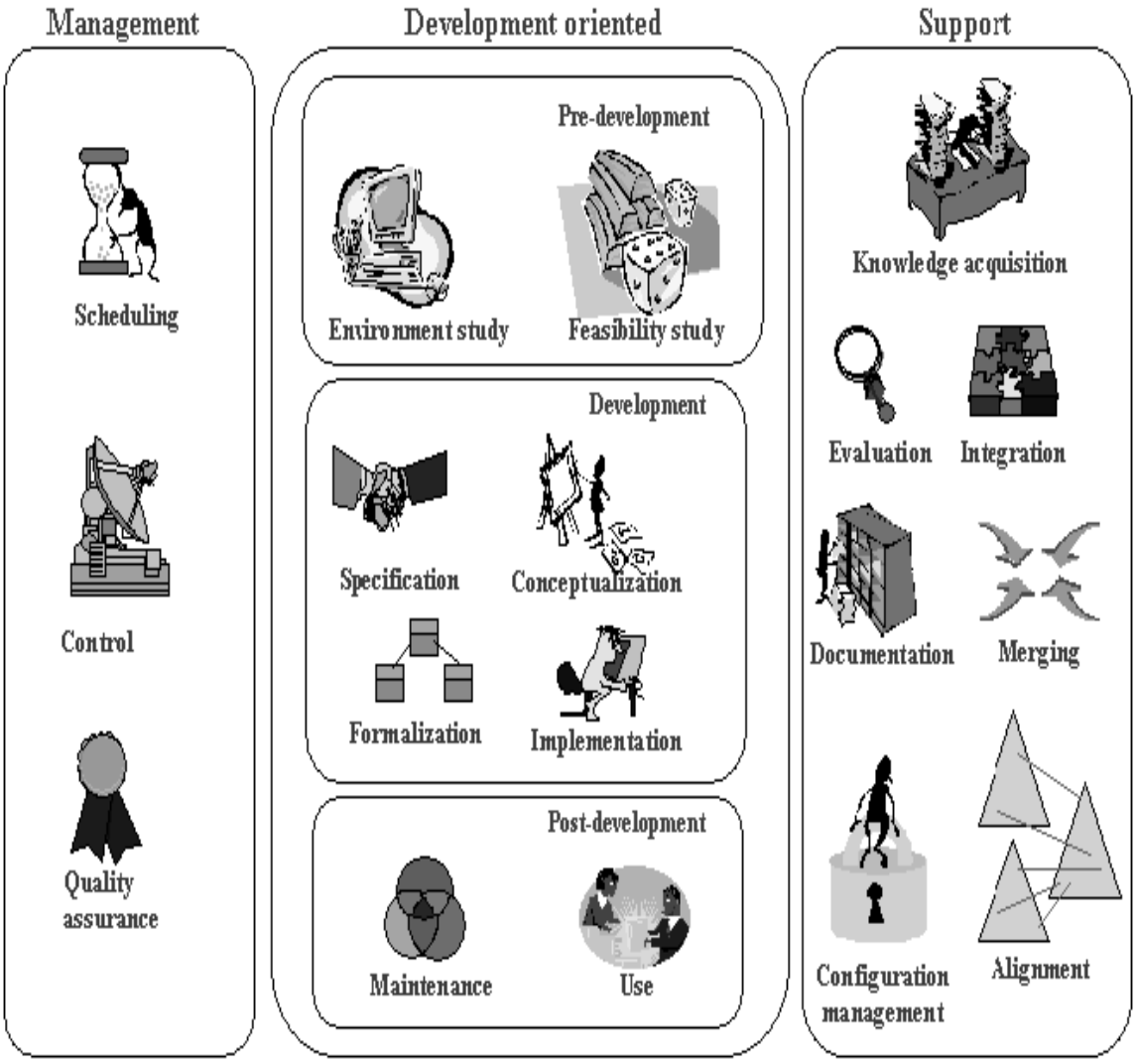


Figure II.2 : Le processus de developpement d'ontologie

II.2. Cycles de vie croisé d'ontologies

Dans METHONTOLOGIE le cycle de vie d'ontologie en construction est basé sur la notion de prototype. Un prototype est construit à la première fois puis il est modifié (ajouter, changer et supprimer des items), le résultat de cette modification est un nouveau prototype, et nous allons maintenant définir l'ordre des activités pour construire un prototype.

Pour chaque prototype METHONTOLOGIE propose de commencer par l'activité de planification pour identifier les tâches à exécuter et ces besoins (temps et ressources), après l'activité de spécification qui est l'entrée pour les activités de développement se commence et en parallèle avec elle certaines activités peuvent être commencées.

C'est vrai que les activités de développement sont exécutées en séquence, mais elles sont exploitées en parallèle avec des activités de management d'ontologie (Contrôle et Assurance de qualité) et des activités de support d'ontologie (Acquisition de connaissance, Evaluation, Intégration, Documentation et Gestion de configuration). [02]

Quand le premier prototype est spécifié, le modèle conceptuel est construit durant l'activité de conceptualisation, puis la formalisation et l'implémentation finissent la construction de prototype. Si quelques anomalies sont détectées on peut retourner à l'activité précédente pour faire une modification.

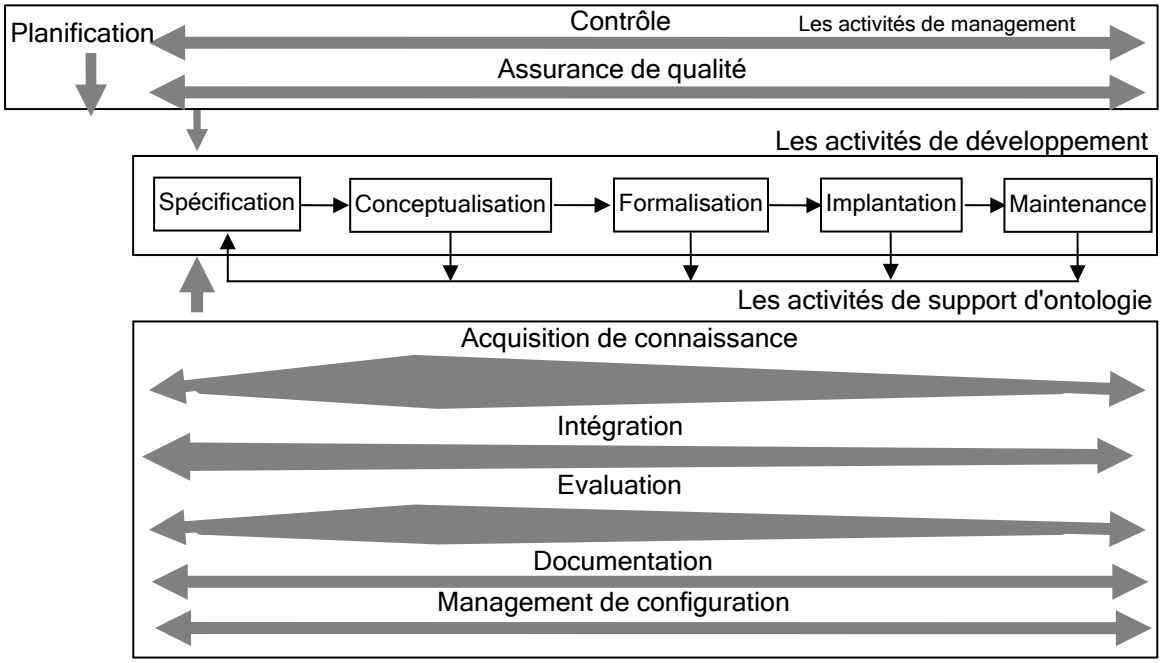


Figure II.3 : Processus de développement et cycle de vie de METHONTOLOGIE

METHONTOLOGIE considère que l'exécution des activités de développement peut invoque l'exécution des autres activités dans des ontologies déjà construites et on parle ici d'interdépendance entre les activités et c'est pour ça on considère les cycles de vie croisés des ontologies.

II.3. Le modèle conceptuel de METHONTOLOGIE

Dans l'activité de conceptualisation, la connaissance assemblée durant l'activité d'acquisition de connaissance est structuré et organisé pour construire une spécification semi formelle, METHONTOLOGIE utilise pour accomplir les objectifs de conceptualisation un ensemble de représentations intermédiaires indépendantes de la représentation de connaissance et de tout langage d'implémentation.

METHONTOLOGIE devise l'activité de conceptualisation en onze tâches illustrée dans la figure II.4. [02]

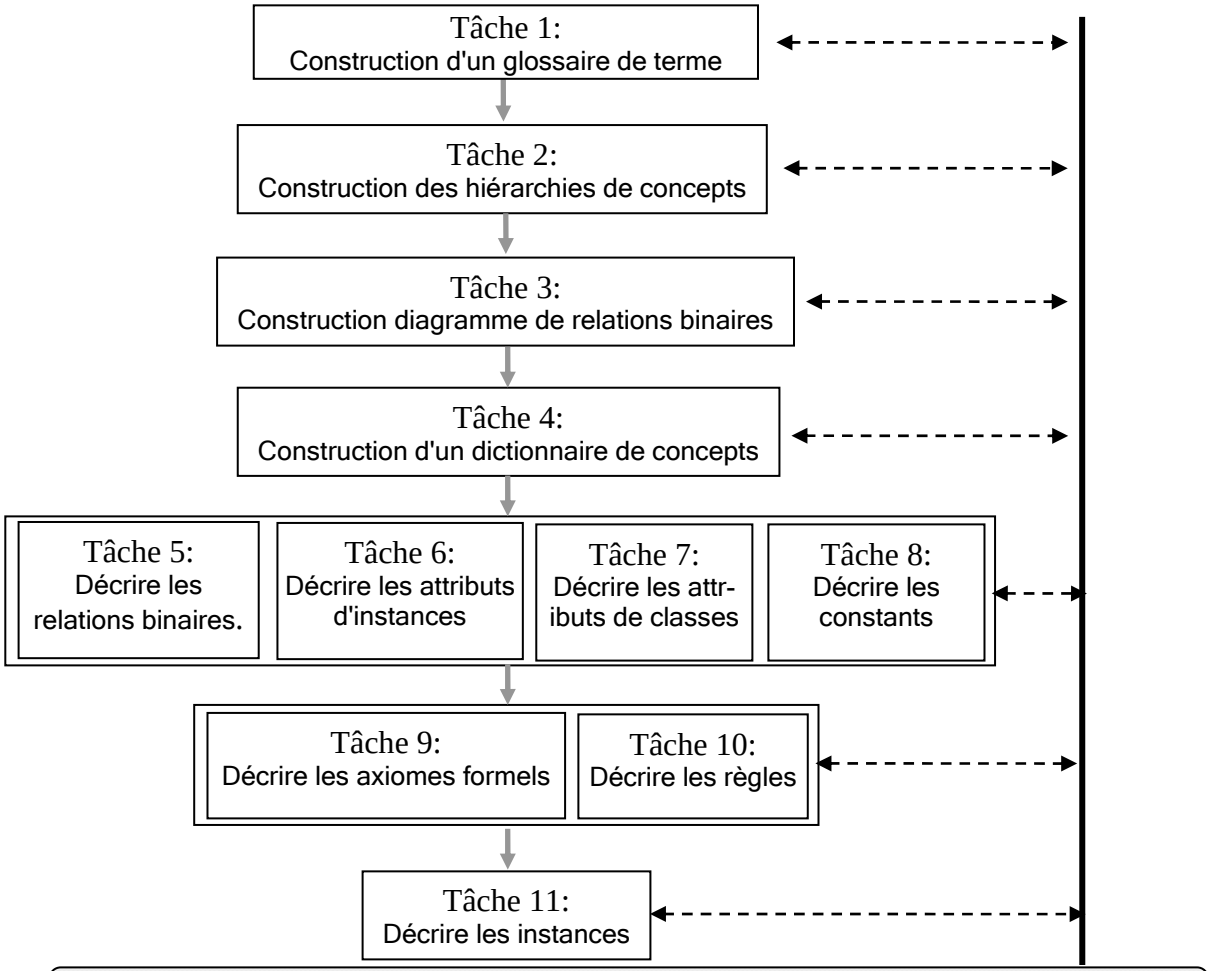


Figure II.4 : Les tâches de l'activité de conceptualisation METHONTOLOGIE

Tâche 1 : Construction d'un glossaire de terme : Construire le glossaire des termes qui identifie l'ensemble de termes à inclure dans l'ontologie, leurs définition en langage naturel, leurs synonymes et acronymes s'ils existent.

Tâche 2 : Construction des hiérarchies de concepts : Construire des hiérarchies de concepts pour classifier les concepts. Le résultat de cette tâche pourrait être un ou plusieurs hiérarchies où les concepts sont classifiés.

Tâche 3 : Construction du diagramme de relations binaires : Construire les diagrammes de relation binaire pour identifier les relations entre les concepts de l'ontologie, ces relations peuvent être avec des concepts d'autres ontologies.

Tâche 4 : Construction d'un dictionnaire de concepts : Construire le dictionnaire de concepts, qui inclut les instances, les attributs, et les relations pour chaque concept.

Tâche 5 : Décrire les relations binaires : Décrire les relations identifiées dans la tâche 2 en détail, on va préciser dans un tableau pour chaque relation le concept source, le concept, le concept cible, la cardinalité source, les propriétés mathématiques et la relation inverse.

Tâche 6 : Décrire les attributs d'instance : Décrire en détail chaque attribut d'instance qui apparaît dans le dictionnaire de concept, le résultat de cette tâche est un tableau où les attributs d'instance sont décrit.

Tâche 7 : Décrire les attributs de classes : Décrire en détail chaque attribut d'instance qui apparaît dans le dictionnaire de concept, le résultat de cette tâche est un table où les attributs d'instance sont décrit.

Tâche 8 : Décrire les constants : Décrire en détail chaque constant et produire un tableau de constants

Tâche 9 : Décrire les axiomes formels: nous devons identifier les axiomes formels nécessaires puis les détailler dans le tableau des axiomes formels, on donne pour chaque axiome: le nom, la description en langage naturel, l'expression formelle en utilisant la logique de premier ordre, les concepts, les attributs, les relations à lesquelles l'axiome se réfère et finalement les variables utilisés.

Tâche 10 : Décrire les règles : nous devons identifier les règles nécessaires pour l'ontologie puis les détail dans le table**au** des règles, on donne pour chaque règle: le nom, la description en langage naturel, l'expression formelle, les concepts, les attributs, les relations à lesquelles la règle se réfère et finalement les variables utilisés.

Tâche 11 : Décrire les instances : donner le tableau des instances qui contient les noms des instances et leur concepts ainsi que les attributs avec ces valeurs si ils sont connu.

II.4. Grüninger et Fox méthodologie

Elle a été développer à l'université de Toronto par Grüninger et Fox, qui sont des experts dans le domaine de construction de gestion des entreprises, cette méthodologie a été utilisée pour construire le TOVE ontologies (c'est pour cela on la

trouve dans les littératures sous le nom le TOVE (TOronto Virtual Enterprise) méthodologie)**.**

Est une méthodologie très formelle, basée sur l'utilisation des questions appelées des questions de compétences (on les appellent des questions parce qu'ils représentent les questions que l'ontologie sera capable de les répondre) pour déterminer le domaine de l'ontologie, cette méthodologie est considérée très utile lorsque le domaine ne contient que peu de connaissances.

Et pour construire une ontologie selon le TOVE méthodologie, le processus présenté dans la figure II.5 doit être suivi [06]:

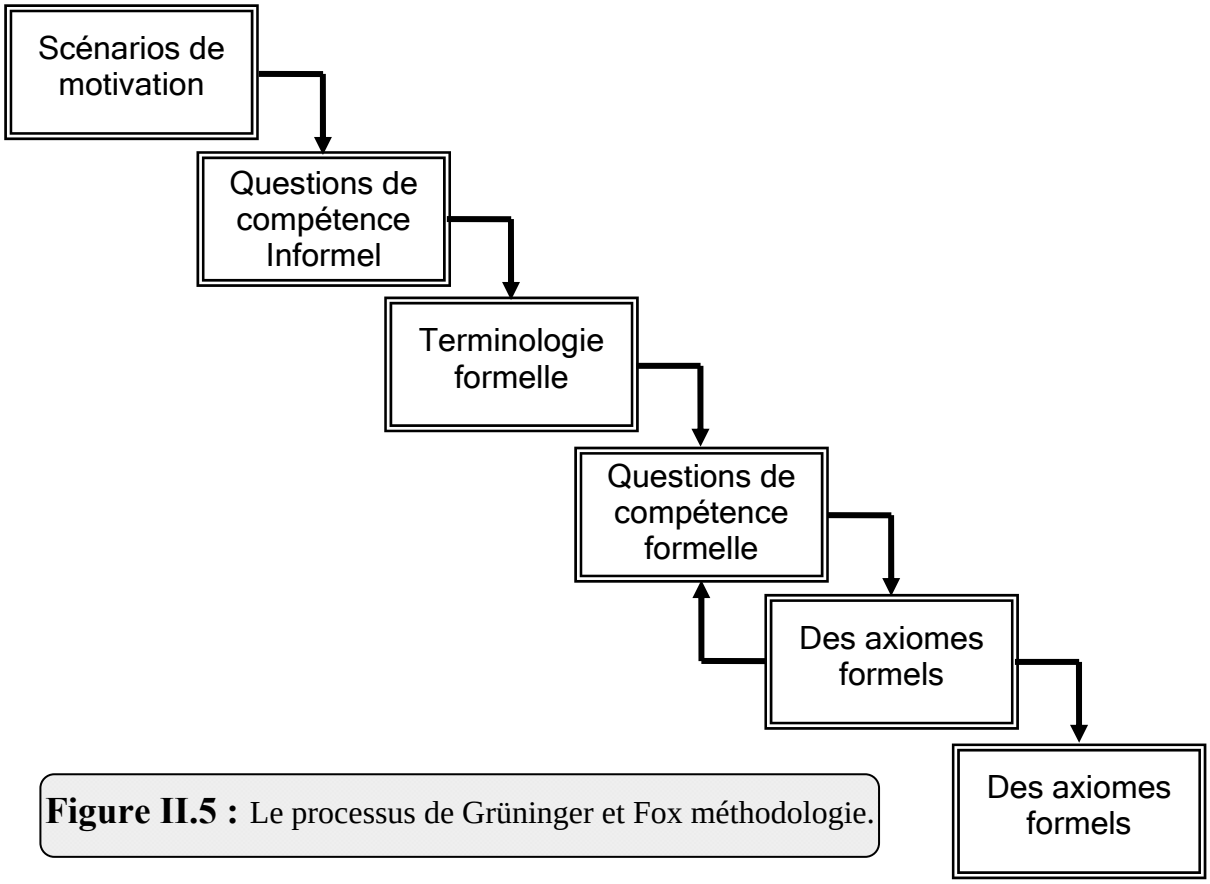


Figure II.5 : Le processus de Grüninger et Fox méthodologie.

II.4.1. Etape 1 : Identification des scénarios de motivations

Le développement de l'ontologie est motivé par des scénarios, qui représentent des problèmes rencontrés dans l'entreprise ou bien des problèmes qui ne sont pas traités par les ontologies actuelles, ces scénarios offrent aussi des solutions intuitives et possibles pour ces problèmes.

II.4.1. Etape 2 : Elaboration des questions de compétence informelle

Elaborer des questions en langage naturel, qui seront répondues par notre ontologie, ces questions ne doivent pas être des requêtes de recherche simples pour capturer mieux la connaissance de domaine.

II.4.3. Etape 3 : Spécification de terminologie

Après avoir extraire les contenus de l'ontologie par les questions de compétence, nous devons les rendre formelles on utilisant les concepts, attributs et les relations de la logique de premier ordre.

II.4.4. Etape 4 : Elaboration des questions de compétence formelle

Les questions de compétences informelles sont formalisées en utilisant la logique de premier ordre.

II.4.5. Etape 5 : Spécification des axiomes formels

Les axiomes exprimés en logique de premier ordre, sont utilisés pour définir les termes de l'ontologie ainsi que leurs contraintes.

II.5. La méthodologie On-To-Knowledge

On-To-Knowledge est le nom d'un projet d'un ensemble de partenaires (l'institue AIFB de l'université de Karlsruhe, le Vrije université d'Amsterdam et British Telekom)[14], qui ont essayés d'améliorer la qualité de gestion de connaissance dans des organisations grandes et distribués. Ce projet contient une méthodologie pour construire des ontologies qui sont en suite utilisés par l'application de gestion de connaissance. C'est pour cela, les ontologies construites par cette méthodologie restent très liées à l'application.

Et comme toute méthodologie, elle est devisée en étapes et comme le montre la figure II.6, chaque étape est la base pour exécuter l'étape prochaine [14].

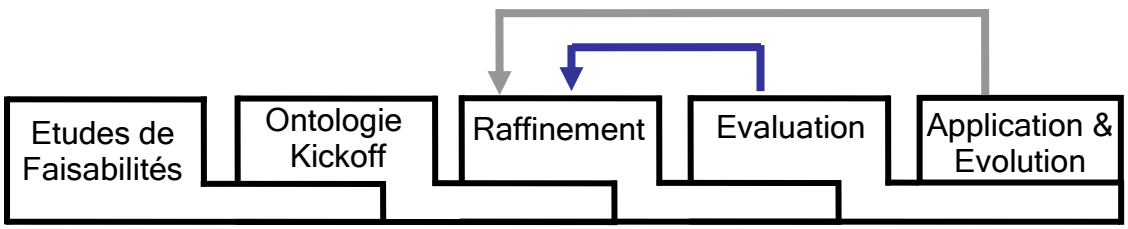


Figure II.6 : La méthodologie On-To-Knowledge.

II.5.1. Etape 1: Etude de Faisabilités

On-to-knowledge emprunte l'étude de faisabilités de la méthodologie commonKADS, le but de cette étape est de prendre une décision pour continuer ou bien arrêter le processus.

Dans l'étape *d'étude de faisabilités* on va identifier les problèmes de domaine ainsi que des solutions potentielles pour ces problèmes, en utilisant trois modèles définis dans commonKADS :

Le modèle organisationnel, le modèle de tâches et le modèle agent[14].

L'étude de faisabilités est considérée comme une base pour l'étape de *kickoff*[14].

II.5.2. Etape 2: Kickoff

Dans cette étape, une première version de l'ontologie est construite, le cœur de cette étape est un document appelé le document de spécification des besoins de l'ontologie (*Ontology Requirements Specification Document* "ORSD"), qui servent comme un guide pour les constructeurs de l'ontologie afin de prendre des décision sur le contenu et la structure hiérarchique de l'ontologie, qui sert comme un guide pour les constructeurs de l'ontologie afin de prendre des décisions sur le contenu et la structure hiérarchique de l'ontologie.

Ce document doit contenir des informations sur [14]:

- *Les objectifs et le domaine de l'ontologie :* le domaine d'application doit être précis le plus possible, qui peut nous aider à identifier des ontologies utilisée dans notre domaine déjà construites. Pour les objectifs, le modèle de tâches défini à l'étape *d'étude de faisabilités* peut sert comme une base pour décrire les objectifs.
- *Les sources de connaissance :* peuvent être des experts de domaine (en utilisant des interviews ou bien des questions de compétences), des ontologies (par la réutilisation des ontologies), des sources internes comme les bases de donnés, et des sources externes comme le web.
- *Les utilisateurs et les scénarios d'utilisation :* ce document contient aussi une description des utilisateurs de l'ontologie, ainsi que en quoi ils sont utilisés.

Le ORSD contient aussi une description sur les applications supportées et les questions de compétences.

Et comme nous avons dit au début le résultat de cette étape est une ontologie semi formelle et le document ORSD.

II.3.3. Etape 3: Raffinement

Elle est composée de deux sous étapes :

L'extraction de connaissance : la capture de connaissance de domaine se fait par les trois approches *Bottom-up, Top-down* et *Midlle-out* :

Les cas d'utilisations et les questions de compétences suivent une approche *Top-down* parce que, on va commencer d'identifier les concepts et les relations les plus génériques.

L'approche *Midlle-out* est plus utile lorsque on veut utiliser des autres ontologies pour construire notre ontologie, et finalement l'approche *Bottom-up* est utilisée lorsque nous voulons faire une analyse automatique d'un ensemble de documents.

Une combinaison entre les trois approches est possible et peut un résultat acceptable avec un coût raisonnable[14].

Formalisation : Formaliser l'ontologie qui est jusqu'à maintenant semi formelle et ne contient que des relations de type *est un,* alors on va enrichir l'ontologie avec des autres relations selon l'analyse de domaine que nous avons fait et finalement on va la formaliser en utilisant un langage d'ontologie.

Durant cette étape une vérification entre l'ontologie formalisée et les besoins spécifiés dans l'étape de *Kickoff* est nécessaire.

II.3.4. Etape 4: Evaluation

L'évaluation de l'ontologie englobe deux activités, dans la première on vérifie est ce que l'ontologie satisfait les besoins des utilisateurs, et elle répondra aux questions de compétences. Dans la deuxième activité on teste l'ontologie dans l'application cible.

Cette étape peut engendrer un retour vers l'étape de *Raffinement,* les auteurs de la méthode suggèrent d'exécuter plusieurs itérations *Evaluation-Raffinement-Evaluation* pour atteindre l'ontologie adéquate[14].

II.3.5. Etape 5: Application & Evolution

- **Application :** c'est le temps pour la mise en œuvre de l'ontologie.
- **Evolution** : l'ontologie doit être sous le contrôle durant son évolution, des règles pour la mise a jour de l'ontologie sont définies et nous devons assurer

la maintenance de l'ontologie en utilisant deux stratégies : la stratégie centralisée et la stratégie distribuée.

- Comme nous avons dit pour l'itération *Evaluation-Raffinement-Evaluation*, une autre itération peut être nécessaire à exécuter selon les auteurs : *Application & Evolution – Raffinement – Evaluation – Application & Evolution*[14].

II.6 La méthode Maedche et ses collègues

Cette méthode départ d'une ontologie de base (SENSUS, WordNet…) et essaye de l'enrichir avec des nouveaux concepts et relations. Ces concepts et relations sont appris en utilisant des techniques de traitement et d'analyse de langages naturels sur des ressources identifiées précédemment par le constructeur. Ces ressources sont un ensemble de documents de domaine décrivent les plus part des concepts et les relations qui seront inclus dans l'ontologie ainsi que la terminologie de domaine. L'ontologie doit être élaguée et focalisée sur un domaine spécifique par un ensemble des approches basées sur la statistique [02].

Cette méthode se déroule sur cinq activités :

II.6.1. Activité 1 : La sélection d'une ressource

Dans cette activité les ressources peuvent être des ontologies de base ou bien des documents, cette activité commence par la sélection d'une ontologie de base qui est utilisée comme une base durant tout le processus de learning, cette ontologie doit contenir les concepts génériques et les concepts spécifiques au domaine [02].

Aussi des documents doivent être spécifiés par l'ontologiste qui sont utilisés durant toutes les étapes pour raffiner et entendre l'ontologie précédente.

II.6.1. Activité 2 : Concepts learning

Cette activité consiste à utiliser un outil de traitement automatique du langage naturel (TALN) pour extraire des nouveaux concepts (concepts génériques et concepts spécifiques au domaine) a partir de texte.

Cet outil utilise "pattern-based" extraction et "conceptual clustering", la selection de l'outil se dépend de la langue cible (Anglais, Français, Espagnol …).

La méthode suggère de lier les nouveaux concepts avec l'ontologie de base en utilisant les relations ("above all" et " Subclass of") [02].

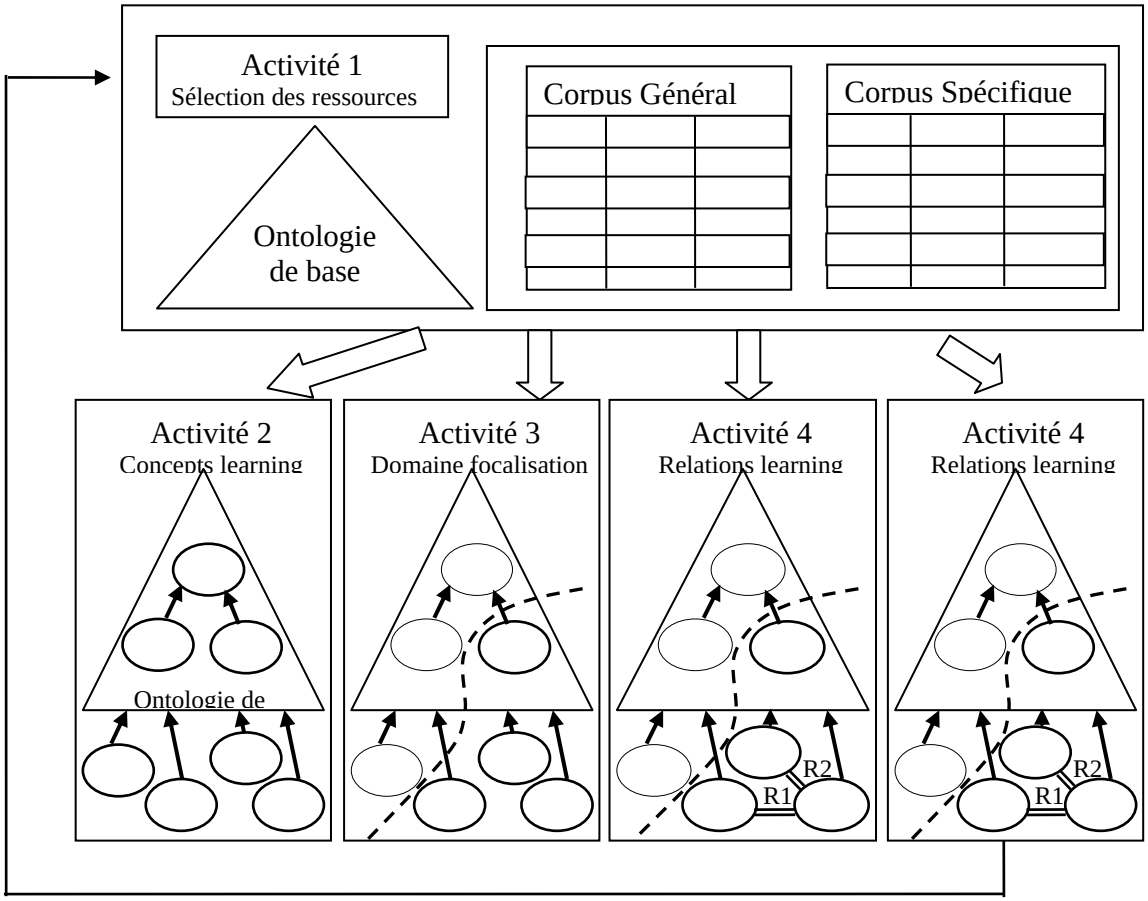


Figure II.7 : La méthode Maedche et ses collègues.

II.6.3. Activité 3 : Domaine focalisation

Elle consiste à élaguer l'ontologie de base enrichi en prenant en compte le résultat d'analyse des termes fréquents dans les deux corpus, générique et spécifique au domaine, et elle consiste aussi à enlever les termes générales les concepts générales.

Les termes les plus fréquents dans les deux corpus doivent être proposé a l'ontologiste pour décider qui parmi eux doit être gardé dans l'ontologie enrichi [02].

II.6.3. Activité 4 : Apprendre des Relations

Ad hoc relations entre les concepts de domaine sont appris par "pattern-based" extraction et les règles d'association [02].

II.6.3. Activité 5 : Evaluation

Alors il est le temps pour évaluer l'ontologie résultat et décider de répéter le processus autre fois s'il est nécessaire [02].

II.7. La méthode Aussenac-Gilles et ses collègues

Cette méthode est basée sur le principe d'extraction de connaissance à partir des documents techniques, la méthode utilise un outil de traitement automatique du langage naturel TALN pour analyser un corpus textuel afin de construire un modèle de domaine. La méthode combine entre les outils d'acquisition de connaissance basée sur le linguistique et les techniques de modélisation pour garder des liens entre le modèle et le texte [28].

Durant le processus de construction de l'ontologie, il est supposé que :

1 – le constructeur (l'ontologiste) doit avoir un connaissance compréhensif sur le domaine, afin que ce dernier peut décider quels termes sont des<termes de domaine et quels sont les concepts et relations qui sont reliés avec ces termes.

2 - l'onologiste doit être en connaissance comment l'ontologie sera utilisée.

La méthode se déroule sur quatre activités : la sélection des ressources, l'étude linguistique, normalisation et formalisation.

Activité 1 : La sélection des ressources : comme nous avons dit, les ressources dans cette méthode sont des documents techniques (corpus textuels), ces corpus doivent être :

1 – des corpus complets : couvrir complètement le domaine d'application.

2 – il est mieux si les corpus sont sélectionnés par un expert de domaine.

Activité 2 : Etude linguistique : Dans cette activité on va sélectionner un outil et une technique linguistiques adéquats à nos objectives.

Activité 3 : Normalisation : dans cette activité on va construire un réseau sémantique représente le modèle conceptuel.

Activité 4 : Formalisation : les concepts et les relations du réseau sémantique seront implémentés dans un langage formel.

II.8. Etude des travaux à base d'agent pour le développement d'ontologie

Les deux termes Ontologie et Système Multi-Agent sont souvent liés, on sais que les systèmes Multi-Agents utilisent les ontologies pour fonctionner, en d'autre termes les agents utilisent l'ontologie comme un langage commun de communication, et comme le langage naturel est un moyen de communication entre les êtres humains, l'ontologie est un moyen de communication pour les agents (on parle ontologie).

Dans le sens contraire, c'est-à-dire les systèmes Multi-Agents qui construisent ou aident à construire les ontologies on trouve le système *Dynamo* (Un système Multi Agent adaptatif Pour la construction d'ontologies à partir de textes) [21] que nous détaillerons dans ce chapitre.

En plus on va jeter un coup d'œil sur trois outils très connus pour la génération automatique d'ontologie, on parle ici de *Text2Onto, OntoGen* et *Terminae.* Pour prendre une idée claire sur le domaine d'ontologie learning et bénéficier de leurs remarques et résultats dans ce domaine.

II.8.1. Dynamo : Un systpème Multi Agent adaptatif Pour la construction d'ontologies à partir de textes [21]

Dynamo est un acronyme pour« DYNAMic Ontologies ». La construction de l'ontologie par Dynamo est semi automatique, car le modèle proposé par *Kévin OTTENS* prend en compte l'intervention de l'ontologue (l'ingénieur de l'ontologie) pour atteindre le résultat finale satisfaisable. Le système est développé en utilisant la méthode ADELFE (Atelier de DEveloppement de Logiciels à Fonctionnalité Emergente), qui est une méthode dédiée au développement de systèmes multi agents adaptatifs, et décrite suivant trois axes fondamentaux : le processus, les notations et les outils. Son processus se déroule durant cinq étapes : les besoins préliminaires, les besoins finals, l'analyse, la conception et le codage.

II.8.1.1. Les entrés de système : la construction de l'ontologie est à partir d'un corpus textuel, qui est une *collection de textes (éventuellement un seul texte) constituée à partir de critères linguistiques ou extra-linguistiques pour évaluer une hypothèse linguistique ou répondre à un besoin applicatif* [Condamines, 2003], mais en réalité le

système n'utilise pas le corpus comme tel, il utilise des résultats d'analyses syntaxiques et terminologiques de ce corpus comme entrée.

L'auteur de ce système a choisi d'utiliser l'extracteur de termes *Syntex*, qui prend en entrée le corpus textuel et donne en sortie un réseau « Tête-Expansion » qui est utilisé par le suite par Dynamo pour construire l'ontologie.

II.8.1.2. Architecture générale :

Le système Dynamo se compose de trois grandes parties (figure II.8) :

- Le réseau de termes : qui est le résultat d'une extraction de terme en utilisant un extracteur de terme sur un corpus textuel.
- Le système multi agent : qui utilise le réseau de terme pour se fonctionner, tel que chaque terme du réseau est représenté par un *Agent Concept*, le rôle de chaque agent est de trouver une position de l'hiérarchie en cour de construction, et pour réaliser ce rôle l'agent repose sur ses compétences personnelles et la coopération avec les autres agents.
- L'interface permettant à l'utilisateur de visualiser et contrôler le processus de classification et ses résultats.

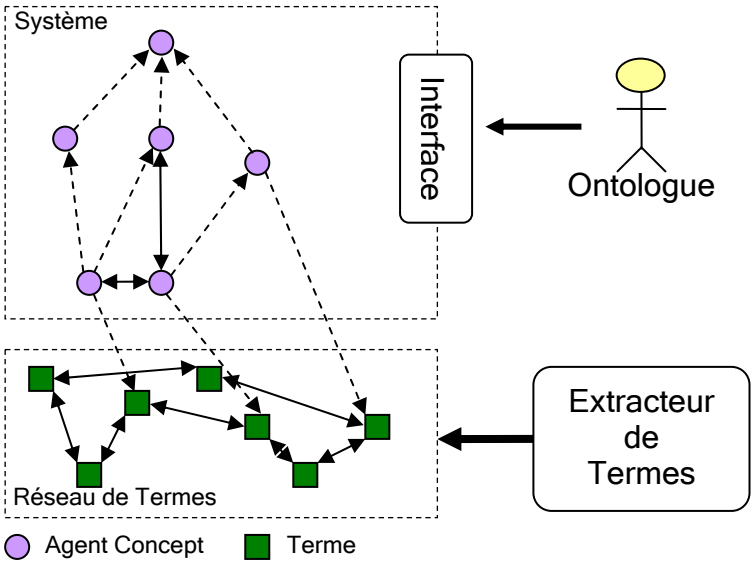


Figure II.8 : Architecture du système Dynamo.

La structure interne d'un agent concept : En respectant la méthode de construction des systèmes multi agents adaptatif ADELFE, un agent doit être décrit par six modules ou bien composants :

Le composant caractéristique, perception, représentation, interaction**,** coopération, et action

II.8.1.3. L'algorithme distribué de classification

Etat initiale : un agent TOP est créé comme une racine de l'hiérarchie. Puis pour chaque terme dans le réseau de termes en créant un agent concept dont le père est l'agent TOP.

L'algorithme se déroule en parallèle au sein de chaque agent pour que l'agent trouve une position dans l'hiérarchie afin d'atteindre une situation d'équilibre globale.

Première étape : la première étape se déroule dans les agents ayant plus d'un frères, chaque agent un message (vote) a son père contient la liste de ces frères par ordre de dissimilarité décroissante.

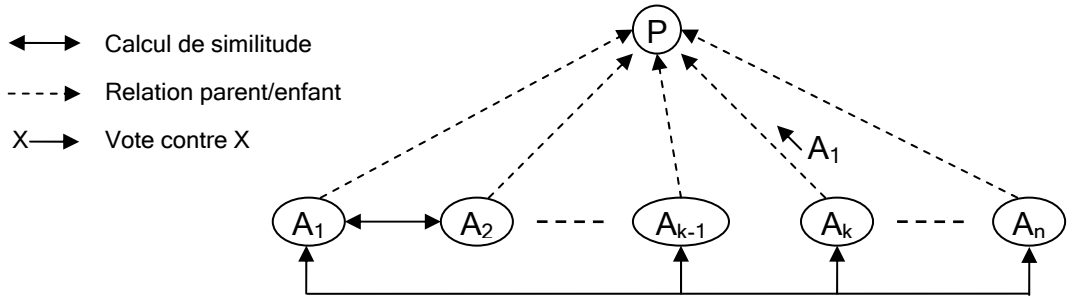


Figure II.9 : Classification distribuée : première étape

Deuxième étape : après que l'agent père reçoit tous les messages de ces fils, l'agent père regroupe ces fils en trois groupes :

- Le fils qui a gagné l'élection c-à-d le fils qui a été choisis par la majorité de ces frères comme un frère dissimilaire (dans notre exemple c'est le fils A_1).
- Les fils qui ont choisis le frère le plus dissimilaire (A_k à A_n).
- Les fils restants.

L'agent père crée un nouvel agent (P'), qui devient un nouveau père pour les agents de deuxième groupe.

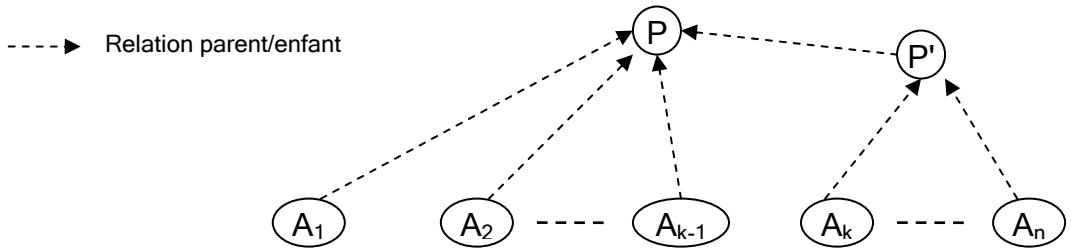


Figure II.10 : Classification distribuée : deuxième étape.

Quand un agent ne possède qu'un seul frère, il arrête l'exécution de cet algorithme, donc le but est d'atteindre un arbre binaire.

Nous avons supposé qu'il est possible de calculer la similarité entre n'importe quel deux termes, mais en réalité ce n'est pas possible car il existe des termes qui n'ont aucune valeur de similarité avec des autres termes, et pour cela nous devons utiliser d'autre moyen de classification. L'auteur a défini cinq règles pour positionner ces termes :

Règle 1. *Quand un agent E est insatisfait de son père P, il évalue a(Fi, E) avec tous ses frères (notés Fi). Celui maximisant a(Fi, E) est alors choisi comme nouveau parent.*

Tel que *a(P, E)* est la fonction d'adéquation entre un parent *P* et un enfant *E*, la régle est appelée l'algorithme de couverture en tête. L'agent dispose aussi d'un mécanisme de décision pour choisir entre l'exécution de l'algorithme de classification, l'algorithme de couverture en tête ou traiter les messages reçus.

Règle 2 Elimination des branches dégradées **:** *Quand un agent P' a des fils (notés A_k), mais aucun frère, alors P' propose à ses fils d'avoir son parent P comme nouveau parent.*

Règle 3 détection des agents inutiles : *Quand un agent P' n'a aucun fils et n'est représenté par aucun terme, il doit se retirer du système.*

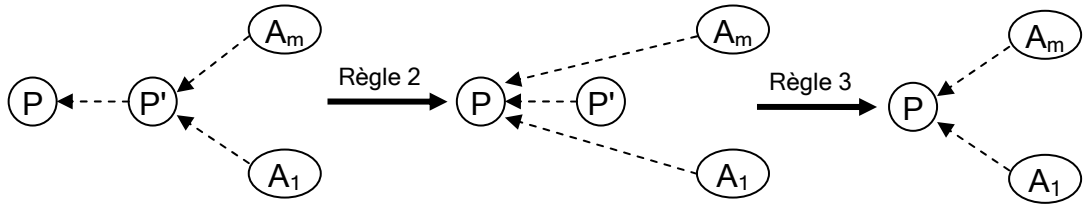


Figure II.11 : Classification distribuée : deuxième étape.

Règle 4 S'éloigner des arbres binaires : *Quand un agent P' a un de ses fils (notés A_k) qui ne respecte plus la propriété sur tolérance $C_{p'}$, alors P' propose à ses fils d'avoir son parent P comme nouveau parent.*

Tel que la propriété de tolérance est :

$$\forall i \in [k;n], \forall j \in [k;n]/\{i\}, \forall l \in [1;k-1]: \text{sim(Ai, Aj)} > \text{sim(Ai, Al)} + C_{p'}$$

Règle 5 S'éloigner des arbres binaires : *Quand un agent P' a un nombre de fils trop élevé (respectivement, trop faible), il abaisse (respectivement, augmente) sa tolérance $C_{p'}$.*

II.8.2. Text2Onto

Text20nto est un outil conçu pour construire des ontologies à partir de textes de manière complètement automatique (voir figure II.12). Il est codé en java et est

composé de modules qui extraient à partir des textes, des concepts, des relations entre ces concepts (relation d'équivalence, hiérarchiques, etc.) et des instances de concepts. Chaque module peut utiliser différents algorithmes et combiner leurs résultats : on peut ainsi combiner des patrons d'extraction "à la Hearst" et une ressource comme WordNet pour construire une hiérarchie. *Text20nto* utilise l'architecture GATE pour pré traiter les textes. Les résultats sont dotés d'une mesure de confiance entre 0 et 1 obtenue à l'aide de différentes mesures combinables (TF.IDF, RTF, entropie) [23].

Le système KASO dont la conception est centrée utilisateur peut être coupler à Text2Onto pour affiner l'ontologie produite à l'aide des méthodes d'acquisition de connaissances telle que la mise en échelle (laddering) et le tri par cartes. La nécessite´ d'avoir recours à des étapes en aval de Text2Onto montre les limites de l'approche tout automatique pour la conceptualisation qui, de notre avis, ne peut se passer de l'intervention humaine [23].

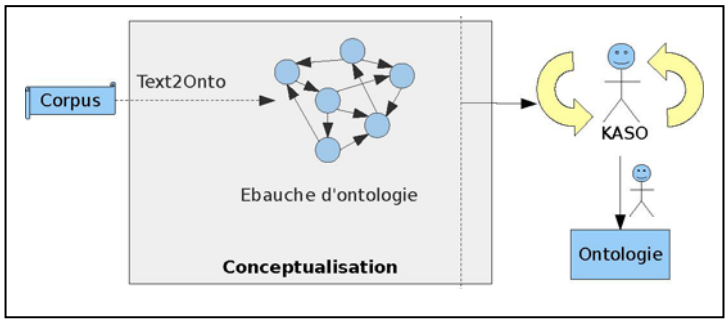


Figure II.12: L'architecture de Text2Onto+ KASO

II.8.3. OntoGen

OntoGen qui est codé en dot net, implémente une approche semi-automatique pour la construction d'ontologies de thèmes (topic ontologies) à partir des collections de documents (voir figure II.13). C'est un outil interactif qui suggère à l'expert du domaine des concepts sous la forme de classes de documents, propose une dénotation et leur associe automatiquement des instances (les documents). Il permet de visualiser l'ontologie en cours de construction. OntoGen exploite des algorithmes de fouille des textes non supervisés (k-means, LSI) ou supervisés (svm active learning) mais toujours selon une approche descendante. A chaque étape le classifieur travaille sur la sous collection associé au concept qui vient d'être construit. OntoGen propose à l'expert, et c'est à ce dernier de choisir la proposition correcte parmi celles qui lui

sont présentés. C'est une approche semi-automatique de la conceptualisation : les outils de classification de documents sont utilisées pour préparer le travail de conceptualisation, l'expert du domaine est guidé dans une démarche descendante mais c'est lui qui construit les concepts et choisit quelles zones de l'ontologie affine. Sur notre exemple provenant du corpus BIT [23].

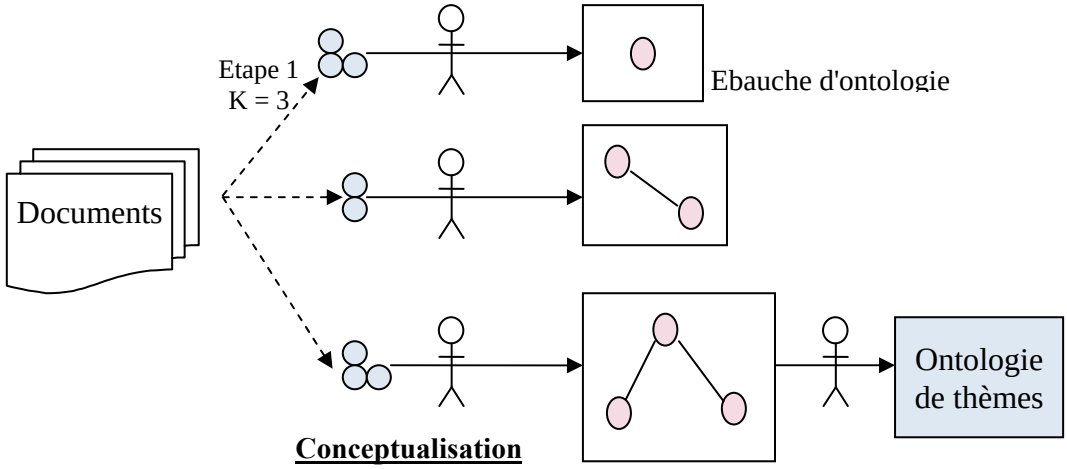


Figure II.13: L'architecture d'OntoGen.

II.8.4. Terminae

Terminae (Aussenac-Gilles et ses collègues, 2008) est une méthode (schématisée sur la figure II.14) supportée par un logiciel. Elle propose de guider l'ontologue dans la conception de l'ontologie. Terminae s'appuie sur les résultats des outils de traitement automatique des langues (extracteur de termes, concordancier, détecteur de synonymie, analyseur syntaxique) pour extraire des éléments dans lesquels l'ontologue puise selon ses objectifs de modélisation. Une fiche terminologique affiche les occurrences d'un terme dans le corpus, une ou plusieurs définitions en langage naturel du ou des concepts terminologiques associés et les termes synonymes. Le concept terminologique dénote´ par le terme est alors construit par l'ontologue. La phase de conceptualisation, passant par ces fiches, reste entièrement manuelle mais elle est assistée : l'ontologue dispose de différentes vues sur le matériau textuel et Terminae offre une traçabilité entre les concepts de l'ontologie en cours de construction et le corpus (voir figure II.14). Cette approche de la conceptualisation n'impose pas de stratégie de construction a contrario de la conceptualisation "guidée" d'OntoGen qui impose de fait une stratégie de construction top-down [23].

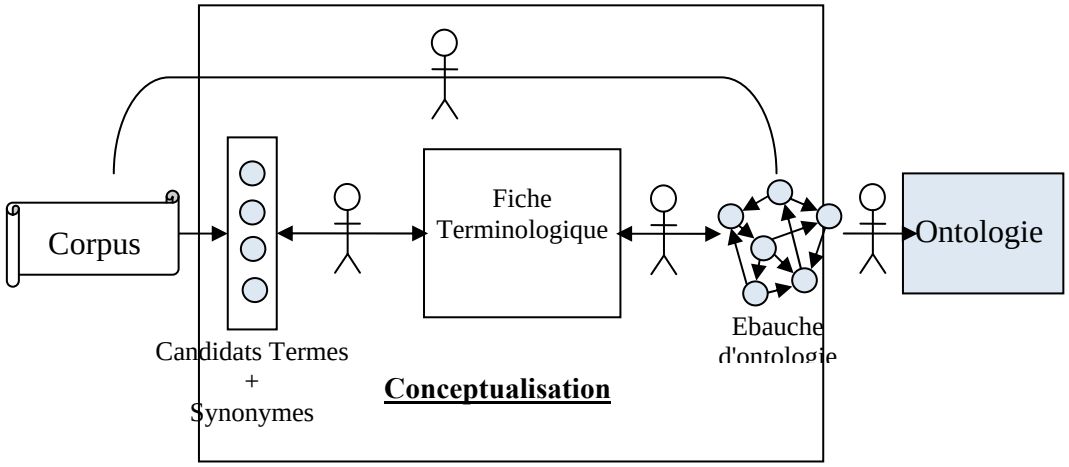


Figure II.14: L'architecture de Terminae.

II.9. Conclusion

Nous avons présenté en détaille dans ce chapitre les méthodes de construction des ontologies, et aussi nous avons présenté des méthodes un peu particulier, celles qui offrent un aide semi-automatique pour l'extraction de connaissance. Les remarques les plus importants que nous pouvons conclure sont :

- Aucune de ces méthodes n'est meilleure que les autres.
- Il existe plusieurs points communs entre ces méthodes, les étapes dans les différents méthodes peuvent être les mêmes mais nommés par des noms variés, ou bien être déférents dans l'ordre d'exécution, ou encore peuvent également être de granularité différents.
- Le processus de construction d'ontologie est un peu analogique au processus dans le domaine de génie logiciel.

Mais quelque soit la méthodologie adoptée, le processus de construction d’une ontologie est une collaboration qui réunit des experts du domaine de connaissance, des ingénieurs de la connaissance, et les futurs utilisateurs de l’ontologie. Cette collaboration ne peut être fructueuse que si les objectifs du processus ont été clairement définis, ainsi que les besoins qui en découlent.

A la fin de ce chapitre nous avons étudié quelques systèmes de construction d'ontologie à partir de texte, parmi les quels DYNAMO est un système multi-agents pour construire d'ontologie à partir de corpus textuel.

Dans le chapitre suivant nous allons développer une approche de construction d'ontologie en basant sur les méthodes et les méthodologies présentées dans ce chapitre, ainsi que nous utilisons cette approche pour la conception d'un système multi-agents pour le processus de construction d'ontologie à partir de texte.

Chapitre III

Une approche basée agent pour le processus génération d'ontologie de domaine

III.1. Introduction

Comme nous avons vu les ontologies ont pris une place très importante dans le domaine de web sémantique, c'est pour cela plusieurs techniques et méthodes sont apparues pour le développement des ontologies, le principal obstacle pour la construction d'ontologie est la grande quantité d'informations disponible pour chaque domaine de connaissance, et pour affronter ce problème les chercheurs ont pensé d'automatiser le processus de construction d'ontologie.

Ontology Learning est l'axe de la recherche qui est apparue récemment et traite la construction d'ontologie à partir de texte, et dans cet axe le domaine de traitement automatique des langages naturelles (TALN) occupe la base de cet axe, parce que avant d'essayer d'extraire la connaissance à partir de texte, on doit traiter ce texte et le préparer afin de trouver les éléments les très importants dans cette énorme quantité d'informations syntaxiques *(les termes)*, et pour que les informations syntaxiques deviennent de la connaissance, il existe plusieurs techniques et algorithmes, parmi les quels on cite *FCA(Formal Concept Analaysis)* et *RCA(Formal Concept Analaysis)* que nous avons utilisé dans notre travail.

Dans le présent chapitre, nous décrivons la méthodologie qui est un ensemble des étapes à suivre durant le processus de construction d'ontologie, ensuite, les deux algorithmes : *Formal Concept Analysis* qui sert a construire un treillis de concepts formels à partir d'un ensemble de termes et Relation Concept Analysis qui sert a identifier les relations entre ces concepts. Finalement, après une présentation détaillée de notre système multi agents pour le processus de construction d'ontologie à partir de texte, on va discuter et analyser un ensemble de résultats.

III.2. Méthodologie

Nous adoptons dans notre approche une combinaison entre les méthodes et les méthodologies que nous avons vu jusqu'à maintenant, alors on va choisir pour chaque étape dans le processus de construction d'ontologie, une étape que nous la voyons appropriée a nos besoins. Ces étapes sont expliquées en dessous (voire la figure III.1):

Etape 1 : Considérer comme une étape de préparation, dans cette étape on va identifier les objectifs de notre ontologie, élaborer un ensemble de questions des compétences et sélectionner les ressources qui nous allons les utilisés durant notre processus :

1.1 **: Identifier les objectifs de l'ontologie**

Dans cette étape on va répondre aux questions suivantes :

- Pourquoi cette ontologie sera construite?
- Quels sont les futurs utilisateurs de l'ontologie et en quoi elle sert?
- Quels sont les termes utilisés dans le domaine d'application?

1.2 **: Elaboration des questions de compétence**

Elaborer des questions en langage naturel, qui seront répondre par notre ontologie, ces questions ne doivent pas être des requêtes de recherche simple pour capturer mieux la connaissance du domaine.

1.2 **: La sélection des ressources :** On sélectionne un ensemble des ressources textuels qui sert comme une base pour notre futur ontologie.

Etape 2 : Construction d'un glossaire de termes : Construire le glossaire des termes qui identifie l'ensemble de termes à inclure dans l'ontologie, leur définition en langage naturel, leurs synonymes et acronymes s'ils existent.

Etape 3 : Construction des hiérarchies de concepts : Construire des hiérarchies de concepts pour classifier les concepts. Le résultat de cette tâche pourrait être un ou plusieurs hiérarchies où les concepts sont classifiés.

Etape 4 : Construction diagramme de relations binaires : Construire les diagrammes de relation binaire pour identifier les relations entre les concepts de l'ontologie, ces relations peuvent être avec des concepts d'autres ontologies.

Etape 5 : Construction d'un dictionnaire de concepts : Construire le dictionnaire de concepts, qui inclut les instances, les attributs, et les relations pour chaque concept.

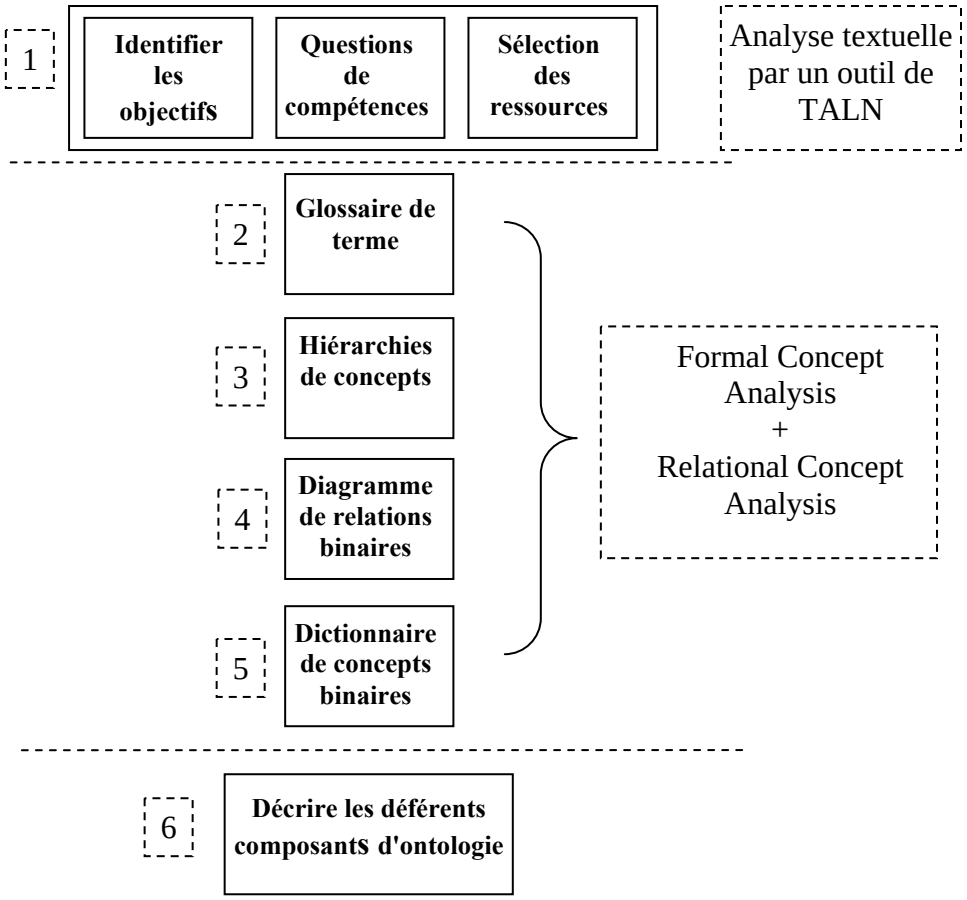


Figure III.1: Les étapes de la méthodologie proposée.

Etape 6 : Décrire les différents composants d'ontologie : tous les composants d'ontologie seront décrites en logique de description.

III.3. Système Multi Agents pour le processus de construction d'ontologie de domaine

Notre système est décomposé en trois sous-systèmes, un sous system pour l'analyse syntaxique, un sous-système pour la construction d'ontologie et le dernier sous-système pour la génération d'ontologie formelle. Ces sous-systèmes sont organisés d'une façon tel que chacun utilise les sorties du précèdent comme des entrées (le figure III.2).

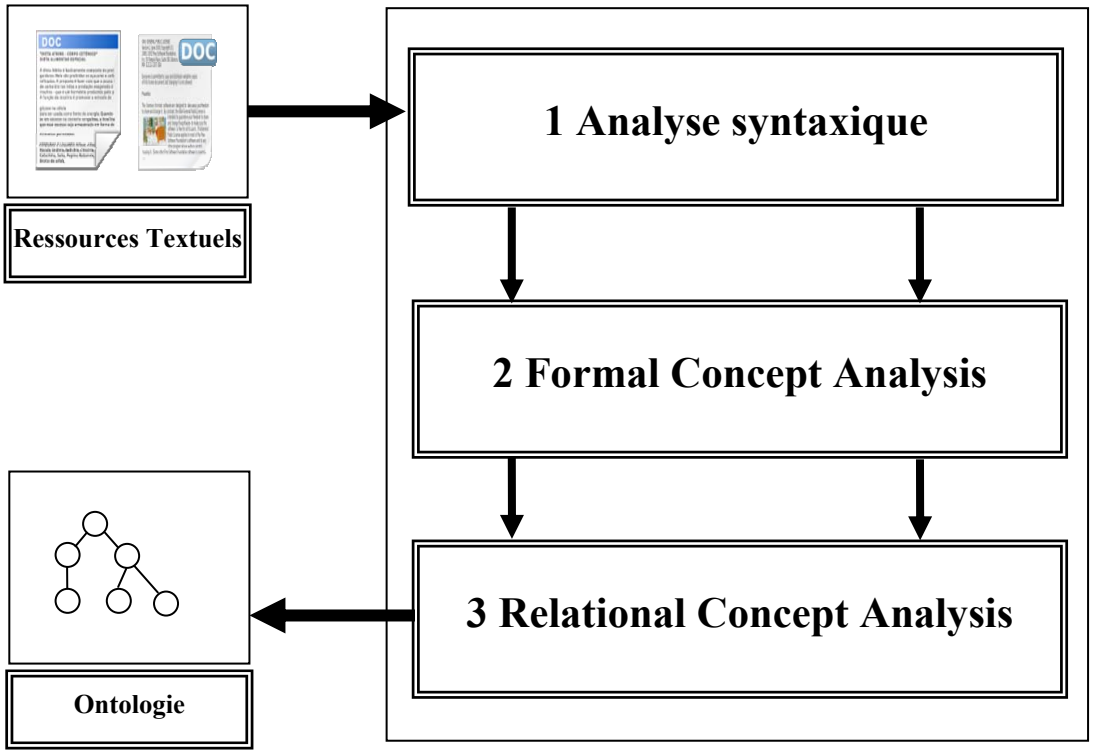


Figure III.2: Les principales étapes de notre système.

Dans ce travail on va essayer de faire une combinaison de ces méthodes et méthodologies que nous avons vu, afin d'arriver à une méthode qui répond à nos besoins.

On va commencer par l'identification des objectifs afin d'essayer de limiter le domaine d'ontologie, puis on va préparer un ensemble de questions de compétences qui peuvent servir comme un critère d'évaluation d'ontologie résultat. Après avoir choisir le domaine d'ontologie on doit collecter un ensemble de ressources qui sont essentiellement des corpus textuels.

III.3.1 Analyse syntaxique

Le rôle de sous système est d'analyser les ressources textuelles afin d'extraire les informations linguistiques les plus utiles possible pour les transformer en des informations conceptuelles à la fin des étapes prochaines.

Pour notre travail nous avons choisis *The Stanford Parser*, un outil de traitement automatique du texte fournit sous forme d'une bibliothèque java développé par le groupe de Stanford pour le traitement du langages naturels *(The Stanford Natural Language Processing Group)* dans l'université de Stanford **[**15**].**

Cet analyseur syntaxique représente les relations grammaticale dans une phrase par un ensemble des dépendances entre les mots de cette phrase pour mieux simplifier la représentation de ces relations grammaticales et les rendre plus compréhensibles par qui n'ont pas une expertise linguistique, les dépendances sont tous (55 dépendances dans la version courante) des relation binaires qui relient *un gouverneur* et *un dépendant*, pour mieux comprendre les Stanford dépendances on donne cet exemple fournis par le groupe de développement [15]

Bell, based in Los Angeles, makes and distributes electronic, computer and building products.

Après avoir exécuter l'analyseur sur cette phrase on obtient l'ensemble des dépendances suivants (figure III.3) :

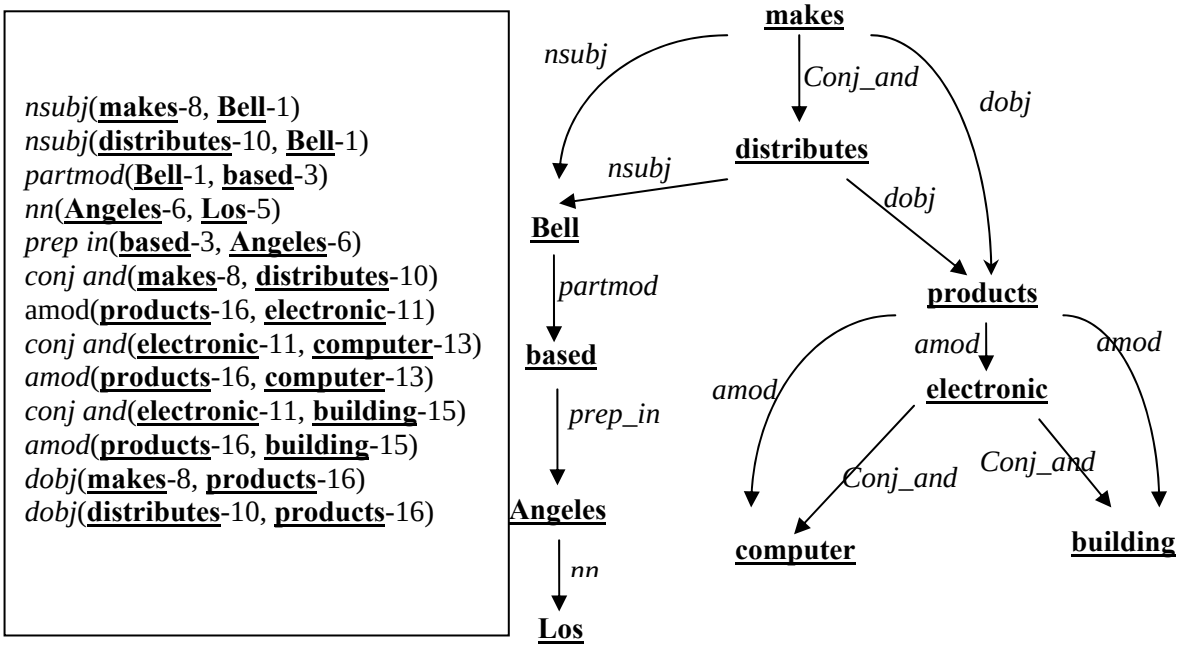


Figure III.3 : Exemple de phrase analysée par Stanford Parser.

Tel que *nsubj* (Nominal SUBJect) : Un sujet nominal est un syntagme nominal qui est le sujet d'une clause syntaxique.

La définition de tous les (55) dépendances peut être consultée dans le site de groupe [15].

III.3.2. Extraction des Concepts (Formal Concept Analysis)

Les méthodes et les techniques pour dériver semi automatiquement une hiérarchie de concepts sont regroupés en deux classes : les méthodes basées sur la similarité et les méthodes basées sur la théorie des ensembles, parmi la deuxième classe on va présenter une approche mathématique qui s'appelle FCA.

L'Algorithme FCA : Naïve Code	
Input :	Contexte Formelle *(G, M, I)*
Output :	Ensemble de concepts formels {(Oi, *Ai),..., (Oq, Aq)}*

C: =0
If (|G|<|M|)
{
For each $O \subseteq G$: *C := C U {{O'', O')}*
}
Else
{
For each $A \subseteq M$: *C := C U {{A', A'')}*
}
Return C

L'Algorithme FCA : Naïve Code

Formal concept analysis peut être considéré comme une technique insupervisée, elle manipule des objets et un ensemble d'attributs appeler respectivement *les objet formels* et *les attributs formels*. La relation entre ces objets et attributs s'appelle *la relation d'incidence*, et elle représente l'information qu'un attribut est valide (vrais) pour un objet.

Les objets, les attributs, et la relation d'incidence sont considérés comme l'entrée d'FCA et peut être représenté sous forme d'un graphe bipartite ou bien une matrice, et on l'appelle *Formel Contexte* [05] [22].

Définition 1 : Formal Context (Formel Contexte)

Une triple *(G, M, I)* est appelé Formel Contexte si et seulement si *G* et *M* sont des ensembles et $I \subseteq G \times M$ est une relation binaire entre les éléments de *G* et *M*. les éléments de *G* sont appelés des objets, et celle de *M* sont appelés des attributs et *I* la relation d'incidence [05], et on a :

Pour : $O \subseteq G$ on définit $O' := \{m \in M / \forall g \in O : (g,m) \in I\}$

Et pour : $A \subseteq M$ on définit $A' := \{g \in G / \forall m \in A : (g,m) \in I\}$

Définition 2 : Formal Concept (Formel Concept) [05] [22] **:**

Un paire *(O, A)* est un formel concept de *(G, M, I)* si et seulement si : $O \subseteq G$, $A \subseteq M$, $O' = A$ *et* $O = A'$.

Pour un formel concept *(O, A), O* est appelé *extension (extent)* et *A* est appelé *intention (intent).*

On peut maintenant définir un ordre entre les concepts formels comme suite :

$(O_1, A_1) \leq (O_2, A_2) \Leftrightarrow O_1 \subseteq O_2 (\Leftrightarrow A_2 \subseteq A_1)$

Le formel contexte *(G, M, I)* et l'ordre entre les concepts forme le **Treillis de concepts formels** noté par $(\beta(G, M, I), \leq)$

III.3.4. Extraction des relations (Relational Concept Analysis)

Relational Concept Analysis RCA est une extension de *Formal Concept Analysis* FCA, son but est d'extraire des relations binaires entre les concepts formels qui forment le treillis de concepts construit par FCA, et pour atteindre ce but RCA se base sur les relations entre les objets (respectivement les attributs) afin d'arriver a des relations entre les concepts. Le problème major dans cette approche est les quantificateurs, c'est-à-dire à chaque fois qu'on veut généraliser une relation on doit vérifier si cette relation est validée pour tous les objets et les attributs de l'extension et l'intention de ce concept.

Plus formellement, RCA prend en entré une structure de donné appelé *Relational Concept Family* (RCF), RCF est un ensemble de contextes binaires $K_i = (G_i, M_i, I_i)$ et un ensemble de relations binaires $r_k \subseteq O_i \times O_j$ tel que O_i et O_j sont les ensembles d'objets de K_i (domaine de relation) et K_j (range de relation) respectivement.

Une relations peut être vue comme une attribut, son cible est un prédicat sur l'ensemble d'objets dans le range et pour cela la relation est exprimée comme une fonction : $r : O_i \rightarrow 2^{O_j}$ [20] [22].

Un nouvel attribut de la forme *r.c* est ajouté à K_i tel que *c* est le concept dans K_j.

On a parlé sur le problème des quantificateurs et comment est difficile de passer d'une relation entre objets à une autre entre concepts :

Alors un objet $O = O_i$ a une attribut de la forme $r.c$, et c est le concept dans K_j, on définit $r(O)$ tous les objets de c qui sont en relation (r) avec O

Si : $r(O) \subseteq Ext(c)$ la relation est dite universelle.

$r(O) \cap Ext(c) \neq \Phi$ la relation est dite existentielle.

Pour mieux comprendre RCA on va présenter un petit exemple expliqué dans [24], le tableau 1 présente un contexte formel qui exprime un ensemble d'article (a, b, c...l) comme des objets et leurs domaines scientifique (SE (Software Engineering), LT (Lattice Theory), MMI (Man Machine Interface)) comme des attributs. En s'appliquant FCA sur cette contexte on obtient le treillis de concepts formels dans la figure III.4.

Articles		IT	MMI	SE
	a	X		
	b	X		
	c			
	d			
	e			
	f			
	g		X	
	h			X
	i			
	j			
	k			
	l			

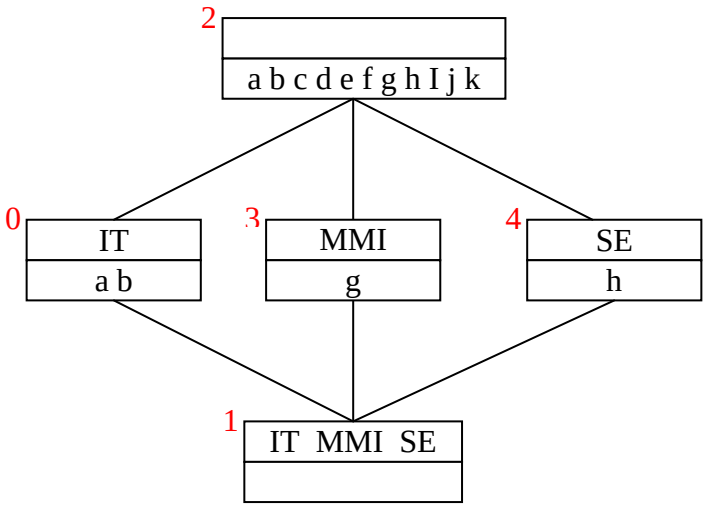


Figure III.4 : Un contexte formel et treillis formel (résultat de RCA).

Et comme nous connaissons entre les articles il y a une relation de référence, c'est-à-dire un article peut utiliser un autre article comme référence, et un article peut développer l'idée déjà exprimée dans un autre. Et la figure III.5 présente ces deux relations, la relation de référence (*Cites*) et de développement (*develops*)

Cites	a	b	g	h
a				
b				
c	X		X	
d		X		X
e				
f				
g				
h				
i	X			
j		X		
k				
l				

	Develops			
	c	d	i	j
a				
b				
c				
d				
e	X			
f		X		
g				
h				
i				
j				
k			X	
l				X

Figure III.5 : Exemple de deux relations pour FCA

On appliquant RCA sur le contexte formel de la figure III.4 et les deux relations présentées dans la figure III.5, on va généraliser les deux relations d'une relation entre les articles à une relation entre les concepts de contexte formel.

On a : *Cites*(c) = (a, g), et *Cites*(d) = (b,h), *Cites*(i) = (a) et *Cites*(j) = (b), on remarque que la relation *Cites* donne toujours des articles qui appertiennent a C2 et C0

Alors on ajoute les deux concepts formels (Cites.C2 et Cites.C0) a notre contexte et la relation *I* doit augmenter en ajoutant :

(C, Cites.C2), (d, Cites.C2), (i, Cites.C2), (j, Cites.C2), (i, Cites.C0) et (j, Cites.C0)

En appliquant le même processus pour la relation *Devlops*, le contexte formel ainsi que le nouveau treille de concepts seront comme suite :

Articles		IT	MM I	SE	Cites		Develops
					2	0	2
	a	X					
	b	X					
	c				X		
	d				X		
	e						X
	f						X
	g		X				
	h			X			
	i				X	X	
	j					X	
	k						
	l						

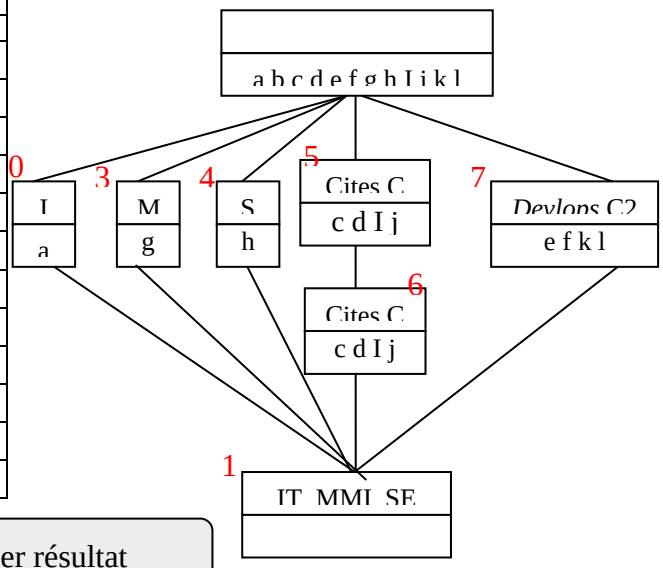


Figure III.6: FCA le premier résultat

A la fin nous avons obtenu un treille de concepts formels plus riche que le précédent, on remarque que la répétition de processus RCA sur la relation *Cites* n'ajoute rien a notre contexte, par contre la répétition sur *Develps* ajoute nouveaux concepts et attributs, Alors on répète l'application de RCA jusqu'à ce que le processus s'arrête automatiquement.

III.3.5. Architecture de Système :

Un système multi-agents est un ensemble d'agents qui évolue dans un environnement commun. Dans [Weiss, 1999], Gerhard Weiss définit l'intelligence artificielle distribuée comme l'étude, la conception et la réalisation de systèmes

multi-agents qu'il présente comme des systèmes dans lesquels des agents intelligents interagissent et poursuivent un ensemble de buts ou réalisent un ensemble d'action. Notre système se compose de six types d'agent, qui communiquent entre eux pour construire une ontologie à partir du texte, la figure III.7 présente le mécanisme de fonctionnement de notre système.

Le système est implémenté en utilisant la plateforme JADE (Java Agent DEvelopment Framework) et la technologie FIPA(http://www.fipa.org/) pour l'échange de message, tel que la seule caractéristique de nos agents est un identifiant (unique dans le système) utilisé pour les envois de messages FIPA, la description de tout les agents et le mécanisme d'échange de messages est expliqués en dessous.

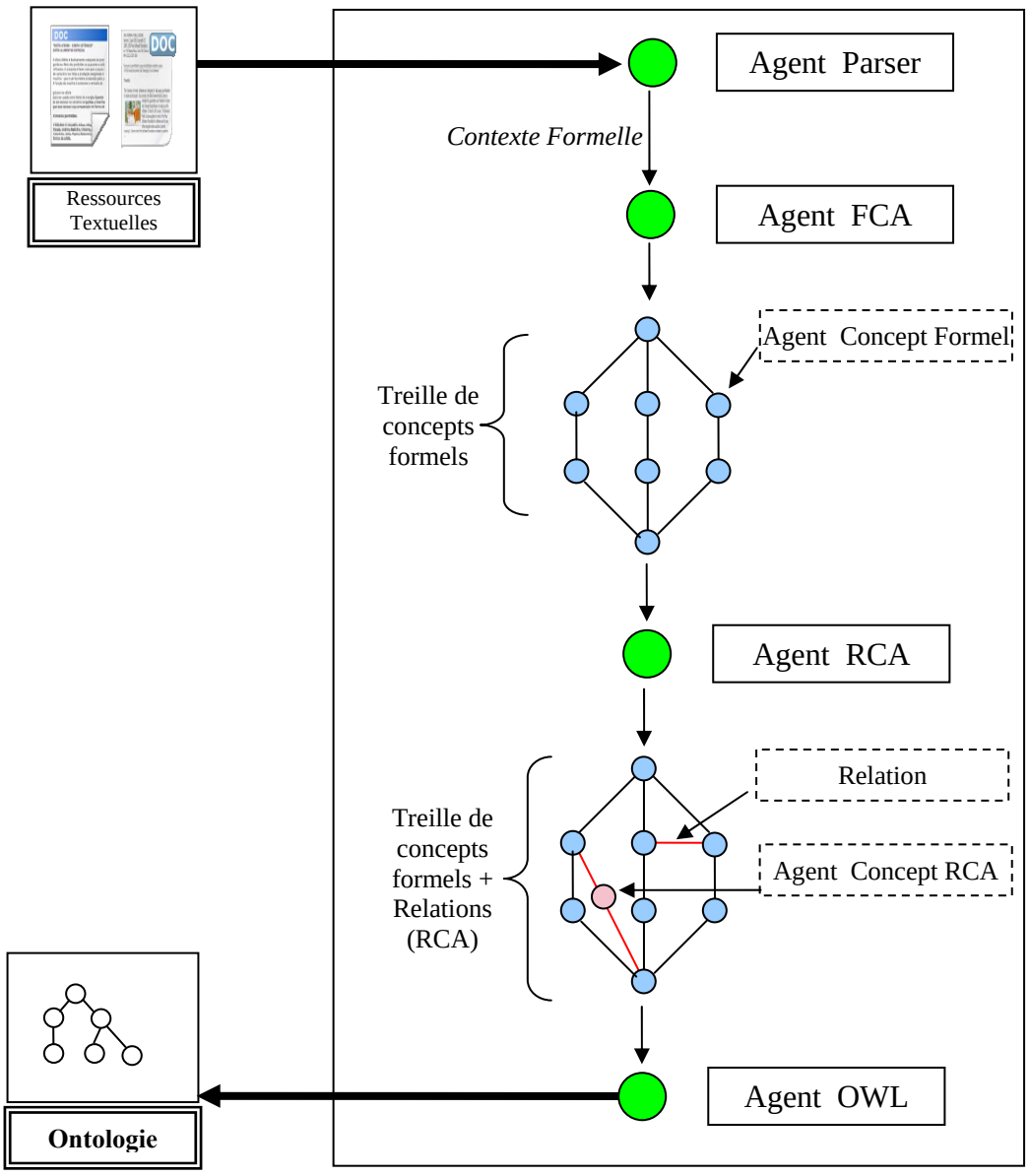


Figure III.7 : Architecture de Système.

1. Agent Parser

Le rôle de l'agent : le rôle de cet agent est d'analyser en ensemble de documents textuels (Corpus de Texte), afin d'extraire un ensemble d'information syntaxique et les insérer ensuite dans une base de données syntaxique, cet agent utilise *The Stanford Parser* [15] pour accomplir son rôle.

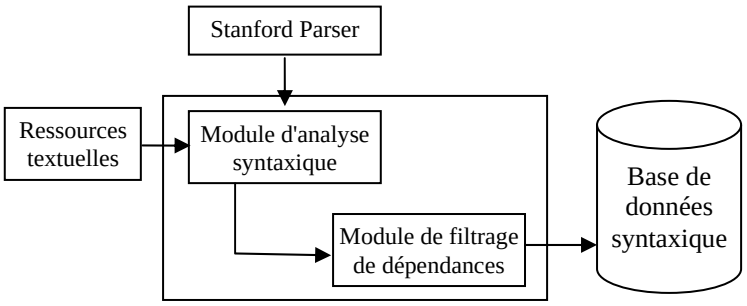


Figure III.8 : les composants de l'Agent Parser

2. Agent FCA

Le rôle de l'agent : Cet agent applique l'algorithme FCA sur le contexte formel (qui est dans la base syntaxique) pour construire un treille de concepts formels, chaque concept dans ce treille est un agent de type *Agent Concept Formel*. L'agent utilise la bibliométrique FCA pour accomplir son rôle.

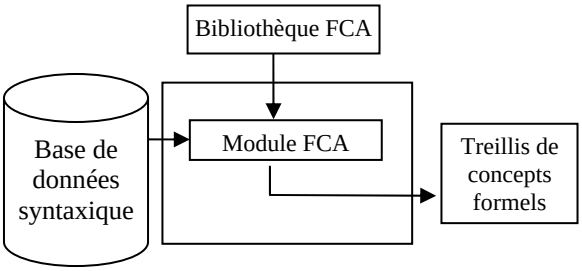


Figure III.9 : les composants de l'Agent FCA

3. Agent Concept Formel

Le rôle de l'agent : cet agent représente un concept formel, il possède une représentation sur l'ensemble d'objets, attributs, les voisins supérieurs et les voisins intérieurs de ce concepts, son rôle est de répondre aux requêtes des autres agents, comme la requête "quels sont tes voisins supérieurs"?

Chaque agent Concept a un connaissance sur lui-même c a d : les objets et les attributs qu'il regroupe, et un connaissance sur l'environnement : les autres agents concept voisins (super voisins et sous voisins).

4. Agent RCA

Le rôle de l'agent : cet agent utilise les relations syntaxiques entre les objets des concepts formels (qui sont dans la base syntaxique) pour essayer de trouver des relations entre les concepts selon le mécanisme déjà expliqué**.**

Ensuite il crée des agents de type *Agent Concept RCA* nouveaux concepts appellent ces relations s'elles existe bien sur.

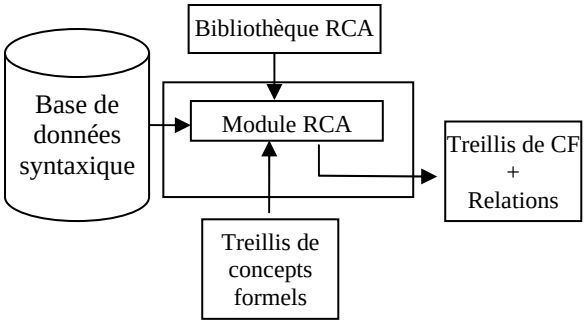


Figure III.10 : les composants de l'Agent RCA

5. Agent Concept RCA

Cet agent est similaire au Agent Concept mais seulement est créé par l'agent RCA.

6. Agent OWL

Le rôle de l'agent : cet agent utilise une bibliothèque de *protégé* pour générer une ontologie formelle en OWL.

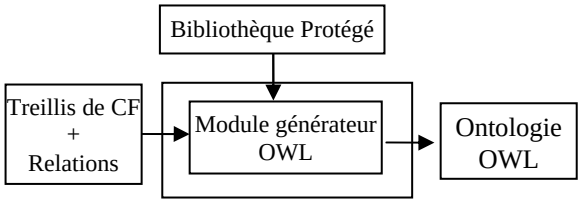


Figure III.11 : les composants de l'Agent OWL

Le fonctionnement de système (figure III.7) et le mécanisme d'envoie de messages (figure III.14) est expliqué dans la suite :

- Au début l'*Agent Parser* se lance et lance avec lui le *Stanford Parser*, ensuite il analyse le document qui contient le corpus textuel, il extrait des informations générales sur le document comme la taille, le nombre des phrases et le nombre de caractères dans chaque phrase, et des informations syntaxique sont les dépendances Syntaxique, puis l'*Agent Parser* filtre les dépendances pour trouver

les quels réponds a nos besoin pour former le contexte et il l'enregistre dans un fichier XML.

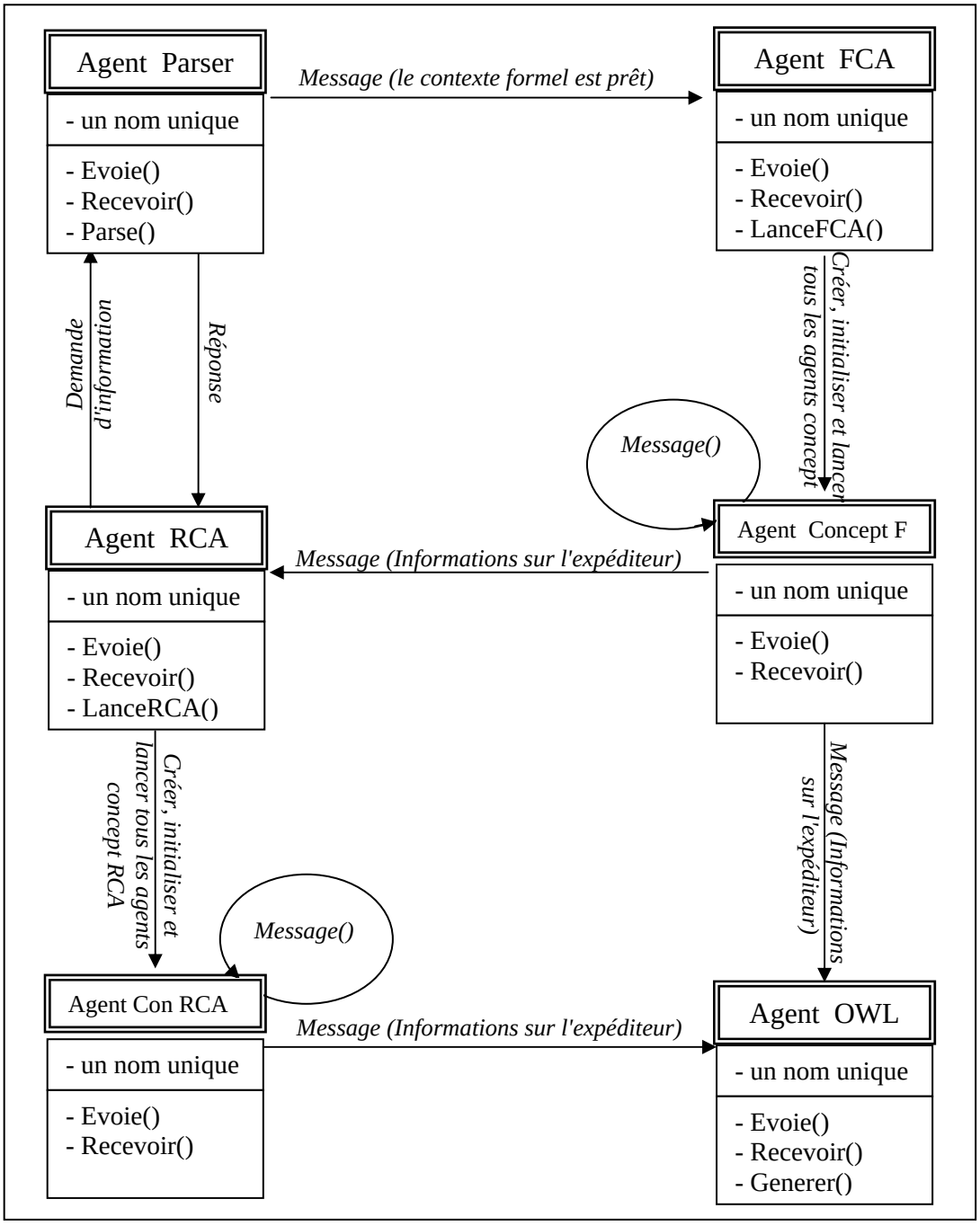


Figure III.14 : Diagramme d'échange des messages dans le Système.

- Après que l'*Agent FCA* reçoit le message de l'*Agent Parser* qu'il le dit que le contexte formel est prêt, il lance l'algorithme FCA qui va construire un treillis de concepts formels, et l'*Agent FCA* va créer un *Agent Concept FCA* pour chaque

concept formel, et il l'initialise tel que chaque agent concept sait les objets, les attributs, les voisins hauts et les voisins bas appropriés pour lui.

- Après la création de dernier agent concept, l'*Agent RCA* se lance, il demande des informations syntaxiques de l'*Agent Parser* concernant des relations entre les objets pour les utiliser dans l'algorithme RCA. Et comme résultat il créer et initialise aussi des Agent représentent des concept formels s'appellent des *Agent Concept RCA*
- Â la fin chaque *Agent Concept FCA* et *Agent Concept RCA* envoient des informations concernant lui-même à l'*Agent OWL* et ce dernier va générer une ontologie OWL compatible avec l'éditeur des ontologies *protégé OWL*.

III.4. Conclusion

Nous avons abordé dans ce chapitre, la problématique de la construction automatique d'ontologies à partir du texte. Pour cela, nous avons déduit une méthodologie de développement d'ontologie et implémenter un système multi agents pour résoudre ce problème en suivant notre méthodologie.

Notre méthodologie constitue en grosso modo de trois niveaux : le niveau syntaxique, sémantique et formel,

Le niveau syntaxique : consiste à utiliser une technique de traitement du langage naturel pour extraire des informations syntaxiques, et nous avons remarqué qu'il existe plusieurs technique pour le faire, et le choix d'une telle technique a un grand effet sur le résultat final dans notre travail. Et on peut voir facilement que ces techniques extraient des relations syntaxiques entre les entités de la même phrase, c'est-à-dire, c'est un peu difficile de découvrir une relation entre un terme dans le début et un autre terme dans la fin du document.

Le niveau sémantique : plusieurs techniques et méthodes sont disponibles pour passer de niveau syntaxique au sémantique, parmi lesquels on a choisit la combinaison de deux algorithmes FCA (Formal Concept Analysis) et RCA (Relational Concept Analysis). FCA consiste à regrouper un ensemble d'objets qui ont un ensemble d'attributs communs pour construire des concepts formels. La première remarque qu'on peut la voir est que ces concepts formels n'ont pas des noms génériques.

Le niveau formel : à ce niveau il existe plusieurs moyens pour coder une ontologie, mais le OWL est le langage le plus récent et le plus utilisé, et bénéfice de la disponibilité de plusieurs outils graphiques qui rend le processus de création d'ontologie plus facile, ces outils offre avec eux des API que nous pouvons les appeler pour créer une ontologie, parmi lesquels on cite le célèbre *protégé* avec ces différents version.

Le chapitre suivant est consacré pour tester notre système et évaluer les résultats obtenus.

Chapitre IV

Implémentation et Evaluation

On suivant notre méthodologie, nous avons implémentés un système multi agent pour la construction automatique d'ontologie à partir du texte, sous la célèbre plateforme de développement JADE,

IV.1. Motivation Technique

JAVA : c'est le langage la plus utilisé dans les plateformes de développement des systèmes multi agents parce qu'il se caractérise par : Simple, Sécurité, portable (Écrire une fois, utiliser partout) et multitâches. Nous avons utilisé la version de JDK1.6

NetBeans : est un environnement de développement intégré (EDI), placé en *open source* par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

JADE : JADE (Java Agent Development Framework) est un plateforme pleinement implémenté en Java. Il simplifie la mise en œuvre des systèmes

multi-agents à travers un middle-ware qui est conforme aux spécifications FIPA et à travers un ensemble d'outils graphiques qui prend en charge les phases de débogage et de déploiement. La plate-forme agent peut être distribué entre les machines (qui n'est même pas besoin de partager le même système d'exploitation) et la configuration peut être contrôlée via une interface graphique utilisateur distante. La configuration peut être encore changé au moment de l'exécution par des agents mobiles d'une machine à une autre, en cas de besoin. JADE est complètement implémenté en langage Java et l'exigence minimale du système est la version 1.4 de java (l'environnement d'exécution ou le JDK).

IV.2. La mis en ouvre du Système

Comme nous avons dit, notre système de construction d'ontologie à partir du texte commence par choisir une ressource textuelle. La figure III.15 représente l'interface principale de ce système, si nous choisissons le fichier textuel le système commence par nous donner la taille et le nombre de phrase dans ce fichier. Jusqu'à ce point notre système est prêt, mais en réalité nous n'avons pas lancé aucun agent pour le moment.

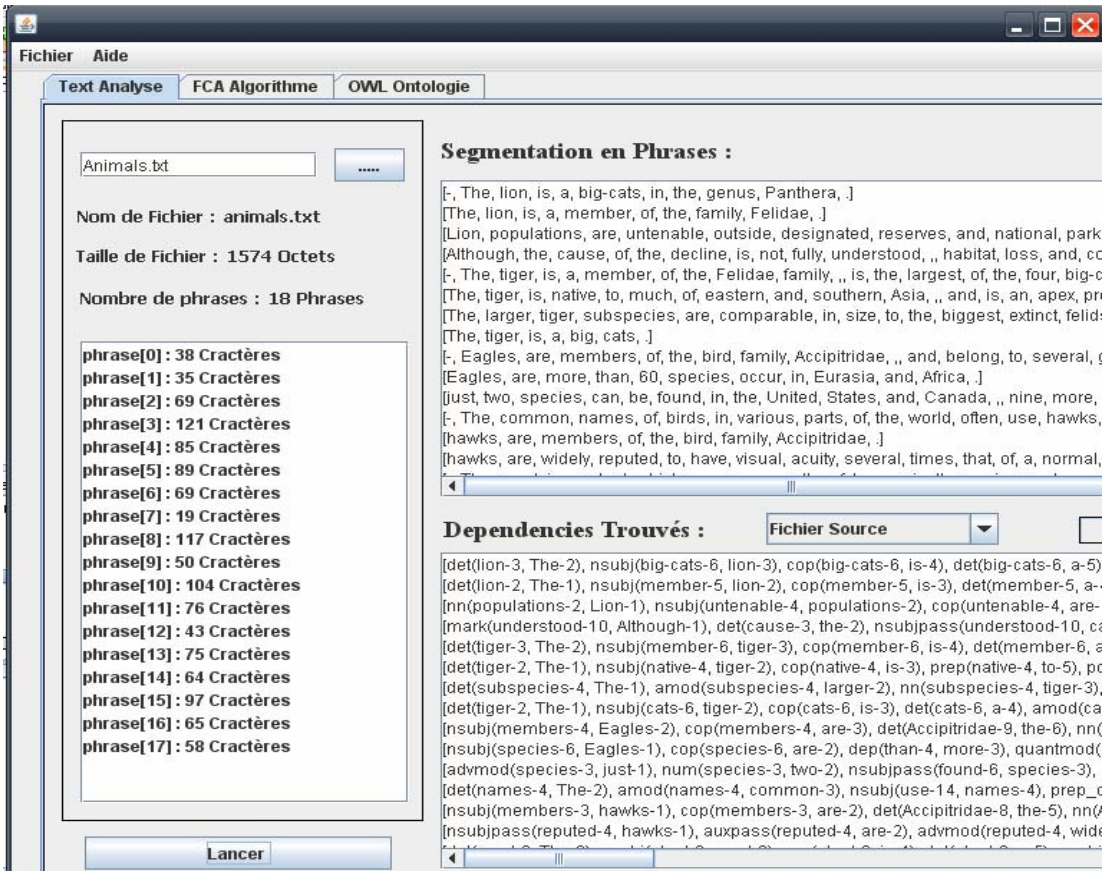


Figure IV.1 : L'interface principale

Pour lancer le système multi-agents nous devons cliquer sur le buttons "*Lancer",* le première agent créé est l'*Agent Parser*, qui doit segmenter chaque phrase dans le fichier source en des entités lexicales et extraire les dépendances syntaxique entre ces entités, plus le nombre de phrases totale, l'interface montre aussi le nombre de caractères dans chaque phrase et une vue échangeable entre : le fichier source, les dépendances syntaxiques et les dépendances syntaxiques sous forme d'arbre.

Chaque fois que l'*Agent Parser* trouve un dépendance syntaxique il l'envoie au deuxième agent dans le système, l'*Agent FCA,* qui va examiner chaque dépendance pour construire un pair formel de forme *(Objet Attribut),* après examiner tous les dépendances, l'*Agent FCA* constitue une base formelle de ces pairs qui sert en suite comme une base pour lancer l'algorithme FCA.

Le résultat de l'application de cet algorithme, est une hiérarchique de concept formel ou treillis formel, tel que chaque concept formel est un regroupement des objets qui ont un ensemble d'attributs communs.

La figure IV.2 représente le graphe de treillis formel engendré par

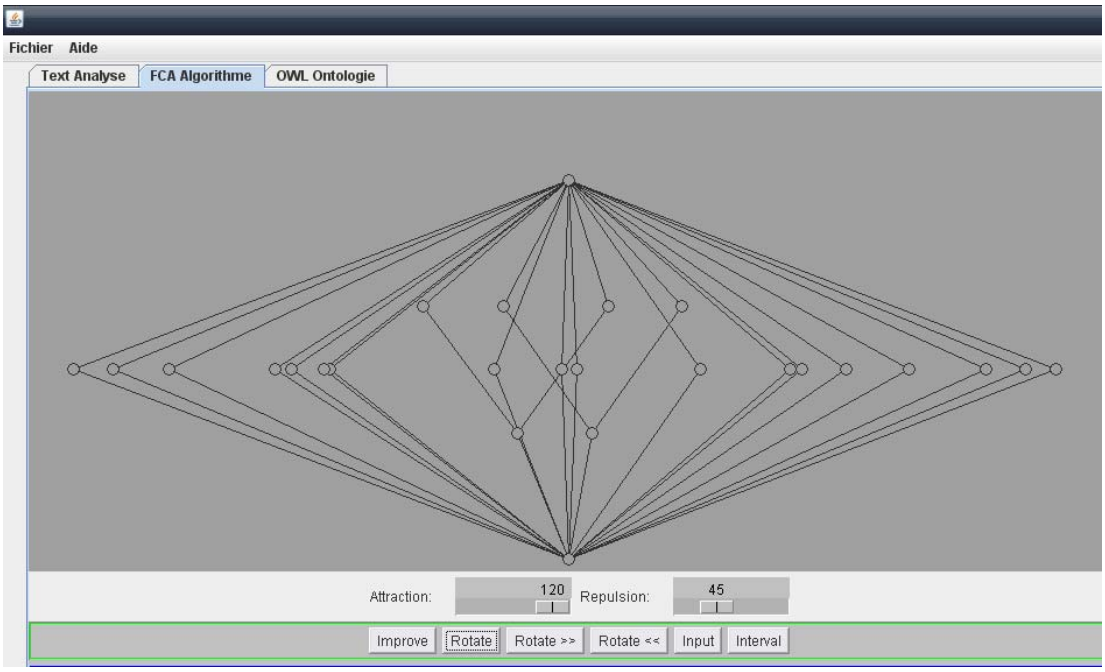


Figure IV.2 : Le treillis de concepts formels

Comme vous voyez dans la figure, chaque nœud est un concept formel et en même temps un agent indépendant qui a son propre nom (Agent AID) dans le système et les arcs sont des relations de *submission* entre les concepts.

Après le lancement de *Agent RCA*, des nouvelles relations sont ajoutées à notre treillis de concepts formels.

Ensuite l'*Agent OWL* qui est depuis le début en état d'attendre les informations de chaque concept formel FCA+RCA, commence a codé notre ontologie en OWL.

Et le résultat sera comme le montre la figure

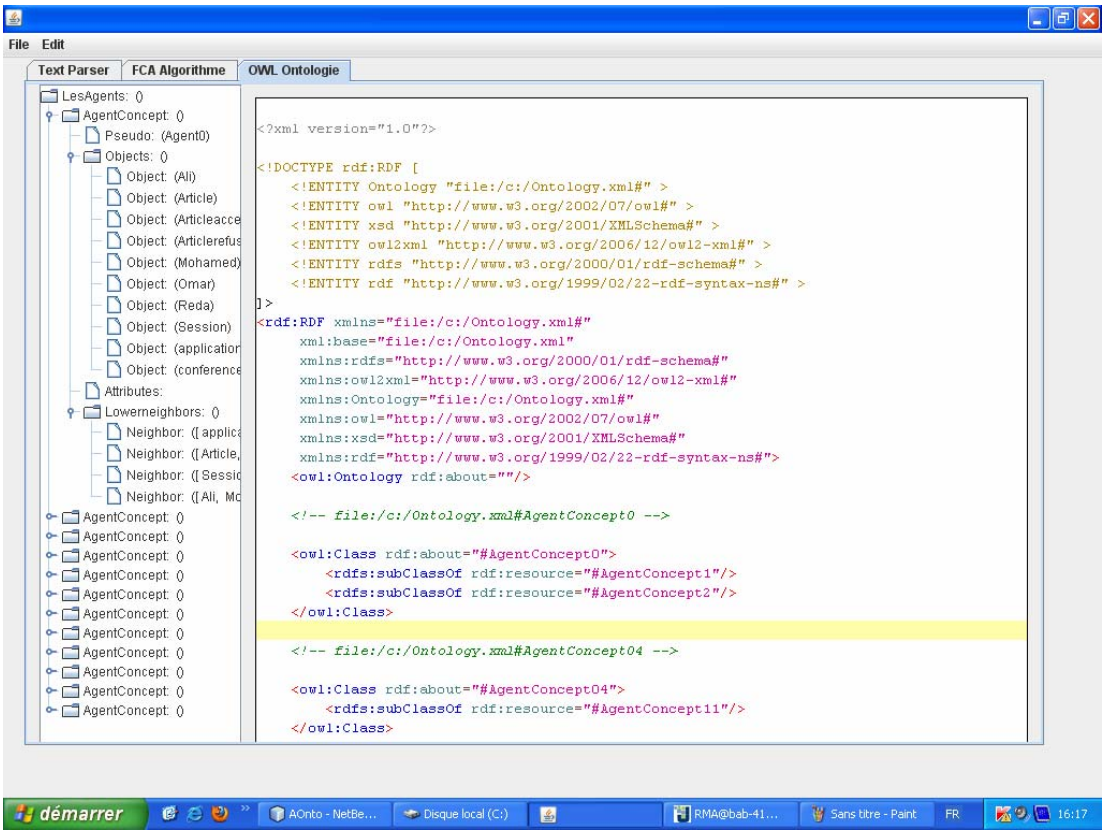


Figure IV.3 : L'ontologie en OWL.

Le coté gauche dans la figure représente l'hiérarchie des concepts en montrant pour chaque concept, ses objets, attributs et ses concepts voisins.

Après avoir tester le système multi agent sur un exemple de corpus textuel représente le domaine des animaux, nous avons obtenu le treillis de concepts formels présenté par la figure IV.2 (ce treillis formel générer en utilisant l'outil de *Ralph Freese* développé au sein de l'université de ***Hawaii).***

Il faut â noter que le processus de construction d'ontologie n'est parfaitement automatique, parce que l'ontologiste doit intervenir pour :

- donner des nom aux concepts et relations, et comme nous pouvons voir dans le figure IV.4 *(lion, tiger)* peut être nommer par le nom *Felidae*.

- pour supprimer les concepts qui véhiculent des informations n'appartiennent pas à notre domaine.

On prends en considération ces derniers tâches et on ajoutant les relations entre les concepts qui figure dans la fenêtre qui montre le fichier OWL, on peut avoir le résultat final montré par la figure IV.4.

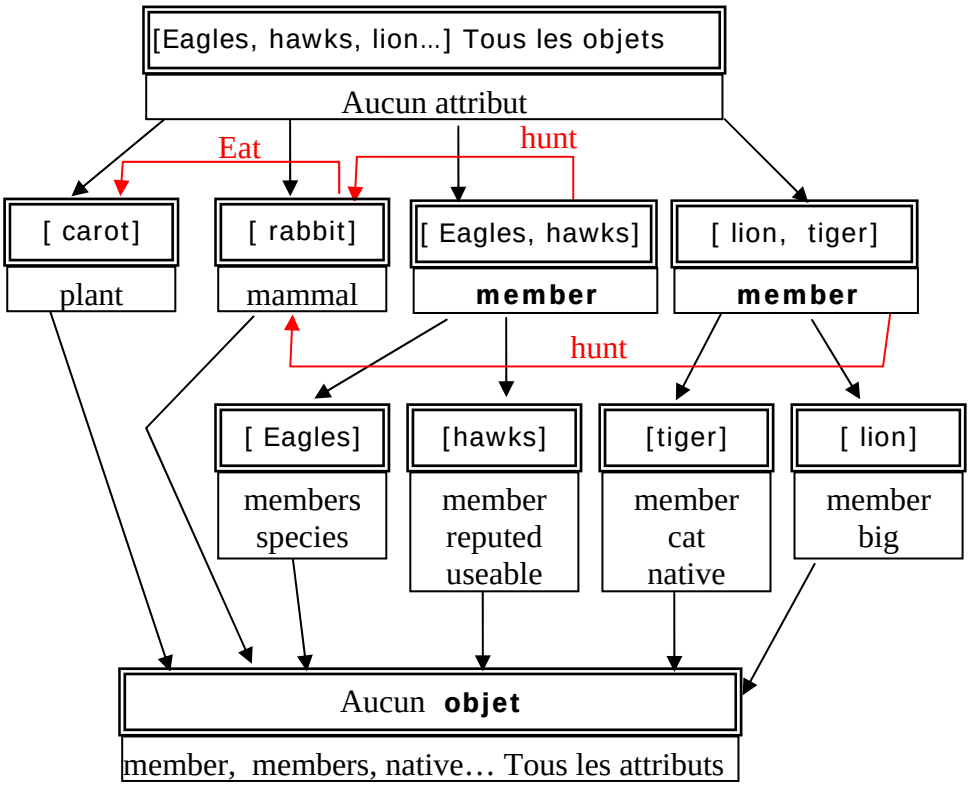


Figure IV.4 : Ontologie résultat pour le domaine des animaux

Un autre résultat qui nous avons obtenu après tester le système sur un corpus textuel représente le domaine de *"la communauté des chercheurs"* est présenté par la figure IV.5.

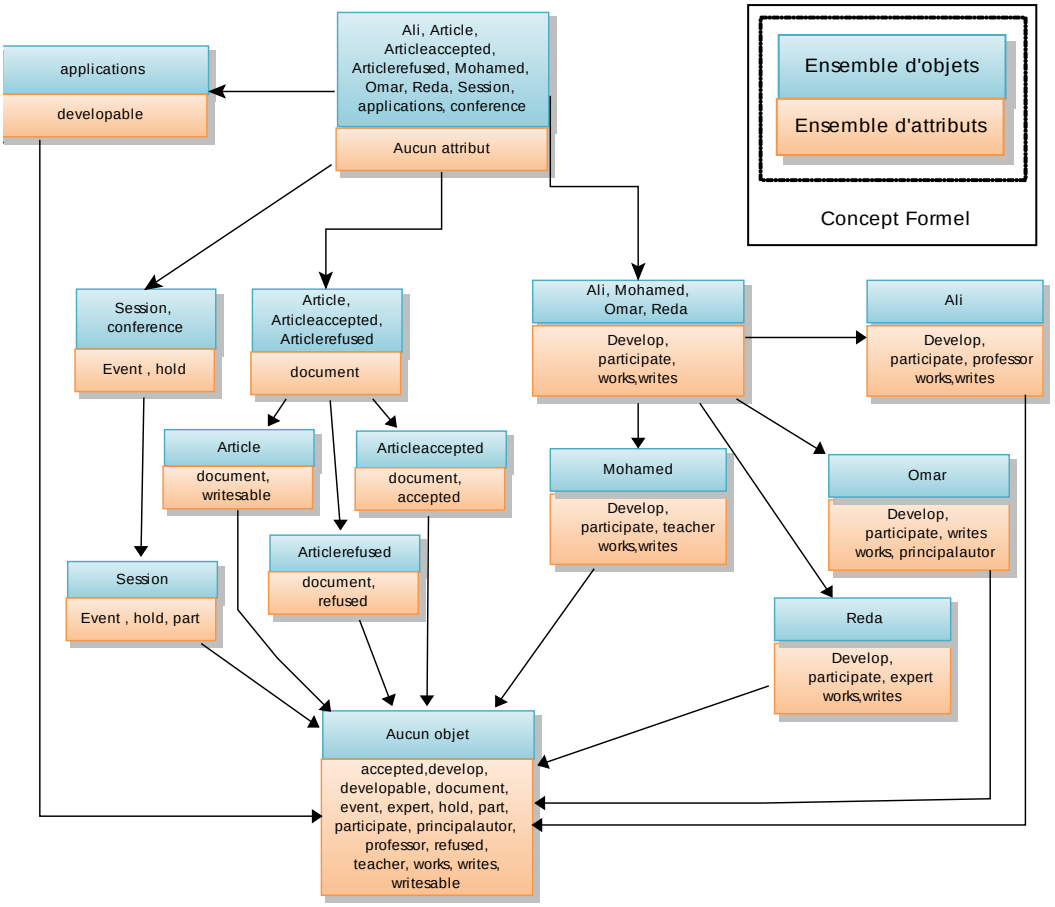


Figure IV.5 : Ontologie résultat pour le domaine de communauté des chercheurs

IV.3. La Communication entre les agents

JADE dispose d'un ensemble des outils graphiques très importants, parmi les quels, le sniffer est un outil pour suivre l'échange des message entre les agents de mon système, quand l'utilisateur décide de *sniff* un agent, le sniffer affiche chaque message reçu ou envoyé pour cet agent.

Alors si on *sniff* les agents de notre système (*Agent Parser, Agent FCA, l'Agent OWL, Agent Concept*…) et on lance le système, le résultat sera comme nous voyons à la figure IV.6.

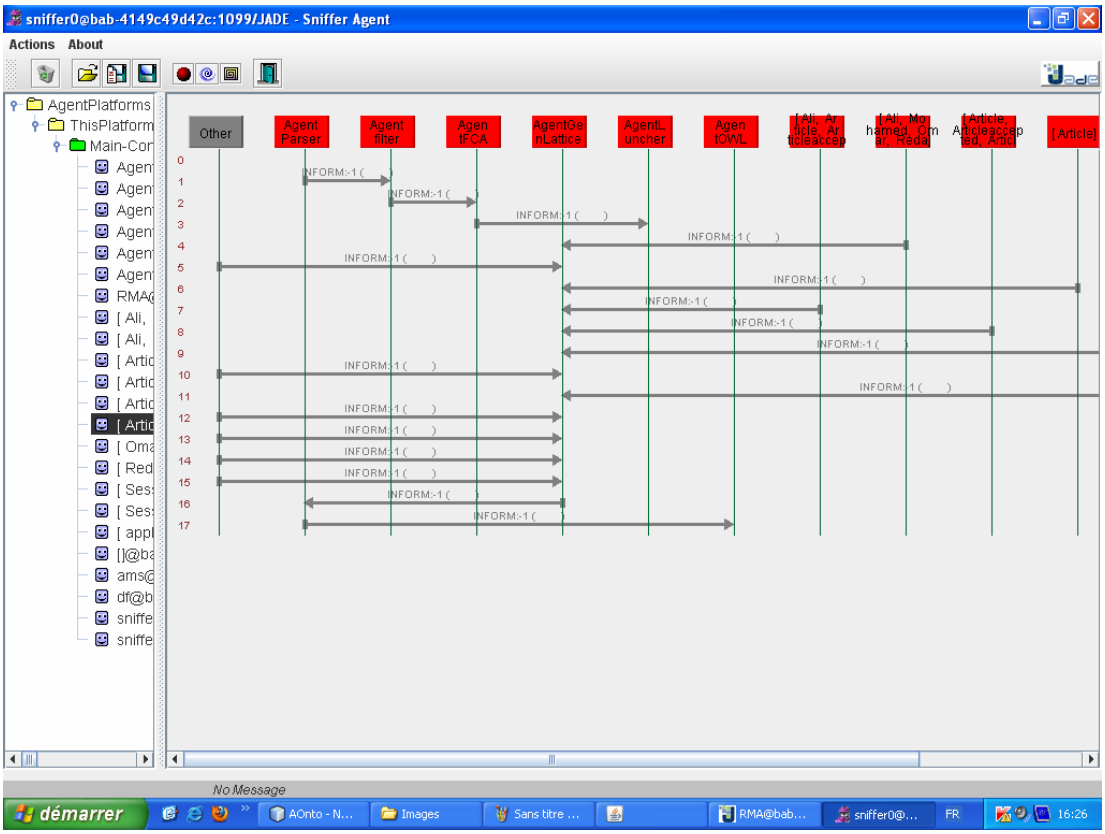


Figure IV.6: Le sniffer : échange des messages entre les agents

IV.4. Conclusion

Dans ce chapitre nous avons implémenté un système multi agents pour la construction d'ontologie à partir de texte, et pour tester ce système nous avons utiliser deux corpus textuels représentent deux domaines différents.

Nous avons vu que le processus de construction automatique d'ontologie a besoin d'intervention d'un être humain pour obtenir des résultat acceptables, et cela peut être considéré comme un barrière pour l'avancement de ce domaine.

Le choix de technologie de multi agents pour implémenter notre travail, a donné une grande facilité et souplesse pour le réaliser, parce que les systèmes multi agents offrent une grande base théorique, et même de coté implémentation, la plateforme JADE représente un ensemble d'outils qui rends la réalisation d'un système multi agents est plus facile qu'on imagine.

Conclusion Générale et perspectives

L’objectif de ce travail est la mise en oeuvre et l’évaluation d'une approche basée agent pour le processus génération d'ontologie de domaine.

Les ontologies sont devenues des entités très importantes et utilisés dans plusieurs domaines, et pour cela nous avons pensé à comment les construire ou comment rendre le processus de leur construction plus facile et rapide. Les deux termes "facile et rapide" nous a conduit vers un autre terme "automatique", et notre question a devenu *"Comment construire une ontologie automatiquement"*. L'ontologie est une représentation de connaissances, et si on veut la construire automatiquement, on ne peut pas baser sur la connaissance qui est dans les cerveaux des êtres humains. Et pour cette raison nous avons pensé à la connaissance codée sous forme textuelle. Et nous avons essayé de répondre à la question *"comment construire automatiquement une ontologie a partir du texte"*.

Au cours de ce mémoire, nous avons rappelé les différentes méthodes et méthodologies pour construire une ontologie, et malgré l'existence de plusieurs méthodes, il n y a pas une qui peut être considéré comme une méthode standard.

Dans le cycle de vie d'une ontologie, notre travail se déroule au cours de l'étape de construction, et nous avons dit qu'il existe plusieurs techniques pour construire une ontologie a partir du texte, dans ce travail nous avons choisi les deux techniques formelles : FCA (Formal Concept Analysis) et RCA (Relational Concept Analysis), et nous avons vu une description détaillée de ces deux algorithmes pour les intégrer avec un outil de traitement automatique du texte afin de générer une ontologie formelle.

Avec tous les avantages qu'ils offrent, les systèmes multi agents sont bien adéquat pour résoudre ce type de problème, alors nous avons utilisé comme nous avons vu dans le chapitre 3, la plateforme JADE pour implémenter le modèle proposé dans le même chapitre.

Avec l'envoie de message, les agents dans ce système échangent les informations nécessaires pour leur fonctionnement, la plateforme JADE offre un modèle très simple pour envoyer et recevoir des messages, ce modèle est compatible avec le ACL (Agent Communication Language) de FIPA qui rend le mécanisme d'envoie de message compréhensible par les êtres humains.

Dans la dernière partie du chapitre 3, nous avons présenté le système que nous avons réalisé ainsi que un exemple de mise en marche, en montrant le graphe des concept formels et l'ontologie OWL ainsi que l'échange de messages entre les agents visualisés par l'outil graphique *sniffer* fournit avec la plateforme JADE.

Perspective

- le système que nous avons réalisé est totalement automatique, une intervention d'un utilisateur peut être introduit à ce système par l'intégration de plusieurs techniques de construire d'ontologie a partir du texte, et le rôle de l'utilisateur dans ce cas est de choisir une combinaison de ces techniques et observer pour arriver a la meilleure combinaison.
- On peut résoudre le problème que les concepts n'ont pas des noms sémantique, par un agent supplémentaire, le rôle de cet agent est d'envoyer une requête qui contient les objets et les attributs de ce concept à une ontologie core (WorldNet) pour recevoir un nom adéquat pour ce concept.

Bibliographie

[01] A Napoli. *"Une introduction aux logiques de descriptions"* Rapport de recherche N° 3314. 1997.

[02] A Goḿez-Pérez, Mariano Ferndndez-Lṕez and Oscar Corcho *"Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web"* Springer 2004.

[03] Borst WN *"Construction of Engineering Ontologies"* Centre for Telematica and Information Technology, University of Tweenty. Enschede, The Netherlands 1997.

[04] F. Furst *"L'ingénierie ontologique"* Rapport de recherche N°02-07. 2002.

[05] Philipp Cimiano *"Ontology Learning and Population from Text Algorithms, Evaluation and Applications"* Springer 2006.

[06] Grüninger M, Fox MS *"Methodology for the design and evaluation of ontologies"* IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing 1995.

[07] Guarino N, Giaretta P *"Ontologies and Knowledge Bases: Towards a Terminological Clarification"* 1995.

[08] Uschold M, M.Grüninger *"ONTOLOGIES: Principles, Methods and Applications"* Knowledge Engineering Review. 1996.

[09] Uschold M, King M *"Towards a Methodology for Building Ontologies"* IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canadab 1995.

[10] Neches R, Fikes RE, Finin T, Gruber TR, Senator T, Swartout WR *"Enabling technology for knowledge sharing"* AI Magazine 1991.

[11] J. F. Sowa *"Conceptual Graphs for a Data Base Interface"* IBM Journal of Research and Development 1976.

[12] Fikes, R *"Multi-use Ontologies"* Stanford University 1998.
http://www.ksl.stanford.edu/people/fikes/cs222/1998/Ontologies/tsld001.htm

[13] T.R. Gruber *"Toward principles for the design of ontologies used for knowledge sharing"* Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University 1993.

[14] York Sure, Rudi Studer *"On-To-Knowledge Methodology"* University of Karlsruhe 2002.

[15] Marie-Catherine de Marneffe and Christopher D. Manning *"Stanford typed dependencies manual"* September 2008.

[16] Bill Swartout, Yolanda Gil *" EXPECT: Explicit Representations for Flexible Acquisition"* the Proceedings of the Ninth Knowledge Acquisition for Knowledge Based Systems Workshop (KAW'95) Banff, Canada, February 26-March 3, 1995

[17] Paul Buitelaar, Philipp Cimiano *"Ontology Learning and Population: Bridging the Gap between Text and Knowledge"* IOS Press 2008.

[18] KASSEL G *"OntoSpec : une méthode de spécification semi-informelle d'ontologies"* journées francophones d'Ingénierie des Connaissances (IC'2002), pages 75-87, 2002.

[19] Brachman RJ "*On the Epistemological Status of Semantic Networks"* Academic Press, London, United Kingdom 1979.

[20] Naouel Moha, Amine Mohamed Rouane Hacene, Petko Valtchev et Yann-Gael Guéhéeneuc *"Refactorings of Design Defects Using Relational Concept Analysis"* 6th International Conference, ICFCA 2008.

[21] KEVIN OTTENS *"Un SMA adaptatif pour la construction d'ontologies à partir de textes"* Thèse de doctorat en informatique, l'Universit´e Paul Sabatier de Toulouse III

[22] Verlag Shaker *"Relational Concept Analysis: Semantic Structures in Dictionaries and Lexical Databases"* (PhD Thesis), Aachen 1998.

[23] Studer R, Benjamins VR, Fensel D *"Knowledge Engineering: Principles and Methods"* IEEE Transactions on Data and Knowledge Engineering 1998.

[24] Mizoguchi R, Vanwelkenhuysen J, Ikeda M "*Task Ontology for reuse of problem solving knowledge"* Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing (KBKS'95) IOS Press, Amsterdam, The Netherlands, pp 46–57 1995.

[25] Lassila O, McGuinness D *"The Role of Frame-Based Representation on the Semantic Web"* Technical Report KSL-01-02. Knowledge Systems Laboratory. Stanford University, Stanford, California 2001.

[26] J. F. Sowa *"Conceptual Structures: Information Processing in Mind and Machine"* Addison-Wesley, 1984.

[27] Aussenac-Gilles N, Biébow B, Szulman S *"Modelling the travelling domain from a NLP description with TERMINAE"* Workshop on Evaluation of Ontology-based Tools (EON2002), Sigüenza, Spain 2002.

[28] Aussenac-Gilles N, Biébow B, Szulman S (2000a) *"Revisiting Ontology Design: A Methodology Based on Corpus Analysis"* 12th International Conference in Knowledge Engineering and Knowledge Management (EKAW'00) Springer-Verlag, Berlin, Germany 2000.

[29] Aussenac-Gilles N, Biébow B, Szulman S *"Corpus analysis for conceptual Modelling"* EKAW'00 Workshop on Ontologies and Texts. Juan-Les-Pins, France. CEUR Workshop 2000.

[30] Kietz JU, Maedche A, Volz R "*A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet"* EKAW'00 Workshop on Ontologies and Texts 2000.

[31] Jean-François Baget, Étienne Canaud, Jérôme Euzenat, Mohand Saïd-Hacid *"les langages du web sémantique"* INRIA Rhône-Alpes, LISI 2003.

[32] Z.Boufaida, cours de magister *"web sémantique"* université de Biskra 2008.

Résumé

Une ontologie est une spécification explicite d'une conceptualisation, ce terme est souvent lié avec celui de Web Sémantique qui est une nouvelle version de web actuel. Il vise à rendre les ressources du web accessibles par des agents logiciels, afin de faciliter leurs utilisations. Les ontologies sont utilisées comme des représentations de connaissances formalisées pour annoter ces ressources web et elles sont aussi utilisées dans la communication entre les systèmes informatique.

Le rôle des ontologies est très important, mais malheureusement leur construction est coûteuse, pour cela nous avons pensé à l'énorme quantité d'informations disponible sous forme textuelle pour l'utiliser pour automatiser le processus de construction d'ontologie. Puisque on traite des textes on est obligé de passer par une étape considérée comme une étape de base pour le domaine de construction d'ontologie à partir du texte, cette étape consiste a faire appel au domaine du traitement automatique de langages naturelles pour extraire des informations syntaxique de texte, puis les transformer en informations sémantiques

Afin d'accomplir cet objectif on a proposé une approche basée agent pour construire une ontologie à partir du texte, et on a implémenté un système multi agents guidé par cette approche qui à partir d'un ensemble de documents textuels nous donne une ontologie formelle sous forme d'OWL.

Mots-clés : La construction automatique d'ontologies, ontologie, Concept formel, Formal Concept Analysis , Relationnal Concept Analysis.

Abstract

An ontology is an explicit specification of a conceptualization, the term is often linked with the Semantic Web which is a new version of the current web aims to make web resources accessible by software agents to facilitate their use, ontologies are used as representations of formalized knowledge to annotate web resources and are also used in communication between the computer system.

The role of ontologies is very important, but unfortunately their construction is expensive, this is why we think of the enormous amount of information available in text format to use it for automate the process of building ontology, and since we deal with texts, we should think about the NLP(), which considered as the basic on ontology construction from text.

To accomplish this goal we proposed an agent-based approach to build ontology from text, and we implemented a multi-agent system guided by this approach, which start from a set of textual documents gives us a formal ontology in form of OWL.

We chose here to use the two formals techniques Formal Concept Analysis FCA and Relational Concept Analysis RCA to move from the syntactic level to the semantic level.

Keywords: Automatic ontology construction, ontology, concept formel, Formal Concept Analysis , Relationnal Concept Analysis.

ملخص

الانتولوجي هي مواصفات واضحة لمفهوم معين, هذا المصطلح يكون غالبا مرتبط مع الويب الدلالي الذي هو نسخة حديثة من الويب الحالي, يهدف الويب الدلالي إلى جعل الموارد الخاصة بالويب متاحة إلى مجموعة من البرامج لتسهيل استخدام هذه الموارد. الانتولوجي تستعمل كتمثيل للمعارف لتمييز موارد الويب و أيضا في التواصل بين الأعوان.

دور الانتولوجي مهم جدا ولكن للأسف عملية بناءها مكلفة جدا, ولهذا السبب فكرنا في كمية المعارف الكثيرة المتوفرة على شكل نصوص لاستعمالها لتأليه عملية بناء الانتولوجي , ولأننا نتعامل مع النصوص وجب علينا المرور عبر مرحلة تعتبر مرحلة أساسية, هذه المرحلة تستعمل تقنيات المعالجة الآلية للنصوص لاستخراج معلومات نحوية و تحويلها إلى معلومات ذات دلالة.

لكي نصل إلى هذا الهدف نحن نقترح طريقة تعتمد على الأعوان لبناء الانتولوجي من النصوص, وكذلك طورنا نظام متعدد الأعوان الذي يتبع هذه الطريقة والذي يعطينا انتولوجي انطلاقا من مجموعة من النصوص.

الكلمات المفاتيح : البناء الآلي الانتولوجي. انتولوجي, المفهوم. تحليل المفاهيم, تحليل العلاقات.