

## Thèse

# **Conception Coopérative d'Ontologies Pré-Consensuelles : Application au domaine de l'Urbanisme**

présentée devant  
L'Institut National des Sciences Appliquées de Lyon

pour obtenir  
le grade de docteur

**Ecole doctorale** : Ecole Doctorale Informatique et Information pour la Société  
(EDIIS)

**Spécialité** : Informatique

Par  
**Abdel Kader KEITA**

Soutenue le 06 Juin 2007 devant la Commission d'examen

### **Jury**

---

ZIMMERMANN Monique	Professeur INSA de Lyon	Présidente
BOUCELMA Omar	Professeur Université Aix-Marseille 3	Membre
LAS CASAS Giuseppe	Professeur Université de la Basilicate	Rapporteur
LAURINI Robert	Professeur l'INSA de Lyon	Directeur
LIBOUREL Thérèse	Professeur Université Montpellier II	Rapporteur
ROUSSEY Catherine	Maître de Conf. Université Lyon I	co-Directrice

## Liste des écoles doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
	<p><i>CHIMIE DE LYON</i></p> <p>Responsable : M. Denis SINOÛ</p>	<p>M. Denis SINOÛ            Université Claude Bernard Lyon 1            Lab Synthèse Asymétrique UMR UCB/CNRS 5622            Bât 308            2<sup>ème</sup> étage            43 bd du 11 novembre 1918            69622 VILLEURBANNE Cedex            Tél : 04.72.44.81.83 Fax : 04 78 89 89 14  <a href="mailto:sinou@univ-lyon1.fr">sinou@univ-lyon1.fr</a></p>
E2MC	<p><i>ECONOMIE, ESPACE ET MODELISATION DES COMPORTEMENTS</i></p> <p>Responsable : M. Alain BONNAFOUS</p>	<p>M. Alain BONNAFOUS            Université Lyon 2            14 avenue Berthelot            MRASH M. Alain BONNAFOUS            Laboratoire d'Economie des Transports            69363 LYON Cedex 07            Tél : 04.78.69.72.76            Alain.bonnafous@ish-lyon.cnrs.fr</p>
E.E.A.	<p><i>ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE</i></p> <p>M. Daniel BARBIER</p>	<p>M. Daniel BARBIER            INSA DE LYON            Laboratoire Physique de la Matière            Bâtiment Blaise Pascal            69621 VILLEURBANNE Cedex            Tél : 04.72.43.64.43 Fax 04 72 43 60 82  <a href="mailto:Daniel.Barbier@insa-lyon.fr">Daniel.Barbier@insa-lyon.fr</a></p>
E2M2	<p><i>EVOLUTION, ECOSYSTEME, MICROBIOLOGIE, MODELISATION</i>  <a href="http://biomserv.univ-lyon1.fr/E2M2">http://biomserv.univ-lyon1.fr/E2M2</a></p> <p>M. Jean-Pierre FLANDROIS</p>	<p>M. Jean-Pierre FLANDROIS            UMR 5558 Biométrie et Biologie Evolutive            Equipe Dynamique des Populations Bactériennes            Faculté de Médecine Lyon-Sud Laboratoire de Bactériologie BP 1269600            OULLINS            Tél : 04.78.86.31.50 Fax 04 72 43 13 88            E2m2@biomserv.univ-lyon1.fr</p>
EDIIS	<p><i>INFORMATIQUE ET INFORMATION POUR LA SOCIETE</i>  <a href="http://ediis.univ-lyon1.fr">http://ediis.univ-lyon1.fr</a></p> <p>M. Alain Mille</p>	<p>M. Alain Mille            UCB Lyon 1            EDIIS            Bâtiment Nautibus            69622 VILLEURBANNE Cedex            Tél : 04.72.44.82.94 Fax 04 72 44 80 53  <a href="mailto:ediis@univ-lyon1.fr">ediis@univ-lyon1.fr</a></p>
EDISS	<p><i>INTERDISCIPLINAIRE SCIENCES-SANTE</i>  <a href="http://www.ibcp.fr/ediss">http://www.ibcp.fr/ediss</a></p> <p>M. Alain Jean COZZONE</p>	<p>M. Alain Jean COZZONE            IBCP (UCBL1)            7 passage du Vercors            69367 LYON Cedex 07            Tél : 04.72.72.26.75 Fax : 04 72 72 26 01  <a href="mailto:cozzone@ibcp.fr">cozzone@ibcp.fr</a></p>
	<p><i>MATERIAUX DE LYON</i>  <a href="http://www.ec-lyon.fr/sites/edml">http://www.ec-lyon.fr/sites/edml</a></p> <p>M. Jacques JOSEPH</p>	<p>M. Jacques JOSEPH            Ecole Centrale de Lyon            Bât F7 Lab. Sciences et Techniques des Matériaux et des Surfaces            36 Avenue Guy de Collongue BP 163            69131 ECULLY Cedex            Tél : 04.72.18.62.51 Fax 04 72 18 60 90  <a href="mailto:Jacques.Joseph@ec-lyon.fr">Jacques.Joseph@ec-lyon.fr</a></p>
Math IF	<p><i>MATHEMATIQUES ET INFORMATIQUE FONDAMENTALE</i>  <a href="http://www.ens-lyon.fr/MathIS">http://www.ens-lyon.fr/MathIS</a></p> <p>M. Franck WAGNER</p>	<p>M. Franck WAGNER            Université Claude Bernard Lyon1            Institut Girard Desargues            UMR 5028 MATHEMATIQUES            Bâtiment Doyen Jean Braconnier            Bureau 101 Bis, 1<sup>er</sup> étage            69622 VILLEURBANNE Cedex            Tél : 04.72.43.27.86 Fax : 04 72 43 16 87  <a href="mailto:wagner@desargues.univ-lyon1.fr">wagner@desargues.univ-lyon1.fr</a></p>

**MEGA**

*MECANIQUE, ENERGETIQUE, GENIE CIVIL,  
ACOUSTIQUE*

<http://www.lmfa.ec-lyon.fr/autres/MEGA/index.html>

M. François SIDOROFF

**M. François SIDOROFF**

Ecole Centrale de Lyon

Lab. Tribologie et Dynamique des Systèmes Bât G8

36 avenue Guy de Collongue

BP 163

69131 ECULLY Cedex

Tél : 04.72.18.62.14 Fax : 04 72 18 65 37

[Francois.Sidoroff@ec-lyon.fr](mailto:Francois.Sidoroff@ec-lyon.fr)

## **Dédicace**

*En la mémoire de mes chers parents !*

## Remerciements

*Je remercie sincèrement Robert LAURINI, Professeur à l'INSA de Lyon, de m'avoir donnée l'opportunité de découvrir la recherche en m'accueillant dans l'équipe spatio-temporelle qu'il dirige.*

*Mes remerciements les plus chaleureux vont à Catherine ROUSSEY, Maître de Conférences à l'Université Claude Bernard de Lyon 1, pour son aide précieuse et sa présence constante.*

*Je tiens à remercier particulièrement Thérèse LIBOUREL, Professeur à l'Université de Montpellier II et Giuseppe LAS CASAS, Professeur à l'Université de la Basilicate (Potenza, Italie), pour avoir accepté d'être les rapporteurs de la présente thèse*

*Mes remerciements vont également à Monique ZIMMERMANN, Professeur au département Génie Civil et Urbanisme (GCU) de l'INSA de Lyon et Directrice du laboratoire Equipe Développement Urbain (EDU), et à Omar BOUCELMA, Professeur Université Aix-Marseille 3 qui ont accepté d'être membres du jury.*

*Je remercie les membres du laboratoire EDU, particulièrement Chantal Berdier Professeur au département GCU et les étudiants Caroline BEAULIEU, Yohan TARDY et Christophe BERTHET, pour leur participation combien productive dans le projet Towntology qui a servi de cadre aux travaux de ma thèse.*

*Je tiens à remercier Samuel GESCHE, élève ingénieur en informatique, pour son implication dans le développement du prototype Towntology.*

*Je remercie David SINUHE et Louhisuo KIRSI, étudiants en stage de master pour leurs apports.*

*Toute ma gratitude et ma reconnaissance vont aux familles Dicko et Touré à Lyon qui m'ont servi de familles d'accueil pendant tous mes séjours à Lyon.*

*Mes derniers mais non moins profonds remerciements vont à ma famille restée au Mali et plus particulièrement à ma chère épouse, qui m'a toujours encouragé et soutenu dans la poursuite de mes études.*

*Je n'oublie pas toutes les personnes et institutions qui ont contribué au bon déroulement de cette thèse.*

## **Résumé**

Une ontologie possède deux définitions différentes suivant le domaine auquel on s'intéresse, en l'occurrence la philosophie ou l'informatique :

- Du point de vue philosophique, une ontologie se définit comme la science de ce qui existe.
- Du point de vue informatique et plus particulièrement ingénierie des connaissances, la définition la plus communément admise est celle de Studer [STUD98] : "*Une ontologie est une spécification formelle, explicite d'une conceptualisation partagée*".

La majorité des travaux et des outils existants concernent la création des ontologies, une fois le consensus réalisé (que nous intitulons ontologies post-consensuelles).

Ainsi, une des étapes clés (importantes) dans le processus de développement d'une ontologie est le consensus entre les différents acteurs. Mais le problème principal est comment atteindre le consensus ?

L'une des caractéristiques essentielles des ontologies est qu'elles fournissent des données consensuelles sur un domaine donné. De nos jours, les ontologies englobent un ensemble si riche de la connaissance que leur compréhension (complète) dépasse celle de n'importe quel développeur ou concepteur seul ou même d'une petite équipe de concepteurs. Le développement d'une ontologie à grande échelle doit être le fruit d'un effort commun de plusieurs experts de domaine et concepteurs de logiciels. Ainsi l'approche collaborative pour le développement d'ontologie est la mieux adaptée pour cette tâche par rapport aux autres approches non collaboratives, vues les caractéristiques mêmes d'une ontologie.

Cependant la plupart des méthodes et méthodologies ne proposent pas de directives pour atteindre le consensus quand les ontologies sont construites de manière collaborative.

La construction des ontologies dans un environnement collaboratif a été et demeure un sujet de recherche continu et ouvert. Il existe un certain nombre d'outils pour le développement collaboratif d'ontologies. Mais aucun n'offre un support complet et adaptable pour toutes les exigences du développement collaboratif.

L'analyse des outils et environnements existants (pour le développement collaboratif des ontologies) ne vérifie pas seulement le fait que le développement collaboratif est la nature inhérente de la conception d'ontologie, mais indique aussi que le développement collaboratif d'ontologie n'est bien soutenu par aucun des outils ou environnements existants. Ceci présente un nouveau challenge pour trouver des outils mieux adaptés pour le développement collaboratif d'ontologie.

L'inexistence (en notre connaissance) d'ontologie pour l'urbanisme et l'aménagement du territoire, ainsi que la complexité et la dimension du domaine de l'urbanisme ajoute à ce challenge un intérêt particulier.

## **Contributions**

Au contraire de la majorité des travaux et des outils existants qui concernent la création des ontologies, une fois le consensus réalisé (ontologies post-consensuelles), nos travaux, réalisés dans le cadre du projet européen Towntology, se situent dans les premières phases de développement d'une ontologie, lorsque le consensus n'est pas encore atteint. Ce travail essaye de répondre à la question d'atteindre le consensus en proposant une méthodologie pour créer une ontologie pré-consensuelle en partant des opinions de divers acteurs. Par exemple, nous insistons sur la nécessité de garder différentes définitions pour le même concept.

Plus exactement, nous proposons :

Une méthodologie de développement d'ontologie pré-consensuelle et une démarche permettant lors de la phase d'acquisition des connaissances à des experts du domaine (dans notre cas des urbanistes) de modéliser les différences de points de vue sur le sens des termes. Notre méthodologie fait intervenir les experts du domaine le plus longtemps possible en leur permettant d'aller vers le consensus. Nous proposons une extension de XML pour stocker notre ontologie.

- Un outil visuel permettant d'éditer, de naviguer dans l'ontologie et de l'interroger.
- Un support pour le développement collaboratif d'ontologie proposant l'ontologie créée comme ontologie d'ancrage pour le travail collaboratif.

**Mots-clés** : Ontologies, conception coopérative des ontologies, Towntology, Groupware, pré-consensus, urbanisme.

## ***Abstract***

An ontology has two different definitions according to the field of interest, in fact philosophy or computer science:

- From the philosophical point of view, an ontology is defined as being the science of what exists.
- From the point of view of computer science and more particularly, knowledge engineering, the definition most commonly accepted is that of (Studer, 1998) : "*an ontology is an explicit formal specification of a shared conceptualization of a field*".

The majority of works and the existing tools relate to the creation of ontologies, once the consensus is carried out (which we entitle as post-consensual ontologies).

Thus, one of the key stages in the process of developing an ontology is the consensus between the various actors. But the principal problem is how to reach this consensus.

One of the essential characteristics of ontologies is that they provide consensual data on a given field. Nowadays, ontologies include a so rich set of knowledge that their full comprehension exceeds that of any single developer or creator or even that of a small team of creators. The development of an ontology on a large scale must be the result of a joint effort from several experts in various fields and software developers. So the collaborative approach for the development of ontology is best adapted for this task compared to the other non collaborative approaches, because of the characteristics, themselves, of an ontology.

However the majority of the methods and methodologies do not propose recommendations to reach the consensus when ontologies are built with a collaborative way.

The construction of ontologies with a collaborative environment was and remains a continuous and open subject of research. There exists a certain number of tools for the collaborative development of ontologies. But none offers a complete and adaptable support for all the requirements of the collaborative development.

The analysis of the existing tools and groupware products (for the collaborative development of ontologies) does not check only the fact that the collaborative development is the inherent nature of the design of ontology, but also states that the collaborative development of ontology is well supported by none of the existing tools or environments. This presents a new challenge to find tools better adapted for the collaborative development of ontology.

The inexistence (as we know) of ontology for urban planning and regional planning, as well as the complexity and the dimension of the domain of urban planning add a particular interest to this challenge.

## **Contributions**

In contrast of the majority of research works and existing tools which relate to the creation of ontologies once the consensus is carried out (post-consensual ontologies), our research works, produced in the context of the European project TOWNTOLOGY, focus on the first phases of development of an ontology, when the consensus is not yet reached. This work tries to answer the question of reaching the consensus by proposing a methodology to create a pre-consensual ontology on the basis of the opinions of various actors. For example, we insist on the need for keeping various definitions for the same concept.

We propose more exactly:

- A methodology of development of pre-consensual ontology and a step allowing the experts of the field (in our case, the urban planners) to model the differences in the various points of view about the meaning of the terms during the phase of knowledge acquisition. Our methodology makes the experts of the field intervene as long as possible by enabling them to tend to some consensus. We propose an extension of XML to store our ontology.
- A visual tool allowing to edit, to navigate in ontology and to query it.
- A support for the collaborative development of ontology proposing the designed ontology as an anchoring ontology for collaborative work.

**Keywords :** ontologies, coopérative development of ontologies, Towntology, Groupware, pré-consensus, urban planning

# SOMMAIRE

<b>PARTIE 1. INTRODUCTION GENERALE .....</b>	<b>11</b>
<b>Chapitre 1. Introduction générale .....</b>	<b>12</b>
1.1 Introduction .....	12
1.2 Problématique.....	13
1.3 Contributions.....	15
1.4 Plan de la thèse.....	17
<b>PARTIE 2. CONTEXTE DE LA THESE.....</b>	<b>18</b>
<b>Chapitre 2. Contexte de la thèse .....</b>	<b>19</b>
2.1 Le domaine de l'urbanisme : complexité et particularité.....	19
2.2 Le Consensus : hypothèse de solution à la complexité de l'urbanisme .....	24
2.3 Les ontologies : quel intérêt pour les urbanistes ? .....	29
2.4 Le projet Towntology .....	31
2.4.1 Les objectifs du projet.....	32
2.4.2 Les différentes étapes du projet Towntology .....	33
2.5 Conclusions .....	34
<b>PARTIE 3. LES ONTOLOGIES ET LE GROUPWARE (ETAT DE L'ART).....</b>	<b>35</b>
<b>Chapitre 3. Qu'est-ce qu'une ontologie ?.....</b>	<b>36</b>
3.1 Les types d'ontologies.....	40
3.2. Les composants d'une ontologie .....	45
3.3 Usages des ontologies .....	47
3.4 Les différentes approches de conception d'ontologie .....	48
3.5 Conclusion.....	50
<b>Chapitre 4. Les méthodes et méthodologies de construction d'ontologie .....</b>	<b>51</b>
4.1 Le processus de développement d'ontologie .....	52
4.2 Les méthodes et méthodologies de développement des ontologies .....	53
4.2.1 La méthodologie METHONTOLOGY .....	53
4.2.2 La méthodologie On-To-Knowledge .....	54
4.2.3. La méthode "Ontology Development 101" .....	54
4.2.4 Les autres méthodes de développement d'ontologies .....	55
4.3. Analyse et comparaison des méthodes et méthodologies .....	57
4.4 Conclusion.....	63
<b>Chapitre 5. Les langages de description des ontologies .....</b>	<b>64</b>
5.1 Les langages d'ontologies dits "traditionnels" .....	65
5.2 Les langages d'ontologie dits "web-standard" ou langages basés sur XML.....	67
5.3 Analyse et comparaison des langages de description d'ontologie .....	69
5.4 Utilisation des langages dans les applications basées sur les ontologies .....	72
5.5 Conclusion.....	74
<b>Chapitre 6. Les Groupwares .....</b>	<b>75</b>



6.1 Aperçu sur les CSCW (Computer Supported Collaborative Work) : le travail collaboratif assisté par ordinateur .....	
6.2 Les dimensions des problèmes pour les Groupwares.....	
6.2.1 Distance et Communication .....	
6.2.2 Gestion de la Documentation et de la Connaissance.....	
6.2.3 Contrôle de version et traçabilité des modifications .....	
6.3 Les outils de groupware .....	
6.3.1 Les outils de communication de base.....	
6.3.2 Les outils de travail partagé.....	
6.3.3 Les outils d'accès au savoir - ou outils de gestion de connaissance .....	
6.3.4 Les outils de workflow .....	
6.3.5 La nouvelle génération des outils de groupware ou collecticiels.....	
6.3.5.1 Les Wiki et Weblog : le renouveau des outils de groupware.....	
6.3.5.2 Les Folksonomies : exemple de communauté sans contrôle explicite .....	
6.3.5.3 Les systèmes de gestion de contenu et PLONE .....	
6.4 Conclusion.....	
<b>Chapitre 7. Les Groupwares pour la construction des ontologies.....</b>	
7.1 Ontolingua Server.....	
7.2 OntoEdit et sa suite OntoStudio .....	
7.3 APECKS.....	
7.4 Système CO4 et protocole CO4 .....	
7.5 Protégé-2000 .....	
7.6 CODE .....	
7.7 DILIGENT .....	
7.8 C-VISTA .....	
7.9 La méthode DELPHI.....	
7.10 Comparaison des outils .....	
7.11 Analyse de la comparaison.....	
7.12 Conclusion.....	
<b>CONCLUSIONS SUR L'ETAT DE L'ART .....</b>	<b>1</b>
<b>PARTIE 4. NOS TRAVAUX ET CONTRIBUTIONS : LA CONCEPTION COOPERATIVE D'ONTOLOGIES PRE- CONSENSUELLES .....</b>	<b>1</b>
<b>Chapitre 8. La conception coopérative d'ontologies pré-consensuelles .....</b>	<b>1</b>
8.1 Notre méthodologie : la méthodologie Towntology .....	
8.1.1 Les activités de planning .....	
8.1.2 Les activités orientées développement .....	
8.1.3 Les activités support.....	
8.1.4 Les activités Groupware.....	
8.2 Le modèle Towntology .....	
8.3 Le langage Towntology .....	
8.4 Les fonctionnalités du système pour la conception collaborative d'ontologies .....	
8.4.1 Navigateur visuel d'ontologie .....	
8.4.2 Editeur visuel d'ontologie : Towntology Editor.....	
8.4.3 Mécanisme de construction du consensus.....	

8.4.3.1 Exemple de workflow de création et d'édition d'un objet : .....	141
8.4.3.2 Exemple de publication d'un objet.....	142
8.4.3.3 Proposition d'argumentations pour traiter des conflits vers le consensus.....	142
8.4.4 Analyse de cohérence locale (interne à une ontologie).....	145
8.4.4.1 Evolution cohérente de l'ontologie (ECO).....	<b>Erreur ! Signet non défini.</b>
8.4.4.2 Réparation des incohérences .....	<b>Erreur ! Signet non défini.</b>
8.4.5 Analyse de cohérence globale (intra-ontologies) .....	146
8.5 Towntology Groupware .....	149
8.6 Conclusions	
<b>Chapitre 9. Implémentation .....</b>	<b>156</b>
9.1 Architecture du système Towntology .....	156
9.2 Les outils ou principales interfaces utilisateurs.....	160
9.2.1 Navigateur visuel d'ontologie : Towntology-Browser.....	161
9.2.2 Editeur visuel d'ontologie : Towntology Editor .....	164
9.2.3 Interface principale : Townto-Plone.....	166
9.3 Les applications du système Towntology .....	160
9.3.1 Les fichiers .....	168
9.3.2 Les classes .....	169
9.4 Structure de données internes pour représenter l'ontologie sous forme de graphe.....	170
9.4.1 Implémentation en JAVA.....	171
9.4.2 Parcours du Graphe .....	175
9.5 Formules mathématiques pour la présentation des concepts (première et deuxième couronne) .....	178
9.5.1 Description axiomatique : .....	178
9.5.2 Représentation des couronnes autour de l'élément actif.....	179
9.6 Conclusion.....	180
<b>Chapitre 10. Expérimentation.....</b>	<b>182</b>
10.1 Conception de l'ontologie de la Voirie [BEAU03].....	182
10.2 Conception de l'ontologie de la Mobilité et Transports [BERT04].....	185
10.3 Les résultats.....	186
10.4 Les difficultés rencontrées par nos experts du domaine et l'apport de l'outil Towntology.....	186
10.5 Conclusions .....	187
<b>PARTIE 5. CONCLUSION GENERALE.....</b>	<b>182</b>
<b>Chapitre 11. Conclusion générale .....</b>	<b>189</b>
11.1 Conclusions .....	189
11.2 Perspectives.....	192
<b>RÉFÉRENCES BIBLIOGRAPHIQUES .....</b>	<b>193</b>
<b>ANNEXES .....</b>	<b>204</b>

## **Partie 1. Introduction générale**

Dans cette partie nous explicitons le sujet traité pour introduire. Ensuite, nous posons la problématique et présentons nos contributions. Enfin nous donnons le plan de la thèse.

## **Chapitre 1. Introduction générale**

Dans ce chapitre nous explicitons le sujet traité comme point de départ. Ensuite, nous posons la problématique et présentons nos contributions. Enfin nous donnons le plan de la thèse.

### **1.1 Introduction**

La construction des ontologies dans un environnement collaboratif a été et demeure un sujet de recherche continu et ouvert.

La conception coopérative d'ontologies pré-consensuelles nécessite, une méthodologie et une démarche qui se situent dans les premières phases de développement d'une ontologie, permettant à des experts du domaine de modéliser les différences de points de vue sur le sens des termes. La démarche devrait faire intervenir les experts du domaine dans la conception de l'ontologie en leur permettant d'aller vers le consensus. Mais atteindre le consensus entre experts n'est pas chose facile.

Le domaine de l'urbanisme est caractérisé par l'implication de plusieurs acteurs dans de secteurs variés, chacun des secteurs ayant son propre vocabulaire. Les données urbaines constituent un ensemble d'informations spatio-temporelles, dont le traitement est une tâche difficile et coûteuse. La meilleure solution d'y faire face passe sans doute par la coopération entre les acteurs : la conception de chaque action dans l'urbanisme est un processus coopératif.

L'une des caractéristiques essentielles des ontologies est qu'elles fournissent des données consensuelles sur un domaine donné. Les ontologies sont utilisées non seulement pour améliorer les communications entre les machines, mais également entre les humains, et entre les humains et les machines. Ainsi, la solution de la coopération entre les acteurs dans l'urbanisme peut se baser sur l'ontologie. Mais comment concevoir cette ontologie ? Et comment atteindre le consensus ? Nous pouvons facilement dire que la conception d'une ontologie est encore un processus coopératif, qui comporte des similitudes avec la conception en proprement parler en urbanisme [LAUR01].

## 1.2 Problématique

Une ontologie possède deux définitions différentes suivant le domaine auquel on s'intéresse, en l'occurrence la philosophie ou l'informatique :

- Du point de vue philosophique, une ontologie se définit comme la science de ce qui existe.
- Du point de vue informatique et plus particulièrement ingénierie des connaissances, la définition la plus communément admise est celle de Studer [STUD98] : "*Une ontologie est une spécification formelle, explicite d'une conceptualisation partagée*". La conceptualisation se rapporte à un modèle abstrait d'un certain phénomène dans le monde en ayant identifié les concepts appropriés de ce phénomène. Explicite signifie que le type de concepts utilisés, et les contraintes sur leur utilisation sont explicitement définis. Formelle se rapporte au fait que l'ontologie devrait être compréhensible (lisible) par une machine. Partagée reflète la notion que l'ontologie capture la connaissance consensuelle, c'est-à-dire, elle n'est pas privée, mais admise par un groupe.

De manière générale, une ontologie contient un vocabulaire formalisé regroupant pour une discipline donnée, l'ensemble des concepts, et de leurs relations. Les définitions associées à chaque concept sont issues d'un consensus entre les différents acteurs et futurs utilisateurs de l'ontologie.

Construire une ontologie est un processus complexe, dont la complexité augmente si les développeurs doivent implémenter directement l'ontologie dans un langage formel, sans avoir à leur disposition un outil d'aide à la construction. Pour faciliter cette tâche, les premiers environnements de construction d'ontologies ont été créés au milieu des années 90. Ces derniers fournissaient des outils pour aider des utilisateurs à effectuer certaines des tâches principales du processus de développement d'ontologie, telles que la conceptualisation, l'implémentation, le contrôle de cohérence, et la documentation. Ces dernières années, le nombre d'outils pour le développement d'ontologies a considérablement augmenté et a été diversifié [GOME02].

La majorité des travaux et des outils existants concernent la création des ontologies, une fois le consensus réalisé (que nous intitulons ontologies post-

consensuelles). Ainsi, une des étapes clés dans le processus de développement d'une ontologie est le consensus entre les différents acteurs. Mais le problème principal est comment atteindre le consensus ?

L'une des caractéristiques essentielles des ontologies est qu'elles fournissent des données consensuelles sur un domaine donné. Cependant la plupart des méthodes et méthodologies ne proposent pas de directives pour atteindre le consensus même si les ontologies sont construites de manière collaborative.

La construction des ontologies dans un environnement collaboratif a été et demeure un sujet de recherche continu et ouvert. Il existe un certain nombre d'outils pour le développement collaboratif d'ontologies.

Comme les ontologies deviennent de plus en plus présentes dans un plus grand nombre d'applications de grande taille et de longue durée de vie, de plus en plus d'ontologies sont développées dans les environnements distribués par des auteurs situés sur des sites géographiquement différents. L'analyse des outils et environnements existants (pour le développement collaboratif des ontologies) ne vérifie pas seulement le fait que le développement collaboratif est la nature inhérente de la conception d'ontologie, mais indique aussi que le développement collaboratif d'ontologie n'est bien soutenu par aucun des outils ou environnements existants. Ceci présente un défi pour trouver des outils mieux adaptés pour le développement collaboratif d'ontologie. Il est vrai qu'Internet et les technologies de communication nous offrent des possibilités de communiquer ; néanmoins, nous avons encore des problèmes pour mettre en place des outils intégrant le travail collaboratif, impliquant des personnes ou des équipes géographiquement distribuées. Il est certain qu'il reste encore du chemin à faire.

L'inexistence (en notre connaissance) d'ontologie pour l'urbanisme et l'aménagement du territoire ajoute à ce challenge un intérêt particulier. Le domaine de l'urbanisme est un vaste domaine avec plusieurs secteurs. En effet plusieurs acteurs sont impliqués, chacun d'eux ayant son propre vocabulaire. Il est reconnu que le vocabulaire urbain est caractérisé par :

- La multiplicité des termes et concepts pour lesquels les frontières sémantiques ne sont pas claires (ou bien définies) ;
- La variation des dispositifs urbains dans différents cultures et pays ;
- Le problème posé par des concepts dans différentes langues.

Comment offrir un cadre de travail unifié et coopératif pour les différents acteurs de l'urbanisme et assurer l'interopérabilité et la coopération entre différentes bases de données urbaines ?

Comment concevoir cette ontologie de domaine sur laquelle sera basée la solution de la coopération entre les acteurs de l'urbanisme ?

### **1.3 Contributions**

A la différence de la majorité des travaux et des outils existants qui concernent la création des ontologies, une fois le consensus réalisé (ontologies post-consensuelles), nos travaux, réalisés dans le cadre du projet Towntology (voir la section 2.5), se situent dans les premières phases de développement d'une ontologie, lorsque le consensus n'est pas encore atteint. Plus exactement, dans cette thèse nous proposons :

- Une méthodologie et une démarche permettant lors de la phase d'acquisition des connaissances à des experts du domaine (dans notre cas des urbanistes) de modéliser les différences de points de vue sur le sens des termes. Notre méthodologie fait intervenir les experts du domaine le plus longtemps possible en leur permettant d'aller vers le consensus.
- Un outil visuel permettant d'éditer, de naviguer dans l'ontologie et de l'interroger. Cet outil permet d'associer des ressources multimédia aux concepts.
- Un support pour le développement collaboratif d'ontologie proposant l'ontologie créée comme ontologie d'ancrage pour le travail collaboratif.

Ce travail essaye de répondre à la question d'atteindre le consensus en proposant une méthodologie pour créer une ontologie pré-consensuelle en partant des opinions de divers acteurs. Ainsi nous construisons un premier prototype d'ontologie appelé ontologie pré-consensuelle. Contrairement aux ontologies post-consensuelles, les différentes définitions d'un même terme sont conservées. Notre travail vise à écrire des ontologies urbaines sous une forme visuelle. Cette visualisation est un apport pour le développement collaboratif. Un format XML est conçu et utilisé pour la description de l'ontologie.

Comme résultat, nous proposons un support collaboratif Townto-Groupware basé sur le prototype Towntology qui est un système permettant d'éditer et de stocker l'ontologie, d'afficher l'ontologie sous forme de graphe visuel, de naviguer dans l'ontologie et de l'interroger.

Concrètement nous avons dans notre ontologie environ 1000 concepts (langue française) et 21 types de relations entre ces concepts.

L'inexistence d'ontologie pour l'urbanisme et l'aménagement du territoire procure à cet outil un fort intérêt, à savoir : satisfaire un besoin de coordination et de coopération entre les acteurs de l'urbain. Cet outil devrait pallier les dérives sémantiques de compréhension entre les interlocuteurs et les systèmes d'information géographiques des divers intervenants dans le champ de l'urbain, permettant ainsi une meilleure coopération. En effet, les mots de la langue française ayant la particularité d'avoir des définitions différentes selon le domaine, cette ontologie permettra de regrouper ces spécificités linguistiques afin de fournir un outil capable de devenir l'articulation entre les différents acteurs du vaste domaine de l'urbain. Cet aspect de l'ontologie devrait permettre d'aller dans le sens de la concertation, dans le but de faire dialoguer les destinateurs et les destinataires autour d'un vocabulaire formalisé.

A terme, lorsque cet outil sera complètement opérationnel, il offrira plusieurs avantages :

- Amélioration des conditions et supports pour l'enseignement de l'urbanisme ;
- Cadre de référence pour les experts, permettant l'indexation de leurs documents ;
- Aide à la recherche d'information ;
- Aide à la formation du personnel ;
- Mises à jour des différents documents d'urbanisme en adaptant le vocabulaire en fonction de l'utilisateur et en organisant ces documents en fonction des besoins des utilisateurs etc.
- Aide pour l'élaboration conjointe des documents d'urbanisme.



Dans cette thèse, nous décrivons notre travail sur la conception collaborative d'ontologies pré-consensuelles appliquée au domaine de l'urbanisme, réalisé dans le cadre du projet Towntology.

Dans la section suivante, nous donnons le plan de la thèse en décrivant les différentes parties et chapitres.

#### **1.4 Plan de la thèse**

Cette section détaille la structure de la thèse, qui est divisée en cinq parties.

Dans la première partie (introduction), nous avons déjà présenté la problématique et les contributions. La deuxième partie de la thèse traite du contexte de la thèse caractérisé par la complexité et la particularité du domaine de l'urbanisme, dont les acteurs dans la recherche du consensus pourraient utiliser des ontologies comme hypothèse de solution. Ensuite, nous présentons le projet Towntology qui a servi de cadre à nos travaux.

La troisième partie de la thèse traite de l'existant (état de l'art) sur les ontologies et les groupwares.

La partie 4 de la thèse relate nos travaux : sur la conception coopérative d'ontologie pré-consensuelle, sur l'implémentation de l'outil Towntology, et sur notre expérimentation avec les experts du laboratoire EDU (Equipe Développement Urbain) du département GCU (Génie Civil et Urbanisme) de l'INSA de Lyon.

La dernière partie, portant sur la conclusion générale de la thèse, présente nos conclusions et perspectives.

## **Partie 2. Contexte de la thèse.**

Dans cette partie nous décrivons le contexte dans lequel nos travaux ont été réalisés.

## Chapitre 2. Contexte de la thèse

Dans ce chapitre nous décrivons le contexte dans lequel nos travaux ont été réalisés. Ce qui nous amène à nous intéresser au domaine de l'urbanisme et de l'aménagement du territoire, avec sa complexité, sa diversité et sa particularité. Ensuite, comme solutions à ces difficultés, nous posons des hypothèses portant sur l'utilisation d'un mécanisme de consensus et de l'ontologie en vue d'atteindre nos objectifs. Enfin, nous présentons le projet Towntology qui a servi de cadre à nos travaux.

### 2.1 Le domaine de l'urbanisme : complexité et particularité

La complexité et la particularité du domaine de l'urbanisme se mesurent dans les faits et vécus des acteurs du domaine. Pour illustrer cette réalité, voyons un exemple : quand François Ascher<sup>1</sup>, un urbaniste avéré, affirme que *"l'urbanisme n'est pas une science, mais un domaine large et complexe où interviennent des disciplines fort différentes"* et qu'il ajoute que *"actuellement l'urbanisme n'est plus le fait de professionnels, c'est un ensemble de dispositifs"*, il se fait critiqué par d'autres urbanistes comme par exemple l'OPQU (Office Professionnel de Qualification des Urbanistes) qui rejette cette affirmation. Cela démontre toute l'ambiguïté et toute la complexité qui caractérisent ce domaine de l'intérieur même.

Aujourd'hui le domaine de l'urbanisme est caractérisé par :

- La diversité des secteurs, basés sur des disciplines aussi disparates qu'interdépendantes, qui le composent ;
- L'intervention d'acteurs différents mais souvent complémentaires par leurs niveaux d'expertise (ou de qualification), leurs activités et leurs terminologies ;
- Des mutations fréquentes aussi bien dans l'espace que dans le temps.

---

<sup>1</sup> François Ascher est professeur à l'Institut Français d'Urbanisme (Paris VIII) et préside le conseil scientifique de l'Institut pour la Ville en Mouvement.

L'urbanisme est un vaste domaine avec plusieurs secteurs (ou sous-domaines) impliquant des acteurs et des disciplines aussi divers que variés. Les secteurs de l'urbain s'intéressent entre autres à la gouvernance, à l'aménagement, aux politiques d'équipement, d'infrastructures ou d'habitat, à la conception architecturale ou urbanistique, à la programmation, au montage et à la gestion de projets. Il est très difficile d'appréhender les limites et frontières entre les différents secteurs. Cet aspect s'accroît encore en présence de concepts, déjà par essence, difficiles de compréhension. De nos jours, le domaine urbain représente des enjeux très importants. Le rôle et l'importance des pouvoirs locaux urbains est reconnu par tous [CAVA02]. Les villes sont devenues le lieu de concentration de la population qui y vit à plus de 80% (pour les pays développés) et des richesses puisqu'elles sont le lieu principal de l'activité économique ; elles sont désormais le moteur du développement économique des pays.

Les acteurs ou professionnels intervenant dans l'urbanisme sont aussi différents que complémentaires par leurs niveaux d'expertise, leurs activités et leurs terminologies. Ces acteurs peuvent être des ingénieurs, architectes, entrepreneurs, maîtres d'ouvrage, maîtres d'œuvre, juristes, coordonnateurs, sociologues, élus, etc.

Concernant le vocabulaire et les terminologies employés dans le domaine et leur organisation, la complexité se situe au niveau des termes employés [BEAU03]:

– **Comment juger de la spécialisation d'un terme et de son appartenance à un sous-domaine (secteur) particulier de l'urbanisme ?** Une des premières difficultés est de savoir si un terme est spécialisé ou plutôt d'ordre général. En effet, pour un acteur (professionnel) qui connaît déjà la plupart des termes, faire la part entre le vocabulaire de l'utilisateur moyen et celui du spécialiste est donc difficile. Plus délicat encore est le classement des définitions par sous-domaines d'utilisation. Dire si un terme est plutôt employé par les aménageurs, les urbanistes ou les ingénieurs relève donc du défi. Les seuls moyens de faire la distinction sont les *dictionnaires* qui soulignent parfois les conflits d'usage. Ainsi davantage de lectures et d'entretiens avec des professionnels sont nécessaires pour amener à des solutions.

– **Comment faire le choix des termes à intégrer :** le choix de conserver ou non un terme est souvent délicat car un terme peut ouvrir la porte à un autre sous-

domaine tout entier. Décider de ne pas conserver un terme pour un secteur peut donc nous priver de l'exploration de ce sous-domaine. En somme, chaque terme fait appel à un autre potentiellement intéressant. De ce fait, le travail d'identification des termes "appropriés" pourrait ne jamais se terminer.

Par rapport aux définitions, la complexité se situe aux niveaux suivants [BEAU03] :

– **La multiplicité des définitions d'un même terme.** Il est étonnant de constater à quel point le vocabulaire urbain est riche de sens. Un même terme a souvent plusieurs définitions, pas toujours clairement établies, ce qui est source de confusions. Par exemple, la notion d'espace public va se limiter aux espaces extérieurs pour certains (rue, place,...) ; pour d'autres il s'agira également des lieux bâtis accessibles au public (gare, centre commercial,...) ; enfin le concept peut également faire référence aux espaces de négociation tels que lieux de réunion, de débats publics,... Difficile, par la suite, de savoir de quoi on parle exactement puisque personne n'a raison et personne n'a tort ! On peut essayer de lever ces ambiguïtés en indiquant les différentes définitions possibles si nécessaire. De cette façon, si chacun sait qu'un terme peut avoir différentes significations, nous sommes plus à même de s'adapter au discours de quelqu'un et de déceler quel sens il donne à un terme. Bien évidemment, cela est loin d'être simple car le vocabulaire fluctue beaucoup. Il évolue sans cesse. Par exemple, le terme "Forum", désignant initialement la place du débat public, s'est vu transformé en lieu virtuel de discussion avec l'arrivée de l'informatique.

– **La complexité de (certains) concepts.** Les concepts sont les termes les plus difficiles à cerner puisque par leur essence même ils sont complexes. Les définitions sont souvent multiples, au contour imprécis et parfois encore au stade de la recherche.

Ainsi, le concept de mixité sociale, mis au grand jour avec la loi SRU<sup>2</sup> (Solidarité et Renouvellement Urbain), regroupe différentes notions encore assez floues du fait de la nouveauté du terme dans la sphère publique. Il faudra donc attendre un peu avant que sa définition se stabilise avec le temps et suite à des recherches poussées.

---

<sup>2</sup> <http://www.aurg.org/sru/sru.htm>

Les concepts sont d'autant plus difficiles à intégrer qu'en plus d'avoir des définitions délicates à manier, les relations dont on dispose pour les lier sont parfois inadaptées. Peut-on dire que la mixité sociale est un problème touchant les groupes sociaux ? qu'elle dépend des groupes sociaux ? qu'elle caractérise les groupes sociaux ? La mixité des modes de transport dépend-t-elle du PDU (Plan de Déplacement Urbain) ou est-elle issue du PDU ? Et la mixité en général, dans tout ça, c'est quoi finalement ?

Aussi certains concepts (abstrait) présentent des difficultés à être illustrés. Le choix des méta données à leur associer n'est pas évident (exemple : Accessibilité, Mobilité, ...). Cela pose le problème du rôle des illustrations.

Par rapport aux relations entre concepts, la difficulté se rapporte à la question de comment décomposer un concept complexe en relations claires ? Cela peut devenir rapidement un travail approximatif ou inextricable.

Enfin par rapport aux mutations, on peut donc noter, au cours de ces deux dernières décennies, sous le coup de la révolution informatique et financière propres à la mondialisation, trois mutations fondamentales, du point de vue des conditions urbaines [SASSOO] :

- **L'apparition de " villes globales"** caractérisées par une fracture sociale doublée d'une fracture spatiale, où s'exprime la violence urbaine, d'une part par la domination des élites "branchées" et d'autre part par les laissés pour compte (immigrants, SDF, chômeurs et autres "pauvres"), dont la violence (délinquance, vols, viols, ...) n'est sans doute que l'expression d'une revendication d'inclusion et de reconnaissance identitaire.

- **La mutation morphologique urbaine** concomitante, entraînant la dissolution et/ou la marginalisation de la " ville compacte" dans la " ville diffuse", une ville-territoire qui aux yeux de nombreux observateurs ne peut plus être qualifiée de "ville", tant elle s'éloigne du modèle politique, économique, social, culturel et spatial de la ville " historique".

- **L'apparition de nouveaux types de centralités urbaines**, dans ces villes éclatées, autour des gares, aéroports et centres commerciaux, basés sur une mobilité accrue, rendue "indispensable et souhaitable" par les valeurs du marketing urbain et d'une société de consommation de masse.

Ainsi la diversité de secteurs basée sur des disciplines disparates, l'intervention d'un nombre élevé d'acteurs différents, et les mutations fréquentes aussi bien dans l'espace que dans le temps représentent un défi pour l'urbanisme, qui doit qui doit s'adapter aux nouveaux besoins. Pour cela, une autre stratégie (par projets) sera préférable à la planification traditionnelle.

En définitive, les mutations en cours ont remis en cause le modèle de la politique urbaine traditionnelle qui confiait aux seules autorités politiques la responsabilité de la gestion de la ville. Les villes doivent s'adapter et trouver de nouvelles démarches de gestion stratégique prenant en compte la complexité et la diversification des régulations, l'imprévisibilité de l'avenir, etc.

De grandes questions se posent alors :

- Comment répondre à la diversité et à la variabilité croissante des besoins des individus et des entreprises, dans un domaine comme l'urbanisme qui ne se conçoit et ne se réalise que dans le moyen et le long terme ?
- Comment organiser et structurer les connaissances du domaine urbain,
- Comment faciliter la communication entre plusieurs acteurs de l'urbanisme manipulant les mêmes types d'objets pour accomplir différentes tâches,
- Peut-on aller vers un consensus entre différents acteurs de l'urbain pour la mise en place d'un cadre unifié de travail et de communication sans ambiguïtés.

Les réponses à ces questions nécessitent des efforts de recherche et d'expérimentation...

Dans la section suivante, nous nous intéressons à l'hypothèse selon laquelle la coopération et l'utilisation d'un mécanisme de consensus peut-elle aider les urbanistes à se mettre d'accord, du moins sur l'essentiel. Consensus ou compromis ?

## **2.2 Le Consensus<sup>3</sup> : hypothèse de solution à la complexité de l'urbanisme**

Un consensus est un accord entre personnes qui implique la notion de consentement. Le terme désigne également l'accord, même non explicite, d'une forte majorité dans une communauté. Un compromis est un arrangement dans lequel des concessions mutuelles sont faites jusqu'à ce qu'un accord partagé puisse être trouvé. La notion de compromis est importante dans notre domaine parce qu'elle renvoie à l'idée que des projets peuvent être discutés et amendés pour tenir compte des intérêts des uns et des autres. L'idée de "projet urbain" met en valeur l'importance d'obtenir une synergie entre les acteurs du projet en vue d'aboutir aux objectifs fixés.

La notion de consensus a déjà été utilisée avec succès dans d'autres domaines pour atteindre des objectifs précis. Voyons un exemple dans la fabrication des normes et standards et un autre en politique, qui pourraient nous inspirer pour le domaine de l'urbanisme.

"La norme est un document établi par consensus et approuvé par un organisme reconnu, qui fournit, pour des usages communs et répétés, des règles, des lignes directrices ou des caractéristiques, pour des activités ou leurs résultats, garantissant un niveau d'ordre optimal dans un contexte donné" [SUTT97].

" Mais le processus de normalisation avec les procédures officielles qui lui sont rattachées est un processus souvent lent, pouvant parfois durer plusieurs années, intégrant différentes phases de validation en vue d'un consensus général : la phase technique confiée à une commission experte, la phase de validation associée à une enquête probatoire, la phase d'homologation par les directions des institutions de normalisation. L'industrie informatique et des télécommunications supporte difficilement ces délais dans un secteur économique en évolution rapide et très concurrentielle. La notion de standard, même si la distinction n'est pas toujours faite, semble alors plus opérante que la notion de norme : elle désigne un accord consensuel entre acteurs économiques, opérationnel rapidement, avant une

---

<sup>3</sup> Attention : lorsque l'on parle de consensus, rappelons-le, il s'agit de consensus sur la définition du vocabulaire en urbanisme, et non pas de consensus sur les objectifs et les moyens, en matière d'aménagement d'une zone urbaine.



validation officielle par les instances normatives comme l'ISO<sup>4</sup>. Norme et standard s'appliquent dans des domaines fortement structurés, ce qui n'est pas le cas de l'urbanisme. La différenciation entre "norme" et "standard" paraît se situer en grande partie au niveau des acteurs en jeu et des procédures de consensus attachées, le renvoi à la langue anglaise pour le monde industriel est d'ailleurs un indice intéressant sur le poids des acteurs à l'œuvre. "

On peut donc faire l'hypothèse que la fabrication des normes mobilise des dispositifs de coordination. Les réunions très formelles font partie du " répertoire des dispositifs" codifié par l'ISO. Elles constituent autant d'"épreuves", pour l'éditeur qui doit trouver la bonne formulation conduisant au consensus, et pour l'animateur du groupe, qui doit modérer les débats et faire en sorte d'arriver au consensus.

Plus concrètement une norme est un ensemble de règles de conformité, édictées par un organisme de normalisation au niveau national ou international. Un standard est un ensemble de recommandations émanant d'un groupe représentatif d'utilisateurs réunis autour d'un forum, comme l'IETF<sup>5</sup> (Internet Engineering Task Force), le W3C<sup>6</sup> (World Wide Web Consortium), le Dublin Core<sup>7</sup>. Le monde de l'Internet, dominé par des acteurs non institutionnels (entreprise, recherche) est né de standards de fait (TCP/IP, http, HTML) efficaces et simples et issus de normes (OSI, SGML) plus complètes mais complexes et impraticables. Les normes en retour s'adaptent aux standards définis.

Les processus d'élaboration des normes et standards sont propres aux communautés ou organisations concernées, ils suivent généralement une séquence générale :

- Expression du besoin.
- Processus d'élaboration et de décision: recherche de consensus, accord d'une communauté.
- Formalisation, validation.

---

<sup>4</sup> <http://www.iso.org/>

<sup>5</sup> <http://www.ietf.org/>

<sup>6</sup> <http://www.w3.org/>

<sup>7</sup> <http://dublincore.org/>

Dans le domaine politique nous nous intéressons à l'exemple suisse. La Suisse, au cours de son histoire, a su exploiter sa grande diversité sociale, linguistique et religieuse, pour créer une véritable démocratie et s'assurer un bien-être matériel enviable. Son secret ? Un art du consensus, qui pourrait nous inspirer lorsque nous désirons gérer efficacement des problèmes autrement que par le conflit.

Qu'est-ce qui fait marcher la Suisse ? On découvre des "Républiques et cantons" formant une confédération, dotée de quatre langues officielles. Alors que les gouvernements de coalition restent sinon honnis, du moins suspects dans d'autres pays, on observe en Suisse un pays dirigé depuis quatre décennies par un conseil fédéral regroupant tous les grands partis politiques. On pourrait donner d'autres exemples, tant la Suisse surprend par sa capacité à concilier les différends, et à tirer le meilleur de sa diversité. C'est donc tout naturellement que le secret suisse se dévoile, lui qui fait cruellement défaut à d'autres : la notion de consensus. Pléthore d'acteurs lors des prises de décision, complexité de celles-ci, et lenteur des changements, sont les prix à payer par le consensus à la Suisse. Alors consensus ou "consensusuisse" ?

L'art du compromis et du consensus n'est certainement pas une faculté naturelle chez l'homme, mais peut s'apprendre notamment grâce à des structures contraignantes, comme l'exemple Suisse tend à le montrer. Cet art trouve tout son sens dans les conflits d'intérêts dans le cas où des stratégies "gagnant-gagnant" existent, et ces situations abondent dans la réalité.

Qu'en est-il du consensus dans le domaine de l'urbanisme ? Est-il possible de mettre d'accord des acteurs aussi nombreux que variés ?

Pour illustrer la complexité du problème, examinons un exemple très simple, concernant le concept "rue". Depuis l'enfance, chacun sait ce qu'est une rue. Apparemment tous les acteurs peuvent atteindre facilement le consensus. En réalité le problème est beaucoup plus complexe. Considérons seulement trois ensembles d'acteurs constitués de "balayeurs de la voirie", de "facteurs de la poste" et d'"employés de la compagnie d'électricité" (voir figure 2.1).

Les balayeurs nettoient des rues ; en France, ils nettoient seulement les rues publiques tandis que des rues privées doivent être nettoyées par des résidents.

Les facteurs distribuent des lettres aux résidents vivant dans des rues publiques et privées. Mais si une rue n'a aucun résident, cette rue n'est pas considérée par des facteurs.

Les employés de la compagnie d'électricité considèrent seulement les rues dans lesquelles leur compagnie fournit l'électricité.

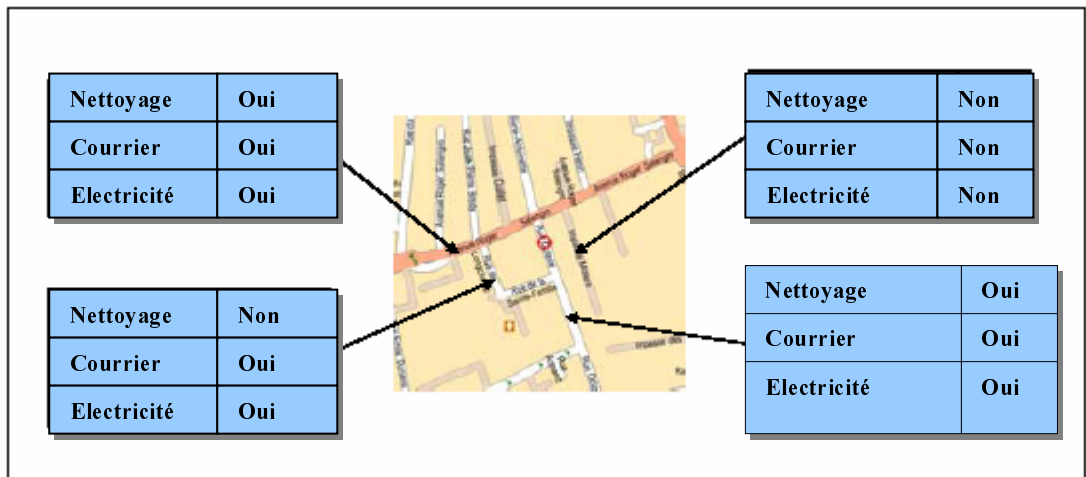


FIGURE 2.1. DIVERS TYPES DE RUES.

Concernant toujours le concept "rue", nous pouvons montrer que ce concept peut avoir diverses définitions selon les points de vue d'acteurs différents (voir figure 2.2).

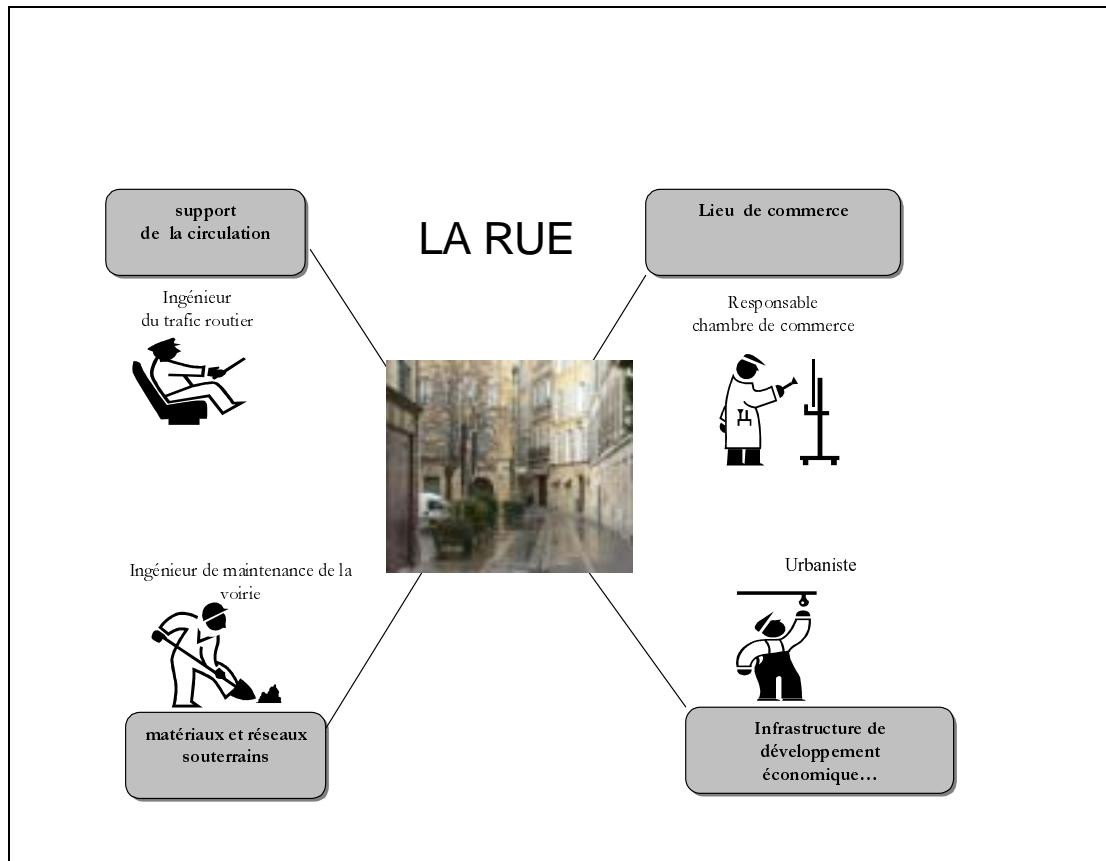


FIGURE 2.2. DIVERSES DEFINITIONS DE RUES.

Par exemple les ingénieurs du trafic routier considèrent la rue comme un support de la circulation.

Les ingénieurs de maintenance de la voirie définissent la rue comme matériaux et réseaux souterrains.

Les agents commerciaux voient la rue comme un lieu abritant les commerces.

Les urbanistes considèrent la rue comme une infrastructure de développement économique et social.

Ces exemples, dans lesquels nous pouvons facilement multiplier les acteurs, prouvent que les dispositifs urbains de base comportent à priori une définition unique, mais en réalité après avoir regardé dans les détails, ils comportent des définitions multiples.

Vu la complexité qui caractérise le domaine de l'urbanisme, quelle solution proposée pour organiser et harmoniser les différentes terminologies employées,

dans le domaine, qui peuvent amener à des ambiguïtés ? Est-il possible d'harmoniser les différents points de vues des acteurs sur les objets du domaine d'étude, qui aboutissent à des définitions différentes d'un même objet ? L'ontologie pourrait-elle être la solution ? Peut-on considérer qu'une ontologie est capable de représenter toute la complexité du domaine de l'urbain ? Les urbanistes ont-ils besoin d'ontologie ? Si oui pourquoi faire ? La section suivante essaye de donner un début de réponses à ces questions.

### **2.3 Les ontologies : quel intérêt pour les urbanistes ?**

Nos travaux ont commencé dans un contexte marqué par l'inexistence (à notre connaissance) d'ontologie pour l'urbanisme et l'aménagement du territoire. Cela a procuré un fort intérêt à nos travaux, dont l'objectif final est de satisfaire un besoin de coordination et de coopération entre les acteurs de l'urbain.

Nous nous sommes inspirés du domaine des Systèmes d'Information Géographique (SIG), où l'ontologie est reconnue comme une initiative de recherches émergente. Davantage de discussions sur la pertinence de l'ontologie dans les SIG sont fournies [SMIT98a] [MARK01]. Le but primaire d'utiliser l'ontologie dans cette discipline est de définir un vocabulaire commun qui permettra l'interopérabilité et réduira au minimum tous les problèmes en intégration de données, à la fois entre systèmes différents et entre utilisateurs et systèmes. L'ontologie est généralement utilisée comme une méthode : pour identifier des concepts, des relations et des règles ([TVER84], [SMIT98b], [SORR99]), pour définir et conceptualiser la connaissance dans un domaine pour la rendre plus facile à modéliser, et pour fournir le vocabulaire et les règles normalisés pour l'application de ce vocabulaire. Dans les SIG, la définition largement citée de l'ontologie est une "spécification d'une conceptualisation" (Gruber 1995), et l'ontologie est appliquée comme la "définition ou l'identification des entités qui peuvent agir en tant que référents pour la capture du monde réel" ([FRAN97], [MARK99], [PEUQ99], [MILT03]. L'ontologie est souvent définie dans les SIG comme la "science de l'existant" [RAUB01]. Les deux approches distinctes suivies dans les SIG sont :

- l'approche philosophique pour l'identification des catégories "top-level" à partir d'une ontologie formelle,

- et l'approche en Ingénierie de Connaissances (IC) portée sur la spécification d'un domaine et basée sur la tâche d'explicitier des actions, des termes et des relations avec une spécification particulière à partir du langage naturel vers des spécifications formelles rigoureuses.

Sowa [SOWA02] définit l'ontologie comme "l'étude globale des catégories des choses qui pourraient exister dans un domaine", alors que Timpf [TIMP02] annonce que les ontologies sont des vues simplifiées du monde que nous souhaitons représenter "pour un certain but", et qu'elles incluent les concepts qui sont appropriés à cette tâche particulière dans le domaine.

Mais la définition de l'ontologie la plus admise dans la communauté de l'Ingénierie de Connaissance (IC) est celle de Gruber [GRUB93]: "*Une ontologie est une spécification explicite d'une conceptualisation d'un domaine.*", qui a été améliorée par Studer [STUD98] : "*Une ontologie est une spécification formelle et explicite d'une conceptualisation partagée.*"

Cette dernière définition repose sur des notions comme le consensus qui pourraient aider les urbanistes à organiser et harmoniser leurs vocabulaires.

En définitive, nous pouvons affirmer que les ontologies sont surtout utilisées dans le cadre de l'interopérabilité des données, et dans les systèmes de conception coopératifs (Groupware). Elles peuvent ainsi contribuer à plusieurs objectifs :

- faciliter la communication entre les hommes et les machines en clarifiant le vocabulaire pour éviter les dérives sémantiques de compréhension entre interlocuteurs.
- structurer et organiser les documents de travail d'une communauté d'utilisateurs pour faciliter leur manipulation et leur recherche [CALA04].
- raisonner par inférence, particulièrement dans de très grandes bases de données.

Ainsi l'intérêt de l'ontologie urbaine serait de faciliter, à terme, l'interopérabilité entre les modèles de bases de données urbaines et entre différents systèmes de conception coopératifs, aussi bien que l'organisation du vocabulaire de l'urbain et

la communication entre acteurs de l'urbanisme. Concrètement cette ontologie permettra de :

- identifier les termes et les concepts utilisés dans différentes activités de l'urbanisme,
- organiser les connaissances du domaine urbain,
- faciliter la communication entre plusieurs acteurs de l'urbanisme manipulant les mêmes types d'objets pour accomplir différentes tâches,
- regrouper les données urbaines provenant de sources hétérogènes.

L'ontologie pourrait donc représenter un réel apport de compréhension sur le domaine complexe de l'urbanisme et son adaptabilité permettrait d'aboutir à un consensus (du moins sur les terminologies) entre différents acteurs du domaine de l'urbain. Nous rappelons que la conception d'une ontologie est encore un processus coopératif, qui comporte des similitudes avec la conception en proprement parler en urbanisme [LAUR01].

C'est ces hypothèses que nous allons essayer de valider dans nos contributions, dans la suite de la thèse. Mais avant, nous présentons le projet Towntology qui a servi de cadre à nos travaux.

## **2.4 Le projet Towntology**

Le projet Towntology, démarré en 2002- 2003, a été d'abord une étude financée par l'INSA de Lyon par l'intermédiaire du bonus qualité recherche (BQR). Il est issu de la collaboration de deux équipes de recherche de domaines différents : les systèmes d'information et le développement urbain pour atteindre le même objectif, la construction d'une ontologie urbaine. Ces deux équipes travaillent au sein de deux laboratoires de recherche basés à Lyon :

- Le Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS). Dans ce laboratoire, se trouvent des compétences en système d'information et aussi en ingénierie des connaissances. Le rôle de l'équipe du LIRIS est la mise en

- place d'un outil d'aide à la conception d'ontologie dans le domaine de l'urbanisme et de l'aménagement du territoire.

- Le laboratoire Environnement Ville et Société (EVS) et plus particulièrement l'Equipe Développement Urbain (EDU). Le but de cette équipe est de fournir les compétences du domaine de l'urbain pour ce projet. Cette équipe joue le rôle d'expert du domaine dont on cherche à modéliser les connaissances.

Le partenariat entre les deux laboratoires permettait au LIRIS de disposer d'un champ d'expérimentation concret pour la mise au point d'outils informatiques spécifiques à la gestion d'ontologie. Mais cela permettait également à l'EDU de disposer d'une organisation de son vocabulaire.

Depuis janvier 2005, le projet Towntology<sup>8</sup> s'est vu transformé en projet européen COST UCE 21 (regroupant actuellement une quinzaine de laboratoires), et qui vise à construire les ontologies urbaines dans différentes langues européennes couvrant la totalité du champ urbain.

#### **2.4.1 Les objectifs du projet**

Le but du projet est de définir une ontologie utilisée à la fois pour enseigner l'urbanisme, proposer aux experts un cadre de référence, indexer leur documentation, aider à la recherche d'information ou la formation du personnel, et finalement faire collaborer des divers intervenants en urbanisme.

Le projet Towntology s'atèle à la construction d'une ontologie urbaine, c'est à dire constituer un vocabulaire formalisé regroupant pour le domaine de l'urbanisme et de l'aménagement du territoire, l'ensemble des concepts, et de leurs interrelations. L'objectif principal de ce projet est la construction de la base d'une ontologie urbaine, autrement dit la collecte des termes utilisés dans cette discipline et leur intégration dans un réseau sémantique. Les buts sont les suivants :

- identifier les termes et les concepts utilisés dans différentes activités de l'urbanisme,
- organiser les connaissances du domaine urbain,

---

<sup>8</sup> <http://www.towntology.net>



- faciliter la communication entre plusieurs acteurs de l'urbanisme manipulant les mêmes types d'objets pour accomplir différentes tâches,
- regrouper les données urbaines provenant de sources hétérogènes,
- construire des ontologies dans d'autres langues européennes.

L'intérêt de notre ontologie est de faciliter, à terme, la communication entre les différents acteurs du domaine de l'urbanisme, aussi bien qu'entre les modèles de base de données et entre différents systèmes de conception coopératifs.

#### **2.4.2 Les différentes étapes du projet Towntology**

Le projet se décompose comme suit :

1. Construire les bases d'une ontologie urbaine, en rassemblant les termes utilisés dans le domaine de l'aménagement et de l'urbanisme, dans un réseau sémantique. (EDU)
  - a- Trouver les ressources adéquates dans le domaine de l'urbanisme. Les ressources peuvent être des bases de données, des dictionnaires, des livres ou tout autre document. Le but est de trouver des ressources facilement exploitables pour construire une terminologie.
  - b- Analyser les contenus des ressources pour sélectionner les termes importants du domaine.
  - c- Faire un premier inventaire des termes et les définir ;
  - d- Sélectionner à partir des termes et de leurs définitions, les concepts structurants et construire un vocabulaire standardisé ;
  - e- Définir les relations nécessaires pour organiser les concepts entre eux;
  - f- Valider ce premier niveau de structuration par des utilisateurs et l'enrichir.
  
2. Elaborer une maquette afin d'en tester la faisabilité et l'intérêt. (LIRIS)
  - a- Analyser les systèmes de construction d'ontologies et choisir un outil ;
  - b- Présenter une ontologie sous forme graphique ;
  - c- Naviguer et interroger une ontologie ;

- d- Stocker une ontologie dans un des langages de représentation des ontologies ;
- e- Proposer un support collaboratif pour la construction d'ontologies.

## **2.5 Conclusions**

Dans cette partie nous avons décrit le contexte, "planté le décor" dans lequel nos travaux ont été réalisés. Ce qui nous a amené à nous intéresser à la complexité et à la diversité qui caractérisent le domaine de l'urbanisme et de l'aménagement du territoire. Ensuite, comme solutions à ces difficultés, nous avons posé des hypothèses portant sur l'utilisation d'un mécanisme de consensus et de l'ontologie en vue d'atteindre nos objectifs, qui est d'offrir aux urbanistes un cadre unifié de travail sans ambiguïté. Enfin, nous avons présenté le projet Towntology qui a servi de cadre à nos travaux.

La partie suivante de la thèse présente l'état de l'art sur les ontologies et le groupware.

## **Partie 3. Les ontologies et le groupware (état de l'art)**

Dans cette partie, nous présentons un état de l'art complet sur les ontologies. D'abord nous donnons les différentes définitions, les différents types, composants et usages des ontologies. Ensuite, nous décrivons les approches, méthodes et méthodologies de construction d'ontologie. Puis nous nous intéressons aux langages de description et outils de construction des ontologies. Enfin nous présentons les technologies de groupware en général et à leurs applications dans le domaine des ontologies.

### Chapitre 3. Qu'est-ce qu'une ontologie ?

Dans ce chapitre, nous donnons les définitions importantes des ontologies à travers des disciplines diverses. Ensuite nous décrivons les différents types d'ontologie existants, les composants et les usages des ontologies. Enfin nous analysons et comparons les approches existantes de conception d'ontologie.

Le mot ontologie tire son origine de la tradition philosophique, mais a trouvé une application dans plusieurs disciplines diverses. L'interprétation du mot ontologie varie selon les différentes disciplines. Les premières divergences ont existé entre la pensée philosophique (ou métaphysique) et l'approche de l'Intelligence Artificielle (IA) (ou système d'information). D'autres confusions dans la définition et l'application de l'ontologie existent dans les distinctions (pas claires) entre ontologie (au singulier) et ontologies (au pluriel), ontologie et base de connaissance, et ontologie et épistémologie.

En logique, le quantificateur existentiel  $\exists$  est une notation pour évaluer que quelque chose existe, mais la logique elle-même n'a pas de vocabulaire pour décrire les choses qui existent. Les ontologies remplissent cet espace, et elles sont utilisées pour étudier l'existence de toutes sortes d'entités, abstraites et concrètes, qui composent le monde [SOWA00]. Dans le dictionnaire anglais d'Oxford [OED03], l'ontologie est définie comme appartenant au "domaine métaphysique" et comme la "science ou l'étude de l'existant". Selon Wordnet [WORD98], l'ontologie peut être définie comme "l'étude métaphysique de la nature de l'être et de l'existant". La première tentative d'explicitation de la réalité par une ontologie a été celle d'Aristote en 340 avant Jésus Christ, dans ses travaux sur les "catégories" [MAN00], adoptée plus tard par Platon. Aristote considère la réalité comme "toutes espèces de quasi-existence et les attributs qui appartiennent à celle-ci" (Métaphysiques IV.1 d'Aristote [GUAR95]). Cette étude d'expliquer la réalité par des concepts, des relations et des règles a depuis lors été reconnue comme ontologie [AUDI95]. Elle a été introduite dans la prominence au 18ème siècle par Christian Wolff (1679-1775), un rationaliste allemand, pour qui l'ontologie était une discipline déductive menant aux vérités nécessaires sur des essences des êtres. Bien que ses opinions sur l'ontologie comme système déductif aient été réfutées plus tard par Kant (1724-1804), avec la rénovation au 20ème siècle de la

métaphysique, l'ontologie ou la pensée ontologique est encore devenue importante, notamment parmi les phénoménologues et les existentialistes. Cette spécification de ce qui "constitue la réalité" forme la base d'une ontologie décrite à partir d'une perspective philosophique [SMIT95b]. Cependant, aucun critère clair pour ce qui constitue l'"existant" et comment l'identifier n'est défini. Même le célèbre critère de Quine [QUIN92] sur l'"existant" ("exister c'est être la valeur d'une variable quantifiée") est critiqué pour manque de spécificité. L'ontologie en philosophie est présentée sous forme de taxonomies ou de classifications hiérarchiques et est représentative du monde dans sa totalité. Dans ce contexte, les travaux de Husserl ([SMIT89], [POLI93], [SMIT95]) sur l'ontologie formelle sont significatifs, fournissant des distinctions entre le matériel et les perspectives orientées logique sur l'ontologie métaphysique ; avec des instanciations significatives dans l'ontologie, et formant la base des traditions formelles dans le développement d'ontologie.

Il existe également des définitions de l'ontologie basées sur un domaine spécifique et dépendantes du point de vue de l'utilisateur. [SOWA00] définit l'ontologie comme "la méthode pour extraire un catalogue des choses ou des entités (C) qui existent dans un domaine (D) selon la perception d'une personne qui se sert d'un certain langage (L) pour le décrire". Contrairement à la perception "réaliste", la perception du domaine spécifique sur l'ontologie est le principe fondamental pour son application en science de l'information et en ingénierie de connaissance (IC). Une des définitions les plus récentes en IC est que "*l'ontologie définit les termes et les relations de base compris dans le vocabulaire d'un domaine donné aussi bien que les règles pour combiner les termes et les relations pour définir des extensions du vocabulaire*" [NECHES91]. Quelques années plus tard, [GRUB93a] définissait une ontologie comme "*une spécification explicite d'une conceptualisation*". Cette définition est devenue la plus référencée en littérature et par la communauté d'ontologie. Basées sur la définition de Gruber, plusieurs autres définitions ont été proposées.

Guarino et Giaretta [GUAR95] ont fourni une autre clarification terminologique entre les différentes utilisations de l'ontologie comme un "système conceptuel informel", un "état sémantique formel", une "spécification d'une conceptualisation", une "représentation d'un système conceptuel par

l'intermédiaire d'une théorie logique", le "vocabulaire utilisé par une théorie logique", et une "spécification d'une théorie logique".

Borst et son équipe [BORS97] ont amélioré la définition de Gruber en affirmant que "les ontologies sont définies comme une spécification formelle d'une conceptualisation partagée". L'équipe de Swartout [SWAR96] déclare que "une ontologie est un ensemble de termes hiérarchiquement structuré pour décrire un domaine qui peut être utilisé comme base squelettique pour une base de connaissances".

La définition de Studer [STUD98], déjà énoncée dans l'introduction de la thèse, fusionne les définitions de Gruber et de Borst, en fournissant une explication des termes généralement employés pour définir les ontologies en IA et IC.

Une ontologie est donc la manifestation d'une compréhension partagée d'un domaine entre un certain nombre d'agents, et un tel accord facilite des communications précises et efficaces qui mènent, à terme, à d'autres avantages tels que l'interopérabilité, la réutilisation et le partage.

Le but principal de l'application d'ontologie en intelligence artificielle n'est pas de servir d'argument déductif pour définir la réalité, mais de permettre une explication des termes et des significations pour définir une base consensuelle pour l'interopérabilité dans le domaine. Par opposition à la perception philosophique, il n'y a pas de "vérité absolue" dans la perception de l'IA ; la spécification dans l'ontologie dépend de l'utilisateur et est développée comme un mécanisme pour atteindre un accord (consensus) sur les différentes interprétations de la réalité. Les deux perceptions sur l'ontologie fournissent différents principes et approches pour la conception et le développement d'ontologie.

Il peut exister d'autres malentendus entre les définitions des mots "Ontologie" et "ontologies". Et il est ironique qu'on propose l'ontologie comme un mécanisme pour résoudre les problèmes sémantiques et arriver à un consensus, et qu'au même moment une compréhension complète et une définition consensuelle pour le mot ontologie lui-même ne soient pas encore une réalité. On pourrait même définir une "ontologie des ontologies". En effet les différentes manières que l'ontologie est comprise, et les différentes manières dont la notion est exprimée (principalement les distinctions entre minuscule/ majuscule et singulier/pluriel) causent la confusion dans la communauté des concepteurs et des chercheurs, et

aucun rapport consensuel n'est encore de mise. Guarino et Giaretta [GUAR95] proposent d'utiliser les mots 'Ontologie' (avec 'O' en lettre capitale) et 'ontologie' (avec 'o' minuscule) pour se référer aux sens philosophique et IC respectivement.

Pour la perception philosophique, l'ontologie est le "concept d'un existant" [FRAN97, SMIT98a, MAN00, MARK00], et l'utilisation des "ontologies" (au pluriel) dans ce cadre est non pertinente. Cependant, dans l'approche IC, pour une perception API, l'ontologie n'est pas représentative d'une vérité singulière, et l'utilisation des "ontologies" (au pluriel) est appropriée. Les sous-distinctions entre ontologie et ontologies existent, mais il est seulement d'usage générale de faire une distinction entre un et plusieurs modèles de description.

Dans cette section nous avons rassemblé les définitions les plus importantes du mot ontologie, bien que d'autres définitions puissent être trouvées dans la littérature de l'Intelligence Artificielle ou ailleurs. Les différentes définitions fournissent des points de vue différents et complémentaires de la même réalité. Certains auteurs fournissent les définitions qui sont indépendantes des processus suivis pour la construction de l'ontologie et de son utilisation dans les applications, alors que d'autres définitions sont influencées par son processus de développement.

Comme conclusion principale de cette section, nous pouvons dire que les ontologies visent à capturer la connaissance consensuelle d'une manière générique, et qu'elles peuvent être réutilisées et partagées à travers des applications de logiciel et par des groupes de personnes. Elles sont habituellement construites d'une manière coopérative par différents groupes de personnes situées dans différents endroits.

L'exemple ci-dessous (Figure 3.1) représente un petit extrait simplifié d'une ontologie des catastrophes naturelles, avec deux types de relations.

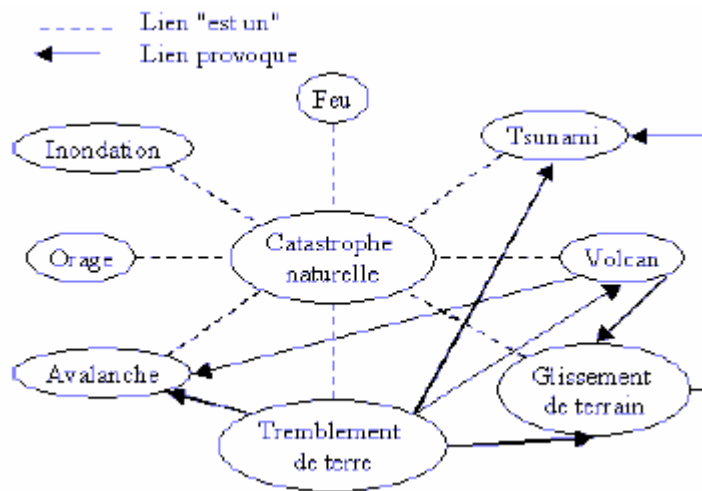


FIGURE 3.1 UN EXEMPLE D'ONTOLOGIE

### 3.1 Les types d'ontologies

Nous pouvons catégoriser les ontologies selon leur formalisation, leur type de conceptualisation, leur propos, leur niveau de complexité, et selon le nombre de points de vue des concepteurs, etc.

Selon l'état de la formalisation, Uschold et Gruninger [USCH96] ont distingué quatre types d'ontologies selon le type de langage utilisé pour les implémenter.

Selon eux, il existe :

- **des ontologies fortement informelles**, si elles sont écrites dans un langage naturel ;
- **des ontologies semi-informelles**, si elles sont exprimées dans une forme restreinte et structurée de langage naturel (c.-à-d. en utilisant des modèles) ;
- **des ontologies semi-formelles**, si elles sont définies dans un langage artificiel et formellement défini (exemple : Ontolingua [FARQ97], OWL [DEAN03]);
- **des ontologies rigoureusement formelles**, si elles sont définies dans un langage avec la sémantique formelle, les théories et les preuves des propriétés telles que la solidité et la perfection.

Notons que selon la définition de [STUD98], une ontologie fortement informelle ne serait pas une ontologie puisqu'elle n'est pas compréhensible par la machine.



L'équipe de Mizoguchi [MIZO95] et de Van Heijst [VANH96] fournissent une typologie exhaustive des ontologies basée sur la nature de la conceptualisation. Nous pouvons les classer comme suit :

- **Les ontologies de représentation de connaissance (RC)** [VANH97], qui définissent des primitives de représentation utilisées pour formaliser la connaissance avec un paradigme donné. Les exemples les plus représentatifs sont la "Frame Ontology" [GRUB93a] et la "OKBC<sup>9</sup> Ontology", toutes les deux sont accessibles sur le serveur Ontolingua. Elles fournissent des définitions formelles de primitives de représentation utilisées principalement dans les langages basés sur les "frames" (c'est-à-dire, des classes, sous-classes, attributs, valeurs, relations et axiomes). Elles permettent la construction d'autres ontologies par des conventions basées sur les "frames".  
En plus de la "Frame Ontology" et de la "OKBC Ontology", nous pouvons citer d'autres ontologies de RC comme celles de RDF(S)[BRIC03], OIL [HORR00], DAML+OIL [HORR01], et OWL [DEAN03].
- **Les ontologies génériques** [VANH97] ou les ontologies communes [MIZO95], qui sont utilisées pour représenter la connaissance commune (consensuelle) réutilisable dans les domaines. Ces ontologies incluent le vocabulaire lié aux choses, aux événements, au temps, à l'espace, à la causalité, au comportement, à la fonction, à la mereologie, etc.  
La "Mereology Ontology" [BORS97] et la gène-ontologie [GENE04] sont les exemples les plus classiques d'ontologie générique. Cette ontologie définit les propriétés que devrait avoir toute décomposition.
- **Les ontologies de haut niveau** ("Top-level ou Upper-level ontologies"), qui décrivent des concepts très généraux et fournissent des notions générales sous lesquelles tous les termes racines dans les ontologies existantes devraient être liés. Le principal problème est qu'il y a plusieurs ontologies de haut niveau et elles diffèrent sur les critères suivis pour classifier les concepts les plus généraux de la taxonomie. Les exemples d'ontologies de haut niveau sont : SOWA [SOWA99], CYC [LENA90], et SUO<sup>10</sup> [PEAS02].
- **Les ontologies de domaine** [MIZO95], qui sont réutilisables dans un domaine spécifique donné. Ces ontologies fournissent les vocabulaires sur les concepts

---

<sup>9</sup> Ontology Knowledge Base Connectivity

<sup>10</sup> Standard Upper Ontology

et leurs relations dans un domaine, sur les activités qui ont lieu dans ce domaine, et sur les théories et les principes élémentaires régissant ce domaine. Il y a une frontière claire entre les ontologies de domaine et les ontologies de haut niveau. Les concepts dans les ontologies de domaine sont habituellement des spécialisations des concepts déjà définis dans les ontologies de haut niveau, et le même principe pourrait se produire avec les relations. Exemple : e-commerce, médecine, science de l'ingénieur (modèles mathématiques), entreprise, chimie, etc...

- **Les ontologies de tâche** [MIZO95, GUAR98], qui décrivent le vocabulaire relié à une tâche générique ou à une activité (comme diagnostiquer, programmer, vendre, etc.) en spécialisant les termes dans les ontologies de haut niveau. Les ontologies de tâche fournissent un vocabulaire systématique des termes utilisés pour résoudre les problèmes liés aux tâches qui peuvent ou ne peuvent appartenir au même domaine. Ce sont des ontologies dépendantes des applications. Les ontologies de tâche de domaine ("Domain-Task ontologies") sont des ontologies de tâche réutilisables dans un domaine donné, mais pas à travers des domaines. Elles se veulent indépendantes des applications.
- **Les ontologies de méthodes** [TIJE93], qui donnent les définitions des concepts et des relations appropriés appliquées pour spécifier un processus de raisonnement afin de réaliser une tâche particulière. Une ontologie sur une activité par les moyens de la décomposition de tâche appartiendrait à cette catégorie.
- **Les ontologies linguistiques (OL)**. Le but de ce type d'ontologie est de décrire les constructions sémantiques plutôt que de modéliser un domaine. Elles offrent tout à fait une quantité hétérogène de ressources, utilisées la plupart du temps dans le traitement de langage naturel. La principale caractéristique de ces ontologies est qu'elles sont liées à la sémantique des unités grammaticales (mots, groupes nominaux, adjectifs, etc). Les exemples de OL sont : EuroWordnet [VOSS99], Sensus [KNIG95], Goi-Taikai [IKEH97], WordNet [FELB98].

Une autre catégorisation des ontologies peut se faire selon leur propos. On distingue :

- **Les ontologies d'application** [VANH97], qui sont dépendantes des applications. Elles contiennent toutes les définitions nécessaires pour modéliser la connaissance requise pour une application particulière. Les ontologies d'application étendent et spécialisent souvent le vocabulaire des ontologies de domaine et de tâche pour une application donnée.
- **Les ontologies de référence**, qui sont proposées dans la perception philosophique comme des théories pour "des réalités indépendamment existantes" [SMIT99]. D'autre part, l'approche de IC suit deux types distincts : d'abord, quand le propos porte sur la conception de la théorie et si l'axiomatisation riche est nécessaire pour définir les engagements ontologiques et en second lieu, quand le propos est le fait d'avoir une ontologie comme une application fonctionnelle en utilisant la terminologie minimum et les taxonomies simples [GUAR95]. Par opposition à formelle, l'ontologie axiomatisée constitue une description détaillée du domaine. Les ontologies de référence permettent de clarifier les définitions entre les communautés en expliquant la signification des termes qui sont inclus dans l'ontologie de haut niveau ("top-level") [BORG02].

La catégorisation selon le niveau de complexité se réfère à celle de Lassila et McGuinness [LASSI01], qui classifie les différents types d'ontologies, allant des plus légères ou "**LightWeight ontologies**" (exemples : catalogues, glossaires, thésaurus, etc) aux plus lourdes ou "**HeavyWeight ontologies**" (exemples : ontologies expressives avec des axiomes et fonctions). Parfois la notion d'ontologie est diluée, dans le sens que les taxonomies sont considérées comme des ontologies [STUD98]. Par exemple, UNSPSC<sup>11</sup>, e-class<sup>12</sup>, et RosettaNet<sup>13</sup>, des propositions pour des standards dans le domaine du e-commerce, et l'Annuaire Yahoo, une taxonomie pour la recherche sur le Web, sont également considérés comme des ontologies [LASSI01] parce qu'ils fournissent une conceptualisation consensuelle d'un domaine donné.

---

<sup>11</sup> <http://www.unspsc.org/>

<sup>12</sup> <http://www.eclass.org/>

<sup>13</sup> <http://www.rosettanet.org/>

La catégorisation selon le nombre de points de vue de concepteurs donne deux types d'ontologies [YILI03] :

- **Les ontologies inspirationnelles**, dans le cas où l'ontologie est conçue selon le point de vue individuel d'un concepteur sur le domaine,
- **Les ontologies collaboratives**, c'est le cas de la conception avec de multiple points de vue de différents concepteurs et acteurs sur le domaine.

Dans nos travaux, nous nous sommes rendus compte que nous pouvons aussi catégoriser les ontologies selon l'état (consensuel ou pas) des définitions (Figure 3.2). Nous distinguons alors :

- **Les ontologies post-consensuelles**, lorsque le consensus est déjà atteint sur les définitions. Ainsi, dans la conception d'ontologie post-consensuelle, l'ontologie est écrite dans un langage tel que les logiques de description dans lesquelles certaines caractéristiques déductives sont importantes pour assurer l'interopérabilité, particulièrement en servant de base à la génération de médiateur. Une spécification principale est qu'il y a seulement une définition par concept.
- **Les ontologies pré-consensuelles**, lorsque le consensus n'est pas encore atteint sur les définitions. Ce type d'ontologies se caractérise par son appartenance aux premières phases de construction des ontologies. Dans ces phases, les définitions sont principalement informelles et multiples et constituent seulement un réseau sémantique. Nous y trouvons différentes terminologies employées qui peuvent donner lieu à des ambiguïtés. D'autre part, la conception d'ontologie pré-consensuelle doit tenir compte de la variété d'acteurs, c'est-à-dire doit aider ces derniers à atteindre le consensus. En d'autres termes, nous devons considérer différentes définitions et les garder (lisibles, éditables) jusqu'à ce qu'un certain accord (consensus) soit atteint.

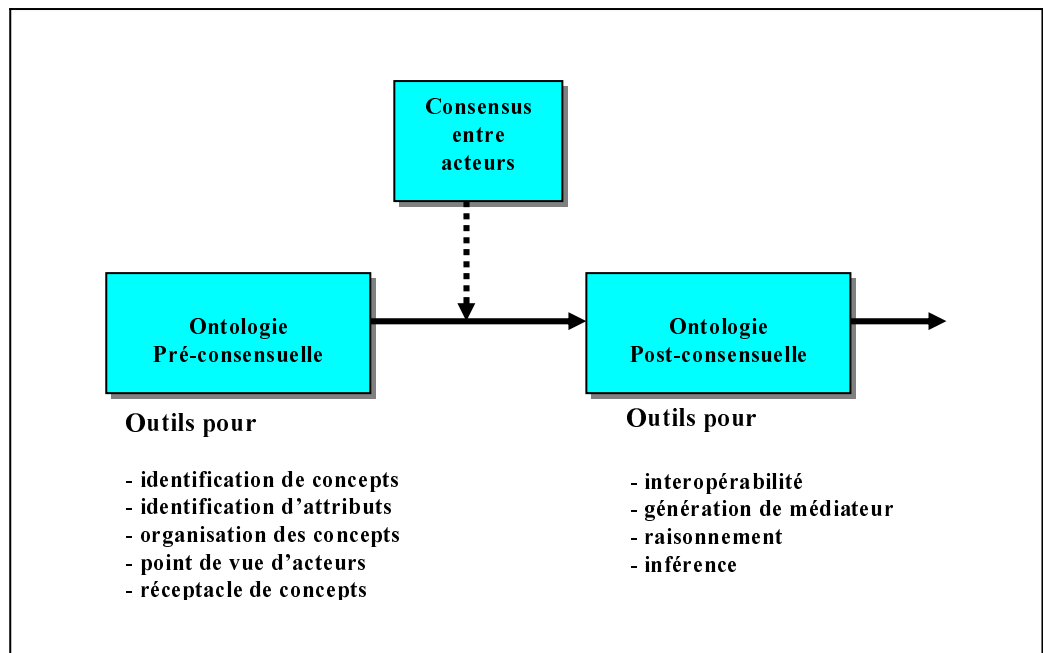


FIGURE 3.2 DIFFERENCES ENTRE DES ONTOLOGIES PRE-CONSENSUELLES ET POST-CONSENSUELLES.

### 3.2. Les composants d'une ontologie

"Une ontologie peut prendre une variété de formes, mais nécessairement elle inclura un vocabulaire des termes, et des spécifications de leurs significations" [USCH98]. Elle inclut les définitions et une indication de la façon dont les concepts sont reliés, qui imposent collectivement une structure au domaine et contraignent les interprétations possibles des termes. Dans cette section, nous nous occupons uniquement des composants de l'ontologie en Ingénierie de Connaissances et en Intelligence Artificielle [GREN04] [FIEL04].

Maedche et Staab [MAED01] déclarent qu'une ontologie comporte un lexique des concepts, des relations et des axiomes. A partir de différentes études [GRUB93, USCH96, STUD98, CHAN99, ASPI01], nous pouvons récapituler les composants primaires d'une ontologie comme suit :

- **Les "classes" ou "catégories"**. Une classe est un ensemble d'objets et la base de la représentation de la connaissance dans une ontologie. Les catégories incluent les concepts [SOWA00], et tandis que les concepts sont liés aux

sémantiques cognitives, les catégories sont universelles dans le (monde) réel. Un concept représente un ensemble ou classe d'entités ou des "choses" dans un domaine.

- **Les "relations"** représentent des types d'interactions entre les classes (ou concepts) dans une ontologie. Les relations peuvent être taxonomiques, associatives, nominales, locatives, fonctionnelles ou causatives. Encore d'autres spécifications axiomatiques pour des relations spécifient des relations formelles de mereologie, d'identité, d'unité, d'essence, de rigidité et d'autres critères de dépendance. Dans le contexte géographique, la topologie et la méréologie sont des relations significatives en raison de la dépendance spatio-temporelle des entités géographiques [SMIT96, VARZ01a].
- **Les "axiomes"** sont utilisés pour modeler les conditions qui sont toujours vraies pour le domaine. Une spécification axiomatique riche est critique en distinguant une ontologie à partir d'une taxonomie terminologique.

Bien que la sémantique ne soit pas explicitement énoncée comme composant primaire dans les spécifications, elle est un aspect crucial dans la conception d'une ontologie à partir d'une classification terminologique. Les sémantiques sont le lien entre le langage et la pensée, et impliquent la signification qui est attachée aux objets et aux concepts du monde réel [JACK83]. Une ontologie peut également être conçue sans inclure la sémantique, mais elle ne sera pas une ontologie explicite. L'énoncé et la considération explicites de la sémantique permet de mieux fusionner et partager des ontologies. Par conséquent, un contenu sémantique permet la construction d'ontologies fiables et consensuelles. C'est particulièrement vraie pour le domaine géographique (et de l'urbanisme), où expliquer la signification est important pour résoudre l'hétérogénéité et les disparités du domaine.

Dans la section suivante nous parlons de l'usage et de l'utilité des ontologies dans les différents domaines.

### 3.3 Usages des ontologies

Les ontologies, au début, ont connu une large utilisation dans le domaine de l'intelligence artificielle.

Aujourd'hui les ontologies sont largement utilisées pour différents buts (traitement de langage naturel, gestion de la connaissance, e-commerce, intégration intelligente d'information, le Web sémantique, etc.) dans différentes communautés (c'est-à-dire, ingénierie de connaissance, bases de données et génie logiciel).

Notons que la communauté de bases de données aussi bien que la communauté de conception orientée objet construisent également des modèles de domaine en utilisant les concepts, les relations, les propriétés, etc., mais la plupart du temps les deux communautés imposent des contraintes moins sémantiques que celles imposées dans les ontologies lourdes.

Dans le commerce électronique, les ontologies sont habituellement utilisées pour représenter les produits et les services qui sont offerts dans les systèmes du e-commerce et qui sont donnés aux utilisateurs dans les catalogues pour consultation [LEGE00].

La vision du Web sémantique [BERN00] est d'ajouter la sémantique compréhensible par la machine (méta- information) au World Wide Web en utilisant une ontologie pour définir et organiser cet espace de méta-information. Le Web sémantique vise à réaliser l'intégration de toutes les sources d'informations sur le Web, permettant la réutilisation des données à travers les applications et rendant la recherche intelligente sur Internet possible.

Comme synthèse, nous pouvons dire que les ontologies possèdent un grand nombre d'applications et d'usages tels que :

- Les systèmes de Recherche d'informations (RI).
- Les inférences, pour découvrir les incohérences sur les connaissances ou les données.
- Les systèmes à base de connaissances dédiés à la résolution de problèmes.
- L'aide au diagnostic : des maladies, des problèmes techniques etc.
- L'interopérabilité des systèmes d'information.

- La coopération des logiciels.
- Les systèmes de traitement automatique du langage naturel.

Dans la section qui suit, nous nous intéressons aux différentes approches de conception d'ontologies.

### 3.4 Les différentes approches de conception d'ontologie

Il existe cinq approches pour la conception d'ontologie : inspirationnelle, inductive, déductive, synthétique et collaborative [YILI03]. Le tableau ci-dessous décrit ces approches.

TABLEAU 3.1 LES APPROCHES POUR LA CONCEPTION D'ONTOLOGIE [YILI03].

<b>Approches de conception d'ontologie</b>	<b>Base de la conception</b>
<i>Inspirationnelle</i>	<i>Point de vue individuel sur le domaine</i>
<i>Inductive</i>	<i>Cas spécifique dans le domaine</i>
<i>Déductive</i>	<i>Principes généraux sur le domaine</i>
<i>Synthétique</i>	<i>Ensemble d'ontologies existantes, chacune fournit une caractérisation partielle du domaine</i>
<i>Collaborative</i>	<i>Multiple points de vue sur le domaine, possibilité de travailler sur une ontologie d'ancrage</i>

Dans l'approche *inspirationnelle*, un concepteur seul prend les décisions de rassembler les conditions de l'analyse de domaine, de la conception et de la vérification de l'ontologie. Le développeur doit être à la fois expert du domaine et expert en conception d'ontologie afin d'assurer le succès de la conception, et surtout, pour assurer l'adoption de l'ontologie par la communauté d'utilisateurs. Ce processus dépend fortement de la créativité d'une seule personne, de son inspiration et de sa perception personnelle du domaine. Cette approche est souvent



appliquée dans un domaine qui peut être bien cerné (compris) par un seul concepteur, où ce dernier est capable de dominer le processus de conception et assurer l'adoption du produit final. Cependant la méthode deviendra aussitôt inefficace, quand on est dans un domaine large avec de la connaissance étendue et que la complexité de l'ontologie commence à croître.

Avec l'approche *inductive*, une ontologie est développée en observant, examinant, et analysant un ou plusieurs cas spécifiques dans le domaine d'intérêt. La caractérisation de l'ontologie résultante pour un cas spécifique est appliquée à d'autres cas dans le même domaine. La conception se base en grande partie sur la généralisation des cas choisis pendant le développement.

Avec une approche *déductive*, des principes généraux sont adoptés d'abord et ensuite sont travaillés et appliqués au domaine cible. L'ontologie résultante peut être vue comme une instance de ces notions générales.

Dans l'approche *synthétique*, un ensemble d'ontologies liées est identifié. Le développeur synthétise alors les éléments de ces ontologies avec les concepts du nouveau domaine cible, produisant une nouvelle ontologie unifiée.

L'approche *collaborative*. La marque d'une ontologie "moderne" est sa grande taille et sa haute complexité ; ces ontologies englobent un ensemble si riche de la connaissance que leur compréhension (complète) dépasse celle de n'importe quel développeur ou concepteur seul ou même d'une petite équipe de concepteurs. Le développement d'une ontologie à grande échelle doit être le fruit d'un effort commun de plusieurs experts de domaine et concepteurs de logiciels. Ainsi l'approche collaborative pour le développement d'ontologie est la mieux adaptée pour cette tâche par rapport aux quatre premières approches, vues les caractéristiques mêmes d'une ontologie.

Aussi, aucune des quatre premières approches n'a un service intégré d'évaluation de la qualité/acceptabilité de l'ontologie résultante. En revanche, l'approche collaborative est basée sur des évaluations diverses et se donne comme contrainte la réduction itérative des objections des participants. Cette approche est basée fortement sur la nature des participants, du degré de leur implication/diligence, et

sur les compétences de l'administrateur (développeur) qui surveille le processus collaboratif.

### **3.5 Conclusion**

Dans ce chapitre, nous avons donné les définitions importantes des ontologies selon la perception de diverses disciplines. Ensuite nous avons catégorisé les différents types d'ontologie, et décrit les composants et les usages des ontologies. Enfin nous avons analysé et comparé les approches existantes de conception d'ontologie.

Dans le chapitre suivant, nous allons étudié les différentes méthodes et méthodologies de construction d'ontologie.

## Chapitre 4. Les méthodes et méthodologies de construction d'ontologie

Dans ce chapitre, nous nous intéressons d'abord au processus de développement d'ontologie. Ensuite nous décrivons, analysons et comparons les différentes méthodes et méthodologies de développement des ontologies .

Dans la littérature, les mots méthodologie, méthode, technique, processus, activité sont bien souvent utilisés de façon aléatoire [DEHO98]. Pour clarifier l'utilisation de ces mots dans le domaine des ontologies, nous adoptons les définitions de l'IEEE. Ainsi les travaux sont présentés uniformément et indépendamment selon que chaque auteur utilise sa propre terminologie.

L'IEEE définit une *méthodologie* comme "une série complète et intégrée de techniques ou de méthodes créant une théorie générale de systèmes de la façon dont un travail bien pensé doit être exécuté" [IEEE90]. Les méthodologies sont largement utilisées en Informatique et Ingénierie de Connaissance. Selon la définition de la méthodologie, les méthodes et les techniques font l'une et l'autre partie des méthodologies. Une *méthode* est un ensemble de "processus ou de procédures ordonné utilisé dans la conception d'un produit ou dans l'exécution d'un service" [IEEE90]. Une technique est "une procédure technique et managériale utilisée pour atteindre un objectif donné" [IEEE90].

Selon la définition de l'IEEE, les méthodes et les techniques sont liées. Au fait, les méthodes et les techniques sont utilisées pour effectuer les tâches dans les différents processus composant une méthodologie. Un *processus* est une "fonction qui doit être exécutée dans le cycle de vie logiciel. Un processus est composé d'activités" [IEEE96]. Une *activité* est "un ensemble de tâches constituant un processus" [IEEE96]. Une tâche constitue la plus petite unité dans un travail. Une *tâche* "est une partie de travail bien définie pour une ou plusieurs personnes dans un projet. Les tâches liées sont habituellement groupées pour former des activités" Pour synthétiser, nous pouvons dire qu'une méthodologie est composée de méthodes et de techniques. Les méthodes sont composées de processus et sont détaillées avec de techniques. Les processus sont composés d'activités. Et enfin les activités regroupent les tâches.

## 4.1 Le processus de développement d'ontologie

Le processus de développement d'ontologie se rapporte aux activités qui sont exécutées pendant la construction des ontologies. Il est crucial d'identifier ces activités si le consensus est à atteindre sur les ontologies construites par des équipes coopératives géographiquement distantes. Le processus de développement d'ontologie identifié par [FERN97] est une proposition basée sur le standard IEEE pour le développement logiciel [IEEE96]. Il est recommandé d'effectuer les trois catégories des activités indiquées ci-dessous et d'orienter clairement le processus des constructions. Ces trois catégories d'activités sont :

- **Les activités de gestion (management)** incluant la programmation, le contrôle et la garantie de la qualité.
  
- **Les activités orientées développement** regroupant les activités de pré-développement, développement et post-développement.

Pendant le pré-développement, une étude de l'environnement est effectuée pour connaître l'environnement dans lequel l'ontologie sera utilisée, les applications dans lesquelles l'ontologie sera intégrée, etc. Aussi pendant le pré-développement, l'étude de faisabilité répond aux questions telles que : est-il possible ou est-il approprié de construire l'ontologie ?, etc.

Une fois dans le développement, les activités à réaliser sont :

- La spécification
- La conceptualisation
- La formalisation
- L'implémentation.

Pendant le post-développement, les activités de maintenance et de (re)utilisation sont réalisées.

- **Les activités support** inclut une série d'activités exécutées en même temps que les activités orientées développement, sans lesquelles l'ontologie ne pourrait être construite. Ces activités incluent l'acquisition de connaissance, l'évaluation, l'intégration, la fusion, l'alignement, la documentation et la gestion de version.

A partir de cette analyse, nous pouvons conclure que le processus de développement de l'ontologie identifie quelles activités sont à exécuter. Cependant, il n'identifie pas l'ordre dans lequel les activités doivent être exécutées [FERN97] [IEEE96]. Le cycle de vie de l'ontologie identifie à quel moment les activités doivent être effectuées, c'est à dire, il identifie l'ensemble des étapes à travers lesquelles l'ontologie évolue, décrit quelles activités doivent être exécutées dans chaque étape et comment les étapes sont reliées.

Dans la suite nous allons décrire les méthodes et méthodologies existantes de développement d'ontologies.

## **4.2 Les méthodes et méthodologies de développement des ontologies**

Une ontologie est toujours liée à une méthodologie de construction, à un outil de construction et avec un langage de représentation d'ontologie. A ce niveau, nous présentons un état de l'art des principales méthodologies et méthodes utilisées pour construire les ontologies à partir de zéro, par la réutilisation d'autres ontologies, par un processus de fusion ou par l'utilisation d'une approche d'étude d'ontologie. Ensuite nous faisons une analyse comparée de ces méthodes et méthodologies. Nous nous intéressons en priorité aux méthodes et méthodologies utilisées dans le domaine de l'ingénierie de connaissance.

### **4.2.1 La méthodologie METHONTOLOGY**

Cette méthodologie a été développée par le groupe d'Ontologie à l'Université Polytechnique de Madrid. METHONTOLOGY [FERN97, GOME98, FERN99] prend ses racines dans les activités principales identifiées par le processus de développement logiciel [IEEE96] et dans les méthodologies d'ingénierie de connaissance [GOME97, WATE86]. Cette méthodologie inclue : l'identification du processus de développement, un cycle de vie basé sur des prototypes évolutifs, et les techniques pour effectuer chaque tâche dans les activités de gestion, de développement et de support.

ODE [BLAZ98] et WebODE [ARPI03] ont été construits pour donner un support technique à METHONTOLOGY. D'autres outils d'ontologie et de suites d'outils

peuvent également être utilisés pour construire des ontologies avec cette méthodologie, par exemple, Protégé-2000 [NOY00], OntoEdit [SURE02a], etc. METHONTOLOGY a été proposée pour la construction d'ontologie par la FIPA ("Foundation for Intelligent Physical Agents"), qui favorise l'interopérabilité à travers les applications.

METHONTOLOGY inclut une méthode de ré-ingénierie [GOME99] pour résoudre certains des problèmes liés à la construction d'une ontologie par la réutilisation d'une autre ontologie.

#### **4.2.2 La méthodologie On-To-Knowledge**

On-to-Knowledge [GOME04] est une méthodologie développée dans le cadre d'un projet dont les partenaires sont l'Institut AIFB de l'Université de Karlsruhe, l'Université Libre d'Amsterdam, et la société British Telecom. On-to-Knowledge recommande un procédé itératif de développement, et comporte quatre phases principales : une phase de spécification de condition, une phase d'amélioration, une phase d'évaluation et une phase d'application et d'évolution. On-To-Knowledge propose l'acquisition des connaissances en spécialisant une ontologie générique. Elle propose de construire l'ontologie en tenant compte de la manière dont elle sera utilisée dans d'autres applications. Par conséquent, les ontologies développées avec cette méthodologie sont fortement dépendantes de l'application. On-To-Knowledge recommande la suite d'outils OntoStudio [SURE02] comme support de développement des ontologies. Le principal auteur est GRÜNINGER [GOME04]. On-To-Knowledge indique clairement les relations entre un ensemble de techniques, méthodes et principes pour chacun de ses processus (par exemple, ordre recommandé, entrée/sorties, inter-dépendances).

#### **4.2.3. La méthode "Ontology Development 101"**

"Ontology Development 101" [NOYN01] a été développée à l'Université de Stanford, elle cherche à construire des ontologies formelles par la reprise et l'adaptation des ontologies déjà existantes, et propose de suivre les démarches ci-après :

- Déterminer le domaine et la portée de l'ontologie

- Considérer la réutilisation des ontologies existantes
- Énumérer les termes les plus importants dans l'ontologie
- Définir les classes et hiérarchie des classes
- Définir les propriétés des classes
- Définir les facettes des attributs
- Construire les instances.

Elle utilise comme support les outils Protégé 2000 [N0Y00] et Ontolingua [FARQ96].

#### 4.2.4 Les autres méthodes de développement d'ontologies

Il existent d'autres méthodes de développement d'ontologies utilisées dans d'autres domaines. Dans ce groupe, nous pouvons citer entre autres :

- **La méthode Cyc** [LENA90] dans le domaine de la microélectronique.
- **Les méthodes de Uschold et King** [USCH95] et de **Gruninger et Fox** [GRUN95], dont toutes les deux cherchaient à modéliser le domaine de l'entreprise. Elles sont moins détaillées et n'ont été testées que dans le domaine des affaires (business).
- **La méthode Kactus** [BERN96], proposée dans le domaine de l'intelligence artificielle, dans le cadre du projet Esprit [KACT96].
- **La méthode Sensus**, proposée par l'équipe de Swartout [SWAR97], est utilisée pour construire la squelette d'une ontologie de domaine à partir d'une grande ontologie, l'ontologie Sensus. Elle a surtout été utilisée dans le domaine de la défense [VALE99].

D'autres méthodes existent, qui ne s'occupent en fait que d'une activité précise du processus de développement. On peut les classer dans les catégories suivantes :

- **Les méthodes d'étude des ontologies ou "ontologies learning"**. Cette catégorie de méthodes s'intéresse à l'activité de l'acquisition de connaissance. Bien que l'un des buts principaux des ontologies soit de réduire le goulet d'étranglement au niveau de l'acquisition de la connaissance, acquérir la connaissance pour la construction des ontologies exige toujours beaucoup de temps et de ressources. Par conséquent, on a pensé à des méthodes d'étude

d'ontologie pour alléger l'effort pendant le processus d'acquisition de la connaissance. Nous pouvons citer les *méthodes de l'équipe de Maedche* [KIET00] et de *l'équipe d'Aussenac-Giles* [AUSS00a]. De telles méthodes sont utilisées dans plusieurs buts : créer une nouvelle ontologie à partir de zéro, enrichir une ontologie existante avec de nouveaux termes, et acquérir la connaissance pour des tâches.

- **Les méthodes d'alignement et de fusion des ontologies.** Comme dit auparavant les ontologies ont pour but de capturer la connaissance consensuelle d'un domaine donné d'une manière générique et formelle pour être réutilisée et partagée à travers des applications et par des groupes de personnes. De cette définition nous pourrions incorrectement déduire qu'il y a seulement une ontologie pour un domaine. En fait, nous pouvons trouver dans la littérature plusieurs ontologies qui modélisent, de différentes manières, le même type de connaissance ou de domaine. Noy et Musen [NOY99] distinguent deux approches pour unifier les terminologies des ontologies : alignement d'ontologie et fusionnement d'ontologie. Les méthodes d'alignement d'ontologie établissent différents types de correspondances entre les ontologies, par conséquent cette option préserve les ontologies originales. Nous pouvons citer la méthode utilisée par *AnchorPROMPT* [NOY01] pour l'alignement d'ontologie.

Cependant, les méthodes de fusion d'ontologies proposent de produire une ontologie unique à partir des ontologies originales. Le processus de fusionnement exige habituellement d'établir des correspondances entre les ontologies afin de les fusionner. Etant donné l'état actuel des choses et dans le contexte du Web Sémantique, il est plus approprié d'établir des correspondances ontologiques entre les ontologies existantes sur le même sujet que de prétendre construire un modèle unifié de la connaissance pour un tel sujet à partir de zéro. Dans cet ordre d'idées, nous pouvons citer la méthodologie *OIGNONS* [GANG99], les méthodes *FCA-Merge* [STUM01], *Chimaera* [McGU00] et *PROMPT* [NOY00] pour fusionner des ontologies.

- **Les méthodes d'évaluation des ontologies.** Des ontologies ne peuvent être réutilisées par d'autres ontologies ou être utilisées par des applications sans



une évaluation préalable de leur contenu d'un point de vue technique. Comme directives pour évaluer des ontologies, les premiers travaux ont été proposés par l'équipe de Gomez-Perez [GOME95, GOME96] ; et ont continué sur l'évaluation de la connaissance taxonomique [GOME99, GOME01]. L'équipe de Guarino a développé *OntoClean* [WELT01], qui est une méthode pour analyser et nettoyer la taxonomie d'une ontologie existante au moyen d'un ensemble de principes basés sur la philosophie. D'autres travaux intéressants sur l'évaluation de base de connaissance avec des ontologies sont ceux de Kalfoglou et de Robertson [KALF99a ; 99b].

Après avoir décrit les différentes méthodes et méthodologies de développement d'ontologie, nous allons les analyser et les comparer.

### **4.3. Analyse et comparaison des méthodes et méthodologies**

Le but de cette section n'est pas de dire quelle méthode ou méthodologie est la meilleure. Comme de Hoog [DEHO98] le disait : *"il est extrêmement difficile de juger la valeur d'une méthode ou méthodologie d'une manière objective. L'expérimentation est naturellement la manière appropriée de le faire, mais elle est à peine faisable parce qu'il y a beaucoup de conditions qui ne peuvent être contrôlées "*. Et si nous extrapolons l'argument que de Hoog fournit pour les systèmes basés sur la connaissance au domaine de l'ontologie, il n'est pas très probable que quelqu'un acceptera de payer deux fois la construction d'une même ontologie complexe avec différentes approches.

Quelle méthode ou méthodologie est la plus utile ou la plus appropriée pour construire notre ontologie ? Pour aider à répondre à cette question nous avons élaboré un cadre pour comparer les méthodes et méthodologies évoquées ci-dessus.

Le cadre de comparaison prend en compte la stratégie de construction d'ontologie des méthodologies et méthodes, leur support logiciel, le processus de développement qu'elles proposent, et comment elles sont utilisées pour développer des ontologies ou dans les applications, projets, etc. Ce cadre est basé sur celui présenté par [FERN02a].

Pour comparer la stratégie de construction des méthodes et méthodologies, nous proposons un ensemble de critères suivants :

- Proposition de cycle de vie. Le cycle de vie identifie l'ensemble des étapes à travers lesquelles l'ontologie évolue pendant sa durée de vie. Il décrit également quelles activités doivent être exécutées à chaque étape et comment les étapes sont liées. Comme exemples de cycle de vie, nous avons :
  - Le cycle de vie incrémental [McCR82]. Selon cette approche l'ontologie se développerait par des couches, permettant l'inclusion de nouvelles définitions seulement quand une nouvelle version est planifiée. Ce modèle empêche l'inclusion de nouvelles définitions si elles ne sont pas planifiées, mais il permet un développement incrémental.
  - Les prototypes évolutifs [KEND95]. Selon cette approche, l'ontologie se développe selon les besoins. Ce modèle permet de modifier, ajouter et supprimer des définitions dans l'ontologie à tout moment.
- Stratégie selon l'application. Ce critère est lié au degré de dépendance de l'ontologie avec l'application qui l'utilise. Considérant ce critère, les méthodologies et méthodes peuvent être classées dans les types suivants :
  - Dépendantes des applications. Les ontologies sont construites sur la base des applications qui les utilisent.
  - Semi-dépendantes des applications. Les scénarios possibles de l'utilisation d'ontologie sont identifiés dans l'étape de spécification.
  - Indépendantes des applications. Le processus est totalement indépendant des utilisations de l'ontologie dans les applications.
- Utilisation des ontologies noyaux. Selon ce critère nous analysons s'il est possible ou non d'utiliser une ontologie noyau comme point de départ dans le développement de l'ontologie de domaine.
- Stratégie pour identifier les concepts. Il y a trois stratégies possibles pour identifier les concepts : partant du plus concret vers le plus abstrait (bottom-up), partant du plus abstrait vers le plus concret (top-down), ou partant du plus important vers le plus abstrait et le plus concret (middle-out). Une explication plus précise de ce critère peut être trouvée dans la description de la méthode de Uschold et King [USCH96].

Concernant les supports techniques, il est important de savoir quels outils donnent le support complet ou partiel aux méthodologies et méthodes. Le tableau 4.3.1 énumère ces outils et suites d'outils.

Concernant les processus de développement d'ontologie, la table 4.3.2 les récapitule dans les différentes méthodologies et méthodes. Chaque cellule de la table peut être remplie de trois types de valeurs. La valeur "++" signifie que la méthode ou la méthodologie décrit comment exécuter chaque tâche dans l'activité proposée, quand le faire, qui doit le faire, etc. La valeur "+" signifie que la méthodologie identifie juste le processus. La valeur "-" signifie que la documentation publique ne mentionne pas l'activité considérée.

**TABLEAU 4.1 COMPARAISON DES METHODES SELON LES STRATEGIES DE CONSTRUCTION DE L'ONTOLOGIE  
[GOME04]**

	<b>Cyc</b> [LENA90]	<b>Uschold &amp; King</b> [USCH95]	<b>Gruninger &amp; Fox</b> [GRUN95]	<b>KACTUS</b> [BERN96]	<b>METHONTOLOGY</b> [FERN97]	<b>SENSUS</b> [SWAR97]	<b>On-To- Knowledge</b> [GOME04]
<b>Proposition de cycle de vie</b>	Prototypes en évolution	Non proposée	Prototypes en évolution ou incrémentale	Prototypes en évolution	Prototypes en évolution	Non proposée	Incrémentale et cyclique avec Prototypes en évolution
<b>Stratégie par rapport à l'application</b>	indépendante d'application	Indépendante d'application	Semi-dépendante d'application	Dépendante d'application	Indépendante d'application	Semi-dépendante d'application	Dépendante d'application
<b>Stratégie d'identification des concepts</b>	Non spécifiée	Middle-out	Middle-out	Top-down	Middle-out	Non spécifiée	Top-down, Bottom-up Middle-out
<b>Utilisation d'ontologie noyau</b>	Oui	Non	Non	Non	Dépend des ressources disponibles	Oui	Dépend des ressources disponibles
<b>Outil Support</b>	Outils Cyc	Pas d'outil spécifique	Pas d'outil spécifique	Pas d'outil spécifique	ODE, WebODE, OntoEdit, Protégé-2000	Pas d'outil spécifique (OntoSaurus)	OntoEdit avec ses plug-ins

TABLEAU 4.2 COMPARAISON DES METHODES SELON LE PROCESSUS DE DEVELOPPEMENT DE L'ONTOLOGIE  
[GOME04]

	Cyc	Ushold & King	Gruninger & Fox	KACTUS	METHONTOLOGY	SENSUS	On-To-Knowledge
Planning	-	-	-	-	-	-	+
Spécification	-	+	++	+	++	+	++
Conceptualisation	-	-	++	+	++	-	+
Formalisation	-	-	++	++	++	-	++
Implémentation	+	+	++	+	++	+	+
Maintenance	-	-	-	-	+	-	+
Réutilisation	-	-	-	-	-	-	+
Acquisition de connaissance	+	+	+	-	++	-	++
Evaluation	-	+	++	-	++	-	+
Intégration	+	+	+	+	+	-	+
Contrôle de versions	-	-	-	-	++	-	+
Documentation	+	+	+	-	++	-	+
Fusion et Alignement	-	-	-	-	-	-	-

Comme synthèse par rapport à l'analyse des méthodologies et méthodes, nous pouvons dire que :

- Le modèle de cycle de vie qui prévaut est celui des prototypes évolutifs, bien qu'il y ait des approches qui ne proposent aucun modèle de cycle de vie.
- Il existe une large diversité de stratégies de développement d'ontologie, avec des approches dépendante, semi-dépendante ou indépendante des applications.
- Nous remarquons aussi une large diversité de stratégies pour identifier les concepts dans la taxonomie, à travers l'approche middle-out qui est la plus communément utilisée.

Aucune des méthodes ou méthodologies ne supportent toutes les activités impliquées dans la construction d'ontologie. Alors que les méthodes de Uschold et King, de Gruninger et Fox et SENSUS ne mentionnent pas de cycle de vie de l'ontologie, et que la méthode Kactus contient également les mêmes omissions, La méthodologie On-To-Knowledge décrit davantage d'activités. METHONTOLOGY est l'approche qui fournit les plus précises descriptions de chaque activité.

La plupart des approches sont portées sur des activités de développement, particulièrement sur l'implémentation de l'ontologie, et elles ne prêtent pas trop d'attention à d'autres aspects importants liés à la gestion, à l'évolution et à l'évaluation des ontologies. Cela est dû au fait que le domaine de la conception d'ontologie est un domaine relativement nouveau. Cependant, une basse conformité aux critères formellement établis ne signifie pas une mauvaise qualité de la méthodologie ou de la méthode. Comme de Hoog [DEHO98] l'a dit, une méthode pas très spécifiée peut être très utile pour un groupe expérimenté.

La plupart des approches présentent des inconvénients dans leur utilisation. Elles n'ont pas été utilisées par des groupes externes et, dans beaucoup de cas, elles n'ont été utilisées que dans un domaine unique.

La plupart des approches n'ont pas un outil spécifique qui leur donne le support technique. En outre, aucun des outils disponibles ne couvre toutes les activités nécessaires dans la construction d'ontologie.

#### **4.4 Conclusion**

Dans ce chapitre, nous avons d'abord décrit le processus de développement d'ontologie. Ensuite nous avons analysé et comparé les différentes méthodes et méthodologies de développement des ontologies .

Les méthodologies décrites requièrent un ingénieur de connaissances qui modélise l'ensemble de l'ontologie. Généralement ces méthodologies sont centralisées et inspirationnelles. Bien que la conception des ontologies soit un processus créatif, il ne peut y avoir d'ontologies identiques faites par des personnes différentes. Les applications potentielles d'une ontologie et le point de vue du concepteur sur le domaine traité, affecteront indéniablement les choix de conception de l'ontologie.

Nous remarquons également qu'aucune des méthodes ou méthodologies mentionnées ci-dessus ne s'intéressent à la construction collaborative et distribuée des ontologies. Nous nous intéresserons aux méthodes qui traitent de la construction collaborative des ontologies dans le chapitre 8.

Dans le chapitre suivant nous nous intéressons aux langages de description des ontologies, car il est indéniable qu'une ontologie est toujours liée à une méthodologie de construction, mais également à un langage de représentation.

## Chapitre 5. Les langages de description des ontologies

L'une des décisions clés à prendre dans le processus de développement d'ontologie est le choix du langage (ou de l'ensemble des langages) dans lequel l'ontologie sera implémentée. Il existe plusieurs langages d'implémentation, et autres langages et systèmes de Représentation de la Connaissance utilisés pour implémenter les ontologies alors qu'ils n'étaient pas spécifiquement créés pour ce but.

D'habitude, le choix d'un langage d'ontologie n'est pas basée sur la représentation de la connaissance et sur les mécanismes d'inférence exigés par l'application qui utilise l'ontologie, mais sur les préférences du développeur. Notre expérience nous indique qu'un mauvais choix du langage utilisé pour implémenter une ontologie peut causer des problèmes une fois l'ontologie utilisée dans une application.

Comme indiqué dans [GOME04], des réponses plus ou moins claires aux questions suivantes permettent d'aider au choix d'un langage approprié :

- Quel(s) langage(s) devrait-on utiliser pour implémenter notre ontologie ?
- Quel caractère expressif le langage doit-il avoir ?
- Quels sont les mécanismes d'inférence attachés au langage ?
- Le langage est-il soutenu par un outil de développement d'ontologie ?
- Le langage est-il approprié pour échanger des ontologies entre les applications ?
- Le langage facilite-t-il l'intégration de l'ontologie dans une application ?
- Le langage est-il intégré dans d'autres langages utilisés pour représenter la connaissance et l'information sur le Web, tels que le HTML et le XML ?
- Y a-t-il des traducteurs qui traduisent l'ontologie implémentée dans un langage source vers un langage cible ?
- Comment de tels traducteurs réduisent-ils au minimum la perte de la connaissance dans le processus de traduction ?

Tous les langages existants n'ont pas le même caractère expressif et ne raisonnent pas de la même manière. Les langages d'ontologie sont basés sur des différents



paradigmes de représentation de la connaissance comme des classes (ou "frames"), la logique de description, la logique de premier (et second ordre), et les réseaux sémantiques, etc... Cet état de fait rend bien plus important le choix correct du langage dans lequel l'ontologie doit être implémentée. En fait, un paradigme de représentation de connaissance peut s'avérer très approprié pour une tâche spécifique mais pas pour d'autres effectuées dans une application.

Par conséquent, avant le codage de l'ontologie nous devrions savoir d'abord ce dont nous avons besoin, en termes d'expressivité et de raisonnement, et puis les langages qui satisfont à de telles exigences. Cette tâche d'analyse et de comparaison des langages d'ontologie, par rapport à leur expressivité et leurs possibilités de raisonnement, est décrite dans [CORC00].

Nous pouvons classer les langages de description d'ontologies existants en deux catégories, à savoir " traditionnels" et liés à internet.

### 5.1 Les langages d'ontologies dits "traditionnels"

Cette catégorie représente un ensemble de langages d'ontologie basés sur l'Intelligence Artificielle et créés la plupart dans les années 1990. Nous pouvons citer entre autres :

- **CycL** [LENA90] basé sur les "frames" et la logique de premier ordre, et utilisé pour la construction de l'ontologie Cyc.
- **KIF** ("Knowledge Interchange Format") [GENE92, NCIT98], désigné comme un format d'échange de la connaissance, et basé sur la logique de premier ordre.
- **Ontolingua** [FARQ97] basé sur KIF, il est aujourd'hui le langage d'ontologie supporté par Ontolingua Server. Ce langage utilise une syntaxe Lisp<sup>14</sup> et est basé sur les "frames" et la logique de premier ordre. Ontolingua était considéré, de facto, comme un standard par la communauté des ontologies dans les années 1990.
- **LOOM** [MACG91] basé sur les logiques de description (LD) et les règles de production, n'était pas prévu pour implémenter des ontologies mais

---

<sup>14</sup> LISP est une famille des langages dont le développement a commencé vers la fin des années 50, pour le traitement de liste dans l'Intelligence Artificielle.

pour les bases de connaissances générales. Ce langage fournit des dispositifs de classification automatique des concepts.

- **OCML** ("Operational Conceptual Modeling Language") [MOTT99], une sorte de "Ontolingua opérationnel", parce que la plupart des définitions qui peuvent être exprimées dans OCML sont similaires aux définitions correspondantes dans Ontolingua. Ce langage a été construit pour développer des ontologies exécutables et des modèles dans les méthodes de résolution de problème.
- **Flogic** ("Frame Logic" ) [KIFE95], développé comme un langage qui combine les "frames" et la logique de premier ordre bien qu'il n'utilise pas Lisp comme syntaxe.
- **OKBC** ("Open Knowledge Base Connectivity") [CHAU98] un protocole né dans le cadre d'un programme de recherche de DARPA. Ce protocole permet d'accéder aux bases de connaissance stockées dans différents systèmes de représentation de connaissance, qui peuvent être basés sur différents paradigmes de représentation de connaissance. Parmi les langages mentionnés ci-dessus seuls Ontolingua, LOOM et CycL sont conformes à OKBC.

La disposition globale de ces langages est montrée sur la figure 5.1.

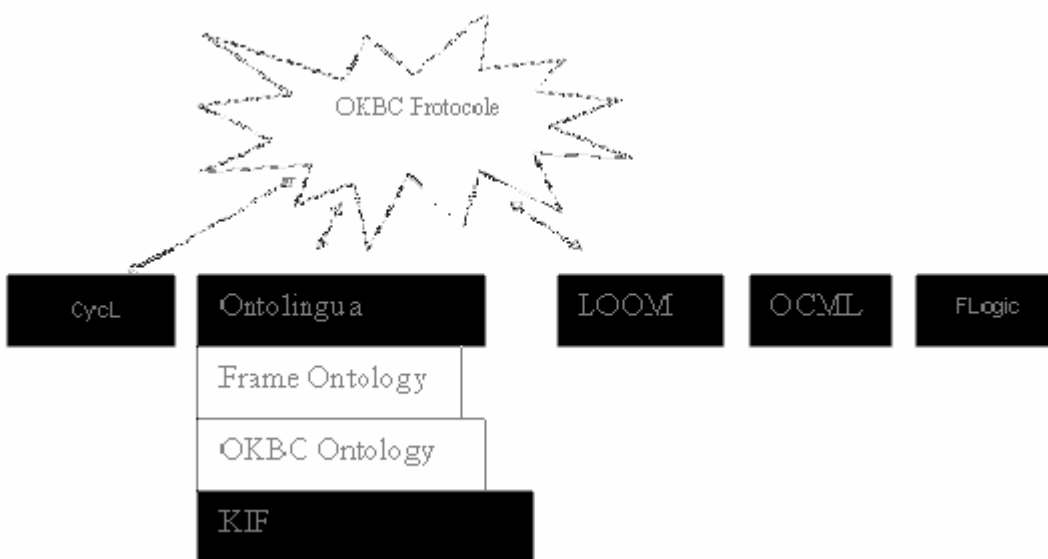


FIGURE 5.1 LES LANGAGES D'ONTOLOGIE " TRADITIONNELS ".  
[GOME04]

## 5.2 Les langages d'ontologie dits "web-standard" ou langages basés sur XML

Le boom de l'Internet a conduit à la création de langages d'ontologie pour exploiter les caractéristiques du Web. De tels langages sont appelés des langages "web-standard". Leur syntaxe est basée sur les langages existants tels que HTML [RAGG99] et XML [BRAY00], dont les buts ne sont pas le développement d'ontologie mais la présentation et l'échange des données respectivement. Les relations entre ces langages sont montrées dans la figure 5.2.

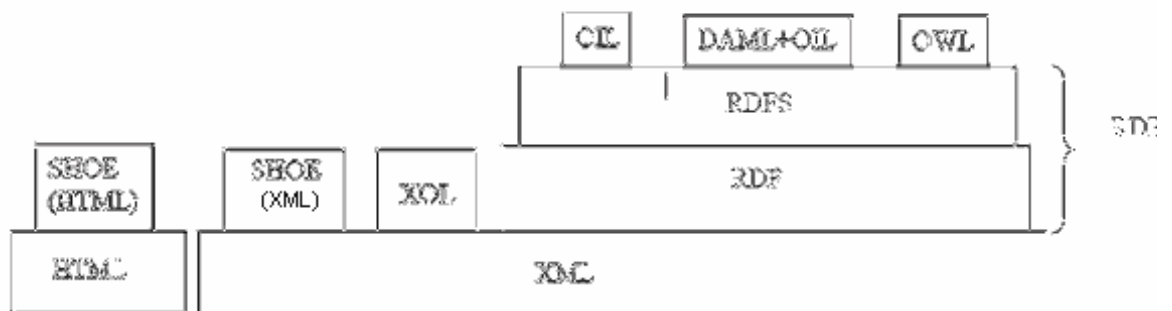


FIGURE 5.2 LES LANGAGES D'ONTOLOGIE DITS "WEB-STANDARD" OU BASES SUR XML. [GOME04]

Dans cette catégorie de langages, nous pouvons citer :

- **SHOE** ("Simple HTML Ontology Extension") [LUKE00], basé sur du web il combine les "frames" et les règles. Il a été construit comme une extension de HTML en 1996. Il utilisait des étiquettes différentes de celles des spécifications HTML, de ce fait permettant l'insertion des ontologies dans des documents HTML. Plus tard sa syntaxe a été adaptée à XML.
- **XOL** ("XML-based Ontology exchange Language") [KARP99], développé comme une transformation en XML d'un petit sous ensemble des primitives du protocole OKCB, appelé OKBC-Lite.

- **RDF** ("Resource Description Framework") [LASS99], développé par le W3C<sup>15</sup> comme un langage basé sur des réseaux sémantiques pour décrire les ressources web. Son développement a commencé en 1997, et RDF a été proposé comme une recommandation W3C en 1999.
- **RDF Schéma** [BRIC03], construit par le W3C comme une extension de RDF avec des primitives basées sur les "frames". Ce langage a été proposé comme une recommandation candidate du W3C en 2000, et ainsi a-t-il subi une révision importante en novembre 2002, de sorte que son document de référence ait été édité comme une ébauche de travail du W3C. Plus tard, il a été révisé en 2003. La combinaison de RDF et de RDF Schéma est connue sous le nom de *RDF(S)*.

Les langages ci-dessus ont établi les bases du Web Sémantique [BERN99]. Et c'est dans ce contexte que trois autres langages ont été développés comme des extensions de RDF(S). Il s'agit de :

- **OIL** ("Ontology Interchange Language and Ontology Inference Layer") [HORR00], développé en début 2000 dans le cadre du projet européen On-To-Knowledge<sup>16</sup>. Il ajoute des primitives de RC basées sur des "frames" à RDF(S) et sa sémantique formelle est basée sur les logiques de description.
- **DAML+OIL** ("DARPA Agent Markup Language") [HORR01], créé plus tard entre 2000 et 2001 par un comité mixte des USA et de l'Union Européenne dans le contexte du projet DAML<sup>17</sup> de DARPA. Il était basé sur la spécification précédente de DALM-ONT, qui a été construit en fin 2000, et sur OIL. DAML+OIL ajoute des primitives de RC basées sur les logiques de description à RDF(S).
- **OWL** ("Ontology Web Language") [DEAN03], créé par le W3C comme le nouvel langage pour Web sémantique. Il couvre la plupart des dispositifs de DAML+OIL et utilise la plupart de ses primitives.

---

<sup>15</sup> World Wide Web Consortium

<sup>16</sup> <http://www.ontoknowledge.org/>

<sup>17</sup> <http://www.daml.org/>

Pour mieux comprendre et mieux juger les langages, nous allons les comparer.

### **5.3 Analyse et comparaison des langages de description d'ontologie**

Tous les langages de description d'ontologie peuvent être jugés selon deux dimensions principales, qui sont fortement liées entre elles : la représentation de connaissances (RC) et les mécanismes de raisonnement.

La dimension de la représentation de connaissance est basée sur la façon d'implémenter différents composants de l'ontologie dans le langage en question, en prenant en compte le paradigme de RC utilisé par ce langage. Celle des mécanismes de raisonnement s'appuie sur les principaux dispositifs des moteurs d'inférence attachés au langage (s'il y en a). Dans le cas des langages basés sur les logiques de description, il peut y avoir un classificateur automatique disponible pour le langage qui met en place automatiquement la taxonomie des concepts à partir des définitions des concepts de l'ontologie. Un autre mécanisme de raisonnement sur lequel il faudrait s'intéresser est la gestion du simple et/ou multiple héritage des attributs de concept et de relations à travers les taxonomies de concepts.

En résumé, les différents langages d'ontologie ont des différents niveaux d'expressivité et de mécanismes d'inférence. Les paradigmes de RC utilisés par ces langages sont divers : les "frames", les logiques de description, les logiques de premier (et second) ordre, les graphes conceptuels, les réseaux sémantiques, les règles de production, les règles déductives, etc. Dans beaucoup de cas, ils sont même basés sur des combinaisons de plusieurs formalismes.

Ainsi comme synthèse, le tableau 5.1 montre les résultats d'une étude comparative entre les langages décrits. Cette comparaison ne vise pas à montrer une liste approfondie des dispositifs de tous les langages présentés, puisque chaque langage est basé sur des paradigmes différents et a ses propres dispositifs. Elle ne vise pas non plus à trancher entre les langages (dans le sens "le langage X est meilleur que le langage Y"). La comparaison vise seulement à aider à mieux comprendre les similarités et différences entre langages et les capacités de chacun. Les cellules du tableau utilise le signe '+' pour indiquer qu'il est un dispositif supporté par le langage, le '-' pour les dispositifs non supportés, et le '+/-' pour

des capacités non supportées, mais qui pourraient l'être par des extensions ou par un travail additionnel. La situation des langages peut changer à cause de leur évolution, spécialement dans les cas de RDF(S) et de OWL.

TABLEAU 5.1 COMPARAISON DES LANGAGES DE DESCRIPTION D'ONTOLOGIE

[GOME04]

	Ontolingua	LOOM	OKBC	OCML	FLogic	SHOE	XOL	RDF(S) OWL- Lite	OIL	DAML+OIL	OWL- DL
<b>CONCEPT</b>											
Attributs											
Attributs d'instance	+	+	+	+	+	+	+	+	+	+	+
Attributs de classe	+	-	+	+	+	-	+	+/-	-	+/-	+/-
Facettes											
Contraintes de cardinalité	+	+	+	+	+/-	-	+	-	+	+	+
<b>TAXONOMIES DE CONCEPT</b>											
Sous classe-de	+	+	+	+	+	+	+	+	+	+	+
Décomposition-Disjointe	+	+	-	+	+/-	-	-	-	+	+	+
Décomposition-Exhaustive	+	-	-	+/-	+/-	-	-	-	+/-	+/-	+/-
Partition	+	+	-	+	+/-	-	-	-	+	+	+/-
<b>RELATIONS</b>											
Relations binaires	+	+	+	+	+	+	+	+	+	+	+
Relations n-aires	+	+	+/-	+	+/-	+	+/-	+/-	+/-	+/-	+/-
Hierarchies de relation	+	+	-	+	+/-	-	-	-	-	-	-
Contraintes d'intégrité	+	+	-	+	+	-	-	-	-	-	-
<b>FONCTIONS</b>											
Fonctions binaires	+	+	+/-	+	+	-	+/-	-	+	+	+
Fonctions n-aires	+	+	-	+	+	-	-	-	-	-	-
<b>AUTRES COMPOSANTS</b>											
Axiomes formels	+	+	-	+	+	-	-	-	-	-	-

La principale conclusion obtenue de l'analyse de ce tableau est liée au pouvoir d'expression différent de chaque langage. Alors que certains langages permettent la représentation d'ontologies lourdes - "heavyweight ontologies" - (avec des axiomes formels, des fonctions, des règles, et d'autres riches modèles de composants), d'autres permettent seulement la représentation d'ontologies légères - "lightweight ontologies" - (avec des concepts, des taxonomies de concepts, et des relations entre eux). Il y a des différences réelles d'expressivité entre les langages dits traditionnels et les langages basés sur XML ou "web-standards". En fait, la plupart des composants peuvent être représentés dans les langages traditionnels (tels que Ontolingua, LOOM et OCML), alors que les langages basés sur XML ne fournissent pas normalement les moyens de le faire. Par exemple, dans le cas des axiomes formels, les langages basés sur XML se fondent sur une "couche logique" additionnelle qui doit être ajoutée, et qui fournira des dispositifs logiques supplémentaires non fournis par les langages eux-mêmes.

#### **5.4 Utilisation des langages dans les applications basées sur les ontologies**

Dans cette section, nous nous intéressons à l'utilisation des langages dans différents types d'applications basées sur les ontologies.

Dans le commerce électronique, les ontologies sont habituellement utilisées pour représenter les produits et les services qui sont offerts dans les systèmes du e-commerce et qui sont donnés aux utilisateurs dans les catalogues pour consultation [LEGE00]. Les besoins représentatifs ne sont pas trop complexes : à la base, il faut besoin des concepts, des attributs, et des relations  $n$ -aires entre concepts. Cependant, les besoins de raisonnement sont habituellement plus grands : si le nombre de produits ou de services offerts dans le système est élevé, les classifications automatiques sont très utiles pour organiser ces produits ou services automatiquement (donc, les langages basés sur les logiques de description sont extrêmement utiles), et un système question-réponse efficace est aussi important dans cet environnement (cela est fourni par la plupart des langages).



Dans le contexte du Web Sémantique [BERN01], et pour l'échange des ontologies entre les applications, les langages basés sur XML sont facilement lus et gérés puisque les bibliothèques pour le traitement de XML sont librement accessibles. Cependant, il n'est pas difficile d'adapter les langages traditionnels à la syntaxe XML, qui pourraient se servir du même genre de bibliothèques. L'avantage principal de RDF(S), OIL, DAML+OIL, et OWL est le support fort qu'ils reçoivent d'autres communautés outre que la communauté d'ontologie, et ceci signifie que davantage d'outils sont disponibles pour éditer, manipuler, et documenter les ontologies.

La création des ontologies de haut niveau ("upper-level ontologies") exige un pouvoir d'expression élevé et la plupart du temps il n'y a pas de grands besoins pour le support de raisonnement. Les ontologies de haut niveau ont été généralement spécifiées dans les langages de logique de description tels que LOOM ou CLASSIC. La base de connaissance Cyc est spécifiée dans CycL [LENA90], qui est un langage basé sur les "frames" et la logique de premier ordre.

En général, les langages basés sur les logiques de description ont été largement utilisés dans les applications qui avaient besoin d'une intégration intelligente des sources d'informations hétérogènes. Par exemple, CLASSIC a été utilisé dans OBSERVER [MENA00] LOOM dans Ariadne [BARI00], CARIN est le langage utilisé dans le projet PICSEL [GOAS00] et OIL a été utilisé dans un processus d'urbanisme au sens de la ré-ingénierie [STUC00]. En plus la plupart de ces langages ont été utilisés pour la recherche d'information. Par exemple, LOOM a été utilisé dans OntoSeek [GUAR99]. La raison principale pour cet usage est leur support d'inférence.

## **5.5 Conclusion**

Dans ce chapitre, nous avons présenté un état de l'art des langages de description des ontologies en les classant dans les catégories de langages dits traditionnels et de langages basés sur XML. Ensuite nous avons analysé et comparé les langages selon leurs capacités de représentation de connaissance et de mécanisme de raisonnement. Enfin, nous nous sommes intéressé à l'utilisation des langages dans les applications et les projets basés sur les ontologies. Nous pouvons dire qu'il n'existe pas de langage pour décrire les ontologies pré-consensuelles, d'où la nécessité de mettre en place un langage qui satisfait à nos besoins.

Dans le chapitre suivant, nous présentons un état de l'art sur les CSCW ("Computer Supported Collaborative Work"), c'est à dire le travail collaboratif assisté par ordinateur, et les Groupwares.

## Chapitre 6. Les Groupwares

En littérature informatique, le terme groupware est souvent assimilé à CSCW (Computer Supported Collaborative Work), en français TCAO (Travail Collaboratif Assisté par Ordinateur). [ELLI91] définit le groupware comme *"un système informatisé qui soutient le travail des groupes de personnes pour un but commun et qui fournit une interface pour un environnement partagé"*. La notion de groupware est le support informatique pour des groupes de travail ou des équipes de projet. Le mot support signifie soutenu et assisté par des logiciels ou matériels spéciaux, par des services de l'information et de la communication, intégrant aussi bien le support de travail en groupe, contrairement au traitement individuel de l'information.

Dans ce chapitre nous donnons un aperçu sur les CSCW et les Groupwares. Ensuite nous nous intéressons aux dimensions des problèmes à résoudre pour la mise en place d'un groupware. Enfin nous décrivons les outils de groupware existants.

### **6.1 Aperçu sur les CSCW (Computer Supported Collaborative Work) : le travail collaboratif assisté par ordinateur**

Le travail de collaboration assisté par ordinateur est un domaine de recherche multidisciplinaire englobant l'informatique, l'intelligence artificielle, la sociologie, et la psychologie. Le CSCW décrit des études sur les outils et techniques de groupware, et sur leurs effets psychologique, social et organisationnel. C'est un terme générique qui combine la compréhension du travail en groupe avec les technologies permettant le travail en réseau, intégrant des services et des techniques (matériel et logiciel ) associés [TERZ03]. En d'autres mots, le terme groupware se rapporte aux systèmes informatiques et se réfère habituellement à des implémentations d'outil, tandis que CSCW se rapporte à tous les impacts autour du groupware.

[DOUG68] s'intéressait déjà aux innovations de la collaboration par écran partagé, faisant participer deux personnes à différents emplacements qui communiquent à travers un réseau par interface audio et visuelle. On inaugurerait

alors le développement du groupware moderne tel que le partage à temps réel d'édition de documents, de messages textuels et des conférences audio et vidéo.

Les CSCW et groupware ont émergé dans les années 80 comme domaine de recherche à la fois pour les développeurs et les chercheurs. Les secteurs concernés étaient entre autres l'économie, la psychologie, les théories d'organisation, et l'éducation. Plusieurs applications de groupware ont été depuis développées et utilisées. Les groupwares ont avancé plus loin ces dernières années avec la croissance rapide de l'Internet et du World Wide Web.

Une catégorisation conventionnelle des systèmes de CSCW selon le temps et l'espace est montrée dans le tableau 6.1.

TABLEAU 6.1. CLASSIFICATION DES SYSTEMES CSCW  
[YILI03]

	Même temps (moment)	Temps (moments) différents
Même place (lieu)	Face-à-face (salles de conférence, salles de classe)	Interaction asynchrone (programmation de projet, outils de coordination)
Places (lieux) différentes	Synchrone distribué (éditeurs partagés, interfaces vidéo)	Asynchrone distribué (e-mail, weblog)

En général, le support coopératif fourni par le système CSCW ou le groupware peut être catégorisé comme suit :

- Stockage des données (exemple : le système de partage de fichier, CVS ou système de contrôle de versions) ;
- Communication synchrone et asynchrone (exemple : e-mail, vidéo/audio conférence, messagerie instantanée, Web log, chat en groupe) ;
- Organisation du travail (systèmes workflow, applications collaboratives édition/graphe, systèmes d'aide à la décision de groupe) ;
- Le groupware avancé qui intègre plusieurs des fonctions citées ci-dessus.

Le groupware fonctionne dans un monde hétérogène où différents médias, systèmes de stockage, et de collaborations planifiés ou impromptus sont utilisés. Dans le monde réel, une variété d'outils et de techniques est souvent utilisé pour exécuter une tâche. Un produit groupware seul qui traite de tous les aspects de la coopération de façon adéquate n'existe pas et ne peut exister.

La typologie dans le tableau 7.1 nous aide à identifier les applications groupware qui posent un défi technique commun. Les développeurs ne lient pas nécessairement l'outil à développer à la catégorisation du temps et de l'espace, parce que les tâches et les activités ne satisfont pas toujours le tableau ci-dessus avec précision. Par exemple, le tableau 7.1 ne prouve pas que certaines de ses activités se recoupent. Le courrier électronique peut être utilisé dans le même endroit et au même moment très efficacement (par exemple quand vous devez parler à quelqu'un en privé pendant que vous êtes dans la même salle et que d'autres sont présents, envoyer un e-mail à cette personne peut être une bonne solution).

Notre travail quotidien conduit souvent à des réunions en présentiel (en face-à-face), à des communications synchrones et asynchrones distribuées, et à de la coordination. Fournir un support de collaboration adéquate pour une tâche exige habituellement un ensemble d'outils.

Les fonctions clés du groupware sont l'esprit de groupe, les interfaces multi-utilisateurs, la gestion des droits d'accès, la communication et la coordination dans le groupe, l'espace d'information partagé et le support d'un environnement hétérogène ouvert, qui intègre les applications mono-utilisateur existantes.

Les applications groupware ont déjà infiltré nos bureaux et maisons. Des systèmes de messagerie instantanée tels que "MSN messenger", "Yahoo messenger" et la messagerie instantanée de AOL ont été largement utilisés pour la communication informelle à la fois au travail et à la maison. En partageant une application, un groupe d'utilisateurs interagit simultanément avec un ou plusieurs programmes et les utilisateurs peuvent tous voir et partager les mêmes résultats. Le World Wide Web, initialement conçu comme un médium passif, permet aujourd'hui des extensions avec des facilités pour soutenir la coopération ; à travers des bulletins web, des calendriers web partagés, des systèmes de planification et de gestion de

projet web qui ont été utilisés pour soutenir l'esprit d'équipe dans différents secteurs.

Aussi bien que les progrès technologiques réalisés dans le développement du groupware, les aspects non techniques, tels que social, psychologique, et économique, sont également importants.

[COLE97] exprime le succès de toute application groupware par l'équation :

$$\boxed{\text{Succès du groupware} = \text{Technologie} + \text{Culture} + \text{Finances} + \text{Politiques}}$$

Avec cette équation Coleman nous rappelle l'importance des facteurs humains et sociaux dans toute implémentation et tout déploiement de groupware. Cependant le succès et l'échec d'une application groupware ne peuvent être sûrement prévisibles [GRUD94]. De façon générale, le développement du groupware exige une bonne compréhension des environnements de travail, des facteurs sociaux, politiques et des facteurs de motivation sur le lieu de travail.

## **6.2 Les dimensions des problèmes pour les Groupwares**

Notre monde devient plus que jamais connecté avec l'utilisation à tous les niveaux d'applications réseau. Depuis que Ray Tomlinson a développé la première application e-mail pour ARPANET en 1971, diverses formes de communication sont apparues, comme la messagerie vocale, la messagerie instantanée, les téléphones cellulaires. Ceci a conduit au fait que la distance ne soit plus un obstacle dans notre travail et notre vie de tous les jours.

Il est vrai que Internet et les technologies de communication nous offrent des possibilités incroyables de communiquer ; néanmoins, nous avons encore des problèmes pour mettre en place des outils intégrant le travail collaboratif, impliquant des personnes ou des équipes géographiquement distribuées. Il est certain qu'il reste encore du chemin à faire.

La notion de distance a été introduite la première fois dans la technologie des logiciels à cause de la globalisation du marché, de la pratique croissante de l'approvisionnement à distance, et pour le besoin d'accéder aux meilleures ressources en dépensant moins. Afin de tirer profit de ces ressources distribuées, l'industrie des logiciels a commencé à décomposer les activités et les ressources

de développement des logiciels, et à les arranger dans des endroits géographiquement distribués. Le développement global de logiciel (GSD pour Global Software Development) étudie l'effet de cette pratique et des complexités auxquelles il est confronté, mais aussi celui des nouvelles variables introduites dans le développement de logiciel, telles que la distance, la communication, les aspects culturels, les aspects de gestion, etc. Il vise à aider à améliorer l'efficacité du développement distribué et la qualité des produits logiciels. Les chercheurs en GSD et en technologie des télécommunications ont trouvé que la communication, particulièrement informelle et non planifiée, est extrêmement importante pour soutenir le travail collaboratif.

Voyons maintenant, comment la difficulté du développement collaboratif se manifeste, et cela dans trois dimensions : celle de la distance et de la communication, celle de la gestion de la documentation et de la connaissance, celle de la traçabilité des modifications et du contrôle de version.

### **6.2.1 Distance et Communication**

La collaboration à distance doit faire face à la perte de riches et subtiles interactions que les équipes sur place utilisent pour coordonner leur travail. La recherche en génie logiciel distribué suggère que travailler sur des sites différents introduit des délais substantiels par rapport au processus de développement. Parce que la communication est réduite, et il est difficile d'établir le contact avec la bonne personne au bon moment, aussi bien que d'avoir une session collaborative effective.

Pour des équipes avec tous les membres travaillant dans un même espace de travail, les communications ad-hoc informelles peuvent se produire n'importe quand pendant le jour de travail (par exemple, dans le couloir ou près de la machine à café) et les membres tirent profit de tout ce temps. Comme la communication informelle passe inaperçue, on a tendance à négliger ses effets et son apport. Cependant, dans une équipe géographiquement distribuée, les chances pour que des membres aient des communications informelles sont réduites, et l'équipe perd les avantages de cette communication. La proximité est donc un facteur incitant à la collaboration spontanée.

Les réunions sont une sorte de communication formalisée ; cependant, la communication ne nécessite toujours pas à se produire dans une forme formelle, hiérarchique. Quand une équipe distribuée collabore, les canaux informels de la coordination sont importants puisqu'ils aident les concepteurs à compléter les détails dans le travail, à manipuler les exceptions et corriger les erreurs. La communication informelle a un impact direct sur le processus de développement. Par exemple, les nouvelles d'une modification peuvent se propager et arriver à chaque membre d'équipe beaucoup plus rapidement qu'en passant par les mécanismes formels de communication, tels que des documents de spécification. Ce phénomène incite potentiellement les travailleurs à réagir plus rapidement aux modifications et par conséquent raccourcit la période de développement.

La coordination est l'acte d'intégrer chaque tâche à chaque unité organisationnelle, ainsi chaque unité contribue à l'objectif global ; l'orchestration de cette intégration exige souvent la communication intense et continue. Le contrôle se rapporte au processus d'adhésion aux buts, aux caractéristiques, et aux normes du projet. La communication inadéquate causée par la distance pose également un challenge sérieux pour la coordination et le contrôle d'équipe.

En résumé, la communication efficace joue un rôle important dans la réussite d'un projet afin de faciliter la coordination et le partage en équipe.

L'importance de la communication informelle a mené à une variété d'outils conçus pour stimuler la conversation occasionnelle (informelle) entre travailleurs sur différents sites. Selon [HERB00], ces outils sont classés principalement en deux catégories :

- Outils support pour compenser l'absence de communication informelle,
- Outils qui apportent l'esprit de groupe au travail et tiennent les membres d'équipe informés de l'état de chaque tâche.

Concevoir des outils pour soutenir et encourager la communication informelle est l'approche principale pour résoudre un manque évident dans le travail de développement distribué. Les chercheurs dans le domaine de GSD ont également



essayé de réduire la quantité de communication informelle exigée dans l'organisation et la conception de leur travail ; bien que ceci n'élimine pas le besoin de communication informelle, le but est de la ramener à un niveau plus maniable [REBE01].

### **6.2.2 Gestion de la Documentation et de la Connaissance**

La communauté de développement de logiciel s'est rendue compte qu'un grand nombre de problèmes peuvent être attribués à la connaissance non-capturée et non partagée [MIKA02]. Cela conduit spécifiquement au besoin de savoir "qui sait quoi", au besoin de la collaboration à distance, et au besoin de sauvegarder les acquis et les meilleures pratiques. L'information et la connaissance obtenues pendant les réunions, les correspondances par e-mail, et par messagerie instantanée nécessitent à être facilement capturées, à être efficacement stockées et partagées. La distribution des ressources et des concepteurs dans l'espace et dans le temps, combinée avec l'évolution dynamique de la connaissance rendent nécessaire l'utilisation des outils pour la gestion de la connaissance dans le développement distribué.

La faible documentation peut aussi mener à une collaboration inefficace. En plus d'avoir la conception et le système bien documenté, maintenir la documentation à jour est aussi important. Pour prévenir des incorrections, incohérences et ambiguïtés, et pour soutenir la maintenance, la documentation doit être actualisée et doit refléter sur quoi les divers équipes ou membres travaillent [HERB01].

### **6.2.3 Contrôle de version et traçabilité des modifications**

Dans le génie logiciel, il existe une variété d'outils pour le contrôle de version, la gestion de code source, et l'historique de la traçabilité des modifications, tels que CVS, RCS, et "SourceSafe" [YILI03]. Pour ce faire, CVS (Control Versioning System) utilise les opérations : *Checkout* pour obtenir une copie locale des fichiers, *Commit* pour envoyer ses modifications, *Update* pour obtenir les modifications des autres, et *Import* pour créer un nouveau module. Il est utilisé

pour gérer l'historique et la traçabilité des modifications du code source. Un exemple de CVS très populaire est le TortoiseSVN<sup>18</sup>.

### **6.3 Les outils de groupware**

Le groupware ou le travail collaboratif assisté par ordinateur rassemble une galaxie d'outils plus ou moins efficaces. Nous distinguons quatre grandes familles d'outils.

#### **6.3.1 Les outils de communication de base**

Leur rôle est avant tout de faire circuler une information entre deux personnes. Ce sont les outils "de première nécessité", d'autant plus que les outils des trois autres familles répondent à un besoin plus particulier, même si ces derniers incorporent nécessairement des fonctions de communication. On peut dire que ces outils sont coopératifs, même si les personnes qui communiquent avec ces outils n'ont pas toujours le sentiment de travailler dans un même but. Dans cette catégorie d'outils nous pouvons citer :

- Le courrier électronique
- La messagerie instantanée
- Le RSS (Really Simple Syndication)
- Le chat
- La visioconférence, etc.

La messagerie instantanée permet de dialoguer instantanément par ordinateur avec un interlocuteur distant connecté au même réseau informatique. Elle requiert l'emploi d'un logiciel client qui se connecte à un serveur de messagerie instantanée. Elle diffère du courrier électronique du fait que les conversations se déroulent instantanément. La plupart des messageries instantanées modernes offrent un système de notification de présence, indiquant si les individus de la liste de contacts sont simultanément en ligne et leur disponibilité pour discuter.

---

<sup>18</sup> <http://tortoisesvn.tigris.org/>

Le RSS (Really Simple Syndication) syndication vraiment simple, est un outil de communication asynchrone. Il consiste en un fichier XML dynamique dont les lecteurs de RSS affichent le contenu qui est mis à jour en permanence. Ce système est très utilisé pour diffuser des notifications, des alertes, de listes d'information. Pour l'éditeur du RSS, cela permet de faire connaître les nouvelles information à des autres intervenants sur le site.

### **6.3.2 Les outils de travail partagé**

Ils permettent à plusieurs personnes de travailler sur un même document ou sur une même application. Le traitement de texte Word possède par exemple un outil qui permet d'annoter un texte pour proposer des corrections. Ce sont des outils de collaboration, puisque les différentes personnes qui les utilisent ont le sentiment d'avancer vers un but commun. Ces outils sont relativement peu employés. Leurs utilisateurs sont souvent de grosses communautés ou entreprises, mais de petites structures peuvent aussi y trouver un intérêt. Dans cette catégorie on recense :

- Le partage d'applications
- L'édition partagée
- Les forums et outils apparentés.

### **6.3.3 Les outils d'accès au savoir - ou outils de gestion de connaissance**

Si un membre a fait l'effort de créer un document, ou de développer une expertise, il peut faire gagner beaucoup de temps à ses collègues. Pourquoi créer plusieurs fois le même document - ou développer plusieurs fois la même expertise - lorsqu'on peut les trouver ailleurs dans l'entreprise ou dans la communauté ? Il faut donc faire en sorte que tous les membres puissent accéder à ces informations. Les entreprises qui y ont recours sont souvent les grandes communautés et les grosses PME. Dans cette catégorie d'outils, nous pouvons citer :

- Les bibliothèques
- Les outils de *peer to peer* (pair à pair)
- Les portails web
- Les listes de diffusion
- Les annuaires électroniques

- Les FAQ
- Les moteurs de recherche, etc.

#### **6.3.4 Les outils de workflow**

Ce sont sans doute les outils les plus spectaculaires. L'intervention du logiciel ne se situe plus au niveau de l'information, de la communication ou de la collaboration, mais au niveau supérieur de la coordination. Nous appellerons workflow la modélisation et la gestion informatique de l'ensemble de flux d'informations et de tâches à accomplir par différents acteurs impliqués dans la réalisation d'un processus, au sein d'une organisation. Le terme de workflow pourrait donc être traduit en français par "gestion électronique d'une chaîne de travail". Le workflow décrit le circuit de validation, les tâches à accomplir entre les différents acteurs d'un processus, les délais, les modes de validation, et fournit à chacun des acteurs les informations nécessaires pour pouvoir enchaîner, de manière automatique, des successions de tâches réparties entre plusieurs utilisateurs [GEOR00].

Ces outils sont utilisés par les communautés de toutes les tailles. Dans cette catégorie, nous pouvons citer :

- Les outils de synchronisation
- Les outils de gestion des tâches
- Les agendas partagés.

#### **6.3.5 La nouvelle génération des outils de groupware ou collecticiels**

Les outils de groupware traditionnels comme les calendriers partagés, les gestionnaires de tâches ou le partage de fichiers sont complétés par ces puissants outils de publication de contenu que sont les weblogs, les wikis, les folksonomies et les plones.

##### **6.3.5.1 Les Wiki et Weblog : le renouveau des outils de groupware**

Les technologies Web s'immiscent de plus en plus au cœur de la collaboration en communauté et entreprise. Si le phénomène des blogs est loin d'être récent (fin des

années 1990 aux Etats-Unis), leur utilisation au sein de la sphère professionnelle française est tout juste émergente.

La logique weblog établit une distinction entre le membre de la communauté, qui possède des droits d'accès et d'écriture, et le lecteur qui n'est pas en mesure de modifier le contenu publié mais simplement d'ajouter des commentaires à l'existant.

Or, la logique wiki (sortes de portails qui s'enrichissent grâce à la contribution des personnes qui les consultent) est toute autre. Elle repose sur une charte de bon fonctionnement communément partagée, et permet, d'ajouter et de modifier tout ou partie d'un document. Le wiki permet un véritable esprit d'équipe par son côté à la fois cohésif, comme l'envie de partager et la complicité entre les membres d'une équipe, et cohérent avec l'adoption de méthodes communes.

***Ce n'est plus l'information qui est au centre du dispositif groupware mais l'individu lui-même : le collaborateur ne partage plus seulement un document mais son expérience.***

Le recours à des weblogs ou à des wikis en communauté ne doit pas être perçu comme une remise en cause de l'existant en matière d'outils de travail collaboratif et doit jouer autant que faire se peut la carte de la complémentarité.

Ainsi, le weblog se présente plus comme un service de communication alors que le wiki est un outil de coopération permettant à des individus de pouvoir travailler ensemble, dans le cas de discussions ou de phases de "brainstorming".

Ces outils de publication viennent compléter l'éventail de solutions existantes qui ne sont pas utilisées de façon optimale.

L'exemple le plus représentatif des Wikis est Wikipedia. Wikipedia<sup>19</sup> est une encyclopédie libre, gratuite, universelle, multilingue et écrite collaborativement sur Internet. En cas de désaccord sur le contenu d'un article, un débat se déroule dans la page de discussion annexée à l'article afin de faire émerger un consensus par la discussion. Cette discussion fait partie intégrante de l'encyclopédie, et sa lecture peut être instructive sur les points délicats à traiter dans le contenu de l'article. Contrairement à une idée souvent répandue, il n'y a jamais de vote pour

---

<sup>19</sup> <http://fr.wikipedia.org/>

déterminer le contenu d'un article, car la retranscription des connaissances ne peut pas être issue d'un processus démocratique. Certains usagers ont des droits supplémentaires, ainsi, les administrateurs ont la capacité de : supprimer des pages, protéger une page, bloquer un contributeur à la suite de vandalismes répétés, ou d'un désaccord sur le contenu tournant à la guerre des mots au détriment de l'article.

De nombreuses personnes jugent une telle encyclopédie vouée à l'échec, puisque n'importe quel vandale peut détruire une page ou répandre de fausses informations. Dans la pratique, ce genre de problème ne semble pas arriver fréquemment : si les utilisateurs réguliers du site sont suffisamment nombreux, ils peuvent corriger ces interventions inopportunes dans un délai réduit chaque fois qu'un vandale agit. En général, celui-ci abandonne en constatant que ses efforts sont inutiles.

### **6.3.5.2 Les Folksonomies : exemple de communauté sans contrôle explicite**

Une folksonomie désigne un système de classification collaborative décentralisée et spontanée. Le terme folksonomie est une adaptation française de l'anglais *folksonomy*, combinaison des mots *folk* (le peuple, les gens) et de *taxonomy* (la taxinomie).

L'intérêt des folksonomies est lié à l'effet communautaire : pour une ressource donnée sa classification est l'union des classifications de cette ressource par les différents intervenants (ou contributeurs). Ainsi, partant d'une ressource, et suivant de proche en proche les terminologies des autres contributeurs, il est possible d'explorer et de découvrir des ressources connexes. A l'inverse des systèmes hiérarchiques de classification, les contributeurs d'une folksonomie ne sont pas contraints à une terminologie prédéfinie, mais peuvent adopter les termes qu'ils souhaitent pour classifier leurs ressources. Ainsi, les personnes forment des groupes informels de manière flexible et construisent la structure de leur communauté à la volée.

Un exemple de folksonomie est Del.icio.us<sup>20</sup>. La principale critique que nous pouvons faire sur ces systèmes est que les utilisateurs peuvent créer un phénomène, qui consiste à multiplier des mauvaises interprétations de

---

<sup>20</sup> <http://www.Del.icio.us>

l'information. Bien qu'il n'y ait pas de règles de définition, ils peuvent introduire des termes synonymes/polysémiques, ou encore introduire des interprétations erronées, occasionnées par des considérations subjectives inappropriées.

Contrairement à la Folksonomie, nous pensons que les groupes de travail ont besoin d'une coordination explicite et effective pour atteindre leurs buts.

Dans la suite, nous allons nous intéresser aux systèmes de gestion de contenu qui peuvent intégrer plusieurs outils de groupware ou de collecticiels.

### **6.3.5.3 Les systèmes de gestion de contenu et PLONE**

Les systèmes de gestion de contenu ou SGC (de l'anglais Content Management System ou CMS) sont une famille de logiciels de groupware pour la conception et la mise à jour dynamique de sites Web. Les SGC partagent des fonctionnalités diverses et permettent :

- A plusieurs utilisateurs de travailler sur un même document ;
- De fournir une chaîne de publication (workflow) offrant par exemple la possibilité de publier (mettre en ligne le contenu) des documents ;
- De séparer les opérations de gestion de la forme et du contenu ;
- De structurer le contenu ;
- D'inclure le contrôle de version CVS.

Les SGC travaillent sur des architectures Client-Serveur, ils intègrent et proposent entre autres la messagerie instantanée, la communication asynchrone RSS, les forums Blogs et Wikis. Ils existent, de nos jours, plusieurs SGC dont les plus connus sont : Graffiti<sup>21</sup>, Drupal<sup>22</sup>, et Plone<sup>23</sup>.

Plone est un système puissant et ergonomique de gestion de contenu, il est construit au-dessus du serveur d'applications Zope. Zope est un serveur d'application web orienté objet écrit dans le langage de programmation Python sous la licence GPL/GNU. Il peut être entièrement géré et optimisé à partir d'une interface Web. Zope publie sur le réseau des objets et des scripts Python qui sont

---

<sup>21</sup> <http://www.Graffit.org>

<sup>22</sup> <http://www.Drupal.org>

<sup>23</sup> <http://www.plone.org>

enregistrés dans une base de données objet, ZODB. Des types d'objets basiques, tels que des documents, des images, des patrons (templates) de page, sont à la disposition des utilisateurs pour être créés et gérés via l'internet.

Des types d'objets spécialisés, tels que les wikis, les blogs, les chats, les galeries de photos, etc., sont disponibles en tant que produits, et il existe une communauté mondiale grandissante de petites entreprises créant des applications web sur mesure à base de Zope.

Comme exemples de portails développés avec la technologie Plone, nous pouvons citer les portails du Parlement brésilien (<http://www2.camara.gov.br/>), de la Ville de Berne (Suisse) (<http://www.bern.ch/>), de Jet Propulsion (NASA) (<http://www.jpl.nasa.gov/>), et du Liris (<http://liris.cnrs.fr/>)

Une des forces de Plone est qu'il propose des interfaces intuitives et facile à prendre en main et à utiliser. De même, Plone permet de créer des espaces de travail, réservés à un utilisateur ou à un groupe d'utilisateurs. On peut alors accorder à ceux-ci des droits particuliers dans ces espaces de travail (lecture, contribution, relecture, etc.).

La politique de sécurité de Zope, sur lequel se base Plone, permet de gérer finement les différents accès possibles à un site Plone. Ainsi, chaque action (par exemple "éditer un contenu" ou "modifier la présentation d'une page") est associée à une permission. Puis, à chaque permission est associée une liste de rôles, suivant le contexte et l'état de workflow courant. Cette gestion des utilisateurs est possible grâce à un ensemble d'interfaces homogènes et faciles à utiliser.

Plone intègre un moteur de workflow basé sur les états. Un exemple de ce type de workflow est un circuit de publication d'un document dans un intranet : un membre de l'intranet rédige un document et le donne à valider à un ou plusieurs membres. Plone peut également intégrer un moteur de workflow basé sur les actions. Dans ce cas, on affecte des tâches à des utilisateurs. C'est une autre façon de voir les workflow, souvent utilisés pour décrire des processus industriels. Le moteur de workflow peut servir à définir des workflows simples ou complexes, et permet de restreindre les actions ("demander la publication", "publier", "rejeter", etc.) possibles pour chaque état ("privé", "en cours de validation", "public").



L'accès aux contenus peut être restreint d'après leur état : par exemple, seul le rédacteur d'un contenu pourra accéder à un document qu'il a créé et déclaré privé. De même, dans un site Internet, seuls les contenus "validés" seraient accessibles à l'internaute non authentifié.

Plone propose un moteur de recherche *full-text* puissant et rapide qui permet de rechercher des mots ou groupes de mots dans tout contenu déposé sur un site Plone, y compris les fichiers. Il propose plusieurs fonctionnalités avancées. Ce moteur de recherche peut également être utilisé pour rechercher les documents créés ou modifiés après ou avant une certaine date (voire entre deux dates), par un ou plusieurs utilisateur(s) spécifique(s) du site, etc. De plus, les résultats de recherche peuvent être affichés dans n'importe quel ordre : pertinence, date de publication, etc. Un fil de syndication RSS (liste ordonnée d'informations) est automatiquement créé à chaque nouvelle recherche. Les mots recherchés sont ensuite surlignés dans le document qui les contient.

#### **6.4 Conclusion**

Dans ce chapitre, nous nous sommes intéressés aux CSCW et groupware en analysant les dimensions des défis auxquels ils sont confrontés tels que la distance et la communication, la gestion de la documentation et de la connaissance, et le contrôle de version et la traçabilité des modifications. Ensuite nous avons décrit les outils de groupware utilisés en général dans les communautés et entreprises. Nous verrons dans la suite comment les défis liés aux groupwares dans le domaine de GSD (Global Software Development) se manifestent dans le domaine de la conception d'ontologie. Le chapitre suivant porte donc sur les outils et les technologies groupware utilisés ou intégrés dans le développement d'ontologie.

## **Chapitre 7. Les Groupwares pour la construction des ontologies**

Dans ce chapitre, nous nous intéressons à l'utilisation des CSCW et des Groupwares dans le développement des ontologies, en décrivant, comparant et analysant les outils et technologies utilisés dans le domaine. Nous nous concentrerons principalement sur les aspects collaboratifs des outils, c'est à dire, nous examinons les faiblesses et les forces de chaque outil par rapport au support collaboratif qu'il propose.

Avant de nous intéresser aux outils de construction collaborative des ontologies, nous avons juger utile de voir ce qui existe déjà comme outils de développement d'ontologie en général. Nous avons pu, dans un premier temps, recenser des outils qui ne s'occupent pas des aspects collaboratifs de la construction d'ontologie. Dans cet ordre d'idée, nous pouvons citer entre autres : KAON [MAED03], Oiled [BECH01], OntoSaurus [SWART97], WebODE [ARPI03], WebOnto [DOMI98]. Une liste exhaustive des outils de construction d'ontologie sera donnée dans les annexes de la thèse.

Concernant la construction des ontologies dans un environnement collaboratif, le nombre d'outils a considérablement augmenté et a été diversifié. Il existe de nombreux logiciels payants ou sous licence GNU/GPL disponibles pour la construction des ontologies. La plupart d'entre eux supportent directement des normes standard d'ontologie (RDFS, OWL), intègrent un moteur d'inférence avec des infrastructures de base de données et sont la réalisation pratique des études méthodologiques.

Les outils présentés ici sont les plus largement utilisés dans la communauté de développement d'ontologie. Ontolingua Server [FARQ96], OntoEdit [SURE02], et APECKS [JENI98] sont des outils construits sur une architecture client-serveur avec différentes approches pour le support de collaboration. Protégé-2000 [NOY00] est un outil "indépendant d'une plateforme" largement utilisé dans le domaine clinique et médical pour créer et éditer des ontologies. Cet outil a commencé par des fonctionnalités soutenant le développement en solo et a évolué vers le support des travaux de collaboration en groupe. Ces outils sont largement répandus dans la communauté d'Ingénierie de Connaissances et offrent des

environnements complets de développement d'ontologies qui proposent, d'une manière ou d'une autre, un support coopératif. Les autres technologies présentées sont des projets qui, en raison de leur récente création, sont seulement parfois au stade des méthodologies sans outils encore développés.

Dans l'état de l'art présenté sur les groupwares, nous avons identifié les services suivants comme les plus pertinents :

- Communication Synchrone / Asynchrone,
- Traçabilité des modifications,
- Gestion d'accès concurrents,
- Contrôle de Versions CVS,
- Hiérarchisation des utilisateurs,
- Édition en ligne.

L'ensemble de ces services nous sert de guide pour pouvoir comparer les méthodes et outils de construction collaborative des ontologies.

### **7.1 Ontolingua Server**

Ontolingua Server [FARQ96] est un système développé à l'Université de Stanford et les services KSL Network. C'est un ensemble d'outils et de services utilisés pour soutenir le processus de réalisation des ontologies partagées par des groupes géographiquement distribués. Ces outils se servent du World Wide Web pour permettre un large accès et fournir aux utilisateurs la possibilité de publier, créer, naviguer et éditer des ontologies formelles stockées sur un serveur d'ontologies. Ontolingua Server permet à plus de 150 utilisateurs de travailler activement en ligne [FARQ96]. Ontolingua recommande d'utiliser la méthodologie *Ontology Development 101*.

Afin de permettre la réutilisation d'ontologies, le serveur stocke une bibliothèque centrale des ontologies, et fournit aux développeurs des droits d'accès à cette bibliothèque. Le serveur soutient la fusion par l'intégration de l'outil Chimaera [MCGU00], les mises à jour et les extensions des ontologies de la bibliothèque.

Le support du développement distribué et collaboratif des ontologies vers le consensus est fourni par l'interface web au moyen des sessions de contrôle d'accès

pour utilisateurs et groupes. Les dispositifs de verrouillage et l'analyse des définitions alternatives des concepteurs multiples facilitent l'accès concourant à une ontologie partagée.

La technologie de Ontolingua Server soutient trois modes d'utilisation :

- Les groupes distribués peuvent naviguer et rechercher des ontologies stockées dans la bibliothèque (réceptacle) en utilisant un navigateur web. Les utilisateurs peuvent également se servir du langage d'Ontolingua pour construire et maintenir des ontologies stockées sur le serveur. Le langage utilisé par Ontolingua est basé sur le format KIF.
- Les applications à distance peuvent interroger et modifier les ontologies stockées sur le serveur à travers Internet en utilisant un API réseau [PETE95].
- Les utilisateurs peuvent traduire une ontologie dans un format utilisé par une application spécifique. Par exemple une traduction IDL (Interface Definition Language) peut produire un fichier d'en-tête IDL qu'un programme conforme de CORBA peut utiliser pour interagir avec une requête d'objet

## **7.2 OntoEdit et sa suite OntoStudio**

OntoStudio [SURE02] est la suite de son prédécesseur OntoEdit. Il a été développé à l'Université de Karlsruhe en Allemagne. OntoStudio est un éditeur d'ontologies pour le Web sémantique, disponible dans des versions de *freeware* et professionnelle (payant). La version professionnelle inclut un ensemble additionnel des modules : OntoKick et Mind2Onto. Ces deux modules supportent la partie collaborative d'OntoStudio. OntoKick repose sur une architecture client serveur qui permet la gestion des sessions, gestion des droits d'accès, gestion des verrous pour les accès concurrents à une même donnée. Mind2Onto supporte les premières étapes de spécification et de raffinement des ontologies avec une approche de *Brainstorming*. Pour sa part Ontokick est utilisé pour la vérification et la validation de cohérence pour les ontologies multiples.

OntoStudio est un outil qui propose comme méthodologie On-To-Knowledge. Nous retenons de OntoStudio sa capacité de stocker et de traiter des ontologies non formelles. Il est structuré sur un langage semi-formel pour décrire une

ontologie. Son module OntoKick importe le fichier XML de Mind2Onto [SURE02].

### **7.3 APECKS**

APECKS [TENN02] (Adaptative Presentation Environment for Collaborative Knowledge Structuring) est un environnement de travail collaboratif, qui a été développé par le département de la Psychologie à l'Université de Nottingham. C'est un serveur d'ontologies avec un support de collaboration permettant à des experts de domaine de créer des ontologies basées sur leur propre perception. APECKS permet aux utilisateurs de comparer leurs perceptions (prototype, conception rationnelle, etc..) à la lumière des discussions au sujet des sources de leurs différences et similitudes. Selon Apecks, l'outil pour le développement devrait soutenir la collaboration entre les concepteurs à travers la communication structurée et non structurée. APECKS essaye ainsi de permettre aux experts d'expliquer les sources de leurs désaccords et de discuter pour atteindre un objectif. Dans APECKS l'accent n'est pas mis sur les résultats qu'il produit, mais plutôt sur le processus : les désaccords et les discussions qui surviennent pendant la construction d'une ontologie consensuelle.

Les serveurs d'ontologies de APECKS sont conçus pour permettre à un certain nombre d'utilisateurs de construire ensemble une ontologie et la communication entre ces concepteurs est soutenue par les mécanismes suivants :

- L'abonnement : les utilisateurs souscrivent à certains centres d'intérêt dans une représentation de la connaissance. Ils sont alors notifiés de toutes les modifications qui se produisent dans ces secteurs.
- L'annotation : les annotations des utilisateurs peuvent être sauvegardées et attachées à tous les concepts ou instances pour des références.
- Les sessions de groupe : les utilisateurs qui ont souscrit et qui travaillent dans la même session de groupe peuvent recevoir la notification synchronisée quand des modifications sont faites par d'autres utilisateurs dans la session.
- La communication synchrone : les collaborateurs peuvent s'envoyer (entre eux) des messages courts, incluant des images et des fichiers d'ontologie.

Le processus conduit à la découverte des détails liés au domaine, et les critères utilisés dans la construction des ontologies sont rendus explicites.

APECKS est adapté pour le travail collaboratif entre groupes de personnes distincts : ingénieurs de la connaissance qui ont l'expertise de construire des ontologies et les experts de domaine qui ont la connaissance du domaine en question. La tâche de l'ingénieur de la connaissance est de construire des ontologies consensuelles de la connaissance d'un certain nombre d'experts. Ceci mène au type d'affirmation [RICE96] que les ontologies, par définition, représentent la connaissance consensuelle, les définitions admises des termes techniques utilisés dans le domaine.

#### **7.4 Système CO4 et protocole CO4**

Le système CO4 (Cooperative Construction of Consensual Knowledge) [EUZE96 ; ALEM98] est conçu pour la construction incrémentale et concurrente d'une base de connaissances. Ce système a été d'abord développé et utilisé dans le domaine de la génétique moléculaire. Le protocole CO4 et l'implémentation du système CO4 supportent la construction collaborative d'une base de connaissance formelle, et permettent à des collaborateurs d'annoter librement, d'exprimer et de manipuler leur connaissance avec du hypertexte, des images, et des données expérimentales [EUZE96]. Le système CO4 utilise un navigateur web comme médium de présentation. Il traite spécifiquement le problème du consensus de la base de connaissance à l'aide du protocole CO4 en intégrant la connaissance à travers plusieurs niveaux des bases de connaissance consensuelles. Capturer l'esprit de corporation par la création coopérative des bases de connaissance et des hyper-documents est la motivation du système CO4.

Les principes de base du protocole CO4 sont dérivés du protocole "peer-reviewing" [PETE96] : avant d'être intégrée dans une base de connaissance consensuelle, la connaissance doit être soumise, analysée et acceptée par la communauté [EUZE96]. On doit s'assurer qu'à la fin du processus de développement, la connaissance stockée dans la base de connaissances soit assez consistante et cohérente de sorte que quiconque puisse l'accepter et l'utiliser avec confiance et facilité. La connaissance informelle est aussi sujette à des soumissions, analyses, et autres.

Les motivations par rapport à la conception dans ce système se situent dans trois aspects :

- La connaissance doit être énoncée aussi formellement que possible. Formaliser la connaissance présente l'avantage de capturer la sémantique, créant une ontologie standard et permet aux agents logiciels de manipuler la connaissance, de faire la recherche et le raisonnement intelligents.
- Tout ne peut être formalisé, et même si tout était formalisé, le système formel peut juste souffrir des problèmes tels que la complexité et l'incohérence [EUZE96]. Ainsi, la connaissance formelle doit être soutenue par une documentation riche, des annotations, des images, des animations et même du matériel video/audio. La connaissance formelle, la connaissance qui n'a pas encore atteint un état formel, et la connaissance qui ne peut pas être formalisée sont toutes bien supportées par les formes informelles de la connaissance.
- Les personnes doivent être soutenues dans la discussion de la connaissance introduite dans la base de connaissances. Dans cette perspective, créer, réutiliser, modifier et maintenir la connaissance devraient être une activité participative de toutes les personnes impliquées [EUZE95,96]. Supporter la discussion et la construction du consensus assure la consistance et la cohérence de la base de connaissances produite, et son acceptation et son utilité dans la communauté d'utilisateurs.

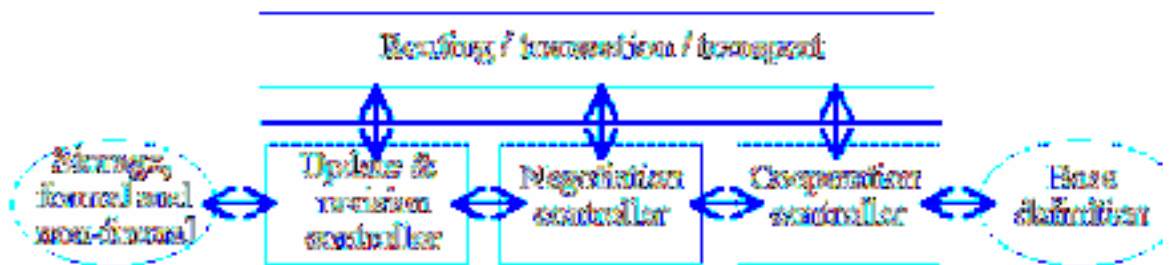


FIGURE 7.4 L'ARCHITECTURE DU SYSTEME CO4. LES RECTANGLES REPRESENTENT LES MODULES DU LOGICIEL, LES CERCLES SONT DES DEPOTS DE DONNEES/CONNAISSANCES ET LES FLECHES REPRESENTENT L'APPEL DE LA FONCTIONNALITE DU PROGRAMME [EUZE96]

## 7.5 Protégé-2000

Protégé 2000<sup>24</sup> [RAYF01] est un système développé à l'Université de Stanford. C'est un outil très populaire dans le domaine du Web Sémantique et très reconnu au niveau de la recherche en informatique. Il est licencié GNU/GPL.

Protégé peut lire et sauvegarder des bases de connaissances dans la plupart des formats d'ontologies : RDF, RDFS, OWL, etc. A l'origine Protégé-2000 est un outil mono-utilisateur de construction d'ontologie, mais sa version bêta de la version multi-utilisateurs est actuellement accessible pour le travail en groupe. Dans cette version, l'ontologie en cours de développement est stockée dans une base de données partagée, où les utilisateurs multiples peuvent lire la même base de données et faire des modifications incrémentales ou des modifications qui ne sont pas en conflit entre elles. Le mot "multi-utilisateurs" est utilisé pour décrire le fait que la version bêta a des fonctions pour soutenir le travail collaboratif limité entre concepteurs.

Protégé-2000 utilise le protocole ouvert de connectivité de base de connaissance OKBC (Open Knowledge Base Connectivity) pour l'interrogation de sa base de connaissances et son interface de construction afin d'atteindre l'interopérabilité avec d'autres systèmes de représentation de la connaissance.

De nombreux utilisateurs préfèrent Protégé pour sa convivialité et l'adaptation de ses *plugins*. Jambalaya [STOR01] est un plugin de Protégé qui permet la visualisation et la navigation dans le réseau sémantique d'une manière très performante.

Protégé recommande d'utiliser la méthodologie Ontology Development 101 pour la construction des ontologies.

## 7.6 CODE

CODE (Collaborative Development Environment for Ontologies) [CAÑA04] est un projet développé à l'Université de Floride. Il cherche à construire des ontologies de façon collaborative par la communication entre des experts. Pour

---

<sup>24</sup> [http://protege.stanford.edu/doc/protege2\\_alpha/index.html](http://protege.stanford.edu/doc/protege2_alpha/index.html), 2003, Sept 16



cela il propose une suite d'outils appelés "Cmap Tools" (ou carte conceptuelle) qui supportent la communication à trois différents niveaux [CAÑA04] :

- Les utilisateurs peuvent collaborer pour partager leur carte conceptuelle sur des serveurs d'accès restreint,
- Les utilisateurs peuvent ouvrir une session synchrone pour éditer un "*concept map*" et finalement ils peuvent ajouter des notes pour la discussion des "*concept maps*" déjà constitués.
- Les "Cmap tools" offrent aussi un schéma des permissions pour effectuer des projets avec « peer review » [PETE96].

## **7.7 DILIGENT**

DILIGENT [VRAN05] (Distributed, Lossely-controlled and evolInG Engineering process of oNTologies) est un modèle qui s'intéresse à l'harmonisation et aux discussions des ontologies. Pendant les discussions les participants partagent leurs arguments d'acceptation ou de rejet sur les décisions à prendre. DILIGENT se base sur des travaux en ingénierie logicielle qui ont montré que la représentation des argumentations aide les utilisateurs à mieux comprendre leurs décisions collectives. Les argumentations sont présentées formellement pour pouvoir détecter des contradictions car plus les ontologies s'élargissent, plus les utilisateurs peuvent argumenter d'une façon contradictoire même sans s'en rendre compte [VRAN05].

## **7.8 C-VISTA**

C-VISTA [DIEN02] a été développé à l'Institut National de Recherche en Informatique et Automatique (INRIA). C'est un modèle pour la construction coopérative d'ontologie, qui propose l'utilisation des points de vue. Ces points de vue servent à définir des propositions formelles et l'indexation des descriptions consensuelles du même objet par différents experts appelés Perspectives. Ces points de vue sont complémentaires et donnent une vision globale de l'objet. Les objets doivent être uniques, mais les caractéristiques visibles s'adaptent aux points de vue de l'expert.

C-VISTA propose aussi l'indexation non consensuelle des descriptions qui correspondent à des approches spécifiques de chaque expert appelés Opinions. Ces vues peuvent avoir des descriptions incomplètes sur l'objet et peuvent être incohérentes au regard de la communauté. Ce modèle propose d'indexer les définitions consensuelles et formaliser les Perspectives par les règles suivantes :

1. Nommer différemment les types de concepts décrits par le même terme par différents experts. Par exemple concaténer l'initiale du nom de l'expert.
2. Si deux types de concept de deux différents points de vue ont des définitions similaires, ils sont associés par un lien d'équivalence.
3. Si deux concepts de différents points de vue ont des définitions compatibles, l'une est incluse dans l'autre, un lien d'inclusion peut être fait entre les deux.
4. Si deux concepts décrivent différentes propriétés d'un concept base et si parmi les experts ces propriétés ne peuvent pas être présentées en même temps sur la même instance, alors un lien d'exclusion peut être fait [DIEN02].

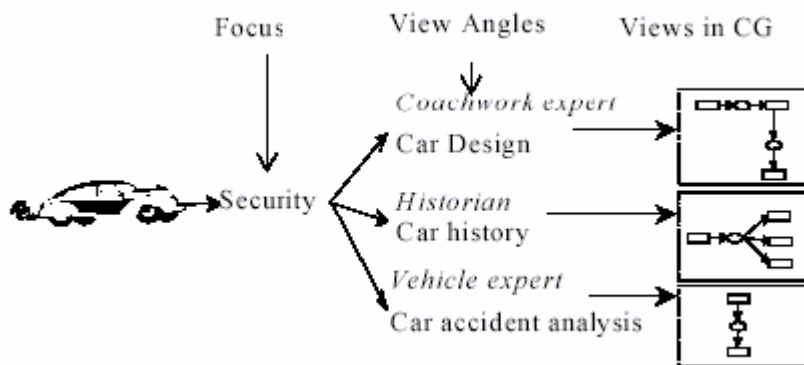


FIGURE 7.5 POINTS DE VUE POUR C-VISTA.

## 7.9 La méthode DELPHI

La méthode Delphi [HOLS02] est décrite comme : "*... une technique formelle pour rassembler et intégrer les points de vue des personnes multiples sur un sujet. Chaque participant fournit indépendamment, par écrit, des vues à un chef, qui prépare un document de synthèse reflétant les vues combinées comme retour pour le prochain round. Dans le deuxième round, les participants fournissent leurs vues écrites indépendantes à la lumière de la synthèse. De cette façon, le chef (d'équipe de conception) essaye de stimuler une convergence (un consensus) des vues à travers les rounds successifs*".

Nous avons quatre phases dans cette approche:

- La préparation : dans la phase préparatoire les critères de conception de l'ontologie, le domaine et la portée de l'ontologie sont définis, et un ensemble de normes pour évaluer l'ontologie sont déterminés.
- Le choix de l'ontologie d'ancrage : cette phase permet de spécifier l'ontologie sur laquelle les participants concentreront leur collaboration.
- La construction itérative : cette phase permet d'identifier les participants au travail collaboratif, d'obtenir leurs critiques et commentaires sur l'ontologie, de mettre à jour l'ontologie en prenant en compte les avis des participants (Notification + Consolidation), et de réitérer jusqu'à ce que le consensus soit atteint
- L'application de l'ontologie. Dans cette phase, l'utilité et les utilisations de l'ontologie résultante sont démontrés.

La figure 7.6 ci-dessous illustre une approche collaborative dans la conception d'ontologie, qui utilise la méthode Delphi.

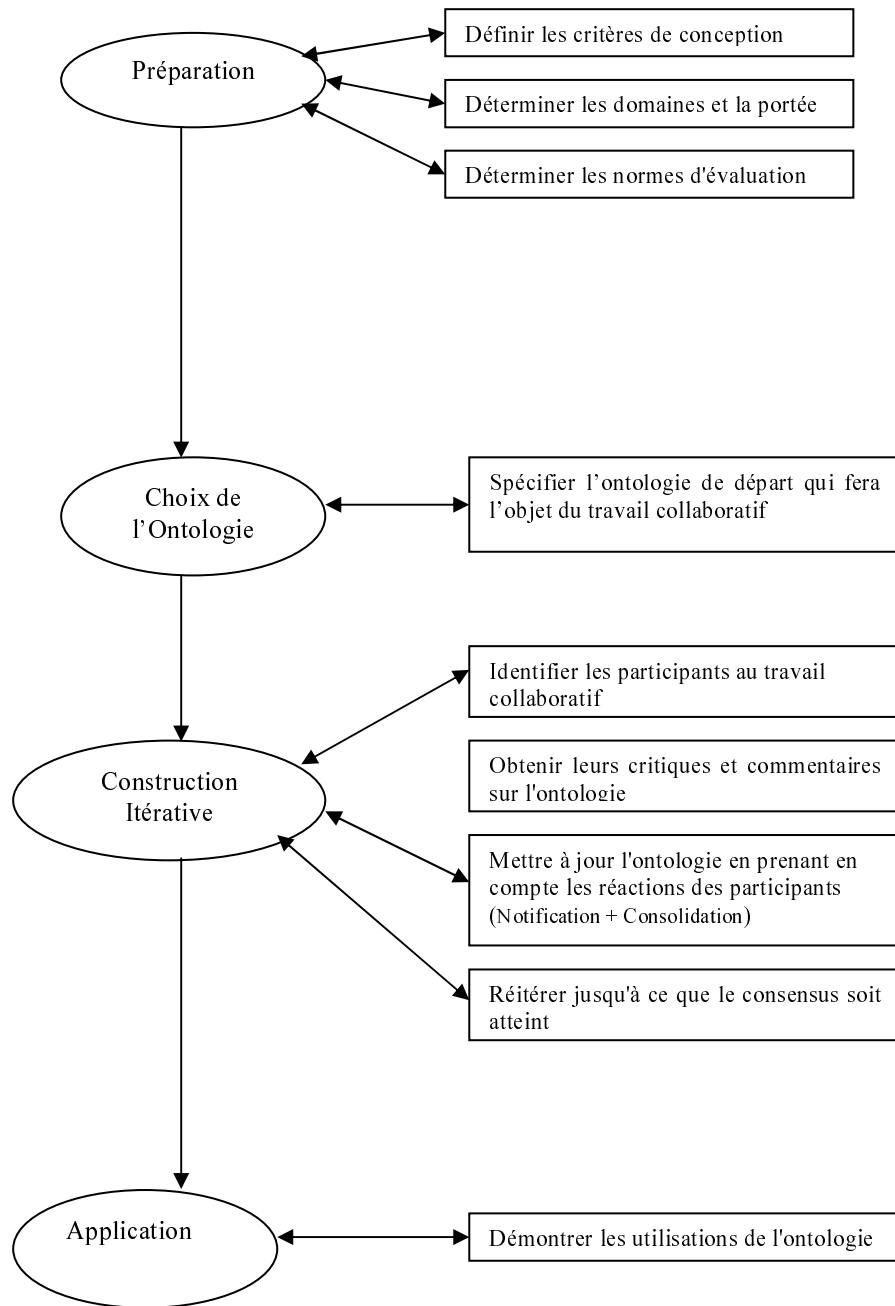


FIGURE 7.6 APPROCHE COLLABORATIVE POUR LA CONCEPTION D'ONTOLOGIE.

Après avoir étudié les différents outils et approches de construction collaborative des ontologies, nous allons dans la section suivante les comparer.

## 7.10 Comparaison des outils

Ontolingua Server et OntoEdit adoptent tous les deux une infrastructure client-serveur pour soutenir la collaboration concurrente et distribuée. Ontolingua Server utilise le protocole HTTP pour faire parvenir des notifications de modifications aux clients ; mais en raison de la nature asynchrone du HTTP, ce mécanisme de notification ne satisfait pas au critère de la collaboration synchrone. Utiliser l'omniprésent navigateur web comme médium interactif pour naviguer et éditer des ontologies sur le serveur Ontolingua attire une grande communauté d'utilisateurs parce qu'il accroît le taux d'acceptation par les utilisateurs, et évite plusieurs pièges de la distribution logicielle [RICE96]. Cependant, l'interface basée sur le web dans Ontolingua impose aussi un ensemble de contraintes à l'interface utilisateur, telle que la non possibilité de réponse de l'application.

OntoEdit utilise un protocole de verrouillage et de transaction et implémente un modèle distribué d'événement sur la base de l'invocation à distance de la méthode Java pour fournir la cohérence et la consistance. Le mécanisme de verrouillage, la gestion de transaction et le support d'inférence dans OntoEdit font de lui la référence parmi plusieurs outils de collaboration. Cependant, utiliser le verrouillage pour coordonner la collaboration crée également quelques problèmes quand il y a une grande équipe de conception. La méthode de verrouillage utilise un algorithme exclusif et mutuel, qui a été longtemps utilisé dans les systèmes d'exploitation, pour coordonner des processus multiples fonctionnant simultanément. Dans le cas de OntoEdit, chaque tâche sur laquelle travaille un développeur est traitée comme un processus en cours, et est suspendue et reprise selon l'accessibilité aux ressources (concepts, instances, sous-arbre hiérarchique d'ontologie). Le processus de collaboration entre les concepteurs d'ontologie et les experts de domaine est réalisé à travers la communication telle que le courrier électronique, les appels téléphoniques, la collaboration en face-à-face, la messagerie instantanée, et même les causeries (informelles) de couloir. Ce n'est pas un processus qui peut être complètement automatisé pour arriver à l'utilisation maximum des ressources. Le meilleur support collaboratif qui facilite à la fois la nature formelle et ad-hoc de la communication entre les développeurs est nécessaire dans ce cas pour soutenir la coordination.

Les développeurs du serveur d'ontologie APECKS ont identifié la communication entre des concepteurs d'ontologie comme une tâche importante et ont essayé de fournir le support explicite pour cette tâche. Les concepteurs de APECKS ont trouvé que, plus de communication entre les constructeurs d'ontologies permettra une discussion plus approfondie sur un domaine, menant à la fois aux ontologies plus riches et de conception rationnelle, qui peuvent être analysées et utilisées pour informer la construction des futures ontologies [JENI98]. Pour APECKS (et comme pour Ontolingua Server), les supports de communication et de collaboration sont à la fois construits comme des extensions des serveurs HTTP, et ils présentent aux utilisateurs une vue de l'ontologie basée sur des "frames" dans le navigateur web. Dans APECKS, tout utilisateur peut définir des ontologies multiples dans un domaine, en représentant différents aspects du domaine ou différentes tâches qui pourraient être effectuées dans ce domaine [JENI98]. Les ontologies sont partagées d'une manière semblable aux bases de connaissances dans CO4, qui sont basées sur le partage et le réseau en groupe. APECKS prévoit l'utilisation des ressources réseau distribuées sur l'espace de travail d'un développeur individuel. Il est conçu aussi bien pour les services client que pour les services serveur, permettant d'accéder aux applications d'acquisition de connaissance par réseau et à d'autres systèmes d'ontologie [JENI98].

Le système CO4 souligne la collaboration et la construction du consensus entre des concepteurs. Il prête, en particulier, l'attention à l'acquisition de la connaissance informelle, et utilise effectivement cette connaissance comme matériels d'annotation à la connaissance formelle, et comme matériels auxiliaires de communication entre les utilisateurs. En utilisant des navigateurs web et le hypermédia pour la présentation, la navigation et l'interaction d'utilisateurs rendent certainement CO4 facile à apprendre et à utiliser, mais la présentation de l'information hiérarchique complexe reste un challenge pour le navigateur web.

Protégé-2000 souligne la réutilisation d'ontologie et l'interopérabilité entre différents outils. L'importance d'un outil par rapport à sa facilité d'utilisation dépend de sa capacité de présentation et de laisser l'utilisateur interagir effectivement avec l'ontologie. Les modèles d'ontologie supportent habituellement

de multiple héritages dans les hiérarchies de concept et hiérarchies de relation ; l'approche standard est l'utilisation des vues multiples ou imbriquées d'arbre avec la capacité d'étendre et de contracter les niveaux hiérarchiques. Le plug-in de visualisation de Protégé fournit divers niveaux d'abstraction sur la structure complexe d'ontologie à travers une interface utilisateur entièrement "zoomable" rendant l'interaction avec l'utilisateur plus intuitive, augmentant la compréhension et permettant une navigation plus facile [RAYF01].

C-Vista et DELPHI proposent des approches intéressantes incluant la gestion des points de vue, mais ne disposent pas encore d'outil supports.

En résumé, nous constatons que les technologies Web sont très utilisées dans les outils de construction de base de connaissances. Nous pouvons affirmer qu'il existe plusieurs méthodes et outils pour la construction collaborative des ontologies, et qu'aucun n'implémente tous les aspects du groupware. La plupart du temps, la meilleure solution dépend du contexte, du domaine d'application et des outils spécifiques de construction des ontologies. Dans le tableau suivant nous synthétisons les services de groupware proposés par chacune des approches étudiées. Le double trait en gras partage le tableau en deux parties, celle des approches existantes avant le début de la thèse et celle des approches apparues pendant la thèse. Un tableau plus détaillé est consultable dans les annexes de cette thèse.

TABLEAU 7.1 COMPARAISON DES OUTILS ET APPROCHES COLLABORATIFS POUR LA CONSTRUCTION D'ONTOLOGIES

	Interface utilisateur	Discussion en ligne	Point de vue	Traçabilité	Possibilité d'annotations	Visualisation en forme de graphe
<b>ONTOLINGUA</b> [FARQ96]	Navigateur Web	Non	Non	Non	Non	Oui
<b>CO4</b> [EUZE96,ALEM98]	Navigateur Web	Asynchrone	Non	Non	Oui	Non
<b>PROTÉGÉ</b> [RAYF00]	Stand-alone	Non	Non	Oui	Non	Oui
<b>DELPHI</b> [HOLS00]	Non	Non	Oui	Oui	Oui	Pas d'information
<b>OntoStudio</b> [SURE01]	Stand-alone	Non	Non	Oui	Oui	Oui
<b>APECKS</b> [TENN02]	Navigateur Web	Asynchrone	Non	Oui	Oui	Non
<b>C-VISTA</b> [DIEN02]	Non (en développement)	Non	Oui	Oui	Oui	Pas d'information
<b>CODE</b> [CANA04]	Stand-alone/ Navigateur Web	Synchrone/ Asynchrone	Oui	Oui	Oui	Oui
<b>DILIGENT</b> [VRAN05]	Web Wiki	Asynchrone	Non	Oui	Non	Non





## 7.11 Analyse de la comparaison

Le développement collaboratif d'ontologie, la construction du consensus dans le processus de développement, l'interopérabilité, et la réutilisation sont les thèmes (ou aspects) communs à tous ces outils. Chacun des outils de développement d'ontologie et de base de connaissances a sa force dans un ou deux de ces aspects, et faiblesse dans d'autres. Une des raisons de cette diversité est que chaque outil provient d'un domaine différent et que chacun est basé sur une méthodologie de conception d'ontologie particulière, ou sur un modèle de processus de développement spécifique.

Les insuffisances dans ces outils au regard du support de collaboration sont :

- Le travail de groupe coordonné (tel que l'édition, la discussion ou l'annotation collaboratives) n'est pas bien soutenu, principalement parce que les systèmes manquent de fonctions pour tenir les concepteurs informés des tâches et des activités des autres.
- Le support de communication contextuel n'est pas pris en compte dans ces outils, forçant les utilisateurs à compter sur des outils à but général, comme des applications de courrier électronique, pour communiquer avec d'autres. Le courrier électronique est bien pour les communications formelles ou semi-formelles, mais considérant que dans beaucoup de cas les utilisateurs doivent établir des liens par e-mail avec l'environnement de développement, le courrier électronique n'est pas de loin le meilleur choix. La documentation reflétant une discussion cohérente est inévitablement dispersée dans les boîtes aux lettres de plusieurs personnes. En théorie nous pourrions associer tous ces documents/vues pour reconstruire la transcription complète et obtenir une image entière du sujet. Dans la pratique, cependant, personne n'a le temps, la compétence, ou la motivation de mener à terme l'ensemble de toutes ces tâches [O'REI99].

La plupart des outils décrits centrent leur support de conceptualisation et d'analyse des incohérences sur l'expérience de l'ingénieur de connaissances. Par exemple, dans Ontolingua et dans Protégé la communication se fait entre ingénieurs de connaissances. On remarque que ces outils travaillent sur la création des

ontologies une fois le consensus déjà réalisé (ontologies post-consensuelles). Ontolingua va plus loin en imposant un vocabulaire disjoint sur les concepts des différentes ontologies qu'il stocke sur son serveur.

Seuls les modèles C-VISTA, Delphi et CODE supportent le traitement des points de vues. Ainsi, un objet de la conceptualisation peut regrouper les différents points de vues des experts. Même si Protégé 2000 reste encore l'outil le plus utilisé, il permet surtout de construire des ontologies post-consensuelles formelles.

## **7.12 Conclusion**

Dans ce chapitre, nous nous sommes intéressé à l'utilisation des CSCW et des Groupwares dans le développement des ontologies, en décrivant, comparant et analysant les outils et technologies utilisés dans le domaine. Nous avons concentré notre étude principalement sur les aspects collaboratifs des outils, en rapportant les faiblesses et les forces de chaque outil par rapport au support collaboratif qu'il propose.

Avant de proposer nos contributions, nous allons conclure sur la partie 3 (état de l'art), en faisant ressortir les manques par rapport à nos besoins de construction d'ontologie urbaine pré-consensuelle.

## **Conclusions sur l'état de l'art**

Dans cette partie, nous avons présenté un état de l'art complet sur les ontologies. D'abord nous avons donné les différentes définitions, les différents types, composants et usages des ontologies. Ensuite, nous avons décrit les approches, méthodes et méthodologies existantes de construction d'ontologie. Puis nous nous sommes intéressé aux langages de description des ontologies. Enfin nous avons étudié les technologies de groupware en général et leurs applications dans le domaine des ontologies.

L'une des caractéristiques essentielles des ontologies est qu'elles fournissent des données consensuelles sur un domaine donné. Cependant la plupart des méthodes et méthodologies ne proposent pas de directives pour atteindre le consensus quand les ontologies sont construites de manière collaborative.

Pour le développement rapide et distribué des ontologies, l'équipe de conception a besoin d'utiliser le support de la technologie collaborative, qui s'intègre au domaine des CSCW.

Le développement distribué d'ontologie continuera à inévitablement relever des défis semblables à GSD (Global Software Development), vu l'importance croissante des ontologies en terme de conception et d'utilisation dans d'autres domaines. Développer une ontologie sur des sites multiples est toujours un domaine relativement peu exploré. Dans certains cas, l'obstacle insurmontable de la distance a forcé le travail de conception à être effectué sur un site central, aux dépens de l'expertise d'autres experts sur des sites distants.

A partir des leçons et des expériences tirées du domaine de GSD, les outils existants essayent tant bien que mal de trouver une solution à la difficulté du développement collaboratif d'ontologies qui se manifeste dans trois dimensions : celle de la distance et de la communication, celle de la gestion de la documentation et de la connaissance, et celle de la traçabilité des modifications et du contrôle de version.

Le développement distribué d'ontologie souffre des problèmes semblables identifiés au niveau des groupwares en général, par exemple, les concepteurs trouveraient des solutions rapides, par rapport au manque de cohérence au niveau

des concepts et des relations, s'ils se trouvaient au même emplacement et s'ils bénéficiaient de la communication informelle. Cela devient beaucoup plus ennuyeux dans le domaine d'évolution rapide de la technologie d'ontologie où les concepteurs géographiquement distribués travaillent sur les tâches dépendantes les unes des autres.

Le support de communication efficace est aussi bien important pour le développement distribué d'ontologie que pour le génie logiciel, parce que les deux ont des processus de développement et des modèles de cycle de vie similaires. L'utilisation simple des applications de messagerie instantanée (MI) dans les environnements de développement d'ontologie pose problème parce que la MI n'est pas intégrée dans l'environnement de développement d'ontologie et il est difficile d'associer le message avec de l'information contextuelle. L'intégration d'un système de messagerie instantanée à un système de développement d'ontologie à la fois aux niveaux des données et de la présentation serait une étape importante vers une messagerie complète et de haute qualité. Implémenter un système de messagerie instantanée à partir de zéro n'est évidemment pas nécessaire ; il serait intéressant d'étudier les divers systèmes ou plate-formes de MI et d'en choisir pour l'intégration.

La gestion du contenu se rapportant aux systèmes utilisés dans le génie logiciel peut être aussi bien utiles dans le domaine de la conception d'ontologie. Une étude des outils utilisés pour la documentation et la gestion de la connaissance dans le génie logiciel nous fournit des perspectives (ou des possibilités) sur la façon dont ils peuvent être utilisés efficacement pour le développement d'ontologie.

Dans le génie logiciel, il existe une variété d'outils pour le contrôle de version, la gestion de code source, et l'historique de la traçabilité des modifications, tels que CVS, RCS, et "SourceSafe". En revanche, les technologies de pointe pour soutenir le contrôle de version et la traçabilité des modifications dans le développement d'ontologie ont encore un long chemin à faire. Les systèmes de contrôle de version dans le génie logiciel ne sont pas directement applicables au développement d'ontologie. Par exemple, l'unité minimum du contrôle de version dans le système CVS est le fichier ; le système n'a pas la notion des concepts, des instances et des

relations, qui sont les éléments de base d'une ontologie, et qui sont sujet au changement de version et à la traçabilité de modifications.

Après avoir posé les hypothèses de solutions à la complexité et à la particularité du domaine de l'urbanisme, et après avoir analysé et évalué les outils de construction coopérative des ontologies, nous proposons, dans la suite de la thèse, nos contributions, notre modèle d'ontologie qui permettent la mise en place d'un cadre de travail unifié pour nos experts de l'urbanisme, leur permettant de coopérer. Notre modèle satisfait à la discussion des définitions pour arriver au consensus. Ces discussions pourront aussi guider l'acquisition des connaissances lors de la construction des ontologies.

## **Partie 4. Nos travaux et contributions : la conception coopérative d'ontologies pré-consensuelles**

Dans cette partie, nous présentons d'abord nos travaux, dans le cadre du projet Towntology, sur la conception coopérative d'ontologies pré-consensuelles. Ensuite nous relatons notre expérimentation avec nos experts du laboratoire EDU (Equipe Développement Urbain). Enfin nous présentons les caractéristiques logicielles de notre prototype dans le chapitre sur l'implémentation.

## **Chapitre 8. La conception coopérative d'ontologies pré-consensuelles**

Après avoir analysé dans les parties précédentes, la complexité et la diversité du domaine de l'urbanisme (qui couvre plusieurs secteurs avec divers acteurs) et étudié les approches existantes en matière de conception d'ontologies, nous décrivons dans ce chapitre-ci nos travaux sur la conception coopérative d'ontologies pré-consensuelles.

La conception coopérative d'ontologies pré-consensuelles nécessite :

- Une méthodologie et une démarche qui se situent dès les premières phases de développement d'une ontologie, permettant à des experts du domaine (dans notre cas des urbanistes) de modéliser les différences de points de vue sur le sens des termes. Notre méthodologie fait intervenir les experts du domaine dans la conception de l'ontologie en leur permettant d'aller vers le consensus.
- Un outil visuel permettant d'éditer, de naviguer dans l'ontologie et de l'interroger. Cet outil permet d'associer des ressources multimédia aux concepts.
- Un environnement pour le développement collaboratif d'ontologies proposant un support pour le groupware.

Les méthodes de construction d'ontologies supposent qu'un cogniticien ou ingénieur de la connaissance travaille sur la construction de l'ontologie en interrogeant des experts. Nous avons vu qu'il existe plusieurs méthodologies et systèmes de développement d'ontologies qui assistent les experts dans le cadre du travail collaboratif en fournissant un support groupware. Dans cette catégorie, nous pouvons citer entre autres : Ontolingua Server [FARQ96], APECKS [JENI98], Protégé-2000 [RAY02], etc. D'autres ne proposent que des approches sans support technique, dans cet ordre d'idée nous pouvons citer : CODE [CAÑA04], DELPHI [HOLS02], C-VISTA [DIEN02].

Dans notre cas, nous sommes dans le pré-consensus et l'outil Towntology, dans un premier temps, sert de support pour la construction de la conceptualisation qui



servira de base, dans un deuxième temps, pour la construction collaborative (vers le consensus) de notre ontologie.

### **8.1 Notre méthodologie : la méthodologie Towntology**

Notre méthodologie repose sur deux étapes importantes. La première consiste à mettre en place une ontologie (dans notre cas une ontologie pré-consensuelle), qui dans la deuxième étape servira d'ontologie d'ancrage pour le travail collaboratif ou groupware vers le consensus.

Notre modèle, inspiré de [FERN97], est construit pour s'intégrer au cycle de vie de construction et de maintenance des ontologies. Il a été complété pour satisfaire les exigences des approches collaboratives.

Le cycle de vie d'une ontologie identifie l'ensemble des phases à travers lesquelles l'ontologie évolue pendant sa durée de vie. Il décrit également quelles activités doivent être exécutées à chaque phase et comment les phases sont liées. La méthodologie Towntology appartient à la catégorie des prototypes évolutifs [KEND95]. Ce qui veut dire que dans notre approche l'ontologie se développe selon les besoins, et notre modèle permet de modifier, ajouter et supprimer des définitions dans l'ontologies à tout moment.

Ainsi notre processus de développement d'ontologie se rapporte aux activités exécutées pendant la construction des ontologies, complétées des activités groupware, comme indiqué dans la figure 8.1. Les activités encadrés en gras sont celles prises en compte par notre méthodologie à l'état actuel.

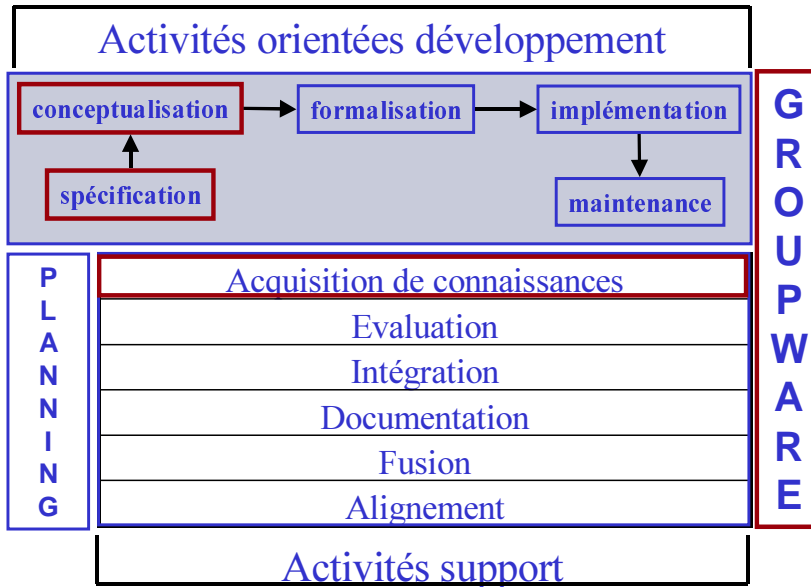


FIGURE 8.1 CYCLE DE VIE DE CONSTRUCTION DES ONTOLOGIES, COMPLETE DE L'APPROCHE COLLABORATIVE

Nous distinguons quatre catégories d'activités qui sont interdépendantes entre elles, c'est à dire qu'une catégorie peut inclure une série d'activités exécutées en même temps que les activités d'une autre catégorie. Les activités ne se suivent pas obligatoirement dans un ordre précis.

### 8.1.1 Les activités de planning

Elles identifient les tâches à être exécutées, leur arrangement, et le temps et les ressources nécessaires pour leur accomplissement. Elles incluent la programmation, le contrôle et la garantie de la qualité. Ces tâches ont été planifiées au début dans le cadre du projet Towntology, suite à un commun accord entre les laboratoires LIRIS et EDU.

### 8.1.2 Les activités orientées développement

Avant de commencer le développement proprement dit (ou phase de pré-développement), une étude a été effectuée pour connaître l'environnement dans lequel l'ontologie sera utilisée, les applications dans lesquelles l'ontologie sera intégrée, etc. Aussi pendant cette phase, l'étude de faisabilité répond aux questions telles que : est-il possible ou est-il approprié de construire l'ontologie ?, etc. Dans cette phase préparatoire, nous définissons les critères de conception de l'ontologie, le domaine et la portée de l'ontologie, et déterminons un ensemble de normes pour évaluer l'ontologie. Les critères de conception ont pour objectif de préserver une certaine cohérence durant tout le processus de développement de l'ontologie. Dans notre cas l'ontologie est cohérente si elle ne contient pas :

- de concepts orphelins, de concepts sans définition, de concepts sans ressources, de relations non utilisées, et d'erreurs de circularité ;
- de concepts redondants et homonymes ;
- et de conflit sémantique entre définitions, etc.

Ces critères sont importants dans le processus de développement de l'ontologie et d'évaluation de son degré de succès. Le domaine et la portée de notre ontologie sont fixés par les objectifs du projet Towntology (voir section 2.4.1).

Une fois dans la phase de développement, l'activité de *spécification* indique pourquoi l'ontologie est construite, quelles sont les utilisations prévues et qui sont les utilisateurs (finaux). En fait, le résultat de cette activité est une description de l'ontologie (d'habitude en langage naturel) qui sera transformée en un modèle conceptuel par l'activité de conceptualisation. Notre objectif, dans le cadre du projet Towntology, est de mettre en place un cadre unifié de travail et de communication sans ambiguïtés pour les acteurs de l'urbanisme qui peuvent être des étudiants, enseignants, chercheurs et professionnels de l'urbanisme.

Notre spécification d'ontologie pré-consensuelle doit tenir compte :

- des différentes terminologies employées qui peuvent amener à des ambiguïtés,
- des différents points de vues sur les objets du domaine d'étude, qui aboutissent à des définitions différentes d'un même objet.

En conséquence, nous devons associer plusieurs définitions aux termes et à partir des définitions qui ne peuvent être considérées comme équivalentes sémantiquement, afin de construire les concepts.

À partir de toutes ces considérations, et pour pouvoir gérer le pré-consensus, nous devons de mettre en place un système qui intègre les besoins des acteurs avant le consensus.

Ainsi les principes de spécification de notre système sont les suivants :

- Définir des termes.
- Définir des points de vue.
- Définir des concepts en associant plusieurs définitions à un terme avec ou sans point de vue.
- Illustrer les définitions par des documents multimédia pour clarifier la définition ou faciliter sa mémorisation.
- Définir les types de relation
- Définir des relations entre concepts.
- Visualiser le réseau des concepts.
- Définir la source des définitions et leurs auteurs.
- Mémoriser le nom de l'auteur de la définition et la date.
- Proposer des fonctionnalités de recherche d'un concept ou d'un terme.
- Proposer des fonctionnalités de recherche, de navigation, de filtrage dans un réseau de relations.
- Proposer des fonctionnalités de contrôle de versions.
- Proposer des fonctionnalités de mise en évidence des incohérences.
- Proposer le support de discussion vers le consensus.
- Etc.

L'activité de *conceptualisation* structure la connaissance du domaine dans des modèles sémantiques [NEWE82]. Nous avons utilisé des vocabulaires existants pour identifier les concepts les plus importants du domaine [BEAU03] [BERT04]. Nous adoptons ainsi la stratégie middle-out pour identifier les concepts en partant du plus important vers le plus abstrait et le plus concret. Comme nous sommes dans le cadre de la construction d'une conceptualisation, le consensus est en cours de construction, donc plusieurs définitions peuvent être associées à un même concept. Notre premier travail a consisté à demander aux experts du domaine leur

conceptualisation pour pouvoir trouver les concepts importants du domaine. Ainsi dans notre approche un concept est un triplet (terme, point de vue, définitions). Les termes homonymes cachent en fait des concepts différents. Donc pour éviter les ambiguïtés, un concept est aussi associé à un point de vue, pour séparer les définitions incompatibles<sup>25</sup>. Les concepts sont liés entre eux par des relations, pour décrire le contexte du concept. Ainsi le réseau de termes construit au départ de l'acquisition des connaissances se raffine petit à petit en un réseau de concepts. De plus il faut mettre en évidence les conflits terminologiques entre différents experts qui peuvent intervenir lors de la conception. Comme dit précédemment, le terme rue ne représente pas le même concept dans le cadre d'un service de distribution de courrier comme La Poste et dans le cadre d'un service de maintenance de la Voirie.

Les méthodes de conception des systèmes d'information ont mis en évidence l'importance des diagrammes graphiques pour faciliter la communication entre les futurs utilisateurs du système et les concepteurs du même système. Ainsi UML [RUMB98] propose différents diagrammes (use case, diagramme de classes) afin de modéliser les données ou les opérations du système. En se basant sur ces méthodes de conception de systèmes d'information, nous avons besoin d'un formalisme non informatique basé sur des diagrammes graphiques compréhensibles par des néophytes en gestion de la connaissance.

Ainsi dans notre cas pour visualiser de manière graphique le sens des termes, sont introduites des relations sémantiques entre concepts. De plus ces relations permettent de formaliser les liens entre les définitions. D'autre part, des images ou d'autres types d'illustrations peuvent aussi servir à clarifier le sens des définitions et surtout à faciliter leur mémorisation.

L'activité de *formalisation* transforme le modèle conceptuel en un modèle formel ou semi-automatique. Comme nous sommes dans le pré-consensus, cette activité n'est pas encore prise en compte dans notre méthodologie. Notre ontologie – dans l'étape actuelle – est juste un réseau sémantique de concepts.

---

<sup>25</sup> Dans la suite de cette thèse, nous appellerons concept un élément de la conceptualisation en cours, qui suivant son état pourra être un concept bien défini, c'est à dire un terme, un point de vue si nécessaire et des définitions sémantiquement équivalentes, ou un concept en cours d'identification, c'est à dire un terme avec un ensemble de définitions qui peuvent être incompatibles .

L'activité d'*implémentation* construit des modèles automatiques dans un langage d'ontologie. Nous avons choisi le langage XML [BRAY00] pour implémenter notre ontologie. Ce choix est expliqué et argumenté dans la section 8.3 qui décrit le langage Towntology.

Après le développement (ou phase de post-développement), l'activité de *maintenance* met à jour et corrige les ontologies si nécessaire par la mise en place des informations de traçabilité, et d'une documentation claire. Egalement pendant le post-développement, l'ontologie est (*re*)utilisée par d'autres ontologies ou applications.

### 8.1.3 Les activités support

Elles incluent une série d'activités exécutées en même temps que les activités orientées développement et les activités groupware, sans lesquelles l'ontologie ne pourrait être construite. Ces activités concernent : l'acquisition de la connaissance, l'évaluation, l'intégration, la fusion, l'alignement, la documentation et la gestion de version.

Le but de l'*acquisition de connaissance* est d'acquérir la connaissance venant des experts d'un domaine donné (dans notre cas les urbanistes) ou à travers un processus (semi) automatique, appelé étude d'ontologie (ou "ontology learning") [KIERT00]. Les méthodologies existantes utilisent différentes approches d'acquisition des connaissances. Par exemple la méthode On-To-Knowledge a étudié l'acquisition des connaissances en spécialisant une ontologie générique [ALEX00a]. L'équipe d'Aussenac-Gilles [AUSS00a] propose d'extraire des documents techniques une terminologie pour débiter la conceptualisation. Utiliser les ressources textuelles conduit d'abord à normaliser le vocabulaire de l'ontologie. Le problème est alors de transformer cette définition textuelle en un ensemble de concepts et de relations par l'induction ou la synthèse. L'équipe de Kogut [KOGU02] a commencé à partir du modèle conceptuel de base de données typiquement spécifié avec le modèle Entité-Relation (E-R) ou UML. Dans ce cas-ci les modèles E-R de différentes bases de données sont intégrés dans une

ontologie unique. Quand on utilise un schéma de base de données, on manipule des objets avec la granularité très fine comme dans l'approche inductive.

Notre approche d'acquisition de connaissance est proche de celle préconisée par l'équipe d'Aussenac-Gilles, sauf que contrairement à eux nous n'avons pas forcément un corpus de documents électroniques. La base terminologique que nous utilisons peut être construite à partir d'interview d'experts, d'analyse manuelle de documents ou d'ontologies.

Toutes les techniques d'acquisitions des connaissances ont besoin d'un ingénieur de la connaissance pour pouvoir débiter l'acquisition des connaissances. Une fois les connaissances identifiées, l'ingénieur les transforme en concept pour construire la modélisation du domaine. Dans notre approche nous cherchons à faire intervenir nos experts du domaine le plus longtemps possible dans la phase d'acquisition de la connaissance pour qu'ils participent et comprennent les choix liés à la conceptualisation du domaine. En effet, ce seront au final les experts du domaine qui vont valider et utiliser l'ontologie réalisée.

L'activité d'*évaluation* [GOME95] porte un jugement technique et humain sur les ontologies, sur leurs environnements logiciels associés, et sur la documentation. Ce jugement est fait dans le respect d'un cadre de référence pendant chaque phase et entre les phases du cycle de vie de l'ontologie. Dans notre cas, nous appliquons à chaque étape une évaluation selon les critères de conception (cohérence) que nous nous sommes fixés.

L'activité d'*intégration* est requise dans la construction d'une nouvelle ontologie par la réutilisation d'autres ontologies déjà existantes. Dans notre cas, dans la première phase de construction de notre ontologie d'ancrage, nous n'utilisons pas d'ontologie existante, nous partons de zéro pour construire notre ontologie. Mais cette activité peut être exécutée dans la phase groupware pour l'intégration d'ontologie dans le réceptacle d'ontologies.

Une autre activité support est la *fusion* [GANG99, NOY01, STUM01], qui consiste à obtenir une nouvelle ontologie en partant de plusieurs ontologies dans le même domaine. L'ontologie résultante peut unifier (rassembler) les concepts, terminologies, définitions, contraintes, etc de toutes les ontologies sources

[PINT99]). Dans notre méthodologie la fusion de deux ontologies ou plus est effectuée lors de l'insertion d'une ontologie dans le réceptacle des ontologies.

L'activité d'*alignement* établit différentes sortes de correspondance (ou liens) entre les ontologies impliquées. Par conséquent cette option préserve les ontologies originales et ne les fusionnent pas. Dans notre approche, cette activité est réalisée dans la phase groupware à travers des concepts passerelles existant dans différentes ontologies.

L'activité de *documentation* détaille, clairement et exhaustivement, chacune des étapes réalisées et les résultats générés.

L'activité de *gestion de version* enregistre toutes les versions de la documentation et du code de l'ontologie pour contrôler les changements et produire les traçabilités. La version d'une ontologie dans notre environnement correspond à sa dernière date de modification. La modification peut concerner les méta-informations de l'ontologie (par exemple : le custodian, le langage, ...) ou les éléments constituant l'ontologie (par exemple : les concepts, types de relations, définitions, ...)

#### **8.1.4 Les activités Groupware**

Notre proposition de cycle de vie de l'ontologie est complétée par le groupware. Ainsi notre méthodologie propose une approche collaborative inspirée de la méthode Delphi [HOLS02], comme montré sur la figure 8.2. Les activités collaboratives incluent trois phases, celle du choix de l'ontologie d'ancrage, celle de la construction itérative de l'ontologie et celle de l'application (utilisation) de l'ontologie résultante. L'ontologie d'ancrage est le réseau sémantique (incluant des points de vue et des définitions multiples) construit dans un premier temps avec l'outil Towntology. Ce réseau sémantique servira de point de départ pour le développement collaboratif dans un deuxième temps.

Une approche collaborative pour la conception et le développement d'ontologie, est un travail (effort) commun reflétant les expériences et les points de vue des personnes qui coopèrent intentionnellement à le produire. La diversité des



personnes et de leurs domaines d'expertise est un atout et permet d'enrichir le contenu de l'ontologie. D'autre part, la coordination du processus de conception et d'analyse peut souffrir si trop de personnes sont directement impliquées. Le processus lui-même s'applique à une ontologie proposée comme point de départ pour des améliorations itératives. Afin d'exécuter une approche de collaboration, un mécanisme de construction du consensus est utilisé.

La figure 8.2 illustre les différentes activités de notre approche collaborative.

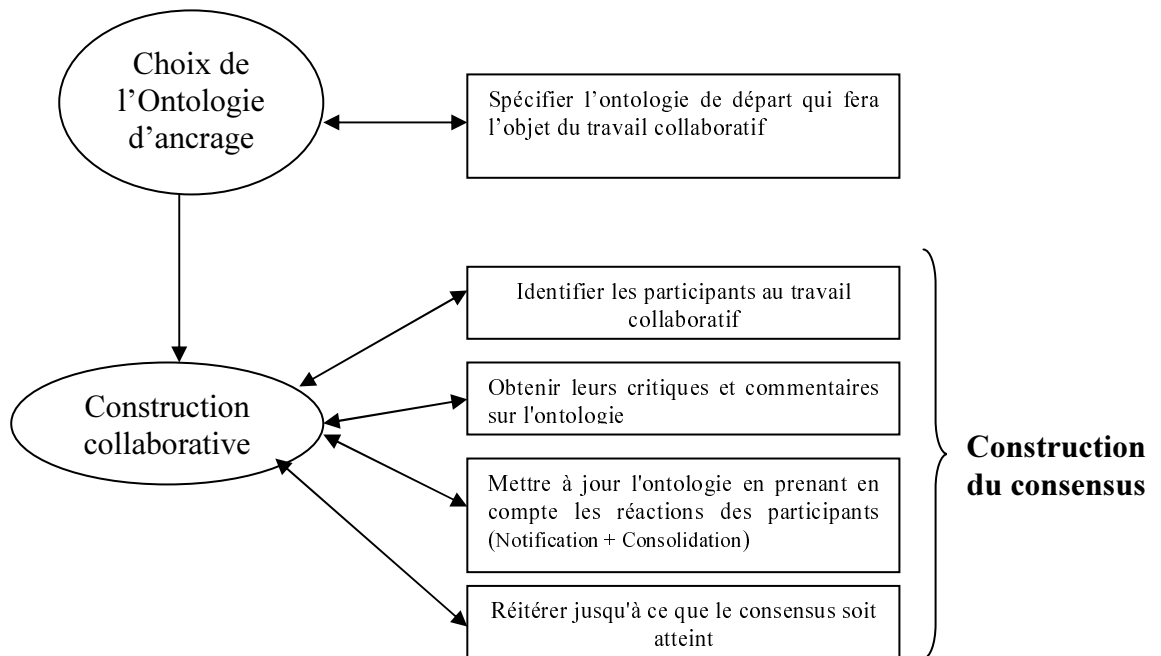


FIGURE 8.2 APPROCHE COLLABORATIVE POUR LA CONCEPTION D'ONTOLOGIE.

La phase du choix de l'ontologie d'ancrage permet de spécifier l'ontologie sur laquelle les participants concentreront leur collaboration. Il peut s'agir d'une ontologie construite avec Townto-Editor ou d'une ontologie soumise dans le format compréhensible par notre système. Cette ontologie servira donc de point d'ancrage pour le développement collaboratif. Quoique notre but soit de répondre à des critères de conception (définis dans l'étape du pré-développement), l'ontologie d'ancrage peut ne pas les satisfaire entièrement, du moins avant le consensus.

L'activité de construction itérative est une technique pour recueillir et intégrer les points de vue de multiples personnes sur un sujet. Elle commence par l'identification des participants au travail collaboratif, et ensuite par la mise à la disposition de ces participants d'un support de construction du consensus (voir la section 8.4.3 sur le mécanisme de construction du consensus). Dans notre cas, les participants sont un mélange équilibré de spécialistes et d'experts en urbanisme, tous sont expérimentés dans le domaine et représentent des points de vue, des expériences, et des milieux divers. Le but de cette phase est aussi de trouver où l'ontologie a des incohérences, les mettre à jour pour surmonter ces imperfections (incohérences), et obtenir des évaluations indépendantes de l'ontologie révisée. Notre système intègre des modules de contrôle de cohérence des ontologies dont il sera question dans la section 8.4.

Dans l'étape de l'activité d'application, la méthodologie démontre l'utilité et les utilisations de l'ontologie résultante comme cadre unifié de travail et de communication sans ambiguïtés et regroupant les critères de conception. Le serveur Townto-Groupware fournit des outils permettant le partage des ontologies. Le mécanisme primaire pour supporter la réutilisation des d'ontologies se fait à travers la bibliothèque d'ontologies qui agit en tant que réceptacle (dépôt) central pour des ontologies réutilisables. Quand on a une ontologie qu'on croie être prête pour l'utilisation par des autres utilisateurs, on pourrait l'éditer sur le serveur. Après que l'ontologie ait été approuvée après avoir passée les contrôles de cohérence, elle est placée dans la bibliothèque sur le serveur et devient accessible pour les autres utilisateurs.

Concernant le support technique, la suite d'outils Towntology (Visual Townto-Browser, Townto-Editor et Townto-Groupware) décrits dans la section 8.7 donne le support complet pour la méthodologie Towntology. Ce support fournit un système de gestion et de stockage des contenus sur un serveur Plone<sup>26</sup>, un système de communication synchrone et asynchrone, et un système d'organisation du travail basé sur les systèmes workflow et les applications collaboratives.

---

<sup>26</sup> <http://www.plone.org>

Les tableaux ci-dessous comparent la méthodologie Towntology avec les deux méthodologies les plus utilisées en ingénierie de connaissances (On-To-Knowledge et METHONTOLOGY) à travers leurs processus de développement et leurs stratégies de construction d'ontologie. Chaque cellule du tableau peut être remplie de trois types de valeurs. La valeur "++" signifie que Towntology décrit comment exécuter chaque tâche dans l'activité proposée, quand le faire, qui doit le faire, etc. La valeur "+/-" signifie que la méthodologie identifie juste le processus et qu'avec un travail additionnel l'activité sera prise en compte. La valeur "-" signifie que la méthodologie ne prend pas en compte (du moins à l'état actuel) l'activité considérée.

**TABLEAU 8.1 PROCESSUS DE DEVELOPPEMENT DE TOWNTOLOGY EN COMPARAISON AVEC D'AUTRES METHODOLOGIES EN IC**

	Plan- ning	Activités orientées Développement					Activités orientées Support							Groupware
		Spécifi- cation	Conceptua- lisation	Formali- sation	Implémen- tation	Mainte- nance	Réutili- sation	Acquisition de connaissance	Evalua- tion	Intégra- tion	Contrôle de versions	Docu- mentation	Fusion et Alignement	
<b>TOWNTOLOGY</b>	++	++	++	-	++	++	+/-	++	+/-	+/-	++	+/-	++	++
<b>On-To-Knowledge</b>	+	++	+	++	+	+	+	++	+	+	+	+	-	-
<b>METHONTOLOGY</b>	-	++	++	++	++	+	-	++	++	+	++	++	-	-

**TABLEAU 8.2 STRATEGIES DE CONSTRUCTION D'ONTOLOGIE DE TOWNTOLOGY EN COMPARAISON AVEC D'AUTRES METHODOLOGIES EN IC**

	<b>Proposition de cycle de vie</b>	<b>Stratégie par rapport à l'application</b>	<b>Approche d'identification des concepts</b>	<b>Utilisation d'ontologie noyau</b>	<b>Outil support</b>
<b>TOWN-TOLOGY</b>	Prototype évolutif	Indépendante de l'application	Middle-out	Non	Outils Towntology
<b>On-To-Knowledge</b>	Incrémentale et cyclique avec prototypes en évolution	Dépendante de l'application	Top-down, Bottom-up, Middle-out	Oui (dépend des ressources disponibles)	OntoEdit avec ses plugins
<b>METHON-TOLOGY</b>	Prototypes en évolution	Indépendante de l'application	Middle-out	Oui (dépend des ressources disponibles)	ODE, WebODE, OntoEdit, Protégé-2000

Pour donner un support à notre méthodologie, il a fallu mettre en place un système qui gère le pré-consensus et permet d'aller vers le consensus par la collaboration. La section suivante traite du modèle de notre ontologie et des spécifications de notre système.

## **8.2 Le modèle Towntology**

Les principes-clés de notre processus de conception collaborative sont d'assurer la communication, la coordination et la coopération entre nos experts. Pour pouvoir gérer les conflits terminologiques qui peuvent intervenir entre différents experts lors de la conception, faciliter le suivi des modifications et de l'évolution des conflits, et enfin aboutir à un consensus, nous proposons les principales spécifications de notre modèle d'ontologie, qui sont les suivantes :

- ***Multiplicité de définitions des concepts.*** Comme chaque expert peut avoir pour un concept donné sa propre définition, nous devons garder toutes ces définitions ; à la fin du processus collaboratif vers le consensus, il y aura plusieurs possibilités. La première des choses serait de synthétiser (combiner) toutes les définitions précédentes dans une définition unique ; cependant notre projection est que cette situation ne sera pas très courante. Une deuxième possibilité serait d'éclater le concept précédent en plusieurs nouveaux concepts, chacun d'eux ayant sa propre définition. L'inconvénient principal sera la multiplication des concepts générant des difficultés spécialement pour nommer ("labelliser") ces concepts. En revenant à l'exemple de la rue (illustré dans le tableau 8.3 suivant), nous ne pensons pas que cela soit acceptable, par rapport aux différents concepts de la rue, pour des ingénieurs de trafic routier, ingénieurs de maintenance de voirie, commerciaux ou urbanistes. Nous admettons qu'un concept puisse être associé à plusieurs termes et que les termes synonymes réfèrent le même concept. À la fin du processus de conception chaque "label" de concept devrait être différent afin d'éviter l'ambiguïté et de créer un vocabulaire normalisé.

TABLEAU 8.3 EXEMPLE DE MULTIPLICITE DE DEFINITIONS D'UN CONCEPT (LE CONCEPT RUE)

Concept	Termes associés	Points de vue	Définitions	Label
Rue	Support	Ingénieur du trafic routier	Rue comme support de la circulation	Rue (Transport)
	Matériaux	Ingénieurs de maintenance de la voirie	Rue comme Matériaux et réseaux souterrains	Rue (Voirie)
	Lieu	Responsable chambre de commerce	Rue comme lieu abritant les commerces	Rue (Commerce)
	Infrastructure	Urbaniste	Rue comme infrastructure de développement économique et social	Rue (Urbanisme)

- Définition de points de vue.** Alors qu'un domaine correspond à une unité thématique, un point de vue est plutôt lié à un type de personne (métier, âge, niveau de formation, etc) ou à un type d'utilisation (une même personne peut avoir un point de vue différent en fonction de la tâche qu'elle cherche à accomplir). Nous admettons dans notre cas qu'un point de vue est un ensemble de définitions, chacune d'entre elles étant liée à un concept différent. Plusieurs acteurs peuvent avoir des analyses et compréhensions divergentes d'un même objet selon leur spécialité ou leur manière d'aborder une tâche. Ainsi notre ontologie doit prendre en compte les différents points de vue de différents acteurs. Selon [DIEN02], un point de vue se compose de vision et d'angle de vue de l'acteur. La vision décrit le contexte du travail de l'acteur. L'angle de vue décrit les caractéristiques d'un acteur ou groupe d'acteurs. Par

exemple, un angle de vue contient le domaine d'application, le niveau d'expertise ou la compétence, le rôle de l'acteur dans l'organisation.

- ***Information hypertexte et multimédia.*** Pour illustrer les concepts, les photos ou les schémas techniques sont souvent plus compréhensibles. Ces informations aideront les experts à comprendre leurs différents points de vue, et atteindre le consensus. La notion du son ou du bruit a été étudiée, mais cette notion a été vite abandonnée, parce que nous avons trouvé qu'il était difficile sinon impossible d'illustrer un concept par le son ou le bruit qu'il provoque. Puisque souvent des définitions des concepts sont liées, quelques liens hypertexte additionnels doivent être utiles.
- ***Traçabilité des définitions et des modifications.*** Afin d'éclaircir l'origine des concepts et des définitions et surtout d'entretenir le suivi des modifications (par qui et quand), nous avons décidé de mettre l'information de traçabilité dans notre modèle. Une possibilité devrait être de définir un métalangage associé avec ces notions. Mais nous avons plutôt décidé de définir un langage unique prenant en charge toutes les méta-données de traçabilité. Ainsi, à chaque définition, trois informations sont ajoutées, la date d'insertion/modification, l'auteur de la définition, et le créateur, c'est-à-dire le nom de la personne qui a inséré la définition dans le système.
- ***Minimisation des types de relation.*** Entre deux concepts, plusieurs types de relations peuvent être définis. Il a été décidé de les réduire, pour avoir un nombre limité. En d'autres termes, quiconque peut insérer de nouveaux types de relation, mais cette personne doit argumenter au sujet de la nécessité d'intégrer ce nouveau type de relation.
- ***Travail sur des ontologies de domaine au lieu d'une ontologie (générale) unique.*** Quand les équipes séparées travaillent à définir ou à concevoir des concepts, il semble plus facile de travailler avec différentes ontologies plus petites. Concernant l'urbanisme, nous pouvons distinguer une ontologie pour la voirie, une pour la mobilité et transports, une pour des risques urbains, une



pour des cadastres, etc. Une ontologie devrait se concentrer sur un domaine spécifique d'intérêt (par exemple la voirie). Ainsi, si nécessaire, une définition ou une relation peut évoquer un concept situé dans une autre ontologie. La conséquence de ce choix sera la nécessité de fournir l'outil pour importer le concept déjà défini dans une nouvelle ontologie.

Ainsi nous montrons ci-dessous (figure 8.3) notre modèle d'ontologie pré-consensuelle. Notre ontologie décrit les concepts d'un domaine d'intérêt spécifique ou d'application. Dans ce domaine, un ID unique identifie un concept. Un concept peut être caractérisé par plusieurs termes, plusieurs définitions et plusieurs ressources multimédia. À la fin du processus de conception d'ontologie, un concept devrait être défini par une définition unique et seulement un terme devrait être "le label" du concept. Des relations binaires lient les concepts et ces relations sont typées. Des photos contiennent les zones clickables, qui sont associées à des concepts.

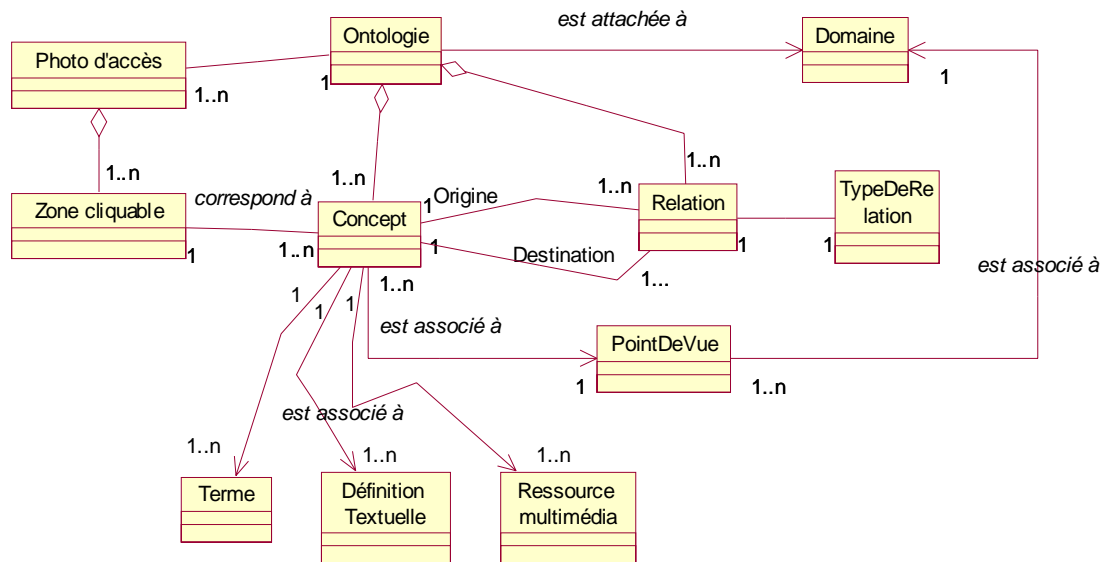


FIGURE 8.3 MODELE UML DE L'ONTOLOGIE PRE-CONSENSUELLE

Ci-dessous la figure 8.4 montre un extrait de notre ontologie sous forme de graphe sémantique. Il contient en même temps des concepts de la voirie et de la mobilité et transports. Pour expliquer la sémantique des concepts nous avons besoin de relations. Ainsi au début, 21 types de relation entre les concepts avaient été déterminés (voir le chapitre 6 sur les expérimentations). Ces types de relation sont utilisés pour établir plusieurs centaines de relations entre nos concepts. Pour spécifier ces relations, nous les avons représentées visuellement dans un réseau sémantique en indiquant le nom de la relation qui lie les deux concepts. Ce réseau sera continuellement enrichi avec de nouvelles relations entre les concepts. Une relation a une direction de lecture : par exemple la relation "est situé sur" entre les concepts "Signalisation horizontale" et "Trottoir" est lue comme : « la signalisation horizontale "est situé sur" le trottoir ».

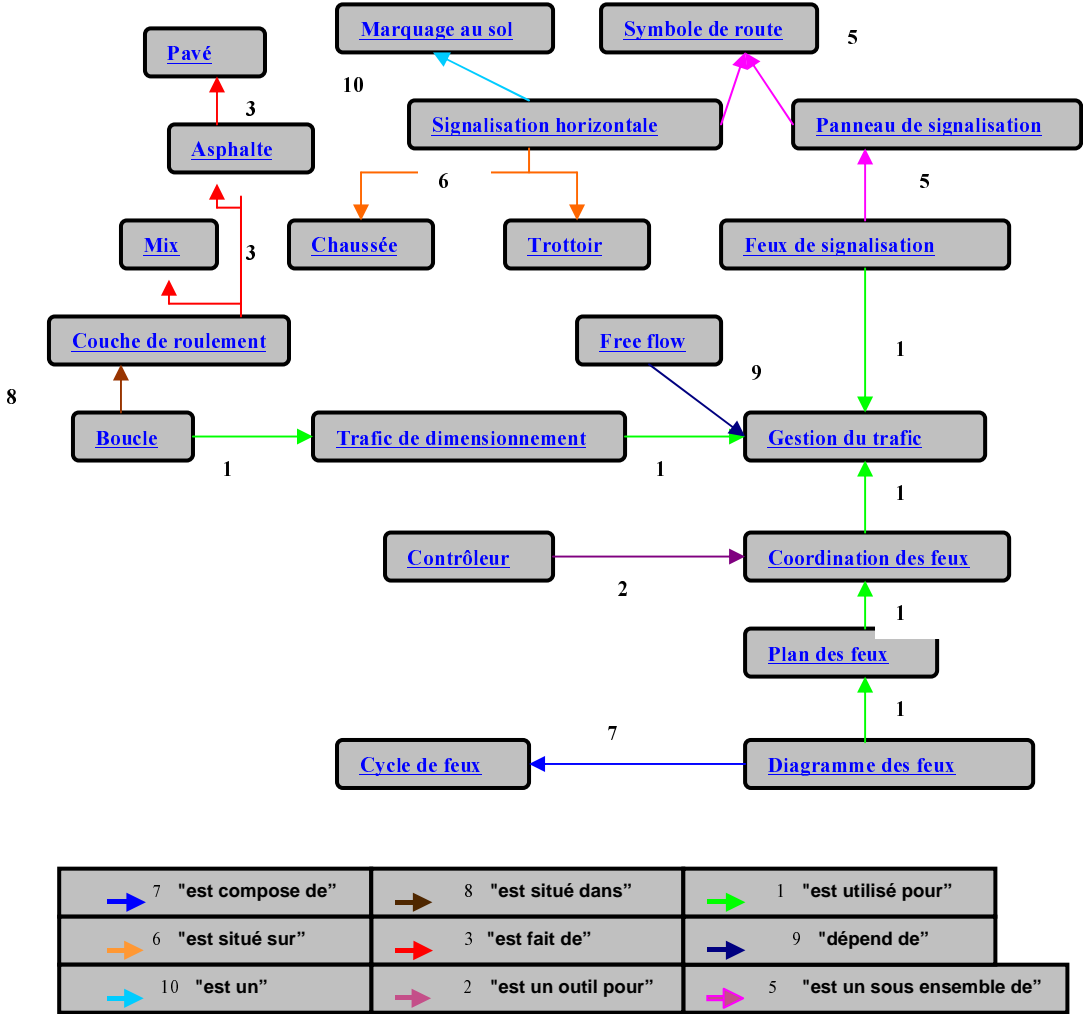


FIGURE 8.4 UN EXTRAIT DE NOTRE ONTOLOGIE [BEAU03]

Après avoir spécifié les besoins de notre modèle d'ontologie, il a fallu penser à trouver un langage pour l'implémenter. La section suivante aborde cette question.

### **8.3 Le langage Towntology**

Dans la section état de l'art, nous avons vu que les langages d'ontologie existants n'ont ni la même expressivité, ni le même type de raisonnement. Les paradigmes de représentation de connaissance utilisés par ces langages d'implémentation d'ontologie sont divers : les frames, la logique de description, la logique de premier (et second) ordre, les réseaux sémantiques, etc. Ce fait rend bien plus important le choix correct du langage dans lequel l'ontologie doit être implémentée. En fait, un paradigme peut s'avérer très approprié pour une tâche spécifique mais pas pour d'autres effectuées dans une application.

Par conséquent, avant le codage de l'ontologie nous devons savoir ce dont nous avons besoin en termes d'expressivité et de raisonnement, et quels langages satisfont à de telles exigences.

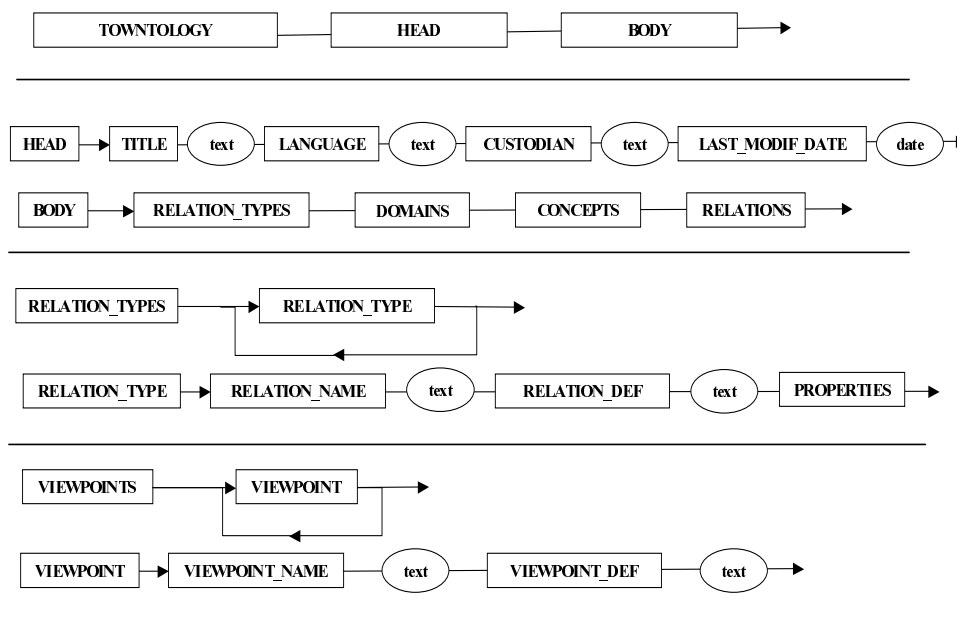
En s'inspirant de [GOME04] et en apportant des réponses plus ou moins claires aux questions sur le choix d'un langage approprié, nous avons comparé et analysé les langages d'ontologie existants selon les besoins de notre spécification [KEIT04].

En effet nous avons dû abandonner l'idée d'utiliser un des langages de description d'ontologie déjà existant car aucun ne répondait à l'intégralité de nos besoins, de nos spécifications. Puisque nous n'avons pas d'ontologie existante, prête à l'utilisation, et à ce stade nous sommes plutôt intéressés par le vocabulaire que par la syntaxe. Ainsi nous sommes dans un processus de construction et de réorganisation de notre ontologie. À cette étape courante nous ne cherchons pas à construire une ontologie formelle. Nous utilisons un modèle simple pour permettre à des non-informaticiens de valider les concepts et d'autres éléments de l'ontologie. Nous ne construisons pas un langage de représentation des ontologies (notre ontologie n'est pas totalement complète, elle est en cours de construction). Par exemple, le OWL est un langage de représentation d'une ontologie déjà construite, complète. Dans notre cas il y a plusieurs définitions pour un concept et

nous sommes dans un processus de choix et de caractérisation de ces définitions... Représentent-elles le même concept, un concept différent etc...? Les défis de la cohérence et de la stabilité sont à traiter plus tard.

Pour toutes ces raisons évoquées, nous avons choisi le standard XML [BRAY00] pour stocker notre ontologie, mais également parce que ce langage nous offre les moyens d'atteindre les objectifs assignés, c'est-à-dire la représentation visuelle de l'ontologie (qui est un chemin vers le système coopératif) et, à terme, l'interopérabilité. XML permet également une utilisation conviviale et adaptée avec le langage de programmation JAVA, que nous utilisons pour notre prototype.

Comme montré dans la figure 8.5, notre spécification XML décrivant le contenu de l'ontologie est divisée en deux parties : head et body. La partie head (en-tête de l'ontologie) contient des informations générales sur l'ontologie. La partie body (corps de l'ontologie) contient les éléments composant l'ontologie comme les types de relation, les définitions de points de vue, les définitions de concept et la liste de relations entre les concepts.



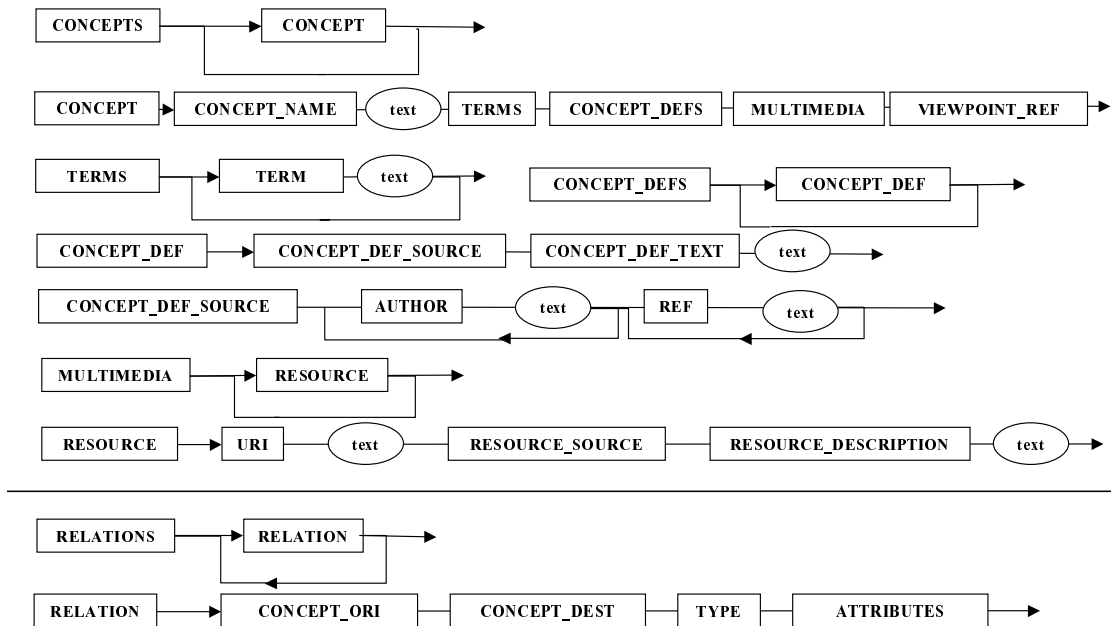


FIGURE 8.5 SPECIFICATION XML DU LANGAGE TOWNTOLGY

La partie Head (en-tête) décrit les metadonnées sur l'ontologie comme :

- Le titre de l'ontologie, c'est à dire le domaine de l'ontologie (exemple : voirie ou transport)
- Le langage dans lequel l'ontologie est écrite (exemple : français)
- Le custodian, c'est à dire l'institution propriétaire de l'ontologie ou son représentant (exemple : LIRIS, EDU ou autres laboratoires)
- La date de dernière modification, c'est à dire la date de création de l'ontologie ou de dernière modification (exemple : 2005/06/25)

Ainsi la structure générale des noms (système) des ontologies est la suivante :

“Townto\_Custodian\_Language\_Title\_LastModifDate”

La structure de "LastModifDate" est : AAAAMMJJ.

Par exemple on pourrait avoir comme nom d'ontologie :

Townto\_Liris\_Francais\_Voirie\_20040728

La partie Body ou corps de l'ontologie décrit les éléments composants de l'ontologie :

- Les types de relation
- Les concepts auxquels sont associés les termes, points de vue, définitions et ressources multimédia.
- Les relations entre deux concepts, avec le type de relation auquel appartient cette relation.

Plus concrètement notre ontologie est un réseau sémantique de concepts reliés par des relations.

Un type de relation est caractérisé par un nom, un créateur (la personne qui l'a insérée), une date d'insertion, et éventuellement par ses propriétés.

Un concept possède un nom (label), de(s) terme(s) clés qui le caractérisent, un point de vue, de(s) définition(s) textuelles et une ressource multimédia qui peut être une photo ou un schéma. Une définition de concept a une source (auteurs, références), un créateur (qui l'a inséré), une date d'insertion, et un contenu textuel. La ressource multimédia possède un Url, une source (auteur(s) et référence) et une description.

Une relation est caractérisée par les deux concepts différents qu'elle relie (concept d'origine et concept de destination), par le type auquel elle appartient, et par son créateur (qui l'a insérée). Une relation peut aussi avoir des attributs (optionnel ou pas).

A titre d'exemple nous présentons un extrait du format XML de notre ontologie "Transports". Le format complet est consultable dans les annexes.

```
<Towntology>
<HEAD>
  <TITLE>Transports</TITLE>
  <LANGUAGE>Français</LANGUAGE>
  <CUSTODIAN>LIRIS</CUSTODIAN>
  <LAST_MODIF_DATE>040625</LAST_MODIF_DATE>
</HEAD>

<BODY>
  <RELATION_TYPES>
  <RELATION_TYPE ID="10001" ORIGINATOR="Christophe BERTHET"
  INSERTION_DATE="040621">
  <RELATION_NAME>dépend de</RELATION_NAME>
  <TERMS />
```

```

<RELATION_DEF />
<RELATION_PROPERTIES SYMMETRIC="false" TRANSITIVE="false"
MAYBEOPTIONAL="false" />
</RELATION_TYPE>
.....
</RELATION_TYPES>
.....
</BODY>

```

Et un extrait de la description XML d'un concept en l'occurrence le concept "Accident de la

route " :

```

<CONCEPT_NAME>Accident de la route </CONCEPT_NAME>
<TERMS />
<CONCEPT_DOMAIN ID="200001" />
<CONCEPT_DEFS>
  <CONCEPT_DEF ORIGINATOR="Christophe BERTHET" INSERTION_DATE="2004/06/21">
    <CONCEPT_DEF_SOURCE>
      <AUTHORS />
      <REF>Glossaires – Promotion Of Results in Transport Research and Learning</REF>
    </CONCEPT_DEF_SOURCE>
    <CONCEPT_DEF_TEXT>Définition utilisée pour les statistiques dans la plupart des pays : il
      s'agit d'une collision ayant lieu sur la voie publique et qui implique au moins un véhicule
      roulant. Sont considérés comme accidents de la route les accidents provoquant
      uniquement des dégâts matériels et les accidents occasionnant des
      blessures.</CONCEPT_DEF_TEXT>
    </CONCEPT_DEF>
  </CONCEPT_DEFS>
  <MULTIMEDIA />
</CONCEPT>

```

Les tableaux suivants comparent le langage Towntology avec d'autres langages d'ontologie basés sur XML à travers leurs capacités et les spécifications de Towntology. Les cellules du tableau utilise le signe '+' pour indiquer qu'il est un dispositif supporté par le langage, le '-' pour les dispositifs non supportés, et le '+/-' pour des dispositifs non supportés, mais qui pourraient l'être par des extensions (par un travail additionnel).

TABLEAU 8.3 LES DISPOSITIFS SUPPORTES PAR LE LANGAGE TOWNTOLOGY

	CONCEPTS	TAXONOMIES DE CONCEPTS	RELATIONS		FONCTIONS	AUTRES COMPOSANTS
			binaires	n-aires		
<b>TOWNTOLOGY</b>	+	+(simple)	+	+	-	-
<b>RDF(S)</b>	+	+(simple)	+	+/-	-	-(sauf instances)
<b>DAML+OIL</b>	+	+	+	+/-	+(binaires)	-(sauf instances)
<b>OWL</b>	+	+	+	+/-	+(binaires)	-(sauf instances)

TABLEAU 8.4 COMPARAISON DU LANGAGE TOWNTOLOGY AVEC D'AUTRES LANGAGES

	RDF(S)	OIL	DAML+OIL	OWL	XOL	TOWNTOLOGY
<b>Basé sur XML</b>	+	+	+	+	+	+
<b>Définitions Hypertextuelles</b>	+	+	+	+	-	+
<b>Possibilité de plusieurs définitions</b>	-	-	+/-	+	-	+
<b>Attributs Multimédia</b>	+	-	+	+	-	+
<b>Inclusion de méta-donnée (pour décrire le contexte de l'utilisation)</b>	+/-	+	+	+	-	+
<b>Traçabilité de la modification</b>	-	-	-	-	-	+



Après avoir identifié les besoins de notre système et effectué le choix du langage dans lequel notre ontologie sera implémentée, nous présentons dans la section suivante les fonctionnalités du système pour la conception collaborative d'ontologies

## **8.4 Les fonctionnalités du système pour la conception collaborative d'ontologies**

Dans cette section nous présentons les fonctionnalités nécessaires pour le développement collaboratif d'ontologies. Elles sont basées sur les fonctions suivantes :

- Navigateur visuel d'ontologie,
- Editeur visuel d'ontologie,
- Mécanisme de construction du consensus,
- Analyse de cohérence locale, et
- Analyse de cohérence globale.

### **8.4.1 Navigateur visuel d'ontologie**

Cette application affiche l'ontologie sous forme de graphe. Un algorithme sur la représentation de l'ontologie sous forme de graphe et sur la présentation des concepts (première et deuxième couronne) est donné dans le chapitre 10 sur l'implémentation. Cette application permet à l'utilisateur de naviguer dans l'ontologie et de rechercher facilement un concept. Il existe plusieurs possibilités d'accès à l'ontologie. D'abord un utilisateur peut trouver un concept par son label (son nom). Ainsi une liste alphabétique de termes peut être affichée afin de sélectionner un concept (voir figure 8.11). En second lieu, un utilisateur peut rechercher un concept par sa représentation visuelle (son image, sa photo). Aussi, des photos incluant des zones clickables peuvent être utilisées comme point d'entrée vers l'ontologie. Chaque zone cliquable est reliée à un concept (voir la figure 8.13). Et enfin, un utilisateur peut rechercher un concept à travers sa relation avec d'autres concepts. Ainsi, le réseau de relations peut être affiché permettant de naviguer entre les concepts.

### 8.4.2 Editeur visuel d'ontologie : Towntology Editor

Cette application contient toutes les interfaces graphiques utilisateurs (GUI) permettant d'éditer l'ontologie. Elle offre le support à l'insertion et à la modification des concepts et des relations, à celles de leurs définitions, et également à l'insertion des informations de traçabilité. Comme précédemment dit, les photos sont un bon chemin d'entrée dans les ontologies. Un module est nécessaire pour insérer de nouvelles photos, pour définir des zones clickables, et pour assigner à ces zones des concepts dans l'ontologie concernée.

### 8.4.3 Mécanisme de construction du consensus

Cette application est une extension de l'éditeur afin de structurer la collaboration entre plusieurs concepteurs d'ontologies en se dirigeant vers le consensus. Rappelons que dans notre approche, nous considérons qu'il y a absence de contradiction, si pour chaque concept il existe une seule définition par point de vue.

La construction du consensus est basée sur la coordination des tâches effectuées par les différents acteurs. Dans notre environnement les acteurs possibles pour le travail collaboratif sont :

- **Le visiteur** : toute personne enregistrée dans la base d'utilisateurs du site, et qui peut consulter tout le contenu publié.
- **L'administrateur** : personne qui fait partie de l'équipe rédactionnelle du site ; il est autorisé à ajouter, modifier et supprimer du contenu dans une (ou plusieurs parties) du site à caractère public (hors de son dossier personnel ). Il relit et valide la publication de contenus qui lui sont soumis, ou les retourne aux auteurs pour correction avant d'accepter de les publier.
- **L'expert** : personne qui peut consulter, soumettre des modifications et les argumenter.
- **Le webmestre** : personne qui a en charge la maintenance du site, notamment la création des dossiers (rubriques) qui structurent le site et la gestion des comptes utilisateurs et de leurs habilitations. Son rôle peut être joué par l'administrateur.

TABLEAU 8.4 LES ACTEURS ET LEURS DROITS DANS L'ENVIRONNEMENT COLLABORATIF

Acteurs	Droits et permissions
Visiteur	Consulter
Administrateur	Ajouter, modifier, supprimer, valider, publier
Expert	Consulter, soumettre, modifier

Dans notre approche collaborative le processus de construction du consensus est basée sur les étapes suivantes :

1. Obtenir les critiques et commentaires des participants par la discussion sur l'ontologie d'ancrage.
2. Mettre à jour l'ontologie en prenant en compte les réactions des participants (Notification + Consolidation).
3. Répéter jusqu'à ce que le consensus soit atteint.

Notre modèle de discussion basé sur le système workflow est mis en application sur un serveur Plone. Le serveur Plone est configuré pour accepter le modèle d'ontologie de Towntology. La technique workflow donne un moyen systématique pour recueillir des suggestions et des critiques sur une ontologie comme une base pour le développement itératif. Dans notre système chaque groupe de travail dispose d'un espace commun à tous ses membres. Ceci permet à des experts de travailler ensemble sur les définitions d'un contenu (élément) de l'ontologie en essayant de se mettre d'accord (consensus), avant de décider d'en publier une version définitive. Ainsi nous faisons émerger un consensus par la discussion à la manière des Wikis.

Le *workflow* de publication mis en œuvre par le système de gestion de contenu définit les états par lesquels le contenu transite selon la politique d'édition et de collaboration, depuis sa création jusqu'à sa destruction.

Le *workflow* de publication de contenu est principalement composé des états suivants :

- *Privé (individuel)* : c'est l'état initial d'un contenu soumis.
- *En attente (discussion)* : c'est l'état intermédiaire par lequel doit passer le contenu devant être publié. Le contenu attend dans cet état jusqu'à faire objet de consensus avant d'être validé par la personne habilitée, et après d'être « publié ».
- *Publié (ontologie)* : c'est l'état dans lequel le contenu a fait l'objet d'un consensus entre les experts et est publiquement accessible, donc publié dans l'ontologie.
- *Rejeté* : c'est l'état dans lequel le contenu n'a pas fait l'objet d'un consensus et est retourné à l'état privé pour amélioration.

La figure ci-dessous décrit la chaîne de publication d'un contenu.

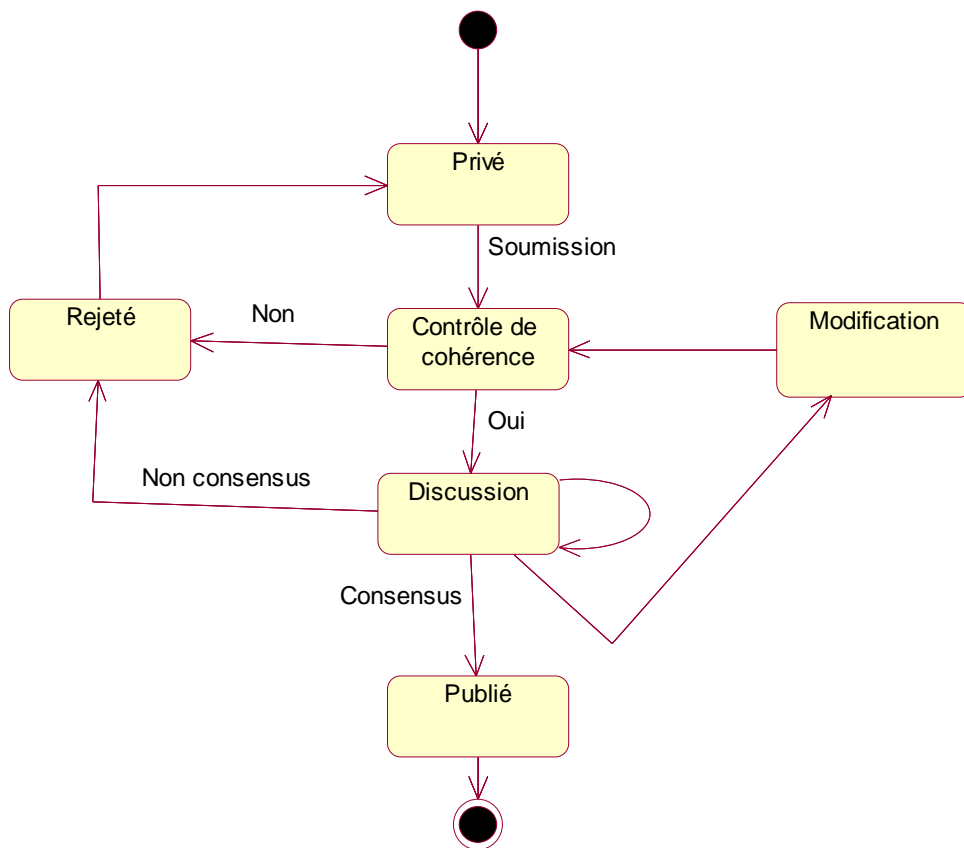


FIGURE 8.6 SCHEMA DE WORKFLOW POUR LES CONTENUS D'ONTOLOGIE.

La mise à jour de l'ontologie en prenant en compte les réactions des participants vient à la suite de la publication dans l'ontologie du nouveau contenu (ou de la nouvelle modification). Cette phase consiste à consolider l'ontologie suite à une modification, grâce au module de contrôle de cohérence interne. L'idée est que l'ontologie soit construite progressivement et aussi qu'elle devienne graduellement stable (et cohérente). Ensuite une notification est adressée à tous les acteurs. Elle consiste à informer les acteurs des modifications (changements) effectuées sur une ontologie tout en introduisant des hyperliens vers ces modifications.

Quand l'ontologie satisfait au contrôle de cohérence interne, elle est publiée (intégrée) dans le réceptacle stockant toutes les ontologies, et le contrôle de cohérence globale est lancé. Le contrôle de cohérence globale ou contrôle de cohérence intra-ontologies consiste à trouver des incohérences entre différentes ontologies du réceptacle et à les corriger, suite à l'intégration d'une nouvelle ontologie.

Si en dernier ressort le contrôle de cohérence globale ne relève aucune incohérence ou imperfection, alors nous pouvons considérer que le consensus est atteint et que nous disposons dans notre réceptacle d'ontologies consensuelles prêtes à l'utilisation.

#### **8.4.3.1 Exemple de workflow de création et d'édition d'un concept :**

*Acteur(s)* : Expert

*Préconditions* :

- L'expert est authentifié par le système.
- L'expert se trouve dans un dossier ou espace où il a les droits requis.

*Scénario* :

- L'expert accède au formulaire de création du concept dans l'éditeur d'ontologie.
- L'expert complète les champs du formulaire avec le contenu textuel et valide.
- Le concept est créé dans son état initial, l'état privé.
- L'expert peut continuer à modifier. En passant par les onglets d'extension prévus par le système, il peut lui associer des métadonnées, des photos, et/ou des fichiers bureautiques en pièce jointe.

### **8.4.3.2 Exemple de publication d'un concept**

*Acteur(s)* : Expert, Administrateur

*Préconditions* :

- Les acteurs sont authentifiés par le système.
- Le concept se trouve dans l'état « privé ».

*Scénario* :

- Via le support de discussion (par l'intermédiaire d'un bouton ou d'un onglet), l'expert soumet le concept pour la publication.
- Lorsqu'il se connecte, l'administrateur voit apparaître le concept dans la liste des objets en attente de validation. Il consulte le concept, il regarde s'il est passé par les étapes de l'analyse de cohérence et de la discussion, il actionne le bouton (ou le lien) permettant de passer le concept à l'état « publié ».

**Alternative :**

Si l'administrateur ne souhaite pas valider le concept pour une raison quelconque, il choisit l'action de le rejeter ; le concept revient dans l'état « privé » et l'expert peut le remanier.

### **8.4.3.3 Proposition d'argumentations pour traiter des conflits vers le consensus**

A titre d'illustration, rappelons que Ontolingua Server [FARQ96] élimine les conflits entre des ontologies dans leur représentation interne par la fabrication du vocabulaire de concept de chaque ontologie totalement disjoint du vocabulaire de concept de toutes les autres ontologies. Dans notre environnement, nous proposons aux experts des types d'argumentations et de résolutions de conflits pour les aider à aller vers le consensus.

Dans notre entendement, un domaine correspond à une unité thématique. Si plusieurs écoles de pensées/conceptions s'affrontent à l'intérieur du domaine, il est préférable de les traiter comme des domaines séparés : le but n'est pas de forcer un consensus artificiel sur les définitions des concepts. Le rôle de notre ontologie n'est pas de normaliser un domaine, mais de donner une représentation

de l'existant. Si on utilise le terme "domaine" pour désigner une vision d'un domaine, à l'intérieur d'une vision, on peut avoir plusieurs points de vue. En prenant ces points de vue deux à deux, il ne doit pas y avoir de contradictions ou d'incompatibilités. Si deux définitions sont incompatibles, cela signifie qu'il n'existe pas d'objet qui puisse les satisfaire toutes les deux.

Dans le cas d'une ontologie, la question qu'on se pose est : « Quelle est la définition du concept x ? ». L'idée est que chacun puisse exprimer son avis (en proposant une définition qui est rattachée au concept) et qu'ensuite, on essaie de se diriger progressivement vers un consensus, par exemple en rattachant chaque définition au point de vue adéquat. Le but est d'obtenir une ontologie où, pour chaque concept, on a au maximum une définition par point de vue ; les différentes définitions d'un concept étant compatibles.

Si le fait d'autoriser que plusieurs définitions soient rattachées à un même concept est intéressant, il ne faut pas oublier que certaines définitions peuvent remettre en cause la position d'un concept dans la hiérarchie. Pour cette raison le fait d'avoir plusieurs concepts désignés par le même terme, à différent emplacement dans la hiérarchie, est possible.

Pour ce qui est de la tâche de résolution de conflit, elle se situe dans le processus de construction de l'ontologie. Alors on s'intéresse à ce qui peut être fait aussitôt pour analyser les conflits de définitions et indiquer quelles opérations peuvent être utilisées pour les résoudre.

On peut utiliser des comparaisons pour analyser les incohérences entre définitions par :

- L'essai de généralisation et de spécialisation des définitions
- L'essai de compatibilité : la compatibilité de deux définitions n'implique nécessairement pas qu'elles représentent le même concept. L'intervention humaine est requise pour compléter le diagnostic, qui est d'identifier les incompatibilités sémantiques, que les différentes analyses ne peuvent détecter. La compatibilité doit être vue comme une règle de contrainte que l'ontologie doit valider pour être dans un état cohérent. Pour que l'ontologie soit cohérente, deux définitions d'un même concept doivent

être compatibles et appartenir à différents points de vue. Cependant durant le développement de l'ontologie, les incompatibilités peuvent être laissées (permises).

En ce qui concerne le processus de résolution de conflits, nous avons admis que l'ontologie est dans un état consensuel si, pour chaque concept, il y a au plus une définition par point de vue, et si toutes ces définitions sont jugés compatibles par les participants au travail collaboratif.

Quand un conflit de définitions se produit, nous proposons trois possibilités pour le résoudre :

- *le consensus* : s'il y a un conflit entre deux définitions, seulement l'une est gardée. Pour cet objectif on peut supprimer l'une d'elles ou combiner (fusionner) les deux en une seule avec par exemple l'opération "union". Dans ce cas il faut une notification indiquant la traçabilité des modifications (par qui et quand).
- *le contraste* : on décide à ce que les deux définitions correspondent à deux différents concepts. On peut alors ou créer un nouveau concept et un nouveau terme pour l'une des définitions ; ou créer un nouveau concept, garder le même terme et le lier à un autre point de vue.
- *les points de vue différents* : les définitions sont gardées mais chacune est liée à un point de vue différent.

Nous pouvons remarquer que les conflits peuvent être résolus en utilisant de simples opérations, par exemple : créer un nouveau concept, associer des définitions à de différents points de vue, combiner des définitions, supprimer une définition. Le choix d'une opération doit être toujours justifié.

Comme argumentations au sujet d'une nouvelle modification, les experts peuvent avoir plusieurs avis possibles :

- Favorable : accord au sujet de la modification.
- Addition : accord avec la modification tout en ajoutant certaines explications



- **Contraste** : désaccord au sujet de la modification tout en détectant une redondance, le concept devrait être fusionné avec un autre
- **Contradiction** : désaccord avec la modification tout en proposant de scinder le concept en deux concepts séparés
- **Incohérent** : désaccord avec la modification tout en détectant une incohérence.

Grâce à ces types d'argumentation, le système de groupware peut vérifier le consensus (l'accord) au sein du groupe de concepteurs et permettre de continuer la discussion.

#### **8.4.4 Analyse de cohérence locale (interne à une ontologie)**

Ici notre objectif est de prévenir, détecter et résoudre les conflits (incohérences) afin que notre ontologie soit construite progressivement et aussi qu'elle devienne graduellement cohérente.

Le processus de construction collaborative de l'ontologie se traduit par :

- la libre création des termes, définitions et concepts
- les délibérations : les concepteurs peuvent donner leur accord ou désaccord en faisant part de leur approbation ou désapprobation.
- l'analyse et la résolution des conflits : la phase d'analyse et de résolution de conflit peut être appliquée localement à une partie de l'ontologie. L'idée est que l'ontologie soit construite progressivement et aussi qu'elle devienne graduellement stable (et cohérente).

Ainsi quand la discussion aboutit à la publication d'une nouvelle modification dans l'ontologie, nous lançons le contrôle de cohérence interne. Le module du contrôle de cohérence syntaxique fournira une assurance de qualité. Par exemple, il analysera si tous les concepts sont reliés à d'autres concepts (connectivité graphique), si chaque concept a une définition simple, si chaque terme est une étiquette d'un concept unique, etc...

A ce niveau nous présentons une tâche d'analyse ou d'évaluation de l'ontologie en produisant une suite de tests qui évalue la cohérence dans l'ontologie en construction. Les tâches majeures conduisent à trouver et à rapporter des

incohérences avérées, des incohérences possibles, et des secteurs de construction inachevée. Le diagnostic peut être arbitrairement complexe, exigeant de l'analyse humaine étendue pour identifier tous les problèmes et de les présenter dans un ordre d'importance approprié. Les outils construits pour fournir le premier niveau de l'analyse, peuvent cependant considérablement aussi bien réduire le coût humain qu'améliorer l'uniformité de l'analyse. Dans notre suite de tests de diagnostic, nous n'essayons pas de trouver tous les problèmes ; nous choisissons juste un sous-ensemble qui est sur le plan informatique viable et motivé par les objectifs recherchés.

Nous proposons un ensemble de diagnostics qui peuvent être lancés séparément (dans les partitions de l'ontologie) ou dans leur intégralité. La liste courante de diagnostics vient de l'analyse des tests les plus utiles que nous exécutons sur les ontologies.

Dans notre cas, l'analyse de la cohérence des ontologies porte sur :

- de simples contrôles d'imperfection (concepts orphelins, concepts sans définition, concepts sans ressources, relations non utilisées, erreurs de circularité, etc)
- une analyse syntaxique (redondances et homonymes )
- une évaluation sémantique (conflit sémantique entre définitions, etc).

La résolution de ces incohérences citées ci-dessus se fait par des opérations élémentaires (telles que : créer, ajouter, modifier, supprimer, combiner) sur les différents éléments concernés.

Dans la section suivante, nous nous intéressons à l'analyse de cohérence globale qui contrairement à l'analyse locale s'occupe de la cohérence entre différentes ontologies.

#### **8.4.5 Analyse de cohérence globale (inter-ontologies)**

Quand une ontologie satisfait au contrôle de cohérence interne, elle est publiée (intégrée) dans le réceptacle stockant toutes les ontologies, et le contrôle de cohérence globale est lancé. Le contrôle de cohérence globale ou contrôle inter-ontologies (entre ontologies) consiste à trouver des incohérences entre différentes

ontologies du réceptacle et à les corriger. Par exemple, quand un concept est en relation avec un concept existant dans une autre ontologie, nous devons vérifier l'existence de ce dernier concept. D'abord on référence le concept dans toutes les ontologies et versions d'ontologies. S'il s'avère qu'il n'y a pas de contradictions (incohérences) entre les différentes définitions du concept dans différentes ontologies, alors on pourrait fusionner les ontologies en une seule (grande) ontologie en se servant de ce concept comme un concept passerelle. Dans le cas contraire (s'il y a des incohérences), on serait amené à effectuer des opérations élémentaires consensuelles pour résoudre les conflits sémantiques. Ensuite on se fait une obligation de faire une notification de modification aux concepteurs et la relation inter-ontologies doit être consolidée, et les définitions additionnelles pour les mêmes concepts doivent être intégrées.

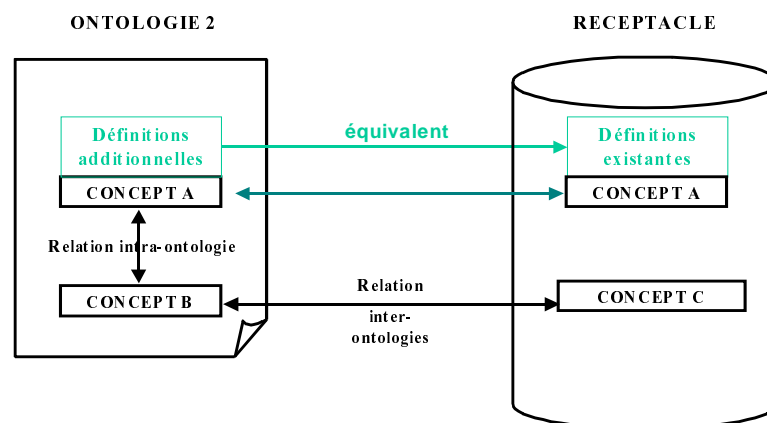


FIGURE 8.7 INSERTION D'UNE NOUVELLE ONTOLOGIE DANS LE RECEPTACLE

Plusieurs cas doivent être considérés à la suite de l'intégration d'une nouvelle ontologie dans le repository :

1. Un concept A (dans l'Ontologie 2) est en relation avec un concept D déjà dans le réceptacle. Dans ce cas-ci, dans l'Ontologie 2, nous avons déjà la connexion R (Concept A → Concept D dans le réceptacle) au moyen d'un

certain type de relation R. Pour la consolidation, nous devons écrire la contre-partie dans le réceptacle, c.-à-d. R (Concept D → Concept A de l'Ontologie 2). En faisant ainsi, nous assurons la continuité du réceptacle.

2. Un concept C est déjà dans le réceptacle, mais il a des définitions additionnelles dans la nouvelle ontologie ; dans ce cas-ci, les définitions additionnelles doivent être ajoutées au concept C déjà stocké dans le réceptacle. L'ajout de définitions additionnelles se fait sous réserve de l'accord du propriétaire de l'ontologie.

Après avoir décrit les fonctionnalités de notre système de conception, la figure 8.8 montre le modèle général pour la construction d'ontologie consensuelle.

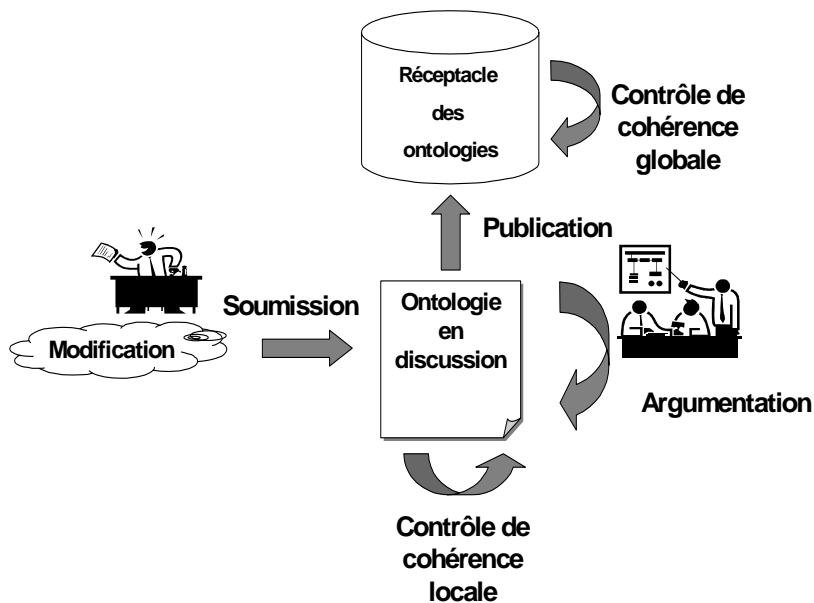


FIGURE 8.8 MODELE GENERAL POUR LA CONSTRUCTION D'ONTOLOGIE CONSENSUELLE

Dans la section suivante nous nous intéressons à l'outil groupware qui offre le support de conception collaborative.

## 8.5 Towntology Groupware

En s'appuyant sur le modèle proposé antérieurement, nous avons développé un outil qui permet le travail coopératif entre les experts pour rendre possible la transformation collaborative des ontologies pré-consensuelles vers des ontologies consensuelles. Townto-Plone est une instance Plone modifiée *ad hoc* sur un serveur Zope qui a été mis à disposition des experts du projet Towntology sur le serveur Web de LIRIS.

La proposition Townto-Plone repose sur le fait que chaque objet de l'ontologie peut être traité comme un document de type « contenu » défini par un seul URL sur le Web. Rappelons-nous que sur les Systèmes de Gestion de Contenu, tout "contenu" peut être soumis au processus de la chaîne globale de publication donc coordonable, discutable et publiable. D'abord nous avons traduit tous les objets : points de vue, concepts, types de relation et relations de l'ontologie.

Notre environnement de développement d'ontologies a un support de construction collaborative distribuée. L'accès distribué est fourni par l'interface Web. Le travail collaboratif est facilité par le contrôle d'accès pour utilisateur et pour groupe, et des sessions multi-utilisateurs.

Nous développons un ensemble d'outils et de services non pas seulement pour le développement d'ontologies par des individus, mais aussi pour atteindre le consensus entre groupes distribués sur des ontologies communes. Ces outils utilisent donc le web pour offrir un large accès et permettre aux utilisateurs de soumettre, créer, naviguer et éditer des ontologies stockées sur un serveur d'ontologies. Ces utilisateurs éventuels seront géographiquement dispersés à travers une variété d'institutions ou d'organisations travaillant dans l'urbanisme.

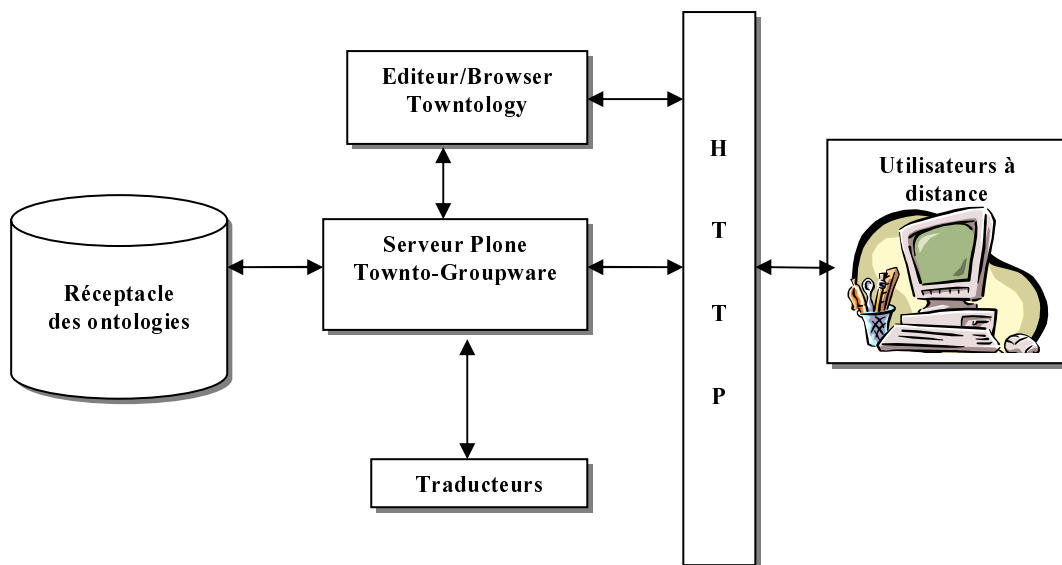


FIGURE 8.9 ARCHITECTURE DE TOWNTO-GROUPWARE

Le serveur fournit l'accès à la bibliothèque d'ontologies, et en s'appuyant sur l'outil Towntology, permet la création de nouvelles ontologies et la modification d'ontologies existantes. Il y a deux modes d'interactions avec le serveur :

1. Les utilisateurs à distance peuvent utiliser leurs navigateurs web pour naviguer, construire et maintenir des ontologies stockées sur le serveur. Le serveur inter-agit avec eux à travers le protocole de transfert hypertexte répandu (HTTP) et le langage hypertexte (HTML) utilisés sur le Web. Ceci rend le serveur accessible à une très large audience. Le serveur permet à de multiples utilisateurs de travailler simultanément en session partagée sur une ontologie. L'éditeur fournit un certain nombre de dispositifs pour supporter le travail collaboratif (par exemple : la construction du consensus, la traçabilité des modifications, etc).
2. Un utilisateur peut traduire une ontologie dans un format utilisé par une application spécifique. La traduction résultante peut être utilisée de plusieurs manières. On pourrait programmer un traducteur. Nous avons deux solutions : la première consiste à mettre en ligne notre DTD et

l'imposer aux utilisateurs qui auraient des ontologies dans d'autres formats, la deuxième solution consiste à faire un logiciel (ou modifier le logiciel existant) pour qu'il lise un certain nombre de représentations comme par exemple KIF et OWL.

Pour rendre l'interface simple, le serveur fournit trois types de ressources : la première concerne les ontologies stockées dans la bibliothèque, la deuxième contient les logiciels du système Towntology tels que Townto-Browser, Townto-Editor, et Towntology-Plone.

Le serveur fournit une grande variété d'autres services à savoir :

- **La gestion des utilisateurs**

Un utilisateur est la représentation informatique d'une personne physique. On utilise généralement un dérivé de son nom pour construire son identifiant (ex : kkeita pour Kader Keita). Un groupe est la représentation informatique d'un groupement d'utilisateurs (ou de groupes). Ces utilisateurs sont réunis dans un groupe parce qu'ils partagent un point commun, par exemple l'appartenance à une même profession. Ainsi, le groupe « Experts » va regrouper tous les spécialistes de l'urbanisme et de l'aménagement du territoire. Dans notre environnement un utilisateur peut appartenir à un ou plusieurs groupes. Utiliser les groupes est donc une façon pratique de distribuer des droits sur le site ou simplement de classer les utilisateurs. Les utilisateurs s'authentifient auprès du site grâce à leur identifiant et leur mot de passe. Ils sont réunis en groupes qui peuvent être calqués sur l'organisation de la communauté. Les comptes utilisateurs peuvent être créés par l'administrateur du site ou fournis par l'intégration d'une source externe de comptes utilisateurs comme un annuaire LDAP<sup>27</sup> (Lightweight Directory Access Protocol).

- **La possibilité de délégation de droits**

La délégation de droits permet à un utilisateur, comme dans la vraie vie (enfin, souvent), de déléguer (prêter) ses droits à un autre utilisateur. Ainsi, en cas

---

<sup>27</sup> <http://fr.wikipedia.org/wiki/LDAP>

d'absence de l'indispensable administrateur, quelqu'un d'autre pourra le remplacer pendant cette période. Plutôt que de donner son identifiant et son mot de passe à cet utilisateur, l'administrateur va simplement lui déléguer les droits de modération du contenu qu'il aura dans telle ou telle rubrique. Il est intéressant de noter qu'un utilisateur ne peut donner (ou déléguer) plus de droits qu'il n'en a lui-même.

- **La sécurité basée sur les rôles**

Un rôle est généralement la représentation informatique d'une fonction dans la communauté.

Les utilisateurs sont porteurs de zéro, un ou plusieurs rôle(s). Les rôles regroupent des permissions techniques qui autorisent l'accès au contenu du site ainsi que des actions au sein du site. Les exemples de permissions nécessaires aux acteurs du site sont :

- consulter du contenu ;
- ajouter, soumettre du contenu ;
- modifier le contenu ;
- valider du contenu ;
- supprimer du contenu.

Au départ, les membres de l'équipe qui s'occupent du paramétrage et de la maintenance du site possèdent le rôle d'administrateur. Ils ont ensuite à s'assurer que les rôles correspondant aux différents profils d'utilisateurs sont paramétrés et affectés aux utilisateurs.

- **La base documentaire ou rubriques d'accès au contenu**

L'objectif principal du site est de permettre la diffusion d'informations aux membres de la communauté Towntology et un accès rapide à ces informations. La production de cette information n'est pas réservée à une élite informatique, mais ouverte à tous, avec, pour certaines informations, un circuit de validation avant publication. Les types d'informations à gérer sont :

- Les actualités : regroupent les articles d'actualité et de revue de presse publiés par la communauté Towntology ;
- Les ontologies : regroupent les ontologies et les éléments publiés sur le site.



- Les événements : regroupent les annonces d'événements relatifs à l'activité de la communauté ;
- Les liens : regroupent des liens annotés vers les sites partenaires ou affiliés, ou les sites de référence.

Ces informations constituent les rubriques du site qui sont des espaces où sont agrégées toutes les informations publiées répondant à un critère donné.

- **Les circuits de validation de publication (workflow)**

Tous les membres peuvent, une fois authentifiés grâce à leur identifiant et leur mot de passe, proposer du contenu dans le site, dans les rubriques pour lesquelles ils auront été habilités à le faire. Un circuit de validation permet à la personne habilitée de valider ou de refuser la publication d'un contenu proposé.

- **L'agenda partagé**

Les événements de la communauté sont consignés dans un agenda partagé par tous. La publication d'un événement dans l'agenda est également soumise à un circuit de validation.

- **Le moteur de recherche**

Le moteur de recherche permet de retrouver les informations disponibles sur le site, en ne présentant à l'utilisateur que les contenus qu'il est habilité à voir. Les résultats de la recherche sont affichés sur une page dynamique. Ces résultats doivent pouvoir être facilement parcourus et reclassés. Ce même moteur de recherche est également utilisé pour afficher dynamiquement les actualités récentes de la communauté et les informations relatives à la rubrique ou au document que l'utilisateur est en train de consulter.

- **Espace personnel pour chaque membre de la communauté**

Chaque utilisateur dispose d'un espace personnel dans lequel il peut mettre en place ses propres fichiers, ou bien travailler ses contributions avant demande de publication.

- **Espaces de travail collaboratif**

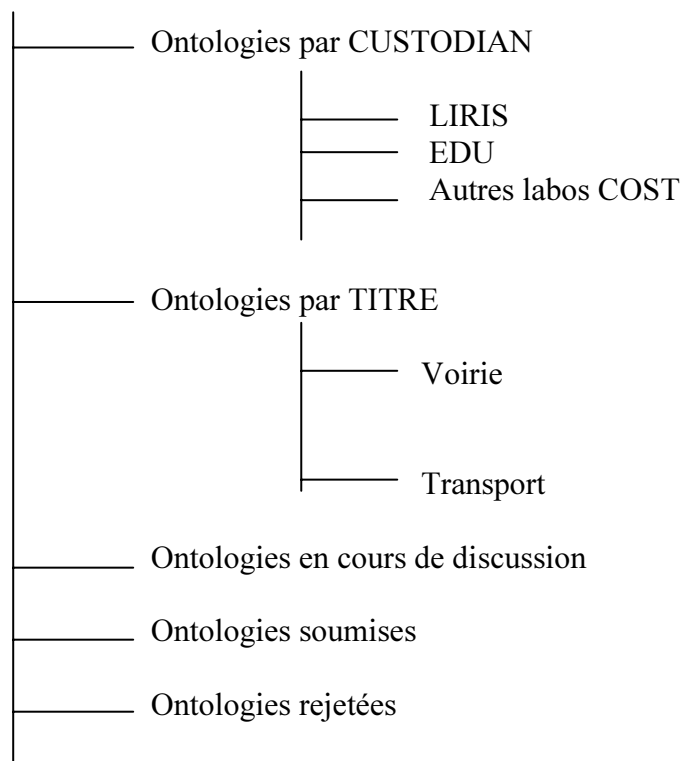
Chaque groupe de travail dispose d'un espace commun à tous ses membres. Ceci permet à des experts de travailler ensemble sur les définitions d'un objet (élément) de l'ontologie en essayant de se mettre d'accord (consensus), avant de décider d'en publier une version définitive. Cela consiste à faire émerger un consensus par la discussion à la manière des Wikis.

- **L'espace de stockage des ontologies**

Nous considérons deux niveaux de stockage des ontologies dans le réceptacle. Nous faisons :

- un répertoire par Custodian (institution propriétaire de l'ontologie)
- un répertoire par domaine (par exemple toutes les ontologies concernant la Voirie seront stockées dans le répertoire Voirie)

#### Réceptacle d'Ontologies



L'arborescence de stockage dans le réceptacle indique qu'il y a possibilité de soumettre des ontologies (envoyées) par courrier électronique qui atterrissent dans le répertoire des ontologies soumises. Dans le cas où ces ontologies seraient décrites dans un format compréhensible par notre système, elles pourraient être

soumises au processus de construction du consensus, c'est à dire à la discussion des experts et aux différents contrôles de cohérence. Sinon elles sont rejetées et placées dans le répertoire des ontologies rejetées.

## **8.7 Conclusion**

Dans ce chapitre nous avons décrit notre approche de conception coopérative d'ontologies pré-consensuelles qui a porté sur : notre méthodologie, notre modèle d'ontologie, le langage dans lequel notre ontologie a été implémentée, les fonctionnalités et les besoins de notre système et de Towntology-Groupware. Dans le chapitre suivant nous donnons les informations techniques sur l'implémentation de notre système, en nous intéressant surtout aux interfaces qui peuvent donner accès aux différents services proposés.

## **Chapitre 9. Implémentation**

Dans ce chapitre, nous nous intéressons aux aspects technologiques de l'outil Towntology. D'abord nous décrivons l'architecture du système Towntology en le comparant avec d'autres outils de construction d'ontologie. Ensuite, nous présentons les outils ou principales interfaces utilisateurs qui donnent accès aux différents services. Après, nous parlons des modules du système en décrivant les fichiers et classes appartenant à ces applications. Et enfin, nous donnons les structures de données internes qui nous ont permis de coder notre ontologie sous forme de graphe, et les formules mathématiques pour la représentation des nœuds (concepts) dans différentes couronnes.

### **9.1 Architecture du système Towntology**

Les aspects technologiques des outils d'ontologie, attribue une attention spéciale à leur architecture logiciel (autonome, client-serveur, n-tier), leurs capacités d'extensibilité, aux langages de programmation dans les quels ils sont implémentés, et à leurs "backends" qui définie comment ils stockent les ontologies (généralement dans les fichiers ou les bases de données).

L'architecture et les capacités d'extensibilité des outils sont étroitement liées. Les premiers outils réalisés (Ontolingua, OntoSaurus, et WebOnto) ont une architecture client-serveur et, par conséquent, sont difficiles à étendre bien que cela ne veuille dire qu'ils ne sont pas extensibles avec d'autres fonctions. Les outils plus modernes tels que OilEd, OntoEdit, et Protégé-2000, ont une architecture 3-tier, où il y a une distinction claire entre les "backends" en charge du stockage de l'ontologie, les modules logiques en charge de la plupart de la logique de l'application, et les "frontends" en charge de toutes les applications interface utilisateur. Ces outils sont plus faciles à étendre que les premiers. OilED permet facilement de changer ses "backends" de sorte que les ontologies peuvent être stockées dans différents formats. OntoEdit et Protégé-2000 permettent facilement d'étendre non seulement les "backends" mais aussi l'interface utilisateur de l'éditeur d'ontologie par l'utilisation de plug-ins. Les suites d'outils KAON et WebODE sont basées sur des architectures de serveur d'application, qui

fournissent aussi des bons dispositifs d'extensibilité, le support multi-utilisateur, des facilités de gestion des accès pour les utilisateurs, etc.

La plupart des outils stockent leurs ontologies dans les fichiers textes. Cela limite la taille des ontologies qui peuvent être construites avec. Seuls KAON, Protégé-2000, et WebODE sont capables de stocker leurs ontologies dans les bases de données, et ainsi de gérer de larges ontologies. Enfin, nous pouvons dire que la plupart des outils ont été développés complètement en Java.

L'outil Towntology a aussi été développé en java, pour ne pas être dépendant d'une plate-forme.

Son architecture est organisée en trois couches :

- Backend, pour le stockage des ontologies : les ontologies sont stockées et sont accessibles sous forme de fichiers XML,
- APIs Towntology : cette couche est constituée des services offerts par l'outil et sa suite. Voir le schéma.
- Applications client : cette couche est constituée des applications Townto-Portal, Townto-Browser et Townto-Editor qui donnent accès à l'ontologie et fournissent des applications pour utilisateur final.

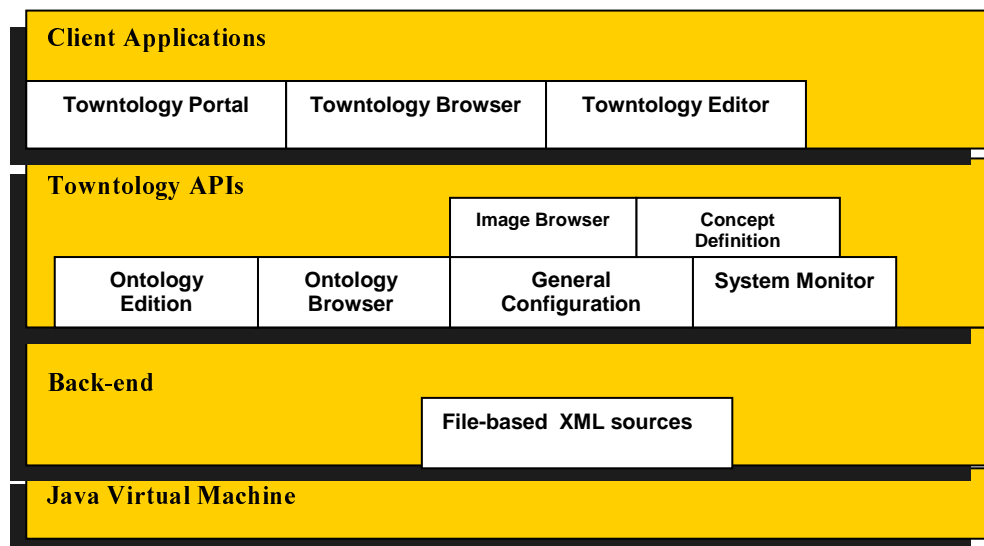


FIGURE 9.1 ARCHITECTURE DE L'OUTIL TOWNTOLGY

Dans le tableau qui suit nous donnons une comparaison - basée sur l'architecture des systèmes - des outils de construction d'ontologie et de l'outil Towntology.

TABLEAU 9.1 COMPARAISON DES OUTILS D'ONTOLOGIE : ARCHITECTURE DU SYSTEME

	<b>KAON</b>	<b>OilEd</b>	<b>OntoEdit</b>	<b>Ontolingua</b>	<b>OntoSaurus</b>	<b>Protégé-2000</b>	<b>WebODE</b>	<b>WebOnto</b>	<b>Towntology</b>
Architecture de l'application	n-tier	3-tier	3-tier	Client/Server	Client/Server	3-tier	n-tier	Client/Server	n-tier
Extensibilité	Serveur d'application	Backends	Plug-ins	--	--	Plug-ins	Serveur d'application	--	Backends
Stockage de l'ontologie	Fichiers DBMS	Fichiers	Fichiers	Fichiers	Fichiers	Fichiers DBMS	DBMS	Fichiers	Fichier XML
Langage de programmation	Java	Java	Java	Lisp	Lisp	Java	Java	Java + Lisp	Java



## 9.2 Les outils ou principales interfaces utilisateurs

La plupart des outils existants et décrits dans la section état de l'art centrent leur support de conceptualisation et d'analyse des incohérences sur l'expérience de l'ingénieur de connaissances. Par exemple, dans Ontolingua [FARQ96] et dans Protégé [RAYF01] la conception se fait entre ingénieurs de connaissances. On remarque que ces outils travaillent sur la création des ontologies une fois le consensus déjà réalisé (ontologies post-consensuelles). Seuls les modèles C-VISTA [DIEN02] et CODE [CAÑA04] supportent le traitement des points vues. Ainsi, un objet de la conceptualisation peut regrouper les différents points de vues des experts. Même si Protégé 2000 reste encore l'outil le plus utilisé, il permet surtout de construire des ontologies post-consensuelles formelles.

Nous nous sommes aussi inspirés des techniques de visualisation des ontologies que intègrent les outils comme GraphViz<sup>28</sup> et Jambalaya [STOR01].

Ainsi les techniques des outils évoqués sont basées sur la visualisation d'ontologie déjà existante, formelle. Dans notre cas, nous n'avons pas besoin de visualiser des instances. Nous sommes dans la phase de la conceptualisation de notre ontologie. À cette étape courante nous ne cherchons pas à construire une ontologie formelle. Nous utilisons un schéma simple pour permettre à des non-informaticiens de valider les concepts et d'autres éléments de l'ontologie. Ainsi nous avons décidé de construire notre outil qui permet d'éditer et de stocker l'ontologie, d'afficher l'ontologie sous forme de graphe visuel, de naviguer dans l'ontologie et de l'interroger. Cette visualisation offre un support pour la construction collaborative de notre ontologie.

Towntology visualise des ontologies enregistrées dans des fichiers XML et les représente sous forme de graphe. Le logiciel propose à l'utilisateur un navigateur et un éditeur d'ontologie. Il offre la possibilité d'éditer et de visualiser des définitions multimédia. L'utilisateur a également la possibilité de consulter des images interactives liées aux différents concepts de l'ontologie. Ensuite il propose un mécanisme de construction de consensus et de collaboration entre les acteurs,

---

<sup>28</sup> <http://www.graphviz.org/>



qui en fait est une extension de l'éditeur visuel. Enfin, l'outil propose deux modules de contrôle de cohérence interne et externe.

### 9.2.1 Navigateur visuel d'ontologie : Towntology-Browser

Après avoir analysé les systèmes de conception d'ontologie existants, nous avons développé un outil qui offre un portail visuel comme entrée dans notre système. Ce portail donne trois possibilités d'accès à l'ontologie, comme montré sur la figure 8.11 :

- Une liste de concepts,
- Des photographies représentant des vues urbaines avec des zones actives
- Le graphe représentant l'ontologie sous forme de réseau sémantique.

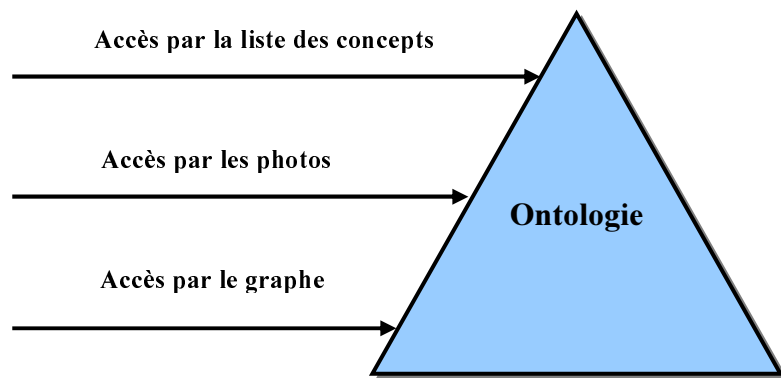


FIGURE 9.2 DIFFERENTS ACCES POUR STRUCTURER LE PORTAIL VISUEL

L'interface du navigateur visuel est donc divisée en trois parties. La partie droite affiche le graphe. Ce graphe est une partie du réseau sémantique de l'ontologie centré autour du concept sélectionné (ou concept courant), il permet de naviguer dans l'ontologie. La partie gauche est une liste déroulante affichant tous les termes utilisés dans le graphe comme labels de concepts présents dans l'ontologie. La partie au centre affiche des photos de zones urbaines.

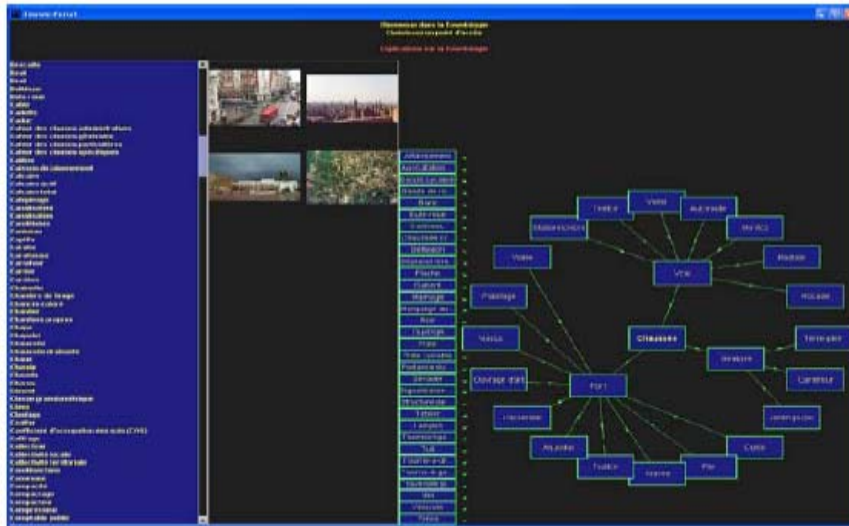


FIGURE 9.3 LE PORTAIL VISUEL : TOWNTOLGY BROWSER.

Le navigateur (portail) visuel Townology-Browser peut donner accès aux interfaces suivantes :

- **La fenêtre définition : "Information frame"**

Dans le navigateur visuel, un clic sur un nœud de concept et un autre sur l'onglet "définition" affichent la fenêtre des définitions ou "information frame" ( voir figure 8.12) contenant toutes les informations sur le concept sélectionné. Cette fenêtre permet de visualiser le(s) définition(s) textuelles et multimédia du concept sélectionné ainsi que les sources de ces définitions. La partie en haut à gauche contient les informations générales sur l'ontologie contenant le concept courant. La partie en bas à gauche contient la liste des relations que ce concept entretient avec ses voisins directs.

Concrètement la figure 8.12 ci-dessous affiche les informations (métadonnées) sur le concept "Espace public". Ce concept a deux définitions et est illustré par une image. Les sources des définitions et de l'image sont également indiquées. Sur la partie gauche de la fenêtre la liste de relations ayant le concept comme argument sont énumérées.



FIGURE 9.4 INFORMATION FRAME

- **La fenêtre de gestion des images interactives : "Image-Browser "**

On y a accès en cliquant sur une photo dans le portail visuel. Cette fenêtre offre la possibilité à l'utilisateur d'accéder à l'ontologie à travers des images de zones urbaines. L'utilisateur sélectionne une zone de l'image à laquelle sont associés des concepts de l'ontologie. En sélectionnant un des concepts, l'utilisateur peut accéder au graphe centré sur ce concept dans Townto-Browser.

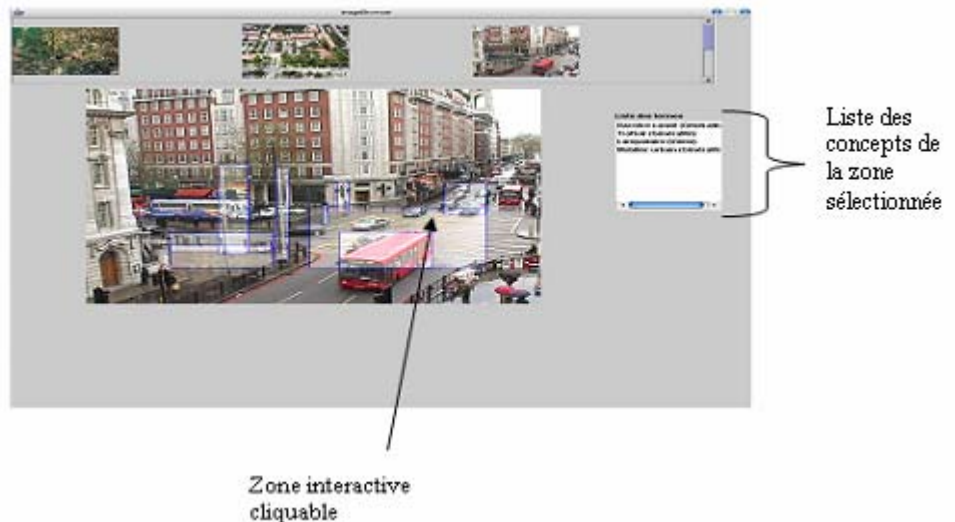


FIGURE 9.5 GESTION DES IMAGES INTERACTIVES

Le système intègre aussi un module pour éditer des photos. Un utilisateur peut éditer une photo représentant des concepts clés de l'ontologie, en divisant la photo en sous-images dans des zones rectangulaires contenant (pointant vers) des concepts. Il devrait être possible d'assigner un concept simple à une sous-image (zone rectangulaire), mais nous avons plutôt choisi d'assigner plusieurs concepts à une zone. En effet autant des dispositifs concrets sont facilement détectables sur des photos, autant il est plus difficile pour des concepts plus abstraits. Par conséquent la solution choisie a été d'assigner différents concepts à une zone rectangulaire unique. Le schéma 8.14 donne un exemple. La description des correspondances entre les sous-images et les concepts est faite par une autre extension de XML.

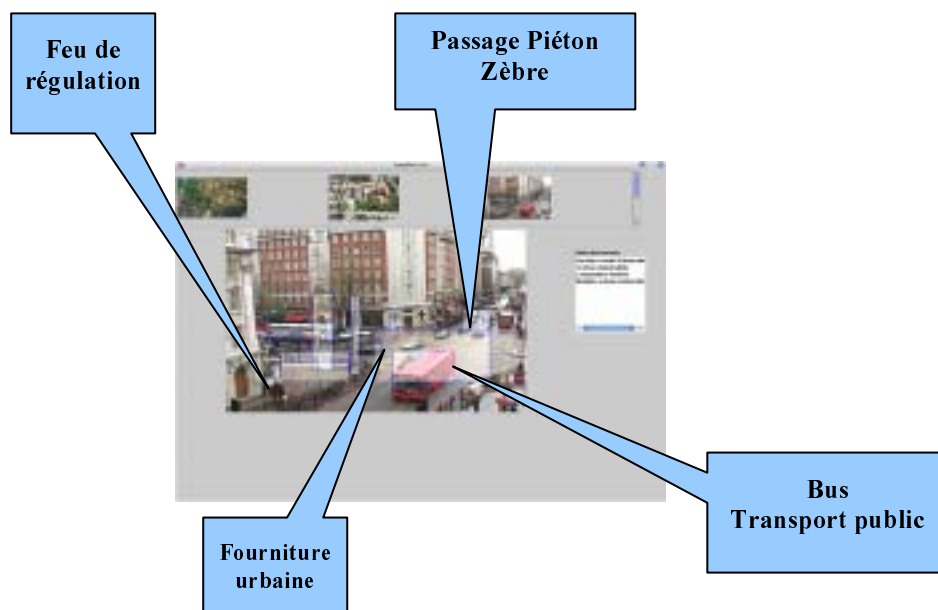


FIGURE 9.6 EXEMPLE DE DIVISION D'UNE PHOTO EN SOUS-IMAGES (ZONES) ET ASSIGNATION DE CONCEPTS POUR ACCEDER A L'ONTOLOGIE.

### 9.2.2 Editeur visuel d'ontologie : Towntology Editor

Cette interface permet de stocker et donner un accès à l'ontologie. Elle permet à l'utilisateur d'insérer et de modifier des concepts et des relations, leurs définitions, et aussi d'intégrer des métadonnées de traçabilité. Elle contient toutes

les interfaces graphiques utilisateurs (GUI) permettant d'éditer l'ontologie. C'est-à-dire elle permet d'éditer les informations principales sur l'ontologie (son titre, sa langue, son propriétaire), et les éléments composants l'ontologie comme les concepts, leurs définitions, les termes liés et leurs ressources multimédia, etc... L'Editeur permet également d'ajouter (créer) de nouveaux éléments à l'ontologie et d'effectuer les mises à jour nécessaires.

Un intérêt tout particulier a été porté à la représentation des définitions dans le logiciel d'édition, mais également aux illustrations associées aux concepts. Par ailleurs, l'outil offre la possibilité de varier et expliciter les sources de définitions. Il permet aussi de bien adapter les niveaux de précision des définitions à l'usage et au public concerné par la base de connaissances.

Enfin l'éditeur Towntology permet à l'utilisateur de consulter, à temps réel, les statistiques sur (entre autres) le nombre de concepts et de relations existants dans l'ontologie en cours, et surtout le nombre de concepts ou de graphes connexes, c'est à dire le nombre de concepts ou de graphes isolés.



**FIGURE 9.7 TOWNTOLLOGY EDITOR**

### 9.2.3 Interface principale : Townto-Plone

Sur cette interface les principales parties contiennent les informations suivantes, comme décrit dans la figure 8.16 :

1. Les onglets : ils affichent la page d'accueil, les nouvelles de la communauté, les Wikis, et le support Townto-Editor.
2. Les informations et les propriétés sur l'utilisateur qui a ouvert la session.
3. L'arborescence des fichiers de stockage des contenus.
4. Les onglets qui affichent les différentes possibilités d'édition des contenus.
5. La notification automatique par RSS (Really Simple Syndication) des définitions soumises à valider. Syndication Vraiment Simple, est un fichier XML dynamique dont les lecteurs de RSS affichent le contenu qui est mis à jour en permanence. Ce système est très utilisé pour diffuser des nouvelles, des alertes, de listes d'information. Pour l'éditeur du RSS, cela permet de faire connaître les nouvelles informations aux autres collaborateurs.
6. Les méta-données concernant le concept courant (nom du créateur, type, dernière date de modification, et état de sa définition).
7. La visualisation du concept courant avec les ressources multimédia associées.
8. Le calendrier des dates importantes de la communauté Towntology.
9. La zone de recherche des contenus.



FIGURE 9.8 INTERFACE PRINCIPALE DE TOWNTO-PLONE

L'interface donne la possibilité de visualiser les états des définitions. En cliquant sur les définitions en discussion, Townto-Plone affiche tous les objets ayant comme propriété d'état "soumis". Ainsi l'utilisateur pourra discuter sur cette définition.

Aussi, nous pouvons visualiser uniquement les objets ayant comme état "publié" en cliquant sur "ontologie publique".

Townto-Plone pourra supporter la méthodologie proposé même si la spécification des objets est définie par des langages de représentation des ontologies comme RDFS. Grâce au fait que le code source de Plone ait une licence GNU/GPL nous pouvons adapter et optimiser tous les processus et services du système. La communauté Zope est en pleine croissance et beaucoup d'entreprises comme des instituts de recherche supportent leur développement ; ceci fait de Townto Plone un outil économique, stable et indépendant.

### 9.3 Les principaux modules du système Towntology

Notre système est composée de trois modules principaux : Towntology.portal, Towntology.browser, et Towntology.editor. Ces modules sont à leur tour composées de fichiers et de classes.

Le tableau suivant décrit les trois prototypes, leurs fonctionnalités et les stagiaires qui ont aidé à les développer.

TABLEAU 9.2 DESCRIPTION DES PROTOTYPES

Prototype	Nom	Fonctionnalité	Stagiaire développeur
1 2003	Portail visuel	Portail visuel Editeur visuel Navigateur visuel	Etudiant de l'INSA de Lyon  S. GESCHE
	Editeur d'image	Editeur d'images	
2 2004	Navigateur visuel	Navigateur visuel	Etudiants de Licence Professionnelle Metinet  F. DUMAS, N. JACQUEMET, A. JUSELME, D. TRICHARD
	Editeur visuel	Regroupe toutes les GUI permettant d'éditer l'ontologie	
3 2006	Townto-Plone	Visualise les états des définitions	Master 2 recherche D. SINUHE

#### 9.3.1 Les fichiers

Les fichiers sont les éléments des modules permettant, entre autres, de lancer le générateur de portail, le navigateur et l'éditeur d'ontologie. Ils permettent aussi la description du portail, la description XML des images, mais aussi de donner la dernière configuration et de lister les termes.



Dans le tableau suivant, nous listons les fichiers en donnant leurs noms et emplacements dans les applications, leur descriptifs et leurs rôles.

TABLEAU 9.3 LISTE DES FICHIERS

Nom et emplacement	Descriptif	Rôle
<b>Towntology/vocab.ini</b>	Vocabulaire de l'application	Ce fichier liste les termes utilisés dans l'interface. On peut être amené à le modifier notamment pour des questions de langue
<b>Towntology/portal.ini</b>	Description du portail	Ce fichier liste les images (.twi) servant de point d'accès pour Townto-Browser. Tout fichier utilisé dans ce cadre doit se trouver dans le répertoire img.
<b>Towntology/terms.ini</b>	Liste des termes (accès rapide)	Ce fichier liste les termes utilisés dans l'ontologie à l'usage du générateur de portails, et il est généré automatiquement.
<b>Towntology/lastus.ini</b>	Dernière configuration	Ce fichier donne la dernière configuration utilisée.
<b>Towntology/browser.exe</b>	Lancement du navigateur	Ce fichier permet de lancer le navigateur visuel (Townto-Browser)
<b>Towntology/portgen.exe</b>	Lancement du générateur de portail	Ce fichier permet de lancer le générateur de portail
<b>Towntology/editor.exe</b>	Lancement de l'éditeur d'ontologie	Ce fichier permet de lancer l'éditeur d'ontologie (Townto-Editor)
<b>Towntology/img/xxx.jpg</b>	Image xxx	Ce fichier présente l'image xxx
<b>Towntology/img/xxx.twi</b>	Description XML de l'image xxx	Ce fichier donne une description XML de l'image xxx
<b>Towntology/img/i_xxx.jpg</b>	Icône de l'image xxx	Ce fichier permet d'afficher l'image xxx

### 9.3.2 Les classes

Les classes sont les composantes qui permettent la mise en place des différentes APIs (Application Programming Interfaces) qui offrent les services aux utilisateurs. Toutes ces classes appartiennent aux applications Towntology.browser,

Towntology.editor, Towntology.ported et Towntology.ontology. Nous aurions pu les classer par rapport à leur appartenance à une application, mais nous avons préféré les classer dans l'ordre alphabétique.

TABLEAU 9.4 LISTE DES CLASSES DANS L'ORDRE ALPHABETIQUE

<u>Browser</u>	<u>FixedParagraphView</u>	<u>LinkFailureException</u>
<u>BrowserManager</u>	<u>FixedParagraphView.FixedFlowStrategy</u>	<u>MultiOntology</u>
<u>BrowserStarter</u>	<u>FixedStyledEditorKit</u>	<u>Ontology</u>
<u>ChoosableInteractiveImage</u>	<u>FixedStyledViewFactory</u>	<u>OntologyCreationException</u>
<u>ChoosableInteractiveImageFactory</u>	<u>Glossary</u>	<u>OntologyFactory</u>
<u>ColorTable</u>	<u>Graph</u>	<u>OntologyPanel</u>
<u>Concept</u>	<u>GraphLayout</u>	<u>Portal</u>
<u>ConceptsOntology</u>	<u>GraphManager</u>	<u>PortalEditor</u>
<u>ConceptsPanel</u>	<u>GraphPanel</u>	<u>PortEdStarter</u>
<u>ConceptsRelation</u>	<u>HintsPanel</u>	<u>Relation</u>
<u>ConnexPanel</u>	<u>ImageEditor</u>	<u>RelationsPanel</u>
<u>Constants</u>	<u>InfoFrame</u>	<u>RelationType</u>
<u>Definition</u>	<u>InfoManager</u>	<u>RelationTypesPanel</u>
<u>DisplayableOntology</u>	<u>IniReader</u>	<u>Resource</u>
<u>Domain</u>	<u>IniWriter</u>	<u>SiteMakerStarter</u>
<u>DomainsPanel</u>	<u>InteractiveImage</u>	<u>SoundHandler</u>
<u>DOMCreationFailureException</u>	<u>InteractiveImageBuilder</u>	<u>Source</u>
<u>EditableOntology</u>	<u>InteractiveImageBuilderFactory</u>	<u>StatsOntology</u>
<u>Editor</u>	<u>InteractiveImageChooser</u>	<u>StatsPanel</u>
<u>EditorManager</u>	<u>InteractiveImageCreationException</u>	<u>StyledTextPanel</u>
<u>EditorStarter</u>	<u>InteractiveImageFactory</u>	<u>SysMon</u>
<u>EventHandler</u>	<u>LastUse</u>	<u>Term</u>
<u>EventListener</u>	<u>LexicographicalComparator</u>	<u>TermsList</u>
	<u>LinkFactory</u>	<u>XMLReader</u>

Dans la suite nous nous intéressons à la partie de l'implémentation qui s'occupe du codage, en particulier à la structure des données pour représenter notre ontologie sous forme de graphe, et aux formules mathématiques pour la présentation des concepts (première et deuxième colonne)

#### 9.4 Structure de données internes pour représenter l'ontologie sous forme de graphe

Le graphe  $G = (X, U)$  est formé d'un ensemble de Nœuds  $X$  et d'un ensemble d'arcs  $U$ .

L'ordre du graphe est son nombre de Nœuds.

Un arc  $u = (x,y)$ ,  $y$  est le successeur de  $x$ .

Soient les ensembles :

**Graphe** : ensemble des graphes orientés,

**Nœud** : ensemble des Nœuds du graphe

La constante **graphevide** définit un graphe sans Nœud.

La description fonctionnelle donne les opérations suivantes :

Ordre (le nombre de Nœuds)	: Graphe	→	naturel
Arc	: Graphe × Nœud × Nœud	→	booléen
D+ (nb d'arcs incidents vers l'ext)	: Graphe × Nœud	→	naturel
D- (nb d'arcs incident vers l'int)	: Graphe × Nœud	→	naturel
Degré (d+ + d-)	: Graphe × Nœud	→	naturel
lème Succ	: Graphe × Nœud × naturel	→	naturel
AjouterNœud	: Graphe × Nœud	→	Graphe
SupprimerNœud	: Graphe × Nœud	→	Graphe
AjouterArc	: Graphe × Nœud × Nœud	→	Graphe
SupprimerArc	: Graphe × Nœud × Nœud	→	Graphe

Pour les graphes valués (arcs portant des valuations) :

AjouterArc : Graphe × Nœud × Nœud × £ → Graphe

Où £ désigne un ensemble de valeurs quelconques.

La valeur d'un arc est obtenue par l'opération :

valeurArc : Graphe × Nœud × Nœud → £

#### 9.4.1 Implémentation en JAVA

L'interface Graphe donne les opérations du type abstrait Graphe

```
public interface Graphe {  
    public int ordre ()  
    public boolean arc (Nœud s1, Nœud s2) ;  
    public boolean  
    public int demiDegréInt (Nœud s) ;
```

```

    public int demiDegréExt (Nœud s) ;
    public int degré (Nœud s) ;
    public Nœud ièmesucc (Nœud s, int i) ;
    public void ajouterNœud (Nœud s) ; throws NœudException ;
    public void supprimerNœud (Nœud s) ; throws NœudException ;
    public void ajouterarc (Nœud s1, Nœud s2) ; throws ArcException ;
    public void supprimerArc (Nœud s1, Nœud s2) ; throws ArcException ;
    public Enumeration grapheEnumeration () ;
    public Enumeration NœudsAdjacents (Somme s) ;
}

```

*GrapheEnumeration* retourne l'énumération de tous les Nœuds du graphe.

*NœudsAdjacents* retourne l'énumération de tous les successeurs d'un Nœud passé en paramètre.

Un graphe est implanté classiquement soit par une matrice d'adjacence, soit par des listes d'adjacence. Le choix de la représentation d'un graphe est guidé par sa densité, mais aussi par les opérations qui sont appliquées. D'une façon générale, plus le graphe est dense (beaucoup d'arcs), plus la matrice d'adjacence conviendra. Au contraire, pour des graphes creux (peu d'arcs), les listes d'adjacences sont plus adaptées. Nous nous intéressons au cas des graphes denses.

### **Matrice d'adjacence**

Une matrice d'adjacence  $n \times n$ , représentant un graphe à  $n$  Nœuds, possède des éléments booléens tels que  $\mathbf{m[i,j]}$  = vrai s'il existe un arc entre  $i$  et  $j$  et faux sinon. Avec cette représentation, la complexité en espace mémoire est  $O(n^2)$ .

	s1	s2	s3	s4	s5	s6	s7	s8	s9
s1	f	v	f	v	f	f	f	f	v
s2	f	f	f	f	f	f	f	f	f
s3	f	f	f	v	f	f	f	f	f
s4	f	f	v	f	f	v	f	f	f
s5	f	f	f	v	f	f	f	f	f
s6	f	f	f	f	v	f	f	f	f
s7	f	f	f	f	f	f	f	v	f
s8	f	f	f	f	f	f	f	v	f
s3	f	f	f	f	f	f	f	f	f

Une classe GrapheMatrice qui implante l'interface Graphe peut utiliser les déclarations :

```
protected int nbNœuds ; // ordre du graphe
```

```
protected boolean [][] matI ;
```

L'ensemble Nœud peut être quelconque, mais l'implantation doit nécessairement offrir une bijection entre le type des indices de la matrice et le type Nœud. Ainsi les fonctions suivantes doivent être définies :

```
Numéro      : Nœud  → int
```

```
Nœud       : int   → Nœud
```

La fonction numéro retourne le numéro de l'indice d'un Nœud dans la matrice d'adjacence, et la fonction Nœud est sa réciproque.

Pour un graphe non orienté la matrice est symétrique. Pour représenter un graphe valué, on choisit une matrice dont les éléments représentent la valeur de l'arc entre deux Nœuds.

L'utilisation de matrice d'adjacence est commode pour tester l'existence d'un arc entre deux Nœuds. La complexité de ces opérations est  $O(1)$ .

```
public boolean arc (Nœud s1, Nœud s2) {
    return matI [numéro (s1)] [numéro (s2)]
}
```

En revanche, le calcul du ième successeur d'un Nœud, ou celui de son demi-degré intérieur ou extérieur, nécessite n tests quel que soit le nombre de successeurs du Nœud. La complexité est  $O(n)$ .

```

public int demiDegréInt (Nœud s) {
    int nbDegrésInt = 0 ;
    for (int i = 0 ; i < nbNœuds ; i++)
        if (matI[i] [numéro(s)]) nbDegrésInt++ ;
    return nbDegrésInt;
}

public int demiDegréExt (Nœud s) {
    int nbDegrésExt = 0 ;
    for (int i = 0 ; i < nbNœuds ; i++)
        if (matI[i] [numéro(s)] [i]) nbDegrésExt++ ;
    return nbDegrésExt ;
}

public Nœud ième Succ (Nœud s, int i) {
    if (i <= 0) throw new NœudException () ;
    int k = 0 ;
    do {
        if (k == nbNœuds) throw new NœudException () ;
        if (matI[s.numéro()] [k++]) i-- ;
    } while (i != 0) ;
    // k est le numéro du ième successeur du Nœud s
    return Nœud (k-1) ;
}

```

La méthode d'énumération des successeurs d'un Nœud donne :

```

public Enumeration NœudsAdjacents (Nœud s) {
    Return new NœudsAdjacentsEnumeration (s) ;
}

```

La classe *NœudsAdjacentsEnumeration* est une classe privée locale à *GrapheMatrice*. Son constructeur fabrique une liste de successeurs, et les méthodes *élément-Suivant* et *finEnumeration* permettent son énumération par réutilisation de l'énumération de liste.

```

private class NœudsAdjacentsEnumeration implements Enumeration
{
    private Enumeration enumNœuds ;
    public NœudsAdjacentsEnumeration (Nœud s) {
        // construire la liste des successeurs de s
        Liste listeSom = new Listechainée2 (Nœud.class);
        int i = 0 ;
        do
            if matI[numero(s)] [i])
                listeSom.ajouter(listeSom.longueur() + 1, Nœud(i)) ;
        while (++i < nbNœuds) ;
        enumNœuds = listeSom.listeEnumeration () ;
    }
    public boolean finEnumeration () {
        return enumNœuds.finEnumeration () ;
    }
    public Object élémentSuivant () throws FinEnumerationException
    {
        if (enumNœuds.finEnumeration() )
            throws new FinEnumerationException ()
        return enumNœuds.elementSuivant () ;
    }
} // fin classe NœudsAdjacentsEnumération.

```

#### 9.4.2 Parcours du Graphe

Il existe deux types classiques de parcours de graphe. Le parcours en profondeur et le parcours en largeur. Ces deux types de parcours font un parcours complet du graphe. Dans notre cas où l'affichage des premières couronnes autour de l'élément actif est demandé, le parcours en largeur semble le mieux indiqué.

- **Le Parcours en largeur :**

On appelle « distance » la longueur du chemin entre deux Nœuds d'un graphe. Le parcours en largeur d'un graphe à partir d'un Nœud origine  $s$  consiste d'abord à visiter ce Nœud, puis à traiter les Nœuds de distance avec  $S=1$  (les Nœuds de la première couronne), puis ceux de distance égale à deux, etc. L'algorithme s'écrit de façon itérative et utilise une file d'attente qui conservent les Nœuds déjà traités par ordre de distance croissante dont les successeurs sont à visiter. Les nœuds parcourus sont marqués afin de ne pas les traiter plusieurs fois. Notons qu'un parcours en largeur parcourt une composante connexe du graphe (composante dans laquelle existe un chemin entre toute paire de Nœuds). Pour un parcours complet d'un graphe à plusieurs composantes connexes on décrit l'algorithme en deux étapes :

**Algorithme** Parcours-en-largeur (G)

{ Parcours en largeur de toutes les composantes connexes du graphe G }

***pour tout* s de G faire**

***si non* marqué (s) *alors***

Plargeur (G , s)

***finsi***

***finpour***

**Algorithme** Parcours-en-largeur (G,s)

{ Parcours en largeur des successeurs du Nœud s }

mettre une marque sur s

f ← filevide

enfiler (f , s)

***tant que non* estvide ( f ) *faire***

p ← premier ( f )

défiler ( f )

***pour tout* x de G tel que  $\exists$  arc (p,x) *faire***

***si non* marqué (x) *alors***

mettre une marque sur x



enfiler (f,x)

*fin*

*finpour*

*fantanque*

La complexité du parcours en largeur est identique à celle du parcours en profondeur, quelle que soit la représentation choisie matrice d'adjacence ou listes d'adjacence.

- **Programmation en JAVA du parcours en largeur du graphe**

L'algorithme de parcours est implémenté par la méthode qui complète l'interface Graphe.

```
public void parcoursLargeur (Opération op) ;
```

Le traitement effectué sur chacun des Nœuds est contenu dans le paramètre op de type Opération. C'est une interface qui déclare l'opération générique exécuter.

```
public interface Opération {  
    public Object exécuter (Object e) ;  
}
```

Lors d'un parcours effectif d'un graphe, on transmet à la méthode de parcours un objet qui donne une implantation particulière de l'interface Opération. Par exemple, pour afficher tous les Nœuds d'un graphe sur la sortie standard, on pourra définir la classe :

```
public class OpérationAfficher implements Opération {  
    public Object exécuter (Object e) {  
        system.out.print (e) ;  
        return null ;  
    }  
}
```

Notons que l'utilisation des méthodes de parcours n'a de sens que si l'ordre de parcours des Nœuds est important. S'il s'agit d'appliquer un traitement sur chacun des Nœuds du graphe dans un ordre quelconque, il conviendra d'utiliser la méthode *grapheEnumération* qui retourne l'énumération des Nœuds du graphe.

## 9.5 Formules mathématiques pour la présentation des concepts (première et deuxième couronne)

Dans cette section, nous donnons la description axiomatique et la représentation des couronnes autour des concepts dans le graphe.

### 9.5.1 Description axiomatique :

$$\forall g \in \text{Graphe}, \forall x, y \in \text{Nœud}$$

- (1)  $x \in g \Rightarrow \neg \exists g', g' = \text{ajouterNœud}(g, x)$
- (2)  $\text{ordre}(\text{graphevide}) = 0$
- (3)  $\text{ordre}(\text{ajouterNœud}(g, x)) = \text{ordre}(g) + 1$  et  $\text{degré}(x) = d^+(x) = d^-(x) = 0$
- (4)  $x \notin g \Rightarrow \neg \exists g', g' = \text{supprimerNœud}(g, x)$
- (5)  $\text{ordre}(\text{supprimerNœud}(g, x)) = \text{ordre}(g) - 1$  et  $\text{arc}(y, x) \Rightarrow d^+(y) = d^+(y) - 1$
- (6)  $\text{degré}(x) = d^+(x) + d^-(x)$
- (7)  $\text{arc}(x, y) \Rightarrow \neg \exists g', g' = \text{ajouterArc}(g, x, y)$
- (8)  $\text{ajouterArc}(g, x, y) \Rightarrow d^+(x) = d^+(x) + 1$  et  $d^-(y) = d^-(y) + 1$
- (9)  $\text{non arc}(x, y) \Rightarrow \neg \exists g', g' = \text{supprimerArc}(g, x, y)$
- (10)  $\text{supprimerArc}(g, x, y) \Rightarrow d^+(x) = d^+(x) - 1$  et  $d^-(y) = d^-(y) - 1$
- (11)  $\forall i \in [1, d^+(g, x)], \text{arc}(g, x, \text{ièmeSucc}(g, x, i)) = \text{vrai}$
- (12)  $\forall i \in [1, d^+(g, x)], y \neq \text{ièmeSucc}(g, x, i) \Rightarrow \text{non arc}(g, x, y)$

### 9.5.2 Représentation des couronnes autour de l'élément actif.

D'abord sont positionnés les concepts de la première couronne, puis ceux de la deuxième couronne. Il y a un nombre maximal de concepts de la deuxième couronne qui sont représentés, les autres sont indiqués par un chiffre indiquant le nombre de concepts non affichés.

1<sup>ère</sup> couronne : on affiche tout

2<sup>ème</sup> couronne : on affiche un nombre maximal de concepts de l'ordre de 20.

S'il y a trop de concepts dans la 1<sup>ère</sup> couronne, on n'affiche pas la deuxième.

L'élément actif a pour coordonnées :

$$X = (W - w) / 2$$

$$Y = (H - h) / 2$$

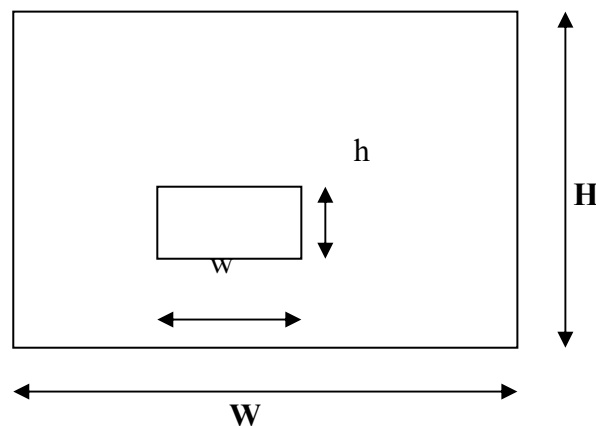


FIGURE 9.9 DISPOSITION DE L'ELEMENT ACTIF

Première couronne : élément d'angle  $\alpha$

$$X = (W-w)/2 + r1*\cos(\alpha)$$

$$Y = (H-h)/2 + r1*\sin(\alpha)$$

Deuxième couronne : élément  $i$  sur  $n$

$$X = (W-w)/2 + r2*\cos(i*2*\pi/n)$$

$$Y = (H-h)/2 + r2*\sin(i*2*\pi/n)$$

$\alpha =$  moyenne des  $(i*2*\pi/n)$  des éléments de la deuxième couronne qui lui sont associés.

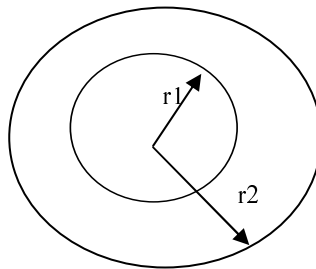


FIGURE 9.10 DISPOSITION DES COURONNES AUTOUR DE L'ÉLÉMENT ACTIF

### 9.5.3 Méthode de disposition des concepts afin d'éviter la superposition des concepts et des flèches

Nous utilisons un algorithme de type champ de force (cas d'un aimant pour positionnement automatique) inspiré du système des Topics Map. Le principe de l'algorithme est que chaque concept attire les concepts qui lui sont liés et qui lui sont éloignés et il repousse les autres, c'est-à-dire les concepts qui ne sont pas liés et les concepts qui sont liés mais trop proches.

Pour éviter la superposition des concepts il y a une distance idéale entre deux concepts suivant s'ils sont liés et suivant la couronne à laquelle ils appartiennent. Chaque concept essaye d'être à distance idéale de tous les autres. Si les concepts ne sont pas liés directement, la distance idéale c'est l'infini (ils s'éloignent infiniment). Entre le concept central et un autre de la première couronne, il y a une distance moyenne. Entre deux concepts de la même couronne, il y a une distance courte. Plus de précisions sont données dans la source de l'algorithme.

Pour les flèches : pour éviter des calculs compliqués, les flèches sont disposées au milieu des arcs. On utilise les fonctionnalités de rotation pour que les flèches pointent dans la bonne direction.

A l'initialisation de l'algorithme tous les concepts sont mis au centre avec un paramètre aléatoire pour éviter la division par zéro. Par "cliquer et tirer" on peut déplacer un concept au cas où la configuration n'est pas stable. Cet algorithme implique que le graphe soit connexe (entre deux concepts on trouve toujours un chemin).

## **9.6 Conclusion**

Dans ce chapitre, nous nous sommes intéressé aux aspects technologiques de l'outil Towntology. D'abord nous avons décrit l'architecture du système Towntology en le comparant avec d'autres outils de construction d'ontologie. Ensuite, nous avons présenté les outils ou principales interfaces utilisateurs qui donnent accès aux différents services. Après, nous avons parlé des modules du système en décrivant les fichiers et classes appartenant aux différentes applications. Et enfin, nous avons donné les structures de données internes qui ont permis de coder notre ontologie sous forme de graphe, et les formules mathématiques pour la représentation des nœuds (concepts) sur différentes couronnes.

## **Chapitre 10. Expérimentation**

Dans ce chapitre, nous relatons nos travaux et expériences avec les experts du laboratoire EDU dans le cadre du projet Towntology. Nous insistons sur les difficultés rencontrées, les résultats obtenus, et l'apport de l'outil Towntology.

Nous avons démarré le projet en envoyant des courriers aux services techniques des villes de plus de 100000 habitants, dans l'espoir qu'ils disposent déjà d'ontologies dans le domaine de l'urbain. Sur 25 courriers, nous avons eu 5 réponses, toutes négatives concernant l'existence d'une ontologie urbaine. A chaque réponse étaient toutefois joints le nom d'une personne prête à nous aider dans notre recherche, ainsi que des documents sur l'organisation de leurs services et sur les systèmes d'information géographique des différentes villes [BEAU03]. Cette source d'information a été peu utilisée. En effet, notre travail n'en était pas encore au stade où nous aurions pu intégrer les documents fournis.

La première année du projet, en 2003, il a été décidé de travailler sur un domaine concret, la voirie. Ensuite la deuxième année, nous nous sommes intéressés au domaine de la mobilité et transports.

### **10.1 Conception de l'ontologie de la Voirie [BEAU03]**

Nous rappelons que notre but était de réaliser une maquette et non d'être exhaustif dans le recensement des termes. Comme dans [AUSS03], nous avons commencé par recenser les objets techniques de la voirie, avec l'ouvrage de Chantal Berdier : Le dictionnaire de la voirie [BERD98]. Nous avons choisi ce point d'entrée car il nous paraissait être le plus accessible pour démarrer notre travail. En effet, en s'intéressant aux objets et non aux notions complexes du domaine, nous étions sûrs de rester sur des notions concrètes et ainsi d'éviter les termes complexes. Ce premier travail sur les objets techniques nous a permis de démarrer sur des bases saines et intelligibles.

Dans un premier temps, nous avons établi un certain nombre de relations entre les termes qui nous paraissaient les plus évidentes. Puis nous avons intégré l'ensemble des termes du dictionnaire de la voirie, à l'exception de ceux qui nous semblaient trop éloignés du domaine de la voirie (termes arboricoles,

informatiques,...). Afin de mettre en relation les termes à intégrer nous avons dû en ajouter d'autres, absents de l'ouvrage, que nous avons définis en interrogeant le grand dictionnaire terminologique (*GDT*).

Après nous nous sommes intéressé à l'analyse de termes plus compliqués comme "espace public", et "mixité" [BEAU03], ... En effet ces concepts sont souvent enclins à des dérives sémantiques parce qu'ils sont plus difficiles à identifier et à définir clairement. Nous avons parfois, pour un même terme, différentes définitions, souvent contradictoires ou correspondant à des champs sémantiques différents.

La première année déjà, plus de 800 termes ont été définis, ce qui veut dire que nous avons plus de 800 concepts (langue française) dans le projet *Towntology*. Pour expliquer la sémantique des concepts nous avons besoin de relations. Ainsi 21 types de relation entre les concepts sont déterminés (voir tableau 9.1). Ces types de relation sont utilisés pour établir plusieurs centaines de relations entre nos 800 concepts. Pour spécifier ces relations, nous les avons représentées visuellement dans un réseau sémantique en indiquant le nom de la relation qui lie les deux concepts. Nous notons que nos experts n'ont pas utilisés beaucoup de relations topologiques. Ce réseau sera continuellement enrichi avec de nouvelles relations entre les concepts.

TABLEAU 9.1 LISTE DES TYPES DE RELATION [BEAU03]

<b>Relation de subsomption ou d'héritage</b>	
Est un	Relation utilisée pour faire le lien entre des concepts spécifiques et leur concept plus généraliste.
Est un sous-ensemble de	Relation regroupant les relations « est un » et « est composé de »
<b>Relations de composition ou d'agrégation</b>	
A pour matériaux	Relation utilisée pour définir le(s) matériau(x) des objets de la voirie, de construction manufacturée, etc.
Est composé de	Relation utilisée pour décrire la composition horizontale, verticale et structurelle
<b>Relation de typage et de structuration</b> (définition des attributs ou définition des types d'un objet)	
Caractérise	Relation définissant les paramètres ou spécificités caractérisant un objet, matériau etc..
<b>Relations topologiques</b>	

Est situé sur	Relation de localisation positionnant un objet par rapport à un autre
Est situé dans	
<b>Relation de causalité</b>	
Est issu de	Relation indiquant la provenance d'un objet, d'un concept ou d'une procédure
<b>Relation d'usage terminologique</b>	
Se dit pour	Relation utilisée pour lier des termes ou expressions spécifiques à un objet ou concept
<b>Relation de dépendance</b>	
Travaille pour	Relation indiquant quel corps de métier travaille pour quel autre
Dépend de	Relation synonyme de « est conditionné par » ou « est la conséquence de ». Cela peut également être une relation entre une procédure et ses décideurs ou une autre procédure, pouvant être caractérisée par la relation « décidé par ».
Est un problème touchant	Relation indiquant quel support est touché par quel dysfonctionnement
<b>Autres relations</b>	
Est un outil pour	Relation générique reliant un concept à un outil
Est un outil (matériel) pour	Relation utilisée pour définir quel moyen matériel est employé à quelle fin ou sur quel support.
Est un outil (étude) pour	Relation liant un outil d'étude (instrument théorique ou procédé) et le domaine pour lequel il est utilisé
Est un outil (informatique) pour.	Relation liant un programme ou logiciel à son domaine d'utilisation
S'oppose à	Relation liant deux concepts qui s'opposent de façon symétrique (flèche bidirectionnelle).
Est utilisé pour	Relation indiquant qu'un objet (et non un outil – étude ou matériel) est employé pour une opération ou fonction particulière
Est une procédure concernant	Relation entre une procédure (ou démarche intellectuelle) et son domaine d'application
Est une opération pour	Relation liant une opération (travaux, traitement, ...) et l'objet ou matériau sur lequel elle agit
Est un document pour	Relation liant un document à son domaine d'application
A une activité concernant	Relation permettant de définir le domaine d'activité d'une personne, d'un groupement de personnes ou d'une science



## **10.2 Conception de l'ontologie de la Mobilité et Transports [BERT04]**

Après la voirie, l'équipe du projet s'est intéressé la deuxième année au domaine de la mobilité et transports. L'objectif à ce niveau était d'intégrer les concepts plus abstraits et plus flous concernant ce domaine à l'ontologie déjà existante. Cela a été fait en trois phases par :

- La construction d'un échantillon de concepts relatifs à la mobilité et aux déplacements,
- La recherche d'un ensemble de définitions associées,
- L'intégration dans le réseau sémantique.

Cette fois-ci pour obtenir un échantillon des termes et concepts les plus caractéristiques du domaine, nous avons choisi d'adresser des questionnaires aux spécialistes du domaine (doctorants, professeurs, chercheurs) du département GCU (Génie Civil et Urbanisme).

La seconde phase a consisté à identifier un ensemble de définitions pour les concepts rassemblés. Certains des concepts avaient, dès le début, plusieurs définitions tandis que d'autres au contraire n'avaient aucune de précise. Il était ainsi nécessaire de mener à bien des travaux de filtrage et de définition. Puis chacune de ces définitions a été organisée selon la structure de la base existante, c'est-à-dire classée selon sa spécialisation et son champ d'activité.

L'organisation des concepts et des définitions dans un réseau sémantique a commencé en s'appuyant sur les relations définies dans la base de la voirie. Il était nécessaire de simplifier l'utilisation de ces relations dans le réseau sémantique. En même temps il fallait les classifier pour essayer d'enlever des ambiguïtés, mais également les généraliser, pour faciliter leur réutilisation et éviter leurs multiplications. Pour pouvoir intégrer ce champ dans le modèle existant, il était alors nécessaire de choisir un certain nombre de concepts passerelles permettant de relier le nouveau réseau sémantique à celui existant.

### **10.3 Les résultats**

La première année, les travaux de nos experts ont abouti à la construction d'un hypertexte terminologique contenant pour chaque terme l'ensemble des définitions trouvées dans la documentation. Pour expliciter les liens entre les termes, une série de fichier power point (un fichier par type de relations) représentant les relations entre les termes ont été construits. La deuxième année l'équipe du Développement Urbain a travaillé à la construction d'une ontologie en utilisant l'outil towntology.

Ainsi la première année, l'ontologie de la voirie contenait 800 concepts en langue française et 21 types de relations identifiées entre les concepts. Et la deuxième année, l'intégration dans la base existante de l'ontologie de la mobilité et des transports a apporté 50 concepts nouveaux et la synthétisation des relations a ramener à 12 types de relations.

### **10.4 Les difficultés rencontrées par nos experts du domaine et l'apport de l'outil Towntology**

La première difficulté rencontrée par nos experts était la définition des concepts, et la conception de liens entre concepts et termes.

La notion de concept a été masquée par l'emploi de l'outil Towntology. En effet le concept apparaît lorsque les points de vues sont associés au terme ou quand les définitions du terme sont jugées similaires. L'ajout du point de vue permet de construire à partir du même terme des concepts distincts sans forcément que l'utilisateur ait conscience du besoin de créer des concepts distincts.

La deuxième difficulté portait sur la définition de point de vue.

L'outil Towntology ne permet pas de résoudre les difficultés liées aux choix des points de vue. En effet il s'agit plus d'un problème de spécification d'ontologie . Lors de cette phase, il faut identifier précisément les différents types d'acteurs et l'ensemble des tâches qui devront être pris en compte dans la modélisation du domaine.

L'une des difficultés majeures pour nos experts a été la visualisation du réseau des relations.

Le premier intérêt de l'outil Towntology concerne les fonctionnalités de visualisation et de navigation dans le réseau des relations. Le fait de voir le voisinage du concept incluant toutes les relations qu'il entretient avec d'autres concepts facilite sa compréhension et permet de voir si une relation a été oubliée. L'outil offre trois fonctionnalités de recherche suivant les préférences de l'utilisateur : recherche dans la liste alphabétique des concepts, navigation dans le graphe, accès au graphe par les images.

Une autre difficulté a été de savoir combien de types de relations sont nécessaires. La possibilité de définir des types de relations dans l'outil towntology et d'expliquer leur signification a donc contribué à relativiser le problème de la recherche de l'équilibre entre le nombre de relations à représenter, et l'intérêt qu'elles présentent.

## **10.5 Conclusion**

Dans ce chapitre, nous avons relaté nos travaux et expériences avec les experts du laboratoire EDU dans le cadre du projet Towntology. Nous avons insisté sur les difficultés rencontrées par nos experts, les résultats obtenus, et l'apport de l'outil Towntology.

La partie suivante de ce rapport porte sur les conclusions générales de la thèse.

## **Partie 5. Conclusion générale**

Dans cette partie, nous présentons les conclusions générales de la thèse. D'abord nous tirons les conclusions de nos travaux et nous nous projetons dans l'avenir en proposant les perspectives.

## **Chapitre 11. Conclusion générale**

Dans ce chapitre nous présentons les résultats de nos travaux et les perspectives à venir.

### **11.1 Conclusions**

Dans la première partie de la thèse, nous avons présenté la problématique du sujet traité.

Dans la deuxième partie, nous avons commencé par décrire le contexte dans lequel nos travaux ont été réalisés. Ce qui nous a amené à nous intéresser à la complexité et à la diversité du domaine de l'urbanisme, qui nous ont permis d'identifier un fort besoin de coopération entre les acteurs de l'urbain. Ensuite, comme solutions aux difficultés relevées, nous avons posé des hypothèses portant sur l'utilisation de l'ontologie et d'un mécanisme de consensus en vue d'atteindre les objectifs recherchés. Enfin, nous avons présenté le projet Towntology qui a servi de cadre à nos travaux.

Dans la troisième partie, nous avons présenté un état de l'art complet sur les ontologies, en décrivant et comparant les approches, les méthodes, et les langages de construction des ontologies. Une large place dans la partie état de l'art a été consacrée aux technologies et outils de groupware en général et à leurs applications dans le domaine des ontologies.

Dans la quatrième partie, nous avons présenté nos travaux et contributions sur la conception coopérative d'ontologies pré-consensuelles. Nous avons terminé nos contributions en relatant notre expérimentation avec nos experts du laboratoire EDU (Equipe Développement Urbain), et en présentant les caractéristiques logicielles de notre prototype.

Ainsi nous pouvons affirmer que la construction des ontologies dans un environnement collaboratif a été et demeure un sujet de recherche continu et ouvert.

A la différence de la majorité des travaux et des outils existants qui concernent la création des ontologies, une fois le consensus réalisé (ontologies post-consensuelles), nous avons proposé une méthodologie qui soutient les activités du

processus de développement se situant dans les premières phases de développement d'une ontologie, lorsque le consensus n'est pas encore atteint.

Plus exactement, dans cette thèse nous avons proposé :

- Une méthodologie et une démarche permettant lors de la phase d'acquisition des connaissances à des experts du domaine (dans notre cas des urbanistes) de modéliser les différences de points de vue sur le sens des termes. Notre méthodologie fait intervenir les experts du domaine le plus longtemps possible en leur permettant d'aller vers le consensus.
- La clarification du concept "ontologie pré-consensuelle" : ce type d'ontologie se caractérise par son appartenance aux premières phases de construction des ontologies. Dans ces phases, les définitions sont principalement informelles et multiples et constituent seulement un réseau sémantique.
- Un outil visuel permettant d'éditer, de naviguer dans l'ontologie et de l'interroger. Cet outil permet d'associer des ressources multimédia aux concepts.
- Un support pour le développement collaboratif d'ontologie proposant l'ontologie créée comme ontologie d'ancrage pour le travail collaboratif.

Ce travail a répondu à la question d'atteindre le consensus en proposant une méthodologie pour créer une ontologie pré-consensuelle en partant des opinions de divers acteurs. Ainsi nous avons construit un premier prototype d'ontologie appelé ontologie pré-consensuelle. Contrairement aux ontologies post-consensuelles, les différentes définitions d'un même terme sont conservées. Comme résultat, nous avons écrit des ontologies urbaines sous une forme visuelle. Cette visualisation est un apport pour le développement collaboratif. Un format XML a été conçu et utilisé pour la description de l'ontologie.

Enfin, nous avons proposé un support collaboratif Townto-Groupware basé sur un serveur Plone qui offre tous les services pour la discussion et la construction collaborative des ontologies.

Concrètement nous avons actuellement dans notre ontologie environ 1000 concepts (langue française) et 21 types de relations entre ces concepts, et un nombre inconnu chez nos partenaires du projet COST.

Cette ontologie essaye de satisfaire au besoin de coordination et de coopération entre les acteurs de l'urbain. Elle devrait pallier les dérives sémantiques de compréhension entre les interlocuteurs et les systèmes d'information géographiques des divers intervenants dans le champ de l'urbain, permettant ainsi une meilleure coopération.

Plus concrètement, notre ontologie offre plusieurs avantages :

- Amélioration des conditions et supports pour l'enseignement de l'urbanisme ;
- Cadre de référence pour les experts, permettant l'indexation de leurs documents ;
- Aide à la recherche d'information ;
- Aide à la formation du personnel ;
- Mises à jour des différents documents d'urbanisme en adaptant le vocabulaire en fonction de l'utilisateur et en organisant ces documents en fonction des besoins des utilisateurs etc.
- Aide pour l'élaboration conjointe des documents d'urbanisme.

Ainsi, nous pouvons affirmer que l'utilisation de l'ontologie est une solution intéressante pour la coopération et la collaboration dans le domaine de l'urbanisme.

En définitive, le Projet Towntology nous a permis de sortir un peu du cadre des projets classiques. En effet, par son double caractère (l'urbain et l'informatique), et par son attrait pour la combinaison de ces deux disciplines, il nous a permis de construire les bases d'une ontologie urbaine. Grâce à ce projet, nous avons découvert les subtilités du vocabulaire dont regorge la langue française et nous avons pu nous rendre compte des difficultés rencontrées par les experts du domaine pour la mise en place d'une ontologie pour l'urbanisme et l'aménagement du territoire.

## 11.2 Perspectives

Nos travaux futurs porteront sur la possibilité d'élargissement à d'autres champs de l'Urbanisme et de la Ville, afin de les intégrer dans notre base. Une autre perspective est la possibilité d'utilisation de l'outil pour des domaines autres que l'urbanisme. La Cité des Sciences de Paris s'était intéressée à notre outil dans le but de l'utiliser pour l'apprentissage des noms d'animaux aux enfants.

Nous devons également améliorer notre prototype. A l'état actuel, le site Townto-Plone propose seulement la couche de Groupware pour la discussion et l'édition collaborative des définitions. Les modules d'analyse de Cohérence n'ont pas encore été implémentés sur Townto-Plone. La résolution des problèmes d'incohérences permettra d'aller vers une ontologie formelle.

Le site ne contient pas encore :

- L'exportation directe du fichier xml vers un document de type "contenu" Plone.
- L'exportation Document Plone vers la DTD du XML Towntology.
- Le module d'analyse de cohérence vers la formalisation et la population de l'ontologie.
- Une visualisation graphique du réseau sémantique sur l'interface Plone.

Plone est un produit sur un serveur Zope et utilise des scripts Python. Zope n'est pas compatible avec des applications Java.

Au niveau international, les participants du projet Towntology ont décidé de se joindre au projet COST qui est un projet européen dont l'objectif est la construction d'une ontologie urbaine complète incluant des termes en plusieurs langues européennes. Notre outil, déjà utilisé par les partenaires du projet COST, continuera à servir d'appui à ce projet.



## Références Bibliographiques

- [ALLE03] ALLEN M. M. "Empirical Evaluation of a Visualization Tool for Knowledge Engineering". Thesis in Computer Science. Victoria: University of Victoria, 2003, pp. 109.
- [ALLE03] ALLEN P. "Requirements Engineering in a Global Software Development: A Case Study", University of Victoria, Victoria May 2003.
- [APEL] APELON [en ligne]. Disponible sur :  
<<http://www.apelon.com/about/about.htm>> (consulté le 19 Juin 2006).
- [ARPI01] ARPIREZ J. C., CORCHO O., FERNANDEZ-LOPEZ M. and GOMEZ-PEREZ A. "WebODE : a scalable workbench for ontological engineering". In Y. Gill, M. Musen and J. Shavlik (Eds), Proceedings of the first International Conference on Knowledge Capture, K-CAP'01, Victoria, Canada, October 20-23, 2001 (New York: ACM Press), pp. 6-13.
- [AUDI95] AUDI R. The Cambridge Dictionary of Philosophy. Cambridge : Cambridge University Press, 1995.
- [AUSS00b] AUSSENAC-GILLES N., BIÉBOW B. et SZULMAN S. "Modélisation du domaine par une méthode fondée sur l'analyse de corpus". In Actes des 9<sup>e</sup> journées francophones d'Ingénierie des Connaissances IC 2000, Toulouse, France, 2000, 293 p.
- [AUSS04] AUSSENAC-GILLES N., CONDAMINES A. "Documents électroniques et constitution de ressources terminologiques ou ontologiques" in Information-Interaction-Intelligence, 2004, vol. 4, No 1.
- [AUSS00a] AUSSENAC-GILLES N. "Revisiting Ontology Design: A Methodology Based on Corpus Analysis". In Design R, Corby O (eds) 12<sup>th</sup> International Conference in Knowledge Engineering and Knowledge Management (EKAW'00), Juan-Lee-Pins, France. (Lecture Notes in Artificial Intelligence LNAI 1937) Springer-Verlag, Berlin, Germany, pp 172-188, 2000.
- [BALA03] BALAKRISHNAN H., FRANS KAASHOEK M., KARGER D., MORRIS R. and STOICA I. "Looking Up Data in P2P Systems", Communication of ACM, 2003, vol. 2, 46 p.
- [BEAU03] BEAULIEU C., TARDY Y. "Projet Towntology : construction d'une ontologie urbaine". PIRD en Urbanisme. Lyon : INSA de Lyon, 2003, pp. 48.
- [BERN01] BERNERS-LEE T., HENDLER JAMES and LASSILA ORA. "The Semantic Web". Scientific American Press, May 2001.

- [BERN96] BERNARAS A, LARESGOITI I, CORERA J. "Building and reusing ontologies for electrical network applications. In Wahster W (ed) European Conference on Artificial Intelligence (ECAI'96). Budapest, Hungary. John Wiley and Sons, Chichester, United Kingdom, pp 298-302.
- [BERT04] BERTHET C. "Projet Towntology : intégration de concepts liés à la mobilité et transports". PIRD en Urbanisme. Lyon : INSA de Lyon, 2004, pp. 64.
- [BIZ2] BIZ2Peer Technologies [en ligne]. Disponible sur : <http://www.biz2peer.com/p2p.htm> (consulté le 02 F2VRIER 2007).
- [BORS97] BORST W.N., AKKERMANS J.M. and TOP J.L. "Engineering ontologies". International Journal of Human-Computer Studies, 1997, vol. 46, pp. 365-406.
- [BRIC03] BRICKLEY D., GUHA RV. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Working Draft. [en ligne] sur : <http://www.w3.org/TR/PR-rdf-schema>, (consulté le 10 Décembre 2005)
- [CAIR03] CAIRNCROSS F., The Death of Distance 2.0: How the communications Revolution Will Change Our Lives. London: Texere Publishing, 2003.
- [CALA04] CALABRETTO S., ROUSSEY C., PINON J.M. "SyDoM : un système de recherche d'information multilingue basé sur des connaissances". In Actes de la 14ème Conférence Ingénierie de Connaissances IC'2004, Lyon, 5-7 mai 2004, pp. 103-114.
- [CAÑA04] CAÑAS A. J., HILL G., et al. "CmapTools: A Knowledge Modeling and Sharing Environment". In Proceedings of the First International Conference on Concept Mapping. Pamplona, Espagne, 2004, pp. 205-230.
- [CARM01] CARMEL E. and RITU AGARWAL. "Tactical Approaches for Alleviating Distance in Global Software Development", IEEE Software 2001, vol. 18, No 2, pp. 22-29.
- [CAVA02] CAVALLIER GEORGES: "Les enjeux de la gouvernance urbaine" [en ligne] sur : <http://www.urbanisme.equipement.gouv.fr/cdu/accueil/bibliographies/gouvernance/notes.htm#not20>, (consulté le 20 Avril 2007)
- [CHAN99] CHANDRASEKARAN B., JOHN R. JOSEPHON, and V. RiCHARD BENJAMINS. "What Are Ontologies, and Why Do We Need Them?", IEEE Intelligent Systems, Special Issue on Ontologies, 1999, vol. 14 (January/February) pp. 20-26.
- [CHAU98] CHAUDRHRI VK., FARQUHAR A., FIKES R., RICE JP. "Open Knowledge Base Connectivity 2.0.3. Technical Report. [en ligne] sur : <http://www.ai.sri.com/~okbc-2.0.3.pdf>, (consulté le 2 Novembre 2006)

- [COLE97] COLEMAN D. "Groupware: Collaborative Strategies for Corporate LANs and Intranets". San Francisco, CA: Prentice Hall PTR, 1997.
- [CORC00] CORCHO O., GOMEZ-PEREZ A. "A Roadmap to Ontology Specification Languages". In Dieng R, Corby O (eds) 12<sup>th</sup> International Conference in Knowledge Engineering and Knowledge Management (EKAW'00), Juan-Les-Pins, France. (Lecture Notes in Artificial Intelligence LNAI 1937) Springer-Verlag, Berlin, Germany, pp 80-96, 2000.
- [CRAN99] CRANFIELD S. and PURVIS MARTIN. "UML as an Ontology Modelling Language", presented at IJCAI'99 Workshop on Intelligent Information Integration, Sweden, 1999.
- [CSCW] Computer Supported Collaborative Work [en ligne]. Disponible sur: <<http://www.telekooperation.de/cscw/>> (consulté le 3 Mars 2007).
- [DEHO98] DE HOOG R. (1998). "Methodologies for building Knowledge Based Systems: Achievements and Prospects". Florida: Liebowitz J (ed) Handbook of Expert Systems, CRC Press Chapter 1, Boca Raton.
- [DEAN03] DEAN M., SCHEREIBER G. OWL Web Ontology Language Reference. W3C Working Draft. [en ligne] sur : <<http://www.w3.org/TR/owl-ref/>>, (consulté le 5 Avril 2007).
- [DESK] Desktop collaboration: project Backgrounder [en ligne]. Disponible sur: <<http://groove.net>> (consulté le 10 Janvier 2007).
- [DEVE02] DEVEDZIC V. "Understanding ontological engineering", Communications of the ACM, 2002, vol. 45 (4(April 2002)) pp. 136-144.
- [DIEN02] DIENG-KUNTZ R., RIBIÈRE M. "A viewpoint Model for cooperative Building of an Ontology". In 13<sup>th</sup> International Conference on Conceptual Structures, ICCS 2002. Springer, Berlin 2002.
- [DIEN05] DIENG-KUNTZ R., CORBY O. "Conceptual Graphs for semantic Web applications". In 13<sup>th</sup> International Conference on Conceptual Structures, ICCS 2005. Springer, Germany 2005.
- [DRUP] DRUPAL - Content Management System [en ligne]. Disponible sur: <<http://www.drupal.org/>> (consulté le 14 Juillet 2007).
- [ELLI91] ELLIS C. A., GIBBS S.J., and REIN G.L. "Groupware: some issues and experiences", Communications of the ACM, Jan 1991, vol. 34, No 1, pp. 39-58.
- [ESPI01] ESPINOSA J. A., KRAUT ROBERT E., LERCH JAVIER F. , SIAUGHTER SANDRA A, HERBSLEB JAMES D., and MOCKUS AUDRIS. "Shared mental models and coordination in large scale, distributed software development". At the 22<sup>nd</sup> Annual International Conference on Information Systems, New Orleans, Louisiana, USA, 2001.

- [EUZE95] EUZENAT J. "Building consensual knowledge bases: context and architecture". At the 2nd international conference on building and sharing very large-scale knowledge bases (KBKS), Enschede (NL), 1995.
- [EUZE96] EUZENAT J. "Corporate Memory through Cooperative Creation of Knowledge Bases and Hyperdocuments". At the 10th Knowledge Acquisition, Modeling and Management for Knowledge-based Systems Workshop (KAW'96), Banff, Canada, 1996.
- [EUZE97] EUZENAT J. "A protocol for building consensual and consistent repositories, Research Report RR-3260, INRIA," September 1997.
- [FARQ96] FARQUHAR A., FIKES R. et RICE J. "The Ontolingua Server: a Tool for Collaborative Ontology Construction" presented at 10th Knowledge Acquisition Workshop, Banff, Canada, 1996.
- [FERN02a] FERNANDEZ M., GOMEZ-PEREZ A. "Overview and analysis of methodologies for building ontologies. The Knowledge Engineering Review, 2002, vol. 17, No 2, pp 129-156.
- [FERN97] FERNANDEZ M., A. GOMEZ-PEREZ et al. "METHONTOLOGY : from ontological art towards ontological engineering". In Proceedings of the Spring Symposium Series on Ontological Engineering (AAAI'97), AAAI Press , 1997.
- [FIEL04] FIELDING J.M., SIMON J., CEUSTERS W. and SMITH B. "Ontological theory for ontological engineering: biomedical systems information integration". In Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR'04), 2-5 June 2004, Whistler, BC.
- [GANG99] GANGEMI A., PISANELLI DM., STEVE G. "An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies. Data & Knowledge Engineering, 1999, vol. 31, No 2, pp 183-220.
- [GENE04] ONTOLOGY CONSORTIUM, 2004, An Introduction to Gene Ontology [en ligne]. Disponible sur: < [www.geneontology.org](http://www.geneontology.org)> (consulté le 15 Août 2006).
- [GEOR00] GEORGAKOPOULOS D., HORNICK M. et al. "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure in Distributed and Parallel Databases". US : Springer, 2000.
- [GOLD03] GOLDSTEIN H. "Collaboration Nation", in IEEE Spectrum 49 (6), vol. 49, June 2003, pp. 49-51.
- [GOME04] GOMEZ-PEREZ A, FERNANDEZ LOPEZ M., CORCHO O. "Ontological Engineering". London : Springer, 2004, 403 p.
- [GOME02] GOMEZ-PEREZ A. (2002). A Survey on Ontology Tools, OntoWeb deliverable D1.3. [en ligne] sur : <URL :<http://ontoweb.aifb.uni-kalsruhe.de/>> (Consulté le 30 avril 2006).

- [GREN04] GRENON P. and SMITH B. "SNAP and SNAP : towards dynamic spatial ontology". *Spatial Cognition and Computation*, 2004, pp. 69-104.
- [GRUB93] GRUBER T. R. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", presented at International Workshop on Formal Ontology, Padova, Italy, 1993.
- [GRUD94] GRUDIN J. "Computer-Supported Cooperative Work: History and Focus," *IEEE Computer*, 1994, vol. 27, No 5, pp. 19-26.
- [GRUD95] GRUDIN J. "Groupware and social dynamics: eight challenges for developers". In *Communications of the ACM*, 1995, vol. 37, pp. 92-105.
- [GRUN02] GRUNINGER M., JINTAE LEE. "Ontology Applications and Design". In *Communications of the ACM*, 2002, vol. 45, No 2, pp. 39-41.
- [GUAR95] GUARINO N. G. "Ontologies and Knowledge Bases: Towards a Terminological Clarification". In *Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing*. Amsterdam: IOS Press, 1995, pp. 25-32.
- [GUAR94] GUARINO N. C., GIARETTA P. "Formalizing Ontological Commitments," presented at 12th National Conference on Artificial Intelligence, Seattle, WA, USA, 1994.
- [HAAS05] P. HAASE and L. STOJANOVIC. "Consistent evolution of OWL ontologies". In *Proceedings of the Second European Semantic Web Conference*, Heraklion, Greece, MAY 2005.
- [HERB99] HERBSLEB J. D. and REBECCA E. GRINTER. "Architectures, Coordination, and Distance: Conway's Law and Beyond", *IEEE Software* (September/October), 1999, pp. 63-70.
- [HERB00] HERBSLEB J. D., MOCKUS AUDRIS, FINHOLT THOMAS A., and GRINTER REBECCA E. "Distance, Dependencies, and Delay in a Global Collaboration", presented at 2000 ACM conference on Computer supported cooperative work, Philadelphia, Pennsylvania, United States, 2000.
- [HERB01] HERBSLEB J. D. and DEEPENDRA MOITRA. "Global Software Development," *IEEE Software*, 2001, vol. 18, No 2, pp. 16-20.
- [HERB03] HERBSLEB J. D. and AUDRIS MOCKUS. "An Empirical Study of Speed and Communication in Globally-Distributed Software Development", *IEEE Transactions on Software Engineering*, 2003, vol. 29, No 6, pp. 112-145.
- [HOLS02] HOLSAPPLE C. W., and K.D. JOSHI, "A collaborative approach to ontology design", *Communications of the ACM*, 2002, vol. 45, No 2, pp. 42-47.

- [HORR01] HORROCKS I., VAN HARMELEN F (eds) (2001). Reference Description of the DAML+OIL (March 2001) Ontology Markup Language. Technical report. [en ligne] sur : <<http://www.daml.org/2001/03/reference.html>>, (consulté le 20 Décembre 2006).
- [IEEE96] IEEE Standard for Developing Software Life Cycle Processes (1996). IEEE Std 1074-1995. IEEE Computer Society. New York.
- [IEEE90] IEEE Standard Glossary of Software Engineering Terminology. IEEE Computer Society. New York Std 610.121990.
- [ISAA02] ISAACS E., ALAN WALENDOWSKI, STEVE WHITTAKER, DIANE J. SCHIANO and CANDACE KAMM. "The Character, Functions, and Styles of Instant Messaging in the Workplace", presented at Conference on Computer Supported Cooperative Work, New Orleans, Louisiana, USA, Nov. 16-20, 2002.
- [KALF99a] KALFOGLOU Y., ROBERTSON D. (1999a). "Use of Formal Ontologies to support Error Checking in Specifications". In Fensel D, Studer R (eds) 11<sup>th</sup> European Workshop on Knowledge Acquisition, Modelling and Management (EKAW'99), Dagstuhl, Germany. (Lecture Notes in Artificial Intelligence LNAI 1621) Springer-Verlag, Berlin, Germany, pp 207-224.
- [KEIT04] KEITA A.K., LAURINI R., ROUSSEY C., ZIMMERMAN M. "Towards an Ontology for Urban Planning: The Towntology Project". In Proceedings of the 24<sup>th</sup> UDMS Symposium, Chioggia, October 27-29, 2004, pp 12.-24.
- [KEIT06] KEITA A. K., C. ROUSSEY, R LAURINI. "Un outil d'aide à la construction d'ontologies pré-consensuelles : le projet Towntology". In actes du 24<sup>ème</sup> congrès de Informatique des Organisations et Systèmes d'Information et de Décision (INFORSID), Tunis 31 Mai-4 Juin 2006, pp 911-926.
- [KIF] Knowledge Interchange Format Manual [en ligne] sur : <<http://www-ksl.stanford.edu/knowledgesharing/kif/>>, (consulté le 17 Septembre 2006).
- [LAUR01] LAURINI R. "Groupware and Public Participation in Urban Planning". Invited paper at 6<sup>th</sup> Symposium on the Role of Information Technology for and in Urban Planning, Vienna, Austria, Proceedings edited by M. Schrenk, February 14-16, 2001, pp. 17-30.
- [LAUR92] LAURINI R., THOMPSON D. (1992) [Fundamentals of Spatial Information Systems](#). Academic Press, Février 1992. 680 p.
- [LIND02] LINDVAL I. R. MIKAEL, SINHA SACHIN SUMAN. "Technology Support for Knowledge Management," presented at Fourth Workshop on Learning Software organizations (LSO), Chicago, Illinois, USA, 2002.
- [LÓPE99] LÓPEZ M. F. "Overview Of Methodologies For Building Ontologies", presented at IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, 1999.

- [MAED01] MAEDCHE A. and STAAB S. "Ontology learning for the semantic Web". IEEE Intelligent Systems, 2001, vol.16, pp. 72-79
- [MAED00b] MAEDCHE A., STAAB S. (2000b). "Mining Ontologies from texts". In Dieng R, Corby O (eds) 12<sup>th</sup> International Conference in Knowledge Engineering and Knowledge Management (EKAW'00), Juan-Les-Pins, France. (Lecture Notes in Artificial Intelligence LNAI 1937) Springer-Verlag, Berlin, Germany, pp 189-202.
- [MAN00] MANN W.R. "The discovery of things: Aristotle's Categories and their Context. Princeton, NJ: Princeton University Press.
- [MARK01] MARK D.M., SKUPIN A. and SMITH B. "Features, objects and other things: ontological distinctions in the geographic domain". In D.R. Montello (Ed.), Proceedings of Conference on Spatial Information Theory: Foundations of Geographic Information Science, COSIT'01, 2001, pp. 488-502 (Berlin: Springer).
- [McGU00] McGUINNESS D. L. "Conceptual Modeling for Distributed Ontology Environments", presented at Eighth International Conference on Conceptual Structures Logical, Linguistic, and Computational Issues, Darmstadt, Germany, 2000.
- [McGU00] McGUINNESS D. L., FIKES RICHARD, RICE JAMES, WILDER STEVE. "The Chimaera Ontology Environment" presented at the Seventeenth National Conference on Artificial Intelligence (AAAI 2000). Austin, Texas, US, 2000.
- [MIZO95] MIZOGUCHI R., VANWELKENHUYSEN J., IKEDA M. (1995). "Task Ontology for reuse of problem solving knowledge ". In Mars N (ed) Towards Very Large Knowledge Bases: Knowledge Building Sharing (KBKS'95). University of Twente, Enschede, The Netherlands. Amsterdam, The Netherlands: IOS Press, 1995, pp. 46-57.
- [NATA97] NATALYA M. S., NOY FRIDMAN. "The State of the Art in Ontology Design". AI Magazine, 1997, vol. 18, No 3, pp. 53-74.
- [NATA01] NATALYA M. S., NOY F., DECKER STEFAN, CRUBEZY MONICA, FERGESON RAY W., MUSEN MARK A. "Creating Semantic Web Contents with Protégé-2000", IEEE Intelligent Systems, 2001, vol. 16, No 2, pp. 60-71.
- [NECH91] NECHES R., FIKES R.E., FININ T., GRUBER T., SENATOR T., SWARTOUT W.R. "Enabling Technology for Knowledge Sharing". AI Magazine, 1991, vol. 12, No 3, pp. 36-56.
- [NOYN01] NOY N. et McGUINNESS D. L. "Ontology Development 101: A Guide to Creating Your First Ontology". Technical Report KSL-01-05Stanford: Knowledge Systems Laboratory, mars 2001.

- [NOYN00] NOY N.F., FERGESON R.W., MUSEN M. A. "The knowledge model of Protege-2000: Combining interoperability and flexibility". In Dieng R, Corby O (eds) 12<sup>th</sup> International Conference in Knowledge Engineering and Knowledge Management (EKAW'00), Juan-Les-Pins, France. (Lecture Notes in Artificial Intelligence LNAI 1937) Springer-Verlag, Berlin, Germany, pp 189-202.
- [ONTO] Ontoprise [en ligne] sur:  
<[http://www.ontoprise.de/home\\_en.htm](http://www.ontoprise.de/home_en.htm),<http://www.mw.com/cgi-bin/dictionary>>, (consulté le 25 Avril 2006).
- [OED03] Oxford English Dictionary. [en ligne] sur internet <URL:  
<http://dictionary.oed.com/>. (Consulté le 20 janvier 2006)
- [PAUL01] PAULY M. "Logic for Social software". Thèse en Informatique. Amsterdam, Pays Bas : Institute for Logic, Language and Computation Universiteit van Amsterdam Plantage Muidergracht, 2001, 210p.
- [PEER] Peer-to-Peer Working Group [en ligne] sur:  
<<http://www.peer-to-peerwg.org/whatis/index.html>>, (consulté le 27 Janvier 2007).
- [PERR03] PERRIN D. "Prompt-Viz: A Visualization Tool for Exploring the Differences Between Ontology Versions". Thesis in Computer Science. Victoria: University of Victoria, 2003, pp. 102.
- [PETE95] PETER T. G., KARP D. "The Generic Frame Protocol", presented at 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, 1995.
- [POLI93] POLI R. "HUSSELER' conception of formal ontology. History and philosophy of Logic", 1993, vol. 14, pp. 1-14.
- [PROT] Protégé project, Stanford University [en ligne] sur :  
<<http://protege.stanford.edu/index.html>>, (consulté le 21 Avril 2006).
- [RAUB01] RAUBAL M. "Ontology and epistemology for agent-based wayfinding simulation". International Journal of Geographic Information Science, 2005, 19, pp. 508-522.
- [REBE01] REBECCA J. D. H., GRINTER E., PERRY DEWAYNE E. "The Geography of Coordination: Dealing with Distance in R&D Work", presented at 23rd international conference on Software Engineering, Toronto, Ontario, Canada, 2001.
- [RICA02] RICARDO G. G., FALBO A., DUARTE KATIA C., CANDIDA ANNA, NATALI C. "Developing Software for and with Reuse: An Ontological Approach", presented at ACIS International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications, Foz do Iguaçu, Brazil, 2002.
- [ROUS04] ROUSSEY C., LAURINI R., BEAULIEU C., TARDY Y., ZIMMERMANN M. "Le projet Towntology : Un retour d'expérience pour la construction d'une ontologie urbaine". In Revue Internationale de Géomatique, France 2004, pp. 98-120 ;



- [ROUS01] ROUSSEY C. "Une méthode d'indexation sémantique adaptée aux corpus multilingues". Thèse en Informatique. Lyon : INSA de Lyon, 2001, 197p.
- [SASS00] SASSEN SAKIA. "Globalisation and telecommunications ". In Urban Forum, Wits Press, South Africa, 2000
- [SCHL03] S. SCHLOBACH and R. CORNET. "Non-standard reasoning services for the debugging of description logic terminologies". In Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI'03, Acapulco, Mexico, 2003. Morgan Kaufmann.
- [SHAW89] SHAW M. L. et GAINES B.R. "Comparing conceptual structures: consensus, conflict, correspondence and contrast" in Knowledge Acquisition, 1989, vol. 1.
- [SMIT98b] SMITH B. "Basic concepts of formal ontology". In N. Guarino (Ed.), Formal Ontology in Information Systems. Amsterdam: IOS Press, 1998, pp. 19-28.
- [SMIT89] SMITH B. "Logic and formal ontology". In J.N. Mohanty and W. McKenna (Eds), Husserl's Phenomenology: A textbook (Lanham: University Press of America), 1989, pp. 29-67.
- [SOWA00] SOWA J. F. "Knowledge Representation: Logical, Philosophical and Computational Foundations". Brooks Cole Publishing Co., Pacific Grove, California. [en ligne] sur internet : <http://users.bestweb.net/~sowa/krbook/index.html> , (consulté le 20 Janvier 2006).
- [STOR97] STOREY M. A., WONG K., FRACCHIA F.D. and MULLER H.A. "On Integrating Visualization Techniques for Effective Software Exploration", presented at InfoVis'1997, Phoenix, USA, 1997.
- [STOR01] STOREY M. M. MARGARET-ANNE, SILVA JOHN, and BEST N. E. CASEY, FERGESSEN RAY, NOY NATASHA, "Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé", presented at Workshop on Interactive Tools for Knowledge Capture, K-CAP-2001, Victoria, BC, Canada, 2001.
- [STUD98] STUDER R., BENJAMINS V.R., FENSEL D. "Knowledge Engineering: Principles and Methods". IEEE Transactions on Data and Knowledge Engineering, 1998, vol. 25(162), pp. 161-197.
- [SURE02a] SURE Y., ERDMANN M., ANGELE J., et al. "OntoEdit: Collaborative ontology development for the semantic web", presented at International Semantic Web Conference 2002 (ISWC 2002), Sardinia, Italy, 2002.
- [SUTT97] SUTTER : Dictionnaire encyclopédique de l'information et de la documentation. Paris : Nathan Université, 1997.
- [SWAR] SWARTOUT R. P. B., KNIGHT K., RUSS T. "Toward Distributed Use of Large-Scale Ontologies". Ontological Engineering, 1997, pp. 138-148.

- [TELL05] TELLER J., KEITA A., ROUSSEY C., LAURINI R. "Urban Ontologies for an Improved Communication in Urban Civil Engineering Projects". At International Conference on Spatial Analysis and GEOmatics, Research & Developments (SAGEO 2005), Avignon, France, June, 20th-23rd, 2005.
- [TENN98] TENNISON J. "APECKS: A tool to support living ontologies", presented at 11th Knowledge Acquisition Workshop (KAW'98), Banff, Canada, 1998.
- [TENN02] TENNISON J., O'HARA K., SHADBOLT N. "APECKS : using and evaluating a tool for ontology construction with internal and external KA support". In International Journal of Human-Computer Studies, 2002.
- [TERZ] TERZIS S. "CSCW & Groupware" [en ligne]. Disponible sur : <<http://www.cs.tcd.ie/Sotirios.Terzis/CSCW.html#COOCOL>> (consulté le 30 Décembre 2006).
- [TOVE] TOVE project publications [en ligne]. Disponible sur : <<http://www.eil.utoronto.ca/enterprisemodelling/papers/index.html>, <http://www.m-w.com/cgi-bin/dictionary>>, (consulté le 25 avril 2006).
- [TVER84] TVERSKY B. and HEMENWAY K. "Objects, parts and categories ". Journal of Experimental Psychology, 1984, vol. 113, pp. 169-193.
- [UDEL99] UDELL J. "Practical Internet Groupware". Sebastopol, CA: O'Reilly, 1999.
- [USCH98] USCHOLD M., HEALY M., WILLIAMSON K., CLARK P. and WOODS S. "Ontology reuse and application ". In N. Guarino (Ed.), Formal Ontology in Information Systems (FOIS'98) Trento, Italy, 6-8 June 1998 (Amsterdam: IOS Press).
- [USCH96a] USCHOLD M. "Building Ontologies: Towards Unified Methodology", presented at the 16th Annual conference of the British Computer Society Specialist Group on Expert Systems, Cambridge, UK, 1996.
- [USCH96b] USCHOLD M., and GRUNINGER MICHAEL. "Ontologies: Principles, Methods and Applications". Knowledge Engineering Review, 1996, vol. 11, No 2.
- [VRAN05] VRANDECIC D., PINTO S., TEMPICH C., SURE Y. "The DILIGENT knowledge processes". In Journal of Knowledge Management, May 2005, Vol. 9, No. 5.
- [WEBO] Web Ontology Language [en ligne] sur : <<http://www.w3.org/TR/2003/WD-webont-req-20030331/#onto-def>>, (consulté le 28 Juillet 2006).
- [WORD] WORDNET: A lexical database for English, Ver. 2.0, Cognitive Science Laboratory, Princeton University [en ligne] sur : <http://www.cogsci.princeton.edu/~wn>, (consulté le 23 janvier 2006).

- [XML] XML: Ontology Building: A Survey of Editing Tools XML.com [en ligne] sur : <<http://www.xml.com/pub/a/2002/11/06/ontologies.html>>, (consulté le 11 Janvier 2006).
- [YILI03] YILING L. "Roadmap for Tool Support for Collaborative Ontology Engineering" [en ligne]. Thesis in Computer Science. University of Victoria Columbia, Canada 2003, 119 p. Disponible sur : <<http://www.cs.uvic.ca/~chisel/thesis/YilingLu.pdf>> (consulté le 20 Decembre 2006)

# ANNEXES

**ANNEXE 1 : INFORMATIONS SUR LES OUTILS ET APPROCHES DE CONSTRUCTION COLLABORATIVE D'ONTOLOGIES.**

<p>Descriptif général</p> <p>Est-ce Open ?</p> <p>Est-ce Web ?</p> <p>Est-ce Web 2.0 ?</p> <p>Est-ce Web 3.0 ?</p> <p>Est-ce Web 4.0 ?</p> <p>Est-ce Web 5.0 ?</p>	<p>Non</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p>	<p>Non</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p>	<p>Non</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p>	<p>Non</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p> <p>OUI (en français) (non) (O)</p>
<p>Descriptif général</p> <p>Est-ce Open ?</p> <p>Est-ce Web ?</p> <p>Est-ce Web 2.0 ?</p> <p>Est-ce Web 3.0 ?</p> <p>Est-ce Web 4.0 ?</p> <p>Est-ce Web 5.0 ?</p>	<p>OUI (en français) (non) (O)</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p>	<p>OUI (en français) (non) (O)</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p>	<p>OUI (en français) (non) (O)</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p>	<p>OUI (en français) (non) (O)</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p>
<p>Descriptif général</p> <p>Est-ce Open ?</p> <p>Est-ce Web ?</p> <p>Est-ce Web 2.0 ?</p> <p>Est-ce Web 3.0 ?</p> <p>Est-ce Web 4.0 ?</p> <p>Est-ce Web 5.0 ?</p>	<p>OUI (en français) (non) (O)</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p>	<p>OUI (en français) (non) (O)</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p>	<p>OUI (en français) (non) (O)</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p>	<p>OUI (en français) (non) (O)</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p>
<p>Descriptif général</p> <p>Est-ce Open ?</p> <p>Est-ce Web ?</p> <p>Est-ce Web 2.0 ?</p> <p>Est-ce Web 3.0 ?</p> <p>Est-ce Web 4.0 ?</p> <p>Est-ce Web 5.0 ?</p>	<p>OUI (en français) (non) (O)</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p>	<p>OUI (en français) (non) (O)</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p>	<p>OUI (en français) (non) (O)</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p>	<p>OUI (en français) (non) (O)</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p> <p>OUI</p>

## ANNEXE 2 : Spécification XML de Towntology

```
<TOWNTOLGY>

<HEAD>
  <TITLE>.....</TITLE>
  <LANGUAGE>.....</LANGUAGE>
  <CUSTODIAN>.....</CUSTODIAN>
  <LAST_MODIF_DATE>.....</LAST_MODIF_DATE>
</HEAD>

<BODY>
  <RELATION_TYPES>
    <RELATION_TYPE ID=... ORIGINATOR=...
    INSERTION_DATE=...>
      <RELATION_NAME>.....</RELATION_NAME>
      <TERMS>
        <TERM>..</TERM>
      </TERMS>
      <RELATION_DEF>.....</RELATION_DEF>

    <RELATION_PROPERTIES SYMMETRIC=... TRANSITIVE=... MAYBEOPTIONAL
    =.../>
      </RELATION_TYPE>
    </RELATION_TYPES>

  <VIEWPOINTS>
    <VIEWPOINT ID=...>
      < VIEWPOINT_NAME>.....</VIEWPOINT_NAME>
      < VIEWPOINT_DEF>.....</ VIEWPOINT_DEF>
    </ VIEWPOINT>
  </ VIEWPOINTS>

  <CONCEPTS>
    <CONCEPT ID=...>
      <CONCEPT_NAME>.....</CONCEPT_NAME>
      <TERMS>
        <TERM>.....</TERM>
      </TERMS>
      <CONCEPT_DOMAIN ID=.../>
      <CONCEPT_DEFS>
        <CONCEPT_DEF ORIGINATOR=...
        INSERTION_DATE=...>
          <CONCEPT_DEF_SOURCE>
            <AUTHORS>
              <AUTHOR>.....</AUTHOR>
            </AUTHORS>
            <REF>.....</REF>
          </CONCEPT_DEF_SOURCE>

        <CONCEPT_DEF_TEXT>.....</CONCEPT_DEF_TEXT>
          </CONCEPT_DEF>
        </CONCEPT_DEFS>
      <MULTIMEDIA>
        <RESOURCE>
          <URL>.....</URL>
          <RESOURCE_SOURCE>
            <AUTHOR>.....</AUTHOR>
```

```
                <REF>.....</REF>
            </RESOURCE_SOURCE>

            <RESOURCE_DESCRIPTION>.....</RESOURCE_DESC
            RIPTION>
        </RESOURCE>
    </MULTIMEDIA>
</CONCEPT>
</CONCEPTS>

<RELATIONS>
    <RELATION ORIGINATOR=... INSERTION_DATE=...>
        <CONCEPT_ORIG ID=.../>
        <CONCEPT_DEST ID=.../>
        <TYPE ID=.../>
        <PROPERTIES OPTIONAL=.../>
    </RELATION>
</RELATIONS>
</BODY>

</TOWNTOLOGY>
```

## FOLIO ADMINISTRATIF

### THESE SOUTENUE DEVANT L'INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE LYON

NOM : KEITA (avec précision du nom de jeune fille, le cas échéant)	DATE de SOUTENANCE : 6 Juin 2007
Prénoms : Abdel Kader	
TITRE : Conception coopérative d'ontologies pré-consensuelles : Application au domaine de l'urbanisme.	
NATURE : Doctorat	Numéro d'ordre : 2007-ISAL-0030
Ecole doctorale : Ecole Doctorale Informatique et Information pour la Société (EDIIS)	
Spécialité : Informatique	
Cote B.I.U. - Lyon : T 50/210/19 /	et bis CLASSE :

RESUME : Une ontologie possède deux définitions différentes suivant le domaine auquel on s'intéresse, en l'occurrence la philosophie ou l'informatique :

- Du point de vue philosophique, une ontologie se définit comme la science de ce qui existe.
- Du point de vue informatique et plus particulièrement ingénierie des connaissances, la définition la plus communément admise est celle de Studer [STUD98] : "*Une ontologie est une spécification formelle, explicite d'une conceptualisation partagée*".

La majorité des travaux et des outils existants concernent la création des ontologies, une fois le consensus réalisé (que nous intituleons ontologies post-consensuelles).

Ainsi, une des étapes clés (importantes) dans le processus de développement d'une ontologie est le consensus entre les différents acteurs. Mais le problème principal est comment atteindre le consensus ?

L'une des caractéristiques essentielles des ontologies est qu'elles fournissent des données consensuelles sur un domaine donné. De nos jours, les ontologies englobent un ensemble si riche de la connaissance que leur compréhension (complète) dépasse celle de n'importe quel développeur ou concepteur seul ou même d'une petite équipe de concepteurs. Le développement d'une ontologie à grande échelle doit être le fruit d'un effort commun de plusieurs experts de domaine et concepteurs de logiciels. Ainsi l'approche collaborative pour le développement d'ontologie est la mieux adaptée pour cette tâche par rapport aux autres approches non collaboratives, vues les caractéristiques mêmes d'une ontologie.

Cependant la plupart des méthodes et méthodologies ne proposent pas de directives pour atteindre le consensus quand les ontologies sont construites de manière collaborative.

La construction des ontologies dans un environnement collaboratif a été et demeure un sujet de recherche continu et ouvert. Il existe un certain nombre d'outils pour le développement collaboratif d'ontologies. Mais aucun n'offre un support complet et adaptable pour toutes les exigences du développement collaboratif.

L'analyse des outils et environnements existants (pour le développement collaboratif des ontologies) ne vérifie pas seulement le fait que le développement collaboratif est la nature inhérente de la conception d'ontologie, mais indique aussi que le développement collaboratif d'ontologie n'est bien soutenu par aucun des outils ou environnements existants. Ceci présente un nouveau challenge pour trouver des outils mieux adaptés pour le développement collaboratif d'ontologie.

L'inexistence (en notre connaissance) d'ontologie pour l'urbanisme et l'aménagement du territoire, ainsi que la complexité et la dimension du domaine de l'urbanisme ajoute à ce challenge un intérêt particulier.

## Contributions

Au contraire de la majorité des travaux et des outils existants qui concernent la création des ontologies, une fois le consensus réalisé (ontologies post-consensuelles), nos travaux, réalisés dans le cadre du projet européen Townology, se situent dans les premières phases de développement d'une ontologie, lorsque le consensus n'est pas encore atteint. Ce travail essaye de répondre à la question d'atteindre le consensus en proposant une méthodologie pour créer une ontologie pré-consensuelle en partant des opinions de divers acteurs. Par exemple, nous insistons sur la nécessité de garder différentes définitions pour le même concept.

Plus exactement, nous proposons :

- Une méthodologie de développement d'ontologie pré-consensuelle et une démarche permettant lors de la phase d'acquisition des connaissances à des experts du domaine (dans notre cas des urbanistes) de modéliser les différences de points de vue sur le sens des termes. Notre méthodologie fait intervenir les experts du domaine le



plus longtemps possible en leur permettant d'aller vers le consensus. Nous proposons une extension de XML pour stocker notre ontologie.

- Un outil visuel permettant d'éditer, de naviguer dans l'ontologie et de l'interroger.
- Un support pour le développement collaboratif d'ontologie proposant l'ontologie créée comme ontologie d'ancrage pour le travail collaboratif.

MOTS-CLES : Ontologies, conception coopérative des ontologies, Towntology, Groupware, pré-consensus, urbanisme.

Laboratoire (s) de recherche : LIRIS

Directeurs de thèse: Robert LAURINI, Catherine ROUSSEY

Présidente de jury : Monique ZIMMERMANN

Composition du jury :

ZIMMERMANN Monique	Professeur INSA de Lyon	Présidente
BOUCELMA Omar	Professeur Université Aix-Marseille 3	Membre
LAS CASAS Giuseppe	Professeur Université de la Basilicate	Rapporteur
LAURINI Robert	Professeur l'INSA de Lyon	Directeur
LIBOUREL Thérèse	Professeur Université Montpellier II	Rapporteur
ROUSSEY Catherine	Maître de Conf. Université Lyon I	co-Directrice