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

Université Mentouri Constantine
Faculté des Sciences de L'ingénieur
Département D'informatique

N° d'Ordre : 32/ B /2010
N° de Série : 06/ Inf /2010

THÈSE

Présentée pour obtenir le diplôme de

Doctorat en Sciences

Spécialité : Informatique

et soutenue publiquement le **03 / 06 / 2010**

Par

BELHADEF HACENE

THÉME

Islam : Système d'information pour l'aide à la décision spatiale basé sur une ontologie

Devant le Jury composé de :

Prof.	Benmohamed Mohamed	Université Mentouri-Constantine	Président
Dr.	Kholladi Mohamed Khireddine	Université Mentouri-Constantine	Rapporteur
Dr.	Chaoui Allaoua	Université Mentouri-Constantine	Examinateur
Dr.	Kazar Okba	Université Mohamed Khider-Biskra	Examinateur
Dr.	Ghoualmi Nacira	Université Badji Mokhtar-Annaba	Examinateur

2010

بسم الله الرحمن الرحيم

ٱلۡحَمۡدُ لِلَّهِ رَبِّ ٱلۡعَٰلَمِينَ (٢) ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ (٣) مَٰلِكِ يَوۡمِ ٱلدِّينِ (٤) إِيَّاكَ نَعۡبُدُ وَإِيَّاكَ نَسۡتَعِينُ (٥) ٱهۡدِنَا ٱلصِّرَٰطَ ٱلۡمُسۡتَقِيمَ (٦) صِرَٰطَ ٱلَّذِينَ أَنۡعَمۡتَ عَلَيۡهِمۡ غَيۡرِ ٱلۡمَغۡضُوبِ عَلَيۡهِمۡ وَلَا ٱلضَّآلِّينَ (٧)

صدق الله العظيم

ملخص

يتضمن مشروع هذا البحث تصميم نظام لضمان وجود التوافق بين أنضمه تقليدية ويستند على انطولوجيا . المهمة التي كلفت بها هي تصميم الإطار ألمفاهيمي للانتقال بين نظام أو نظم المعلومات التقليدية التي تصممها وتنفذها المناهج التقليدية مثل نموذج ألمفاهيمي الكيان ،العلاقة ونموذج قواعد البيانات العلائقية ،نحو واحد أو أكثر من النظم القائمة على الانطولوجيا ، والفكرة هنا هي ضمان الانتقال ، بتطبيق مجموعة من القواعد تحول بين الزوج (نموذج الكيان ، نموذج علائقي) و الانطولوجيا ، مع المحافظة على مبدأ النظام التقليدي عن طريق التقليل من الخسارة القصوى في المعلومات ومعانيها . الهدف من هذا التحول هو تحقيق التوافق البيني (فهي الخطوة الثانية بعد التحول) بين نظم المعلومات العاملة في المجالات المشتركة أو الكامنة، ولكن في سياقات مختلفة، وتشترك في مجموعة من المفاهيم.

البنية ومفهوم الانطولوجيا ، يساعد على تحقيق هذه الفكرة من خلال مقارنة (مطابقة) بين اثنين من الانطولوجيا لاكتشاف واستخراج نظائر المفاهيم و بالتالي توحيد كون من الخطاب ، بإنشاء انطولوجيا موحدة . وتستخدم المفاهيم المتناظرة لصياغة الاستفسارات وإعداد أجوبة مفهومة في السياق الذي تم استخدامها فيه. مساهمتي الخاصة في هذه النقطة ،هي إنشاء طريقة مبتكرة للمطابقة المزدوجة بين اثنين من الانطولوجيا.

وقد أجريت دراسة حالة على نظام معلومات خاص بإدارة لوحات الإشهار و دلك لإظهار فائدة الانطولوجيا في مجال استكشاف وإدارة الأشياء المكانية في نظام المعلومات الخاص بصنع القرار في الحالات التي تتطلب استجابات دقيقة وحادة.
وقد تم تنفيذ هذه الانطولوجيا من خلال البرمجية "Protégé2000 " و باعتماد لغة مواصفات
OWL-DL

RESUME

Le projet de recherche de cette thèse a porté sur la conception d'un système capable d'assurer une interopérabilité entre des SI classiques en se basant sur une ontologie, donc la tâche qui m'a été affecté c'était de concevoir un cadre conceptuel de migration entre un ou plusieurs systèmes d'information classiques conçus et implémentés par des approches classiques tel que le modèle conceptuel Entité-Association et le modèle relationnel des bases de données, vers un ou plusieurs systèmes basés sur une ontologie, l'idée ici est d'assurer un passage, en appliquant donc un jeu de règle de transformation entre la paire (modèle E/A, le modèle relationnel) et l'ontologie tout en conservant le principe de fonctionnement du système classique en minimisant au maximum la perte d'informations et leurs sens. L'objectif visé par cette transformation est la réalisation de l'interopérabilité (c'est la deuxième étape après la transformation) entre des systèmes d'informations travaillant sur des domaines communs ou sous-jacents, mais dans des contextes différents et qui partagent un ensemble de concepts.

La structure et la sémantique de l'ontologie aide à réaliser cette idée en faisant un appariement ou une mise en correspondance (matching) entre deux ontologies pour détecter et extraire leurs concepts homologues et cela pour unifier par la suite leur univers de discours en créant une ontologie globale du domaine. Les concepts homologues sont utilisés pour formuler des requêtes et établir des réponses compréhensibles dans le contexte, là où elles sont utilisées. Ma contribution dans ce point, s'exprime par la création d'une nouvelle méthode de mise en correspondance bidirectionnelle entre deux ontologies.

Une étude de cas a été réalisée sur un système d'information pour la gestion des panneaux publicitaires, pour montrer l'intérêt de l'ontologie dans le domaine de la localisation et la gestion des objets géographiques dans un système d'information pour l'aide à la prise de décision face à des situations demandant des réponses précise et tranchantes. L'implémentation de l'ontologie s'est effectuée sous l'éditeur d'ontologies Protégé2000 et par le langage de spécification OWL-LD.

Mots clés : Interopérabilité, migration, mise en correspondance, ontologie, SI

ABSTRACT

The research project of this thesis focused on the conception a system able to ensure interoperability between classical information systems, based on an ontology; the task assigned to me, was to design a conceptual framework for migration between one or more classical information systems designed and implemented by classical approaches such as Entity-Relationship conceptual model and relational model of databases, towards one or more systems based on an ontology; the idea here is to ensure passage, applying a set of transformation rule between the pair (model E / A, the relational model) and ontology while preserving the operating principle of the classical system by minimizing the maximum loss of information and their meanings . The objective of this transformation is the realization of interoperability between information systems working on a common domain or underlying, but in different contexts and they share a set of concepts (this is the second step after the transformation).

The structure and the semantics of ontology helps to implement this idea by making a matching between two ontologies, for detecting and extract their counterparts concepts and to unify their universe of discourse, by creating a global domain ontology. the concepts counterparts are used to formulate queries and prepare responses understandable in the context where they are used. My contribution to this point expressed by the creation of a new bidirectional method between two ontologies.
A case study was conducted on an information system for management of publicity panels, to show the interest of the ontology in the field of location and management of spatial objects in an information system for the aid decision in situations requiring precise and sharp responses. The implementation of the ontology was done under the ontology editor Protégé2000 and the specification language OWL-DL.

Keywords: Interoperability, matching, migration, ontology, IS

Dédicace

A la mémoire de mes parents,

A ma femme et ma petite Acil,

A toute ma famille

A ma belle famille,

Et tous ceux qui m'aiment…

Remerciements

La réalisation d'une thèse de doctorat est une tranche de vie à part entière qui s'est nourrie de nombreuses et diverses influences. Que Dieu le Tout Puissant reçoive ma gratitude pour m'avoir donné la force morale et physique, la volonté et le courage pour l'achever.
Beaucoup de personnes ont participé directement ou indirectement à sa concrétisation et méritent d'être remerciées, quel que soit, le degré de leur contribution.

Je tiens à remercier mon encadreur, le Docteur Kholladi pour sa patience, conseils et encouragements. Je remercie aussi vivement les honorables membres du jury qui ont accepté d'évaluer ce travail. Je remercie, en l'occurrence, le Professeur Benmohamed Mohamed, Université Mentouri -Constantine, pour avoir accepté de présider le jury, le Docteur Chaoui Allaoua, Université Mentouri –Constantine, le Docteur Kazar Okba, Université Mohamed Khider-Biskra et le Docteur Ghoualmi Nacira, Université Badji Mokhtar-Annaba, pour avoir accepté de faire parie du jury de soutenance.

Table des matières

Table des figures

Liste des tableaux et des formules

Formules

Chapitre 1 :

Introduction générale

1. Introduction

D'une manière générale, dès que l'on se trouve placé dans une situation où diverses actions sont envisageables, il convient de décider de celle qu'il est nécessaire de choisir. C'est en ce sens que le terme de « décision » est utilisé. En théorie, une décision doit être prise en connaissance de cause, donc à partir d'informations suffisamment pertinentes pour la construire, en éliminant le plus possible les risques d'échec. La prise de décision est devenue une tâche primordiale dans n'importe quelle entreprise, surtout dans le domaine du business. Les systèmes d'aide à la décision (SAD) sont devenus la clé gagnante des entreprises: Une bonne décision implique une bonne continuité et durabilité de l'entreprise.

Historiquement, dans le domaine de la géographie, pour prendre des décisions, l'homme utilisait les cartes géographiques et les informations qu'elles contenaient. La construction des cartes passait par plusieurs étapes, de l'information obtenue par des observations faites sur le monde réel par des êtres humains qualifiés d'experts, à la représentation schématiques de ces informations et ceci afin d'offrir aux décideurs qu'ils soient des dirigeants, des experts ou de simples personnes un moyen commode pour la prise de décision.
La vision actuelle de la décision par cartes géographiques a changé avec l'apport de l'outil informatique qui a amélioré en matière de qualité et de rapidité le processus de réalisation de ces cartes. Il a permis aussi une représentation plus précise et réaliste du monde réel, mais un certain nombre de questions restent encore en suspens, notamment autour du traitement général de la vision de l'information.

Un système d'aide à la décision environnemental spatio-temporel doit permettre l'analyse de phénomènes naturels et anthropiques ; les données alimentant ces systèmes proviennent de diverses sources et moyens d'acquisition. Le nombre et le type de capteurs sont variés et hétérogènes, par suite la restitution, des informations et leurs échanges posent un vrai défi en terme d'interopérabilité. Un architecte expert de SAD doit pouvoir construire et mettre à disposition des utilisateurs des éléments d'informations en provenance soit d'observations, soit de données patrimoniales. Pour cela il faut disposer d'une chaîne de traitements depuis l'observation jusqu'à la restitution.

Le concept de Systèmes Interactifs d'Aide à la Décision (SIAD) s'est formalisé dans la littérature dans les années 70, ce sont des outils spécifiquement développés pour supporter la prise de décision. La conception de tels systèmes implique l'utilisation de techniques issues de divers domaines comme l'informatique, la recherche opérationnelle, l'intelligence artificielle, l'ingénierie logicielle, l'interaction homme-machine et les télécommunications. Les SIAD s'avèrent particulièrement utiles pour aider à trouver des solutions appropriées à des

problèmes complexes, de grande dimension et ayant des objectifs fortement dépendants des préférences de l'utilisateur.

Les Systèmes d'Information Géographique (SIG) sont aussi considérés comme des SIAD qui sont utilisés dans des domaines très divers tels que l'aménagement du territoire, le calcul d'itinéraires ou la gestion des risques naturels. Le développement d'Internet et du Web affecte également les SIG qui doivent être utilisables via ce réseau par des utilisateurs de profil très divers avec des besoins différents en termes de manipulation, de visualisation et de performances. La conception des SIG doit donc permettre la modélisation de l'ensemble des propriétés sur lesquelles le système se basera pour réaliser tâches. Les SGBD ont montré qu'en respectant l'indépendance entre le niveau conceptuel et le niveau externe, il était possible de proposer différentes vues d'une base de données. Concernant les SIG, le problème est plus complexe car il est nécessaire de prendre en compte les aspects spatiaux, les aspects temporels et les règles qui régissent l'évolution des données, l'une des solutions prometteuses à ces problèmes, est la création d'ontologies.

L'ontologie est un mot qui a fait couler beaucoup d'encre durant des siècles et qui était une énigme pour les philosophes. Durant les dernières décennies, cette idée a été importée dans le monde de l'informatique et en particulier le domaine de l'IA et l'ingénierie de connaissances. Aujourd'hui, les ontologies occupent une place pivot dans de nombreux domaines. De l'intelligence artificielle au Web sémantique, du génie logiciel à l'informatique décisionnelle.

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 : "***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. L'ontologie est généralement employée pour raisonner à propos des objets du domaine concerné après avoir modélisé un certain ensemble de connaissances relevant du domaine en question. Donc, l'ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que les relations entre ces concepts. Néanmoins, il existe une nuance entre terminologie et ontologie. Il faut savoir que dans une terminologie on s'intéresse aux mots et au sens, c.-à-d., aux distinctions entre ces

mots, tandis que dans une ontologie, on s'intéresse plutôt à la notion de concept et à leurs relations de dépendance.

Les ontologies sont actuellement omniprésentes dans de nombreux domaines tels que l'intégration de systèmes d'information, l'extraction de connaissances, la recherche d'information par le contenu, le web sémantique… Elle offrent une référence pour l'interprétation correcte des données et sont au cœur de nombreuses approches. Elles jouent aussi un rôle essentiel dans le domaine spatial. En effet, du fait de la disponibilité croissante de l'information mise à disposition des utilisateurs, de la complexité inhérente aux données géographiques et du coût élevé d'acquisition de ces données, le partage et la compréhension des données spatiales est un enjeu primordial. De cette façon les ontologies spatiales jouent aussi un rôle important pour la formalisation des données spatiales, le partage des données, l'intégration des systèmes d'information géographique.

Traditionnellement, pour chaque logiciel à construire, nous concevons une conceptualisation du domaine qui permet de définir les concepts et les relations à implémenter dans l'application. Les ontologies peuvent être utilisées pour faciliter la spécification et construire des applications réutilisables. Alors l'idée principale se résume par la mise en œuvre d'un vocabulaire commun pour spécifier les besoins pour une ou plusieurs applications.

Malheureusement, les applications développées en utilisant les méthodes traditionnelles de spécification ne permettent pas d'avoir localement les ontologies explicites. Alors, nous nous sommes intéressé à la définition des ontologies qui permettent de partager explicitement les connaissances du domaine. Dans cette thèse nous décrivons une démarche pour migrer une application traditionnel ou classique vers une autre à base d'ontologie locale, tout en gardant le principe de fonctionnement de l'application source.

Les ontologies locales sont définies, contrairement aux ontologies globales de l'entreprise, pour représenter les spécifications et la conceptualisation du domaine et des processus que le système doit implémenter. Néanmoins, les ontologies globales de l'entreprise sont difficiles à concevoir, puisqu'il est beaucoup plus difficile de modéliser le secteur des activités de l'entreprise que de modéliser une activité en particulier.

Nous nous intéressons dans notre travail au développement automatique des ontologies locales via un processus de transformation, ainsi une opération manuelle d'enrichissement sémantique pour arriver au problème de l'interopérabilité, qui est assuré par alignement ou une mise en correspondance des ontologies développées.

2. Problématique et motivation

Il y a de multiples raisons pour lesquelles la difficulté de compréhension dans un dialogue apparaît, nous empêchant de faire comprendre à l'autre ce que l'on désire exprimer. Bien souvent, nous passons par de petits dessins pour mieux exprimer les choses, ces petites représentations plus ou moins formelles permettent un accord sur l'interprétation à adopter pour nous comprendre. Pour éviter au maximum les ambiguïtés de compréhension, nous utilisons donc naturellement une convention entre les individus prenant part au dialogue, passant par un formalisme qu'il soit graphique ou linguistique. C'est dans ce dernier formalisme que les ontologies interviennent.

Aujourd'hui le partage sémantique entre les systèmes d'information est devenu un challenge dans les entreprises. En effet, ces systèmes constituent les fondements de la gestion du métier de l'entreprise, mènent les stratégies économiques et décisionnelles, et gèrent la communication avec les partenaires. Pour ces raisons, ces systèmes doivent être amenés à fonctionner ensemble pour permettre d'atteindre les objectifs visés par l'entreprise. Nous constatons que le partage sémantique représente une réelle barrière empêchant de faire interopérer et réutiliser ces systèmes à travers une architecture pratique.

Dans un système d'information ouvert incorporant plusieurs systèmes hétérogènes (dans le sens du vocabulaire utilisé), c'est-à-dire qu'ils n'ont pas forcément été conçus au même moment, ni par les mêmes organisations mais qu'ils travaillent dans des domaines proches ou identiques. Ainsi, pour qu'un système puisse découvrir et exploiter les services disponibles, il doit être capable de communiquer avec d'autres systèmes. C'est dans ce contexte que nous avons étudié l'apport de la notion d'ontologie pour améliorer l'interopérabilité dans un système ouvert. Dans notre proposition, chaque système possèdera sa propre ontologie décrivant formellement ses connaissances.

Peu d'ontologies sont actuellement actives dans les architectures des entreprises à cause de la complexité et le manque du personnel expert qualifier, dans le domaine de l'ingénierie de connaissances, malgré les points positifs apportés par la notion de l'ontologie dans les SI, surtout en ce qui concerne le problème de l'interopérabilité et la recherche d'informations.

Le but de ce travail de recherche, est de concevoir un système capable d'assurer l'interopérabilité entre des systèmes d'informations classiques, en faisant une migration vers des systèmes ontologiques (ou basés sur une ontologie), puisque seules les ontologies semblent capable pour aider à résoudre ce type de problèmes. Nous proposons donc un modèle d'expression de mappage pour exprimer le processus de migration et de transformation d'un SI. Un médiateur se basant sur un jeu de règles a été élaboré pour assurer

ce passage, qui traduit le schéma de la base de données en éléments de l'ontologie. Le processus de migration vers l'ontologie, n'assure pas le contenu sémantique dans les concepts de l'ontologie issue de l'étape de transformation, donc un module supplémentaire et complémentaire a été prévu pour enrichir l'ontologie par des méta-données exprimant l'information sémantique. Cette étape d'enrichissement n'est pas automatique, mais elle est assurée par un expert du domaine. La transformation ne représente pas l'objectif principal de la thèse, bien qu'elle puisse être utilisée dans pas mal de domaine, tel que l'intégration de données dans les SI.

L'architecture proposée est constituée de trois grands modules, étant la transformation, l'enrichissement (facultatif) ainsi un module de matching (ou mise en correspondance) entre les ontologies résultats du processus de transformation. A propos de ce dernier module, nous avons proposé une nouvelle méthode de mise en correspondance bidirectionnelle pour détecter et extraire les concepts homologues, constituant par la suite une ontologie globale du domaine partageable entre les différents SI, et cela pour unifier par la suite leur univers de discours. Les concepts homologues sont utilisés pour formuler des requêtes et établir des réponses compréhensibles dans le contexte, là où elles sont utilisées. La méthode proposée se caractérise par une vérification bidirectionnelle assurant la contrainte d'unicité des concepts homologues ainsi l'optimisation des résultats obtenus. Trois types de critères sont utilisés au cours du processus de matching, pour mesurer la similarité entre les concepts de deux ontologies, étant la mesure lexicale, mesure structurelle et mesure sémantique.

L'apport de l'ontologie dans le domaine des SI, peut être résumé par les points suivants :

- l'ontologie modélise un domaine et fournit un vocabulaire pour spécifier les besoins d'une (ou plusieurs) application(s) cible(s).
- promouvoir la réutilisation de connaissances dans plusieurs applications,
- faciliter la maintenance de logiciels grâce à une représentation explicite de l'ontologie sur laquelle ils sont basés,
- permettre à plusieurs applications d'accéder, par le partage ou l'échange, à des données communes.
- plusieurs traducteurs bi-directionnels sont développés pour faire le lien entre les structures de données propres aux applications et le format commun de l'ontologie. Ces traducteurs peuvent être utilisés pour représenter le sens de la requête et d'effectuer des inférences sur les informations décrivant le contenu des ressources (les méta-data), afin d'améliorer la qualité de la recherche.
- réduire le coût des applications multiples en leur donnant un accès à des données communes et de faciliter l'interopérabilité.

3. Plan de la thèse

La présente thèse est organisée en six chapitres :
Le premier chapitre présente une introduction générale sur le contexte du travail de ce projet de recherche, ainsi une explication de la problématique et les motivations des solutions proposés.
Le second chapitre¸ donne un aperçu général sur les principales fonctionnalités des systèmes d'information géographique (SIG), ainsi, une simple introduction aux concepts liés aux SIG est fournie. Pour cela, nous commençons par donner des définitions sur l'information géographique, puis nous entamons le principe de ces systèmes en décrivant brièvement leurs différentes fonctionnalités. Enfin, nous donnons quelques exemples de logiciels SIG, ainsi des perspectives de développement pour améliorer leur fonctionnement afin qu'ils puissent répondre aux besoins et exigences du marché. Dans le troisième chapitre, nous décrivons la pierre de base sur laquelle se base le sujet de cette thèse, c'est la notion d'ontologies, notamment les ontologies spatiales en précisant les différents points de vue sur cette notion et plus particulièrement les ontologies en ingénierie des connaissances. Ensuite, les différents éléments dont elle est constituée et les besoins auxquels elle répond sont décrits. Nous citons aussi les différentes classifications et les principales méthodologies et le processus de construction d'ontologies en précisant leur cycle de vie et nous terminons par les ontologies spatiales décrivant des entités spatiales pouvant être particulièrement importantes dans une ontologie, les entités spatiales constituent généralement des objets de référence décrits par un ensemble d'attributs descriptifs et une spatialité c'est-à-dire des attributs spatiaux et des relations spatiales. Ces dernières comprennent les relations topologiques d'inclusion, de superposition, de contiguïté, etc. ainsi que les relations de position relative (au-dessous, au-dessus, parallèle, perpendiculaire, etc.). Les attributs spatiaux incluent les dimensions des entités dans le système de référence : hauteur, aire, orientation, sens, etc., la forme des entités (lignes, point ou polygones) et le système de référence. Dés le quatrième chapitre, nous allons aborder le noyau ou le cœur de l'ontologie qui est le(s) formalisme(s) de description ou ce que nous appelons le langage de description ou de spécification, nous distinguons plusieurs types de formalismes pour représenter une ontologie, et différents langages pour chaque type, par exemple les graphes qui sont représentées par les réseaux sémantiques et ceux qui sont orienté-web comme RDF et RDF Schéma étant basés sur la syntaxe XML. Le deuxième type ce sont les logiques de description qui sont basés sur la logique du premier ordre, dans ce chapitre nous introduisons une grande partie sur les logiques de description qui jouent un rôle primordial pour la représentation sémantique dans l'ontologie en introduisant la notion de ABox et TBox. Nous parlons aussi dans ce chapitre sur le langage d'ontologie pour le web sémantique et ses différents types qui se basent sur une nuance de niveaux d'expressivité; en arrivant à la fin de ce chapitre nous présentons deux moteurs d'inférences qui sembles les plus

fiables et utilisés ainsi quelques langages d'interrogation et des environnements de développent pour les ontologies.

Maintenant pour entrer un peu dans l'histoire des systèmes d'aide à la décision (SAD), autour desquels s'inscrit le sujet de cette thèse. C'est le cinquième chapitre qui a été consacré pour cette notion, donc un SAD est un système informatique procurant une aide dans le processus de la prise de décision face à une difficulté ou à une modification de l'environnement. Il sert à résoudre un problème qui se pose à l'organisation ou à l'individu. Nous commençons ce chapitre par la présentation de la théorie de décision, définitions, classification et processus de prise de décision, sont les éléments constituant cette partie, par la suite nous décrivons la structure d'un SAD, notamment les systèmes interactifs d'aide à la décision (SIAD) et ceux qui sont intelligents et à caractère spatial. A la fin de ce chapitre nous présentons un exemple d'un SAD pour la gestion des panneaux publicitaires basé sur une ontologie, qui a été conçu au sein du service de la voirie de la ville de Constantine, et ce pour montrer l'intérêt d'ontologie dans les SAD en assurant l'interopérabilité et en facilitant la recherche.

Dans le dernier chapitre nous présentons une description de l'architecture générale de notre système, nous commençons par l'introduction des notions de l'interopérabilité et la réutilisation dans les SI, nous présentons aussi le processus de mise en correspondance (matching) et les mesures de similarités utilisées pour détecter les concepts homologues, nous venons par la suite à la description détaillées du système proposé et ses différents modules, notamment le module d'annotation sémantique qui est un module supplémentaire et complémentaire en même temps pour compléter le processus de création de l'ontologie issus du processus de transformation du modèle E/A et le modèle relationnel. Pour cela l'ensemble de règles est illustré en détail dans ce chapitre ainsi notre méthode de mise en correspondance que nous avons proposé, se caractérisant par une comparaison bidirectionnelle entre deux ontologies.

Enfin, notre thèse est achevée par une conclusion générale résumant notre contribution et décrit les perspectives d'avenir.

Chapitre 2 :

Les SIG

Le SIG évolue d'une approche orientée vers les bases de données à une approche orientée vers les connaissances.

Jack Dangermond, Président d'ESRI[1]

[1] ESRI : **E**nvironmental **S**ystems **R**esearch **I**nstitute, une société éditrice de logiciels de systèmes d'information géographique : http://www.esrifrance.fr/

2.1 Introduction

L'objectif de ce chapitre est de donner un aperçu général sur les principales fonctionnalités des systèmes d'information géographique (SIG). Ainsi, une simple introduction aux concepts liés aux SIG est fournie. Pour cela, nous commençons par donner des définitions sur l'information géographique. Ensuite, nous décrivons brièvement leurs différentes fonctionnalités. Enfin, nous donnons quelques exemples de logiciels SIG.

Des questions comme, C'est quoi un SIG ? Pourquoi un SIG et non pas un SI? Pour quels types d'applications ? Pour quels type d'utilisateurs ? Comment sont conçus ces systèmes ? Quelles sont les technologies mises en œuvre ? Comment peut-on faire communiquer ces systèmes ? Quel avenir pour les SIG ? Autant de questions auxquelles répond le présent chapitre [1].

2.2 Historique

Les S.I.G sont apparus suite à l'informatisation de la production cartographique qui avait commencé à la fin des années 60 par « Tomlinson » considéré comme le père du SIG moderne, il a fait son premier essaie sur l'installation de plantations forestières en Afrique de l'Est. En 1963 un système d'information géographique du Canada (SIGC) a été établi par l'agence de remise en valeur et d'aménagement des terres agricoles, c'était le premier SIG opérationnel de gestion des ressources terrestres qui est géré aujourd'hui par le "Environnement Canada[2] ". A partir des années 70, c'était la diffusion des SIG surtout dans le domaine civil, par exemple en 1969 le développement d'un SIG à Paris par l'atelier parisien d'urbanisme (APUR) qui a mis en place une banque de données urbaines sur Paris à partir d'informations administratives référencées à l'îlot. Dans ces années, y avait aussi une informatisation des données urbaines, l'entrée de la télédetection dans le domaine civil et l'apparition des fondations de sociétés spécialisées (Esri,Intergraph...). Dans les années 80 (fin), le développement de l'ordinateur individuel a permis la démocratisation et la diffusion des SIG à grande échelle [2]. Mais c'est à partir des années 1960 que l'informatisation de la production cartographique qui a donné lieu à l'avènement d'une nouvelle technologie, dénommée géomatique. Progressivement, les données cartographiques nécessaires à la géomatique ont dû, pour être pleinement exploitables, s'organiser en bases de données. Ainsi, l'exploitation combinée de plusieurs bases de données a conduit à la notion de système informatique capable d'en assurer la synthèse, la gestion et l'archivage. Donc ce n'est que progressivement, au cours des années 80, que la notion de Système d'Information Géographique s'est imposée comme l'objectif général de la géomatique et à la fin des années

[2] http://www.ec.gc.ca/

90, il y avait une explosion technologique en utilisant des progiciels dédiés comme **ArcInfo** qui a été utilisé au début sur des stations Unix, puisque il était vraiment inimaginable de traiter ces données sur un PC de bureau, et bien sur l'avènement de la technologie Internet a créé une révolution dans le domaine.

2.3 Information géographique

Toute information sur l'existence ou sur les caractéristiques d'un objet ou d'un phénomène repéré par sa localisation, peut être considérée comme une information géographique [3]. Donc une information géographique est une information relative à la localisation d'un objet ou d'un phénomène du monde terrestre. Ces objets sont décrits par leur nature, leur aspect, leurs relations éventuelles avec d'autres objets ou phénomènes. C'est la définition qui est adoptée par le CNIG[3] qui définit l'information géographique comme l' "*information qui est reliée à une localisation sur la Terre, exprimée par rapport à un système de référence*".

L'information géographique s'apparente à la relation entre des données descriptives d'un objet et de sa localisation géométrique sur la terre. Il y a ainsi une mise en correspondance entre deux types de données et de deux modes de représentation, le texte géographique qui est chargé de la sémantique et la carte géographique.

2.4 Descripteurs de l'information géographique

La numérisation de l'information géographique permet de décrire l'information en terme d'objets géographiques disposant d'une composante sémantique ou descriptive (*Quoi?*) et d'une composante spatiale ou géométrique (*Où?*), plus une information topologique décrivant la relation entre les objets se trouvant dans le même territoire (*Comment ?*).
La composante sémantique, thématique ou descriptive sert à décrire les informations qualitatives et quantitatives des objets géographiques. Au niveau informatique, ces informations sont des données alphanumériques représentées sous forme d'attributs et de leurs valeurs, par exemple le nom d'une route et le nombre de ses voies.

La composante spatiale ou géométrique, sert à géoréférencer un objet géographique, c'est à dire à lui donner des valeurs décrivant son aspect graphique, sa localisation, sa forme, ses limites et ses dimensions ; ces informations sont représentées sous forme de point (Ecole, Agglomération, Puit,…etc.), lignes (Route, Oued,…etc.) ou polygone (Wilaya, Commune,…etc.)
La composante topologique sert à décrire les relations entre les objets géographiques du même territoire, elles sont représentées par le mode vecteur ou raster (voir paragraphe 2.5).

[3] CNIG : Conseil national de ***l'information géographique : http://www.cnig.gouv.fr/***

Par exemple, une ville peut être vue comme un objet géographique qui possède des coordonnées traduisant sa position (information géométrique), elle se trouve sur le littoral (information topologique) et elle est décrite par son nom, sa surface et sa densité populaire (information sémantique)[1]. La Figure 2.1, illustre la représentation d'information spatiale et sémantique de deux objets géographiques se trouvant dans le département de l'Orn-France [4]

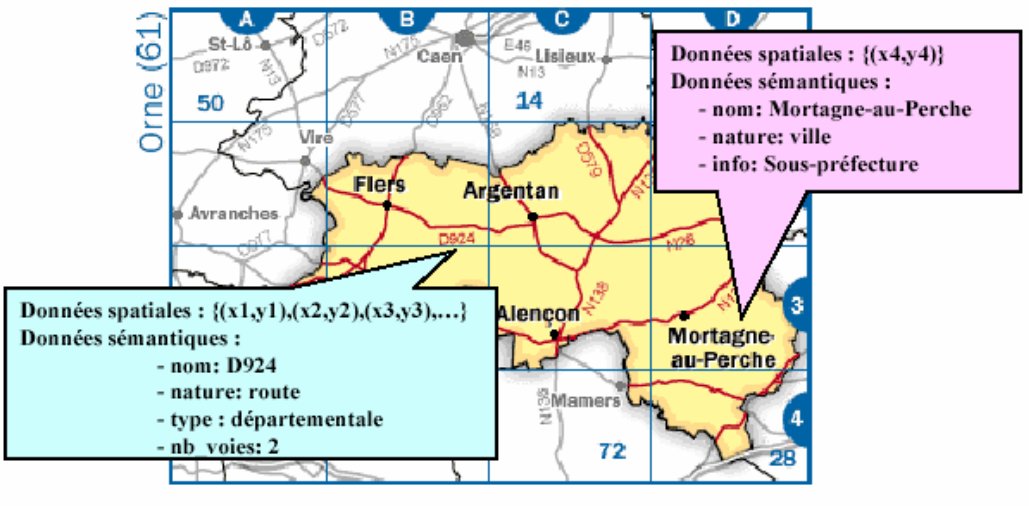
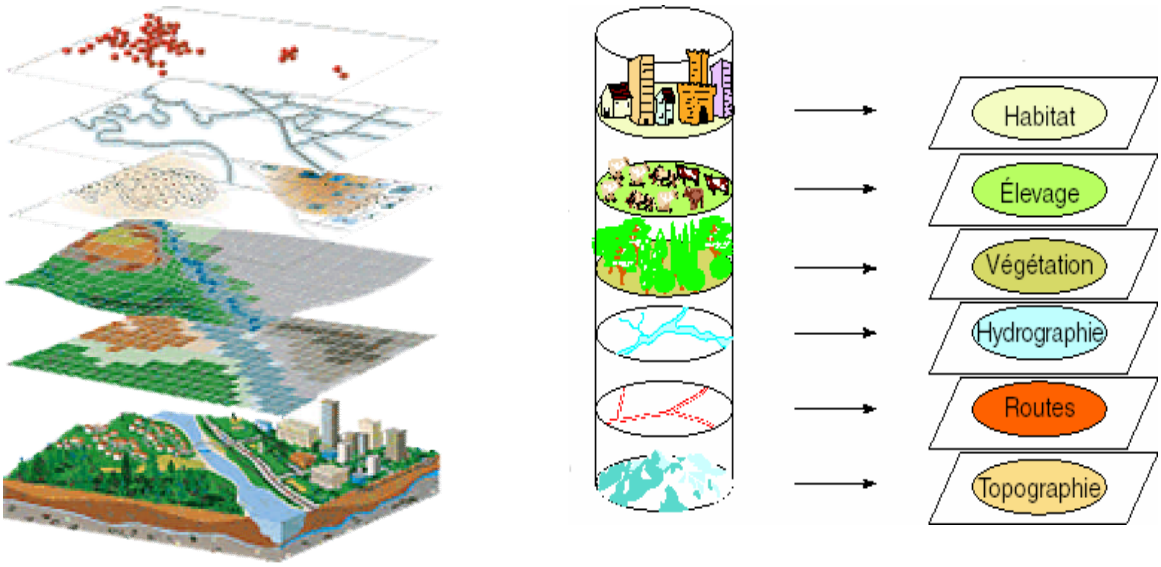


Figure 2.1 : Représentation graphique de deux objets d'une carte.

2.5 Mode de représentation de l'information géographique

L'information est dite géographique lorsqu'elle se rapporte à un ou plusieurs lieux de la surface du globe terrestre. Cette information est structurée en couches superposées, où chacune englobe les objets de même nature ou du même domaine d'intérêt (figure 2.2).



[web 1] [5]

Figure 2.2: Représentation de l'information géographique en couches

Mise à part la représentation en couche, il existe deux types (modes) pour représenter l'information géographique, le mode Vecteur et le mode Raster (Maillé), le choix d'un tel mode dépend d'un côté, de la disponibilité des données, et de l'autre côté, du niveau de détail de représentation exigé.

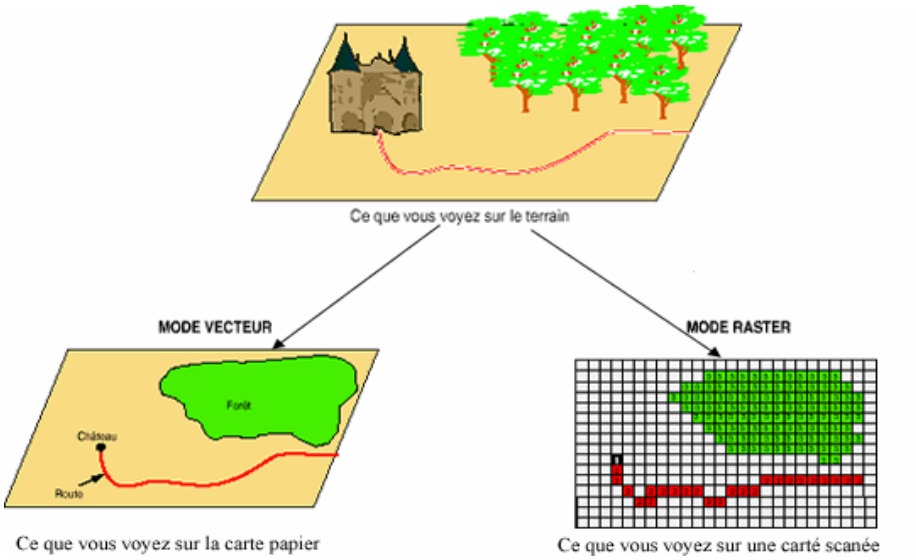


Figure 2. 3 : Représentation en mode Vecteur et Raster

2.5.1 Mode Vectoriel

Le mode vectoriel (figure 2.4) représente le contour, l'axe ou le centre d'objets géographiques. Il est couramment associé à la question "*Où se trouve tel objet géographique*?"[4]. Chaque objet géographique est défini par des données sémantiques auxquelles sont associées des données spatiales pour la localisation. Les données vecteur sont un ensemble d'objets géographiques localisés par des coordonnées (X, Y) et représentés chacun par des primitives graphiques de type: point, ligne, polygone.

Les points servent à la représentation de symboles ponctuels, ils définissent des localisations d'éléments séparés pour des phénomènes géographiques trop petits (pour être représentés par des lignes ou des surfaces) soit pour lesquels le contour est inconnu ou peu intéressant, par exemple une ville, école, arbre …etc. Les points peuvent représenter également des intersections entre lignes. Un point peut être représenté par au minimum d'une paire de coordonnées (x,y), l'altitude pouvant y être ajoutée (x,y,z).

Les lignes représentent les formes des objets géographiques trop étroits pour être décrits par des surfaces ou des objets linéaires qui ont une longueur mais pas de surface large, par

exemple les routes, rivières, voies de chemin de fer, …etc. Une ligne est représentée par une suite de points, eux-mêmes identifiés par leurs coordonnées {(x,y)} ou {(x,y,z)}.

Les polygones, servent à représenter la forme et la localisation d'objets homogènes comme des pays, des parcelles, des types de sols,…etc. Un polygone est une face délimitée par une succession de lignes fermées.

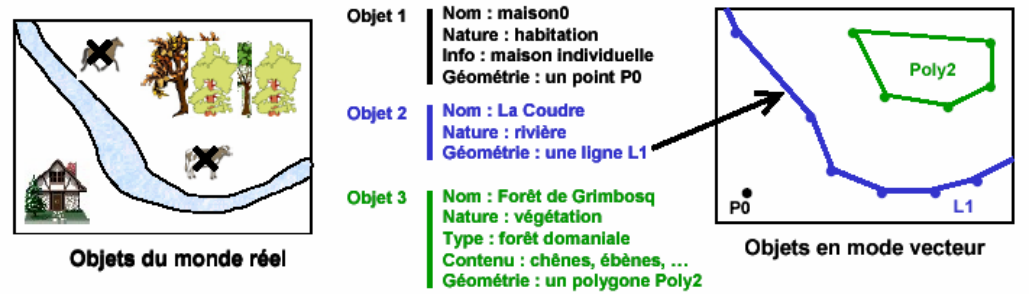


Figure 2.4 : Représentation d'objets du monde réel en mode vecteur [4]

Le mode vecteur permet de représenter des relations d'adhérence et d'inclusion pouvant exister entre les objets c'est ce qu'on appelle la **topologie.** Les relations d'adhérence permettent d'identifier des intersections entre des lignes (ou arcs) et des adjacences entre des faces (zone ou région) ou des faces et des lignes. Les relations d'inclusion s'appliquent sur les faces pour localiser une ligne, un point ou même une surface. La Figure 2.5 regroupe plusieurs relations topologiques, elle montre une intersection entre la route A et la route B au point O. Une inclusion est représentée par un étang E avec une forêt F. Une relation d'adjacence est présente entre la forêt F et la culture C. La route A est adjacente à la forêt F et à la culture C.

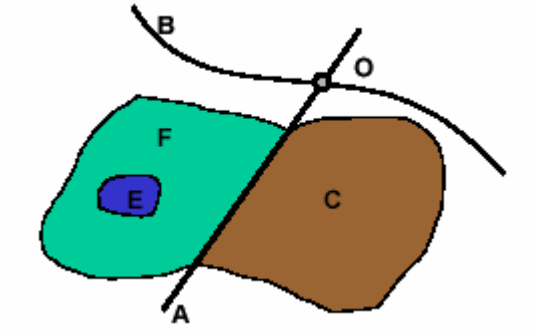


Figure 2.5 : Ensemble de points, lignes et faces décrivant des relations topologiques [4]

La topologie permet de décrire formellement une partie des relations spatiales entre des objets géographiques qui sont utilisées dans le raisonnement, par exemple : *dans, touche, longe, hors de, sur, croise, entoure, se connecte,* etc. Ainsi, sur l'exemple précédent, nous pouvons affirmer que l'étang E est dans la forêt F (inclusion), que la forêt F est à côté de la culture C,

que la route A croise la route B, etc. Par contre, il serait plus difficile d'affirmer que la forêt F est proche de la route B.

2.5.2 Mode Raster (Maillé)

La réalité est décomposée en une grille régulière et rectangulaire, organisée en lignes et en colonnes, cette réalité peut être obtenue à partir d'images satellitaires, de photographies aériennes numériques ou de modèles numériques de terrain. Le mode **maillé** (matriciel ou raster) représente cette réalité (espace) dans un maillage ou une grille à deux dimensions, où chaque cellule (maille) représente la résolution spatiale qui correspond à un pixel ayant une intensité au niveau de gris ou en couleur. La juxtaposition des points recrée l'apparence visuelle du plan et de chaque information. Par exemple, une forêt sera représentée par un ensemble de points d'intensité identique (figure 2.6).

Pour localiser un objet sur la grille, les coordonnées sont les numéros de lignes et les numéros de colonnes. Ainsi, une série de numéros de lignes et de colonnes peut représenter une géométrie d'un objet. Le mode maillé est souvent associé à la question "*Qu'est ce qui se trouve à cet endroit?*" [4] et il couvre tout le territoire. En effet, le mode maillé représente des données géométriques relatives à une portion de l'espace, auxquelles des données sémantiques sont associées. La figure 2.6 montre des objets du monde réel sous forme maillée en associant à chaque case des données sémantiques ((1) Maison individuelle, (2) Rivière et (3) Forêt domaniale).

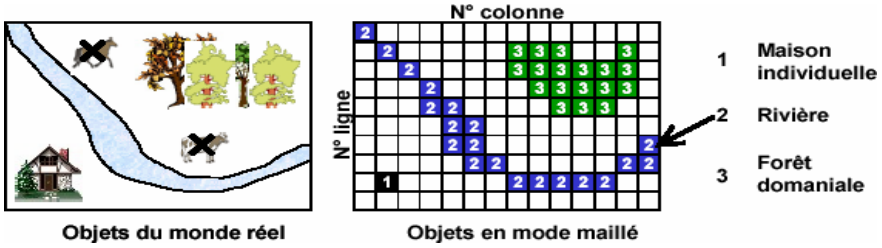


Figure 2.6 : Représentation d'objets du monde réel en mode Raster [4]

2.5.3. Relations spatiales entre objets

Les relations spatiales entre objets sont, soit de type booléen (intersection, inclusion, adjacence), soit de type flou lorsque les critères de la relation doivent être précisés (par exemple, la notion de proximité). Dans ces deux cas les relations peuvent être soit explicites (recalculées à chaque usage) ou implicites (calculées une fois et stockées). Les deux modes de relations spatiales sont utilisées entre les objets. Bien que, par nature, la description des formes issues du mode raster soit implicite, il est possible de rendre leurs relations explicites. C'est la théorie des graphes qui fournit les outils nécessaires à cette transformation [6].

2.6 Systèmes d'information géographique (SIG)

Plusieurs définitions ont été écoulées sur la notion d'un SIG, **mais toutes incluent l'ensemble de ces différentes fonctions. Donc généralement** un SIG, est un outil informatique capable de créer, saisir, transformer, afficher, analyser et stocker de l'information géographique, c'est ce que nous appelons le processus d'informatisation (figure 2.7). Ce système permet aussi d'organiser et de présenter des données alphanumériques spatialement référencées, en vue notamment de produire des plans et des cartes. C'est pour cela l'information géographique est considérée comme le noyau des systèmes d'information géographique. Si elle est bien organisée, entretenue, représentée et bien sûr gérée en permanence, dans ce cas là, elle va apporter une idéale aide à la décision.

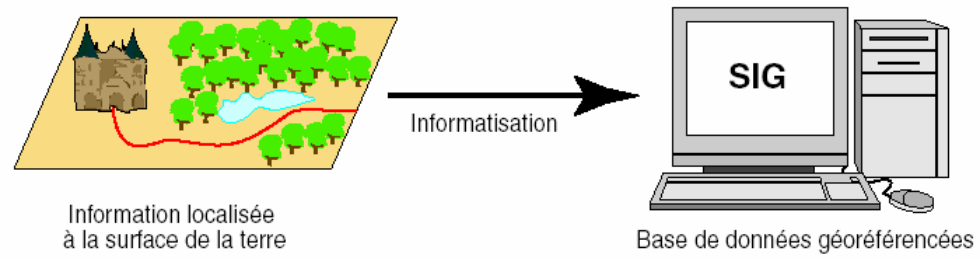


Figure 2.7 : Représentation d'un SIG

Deux autres définitions un peu précises : [web 2].
- La définition américaine émane du comité fédéral de coordination inter-agences pour la cartographie numérique (FICCDC, 1988) : Un système d'information géographique est un "système informatique de matériels, de logiciels, et de processus conçus pour permettre la collecte, la gestion, la manipulation, l'analyse, la modélisation et l'affichage de données à référence spatiale afin de résoudre des problèmes complexes d'aménagement et de gestion".

- La définition française est due à l'économiste Michel Didier (1990), dans une étude réalisée à la demande du CNIG : Un système d'information géographique est un "ensemble de données repérées dans l'espace, structuré de façon à pouvoir en extraire commodément des synthèses utiles à la décision".

Le but d'un SIG est d'offrir un système capable de répondre aux diverses questions de base sur les informations géographiques [1] [4], et ce pour aider les utilisateurs des SIG à la prise de décision face à des problèmes ou à des situations critiques :

- **Où** ? : Où se trouve un tel objet? où se produit tel phénomène, cette question permet de mettre en évidence la répartition spatiale d'un objet géographique (statique ou dynamique). Par exemple : ou se trouve la tour administrative de l'université Mentouri de Constantine, où se produit exactement l'inondations dans la ville d'ElOued-Algérie.

- **Quoi** ? **Qui**? **Quel**(le)? : Que trouve-t-on à tel endroit ? Il s'agit de mettre en évidence tous les objets ou phénomènes présents sur un territoire donné. Quel est le nombre de voies d'une tette route? Quelles sont les zones sensibles en cas d'avalanches ou de glissement de terrain? Trouver les zones favorables à la culture du riz.
- **Comment** ? : Quelles relations existent entre les objets et les phénomènes ? C'est la problématique de l'analyse spatiale. Ici, une référence est faite pour l'analyse spatiale. Par exemple : Comment relier une agglomération à un réseau d'assainissement, Comment limiter les accidents? comment se répartissent les eaux souterraines?
  - Les 4 dimensions de l'analyse spatiale sont :
    - Proximité (qu'est-ce qui est proche de quoi?)
    - Contiguïté (qu'est ce qui touche quoi ?)
    - Limologie[4] (qu'est ce qui limite quoi ?)
    - Superposition (qu'est ce qui se superpose ?)
- **Quand** ? : A quel moment des changements sont intervenus? cette question informe sur l'aspect temporel des objets géographiques, donc c'est la problématique de l'analyse temporelle. Par exemple : Quand est-ce que les ressources en eau de la nappe de Mitidja sont devenues salées ? Depuis quand le barrage de Beni-Haroun existe-t-il ? Quand est-ce que une telle autoroute a été mise en service?...
- **Et si**? : Que se passerait-il si un tel scénario d'évolution se produisait et quelles conséquences cela aurait. Cette dernière question permet de faire des prévisions ou des simulations sur l'avenir (= projection dans l'avenir, simulation, étude de projet, étude d'impact…). Par exemple si le niveau de la mer augmente de 2 mètres, quelles seront les villes touchées ?

Question	Objet d'étude	Exemple d'application
- Où se trouve cet objet, ce phénomène ? - Quoi ? : A cet endroit que trouve-t-on ?	Localisation	Inventaire localisé Analyse thématique
Comment ?	Répartition	Analyse spatiale
Quand ?	Evolution	Analyse temporelle
Et si ?	Modélisation	Simulation Étude d'impact

Tableau 2.1 : Tableau récapitulatif des cinq questions

2.7 Fonctionnement et dimensions d'un SIG

Un SIG contient généralement plusieurs types d'objets géographiques, organisés en thèmes et affichés sous forme de couches (figure 2.2). Chaque couche contient des objets de

[4] Etude des problèmes liés aux frontières, à l'établissement des frontières (Dictionnaire Reverso : http://dictionnaire.reverso.net/francais-definition/limologie)

même type (routes, bâtiments, cours d'eau, limites de communes, …etc.). Chaque objet est constitué d'une forme (géométrie de l'objet) et d'une description (appelé aussi sémantique).
Le niveau géométrique est la description de la position et de la forme des objets. La position peut s'exprimer par la latitude et la longitude des objets (ou des points qui composent ces objets) ou par des coordonnées x, y dans un système de projection. Les objets peuvent être identifiés sous forme de points (villes, entreprises,..), de lignes (routes, chemins de fer,...) et de polygones (communes, occupation du sol,...). A chaque objet est attribuée une fiche contenant des informations de type alphanumérique, ces informations décrivent l'objet (par exemple : nom de la ville, code de la commune, type de l'occupation du sol,...) et qui sont stocker sous forme de fichiers sauvegardés dans une base de données.

Généralement un SIG réunit un environnement de visualisation performant et une puissante infrastructure d'analyse et de modélisation des données spécialement adaptées à la géographie. Chaque SIG intègre plusieurs applications cartographiques bidimensionnelles (2D) ou tridimensionnelles (3D). Il offre ainsi une gamme complète d'outils permettant le traitement des informations géographiques à l'aide des trois volets, comme l'illustre la figure 2.8.

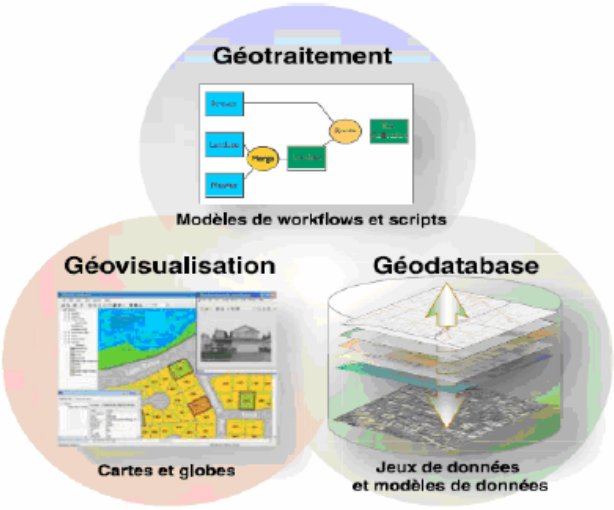


Figure 2.8 : les trois volets d'un SIG [7]

2.7.1. Volet géo-database : Un SIG se connecte à une base de données spatiale. Cette dernière décrit les objets manipulés géographiquement et sémantiquement.

2.7.2. Volet géo-visualisation : Un SIG est un ensemble de cartes intelligentes et de vues qui montrent les entités et leurs relations à la surface de la terre. Il est possible d'élaborer différentes vues cartographiques comme " des fenêtres ouvertes sur la base de données géographique ", afin d'effectuer des requêtes, des analyses et de modifier les informations géographiques.

2.7.3. Volet géo-traitement : Un SIG comprend des outils de transformation des données qui produisent des informations à partir des jeux de données existants. Les fonctions de géo-traitement appliquent sur les données de base, des fonctions analytiques, et écrivent les résultats dans de nouveaux jeux de données.

2.8 Fonctionnalités d'un SIG

Un SIG est un ensemble puissant d'outils pour saisir, conserver, extraire, transformer et afficher les données décrivant le monde réel [8] Cette définition résume parfaitement les cinq fonctionnalités (les cinq A) d'un système d'information géographique qui sont : Abstraction, Acquisition, Archivage, Analyse et Affichage de données à caractère spatial.

2.8.1. Abstraction: Abstraire Revient à concevoir un modèle qui organise les données par composants géométriques et par attributs descriptifs ainsi que l'établissement des relations entre les objets, les outils de définition des données et de conception du schéma conceptuel des données peuvent y être intégrées. Le découpage en couche est une solution envisagée pour un regroupement des objets géographiques homogènes (figure 2.2).

2.8.2. Acquisition : Acquérir revient à alimenter le SIG par les données. Les fonctions d'acquisition consistent à entrer d'une part, la forme géométrique des objets et d'autre part leurs attributs et relations**.** Ce module intègre deux types d'outils, un pour l'importation de données de différentes sources, et l'autre pour la numérisation.

2.8.3. Archivage : Archiver consiste à sauvegarder les données manipulées d'une manière bien organisée et structurée, au sein d'une base de données, et ce pour y faciliter l'accès quand il y aura besoin. Ce module s'appuie sur le support de stockage d'informations utilisé lors de l'acquisition.

2.8.4. Analyse: Ce module permet de répondre aux questions que l'on pose à un SIG, donc un SIG devient alors un tableau de bord cartographique et un outil d'aide à la décision.

2.8.5. Affichage : Afficher pour percevoir les relations spatiales entre objets et pour visualiser les données tout en faisant lien avec leurs localisations spatiales, un mode hypertexte peut être retenu.

2.9 Composants d'un SIG

Les SIG comportent trois types de composantes : technologiques (matériels et logiciels), informatives (base de données géographiques) et infrastructurelles (personnel, installation, services de support) » [8]. Cela fait qu'un Un systèmes d'information géographique est constitué de cinq composants majeurs :

2.9.1 Composantes matérielles : Les SIG fonctionnent aujourd'hui sur une très large gamme d'ordinateurs, des serveurs de données ou ordinateurs de bureaux connectés en réseau ou utilisés de façon autonome. **Les périphériques** Reliés aux ordinateurs permettent d'assurer diverses fonctions et deviennent de plus en plus indispensables :

- Matériel d'acquisition des données : scanner, table à digitaliser...
- Matériel de stockage des données : disques durs, CD Rom, disquettes, DVD...
- Matériel de visualisation des données : écrans traditionnels, écrans plats, portables...
- Matériel d'impression des données : imprimantes, traceurs...

2.9.2 Composantes logicielles : Les logiciels S.I.G offrent une panoplie d'outils et de fonctionnalités qui permettent de stocker, d'analyser et d'afficher des données géographiques (voir paragraphe 2.8)

2.9.3 Données : Les données sont certainement la composante la plus importante d'un S.I.G. Les données géographiques et les données tabulaires associées peuvent soit être constituées et élaborées en interne par un cartographe, soit acquise auprès de producteurs de données.

2.9.4 Humaines : Un S.I.G étant avant tout un outil, il s'adresse à une très grande communauté d'utilisateurs. De ceux qui créent et maintiennent les systèmes, jusqu'aux personnes qui utilisent la dimension géographique dans leur travail quotidien :

- Les techniciens et ingénieurs chargés de la conception, de l'entretien et de la gestion du S.I.G,
- Les techniciens utilisant quotidiennement le S.I.G dans leur travail,
- Les décideurs utilisant le S.I.G comme moyen d'aide à la décision.

Cependant, avec le développement des S.I.G sur Internet, le nombre d'utilisateurs potentiels ou réels de S.I.G augmente chaque jour.

9.9.5 Méthodes : La mise en oeuvre et l'exploitation d'un S.I.G ne peut s'effectuer sans le respect et l'application de certaines méthodes, règles et procédures. Ces méthodes permettent une utilisation rigoureuse et cohérente du matériel, des logiciels et des données du S.I.G par l'ensemble des utilisateurs et cela afin de répondre aux objectifs fixés au préalable dans tout projet.

2.10 Domaines d'application des SIG

A l'heure actuelle, les SIG touchent pratiquement la plupart des domaines et résolvent des problématiques de plus en plus complexes. Si l'on essaie de caractériser les questions auxquelles un S.I.G est censé pouvoir répondre, on est vite confronté à la multiplicité des domaines d'application possibles [web 3]:

- **Aménagement du territoire :** Schémas de Cohérence Territoriale (SCOT, Plan Locaux d'Urbanisme (PLU), choix de tracés routiers, autoroutiers ou ferroviaires, études d'impacts….
- **Gestion urbaine :** Gestion de la voirie, des réseaux de distribution, des espaces verts, du patrimoine, de la sécurité, simulation d'insertion de projets architecturaux….

- **Circulation et conduite automobile (Transport):** choix d'itinéraires, suivi de flottes de véhicules, aide à la conduite assistée par ordinateur.
- **Agriculture :** génie rural, gestion des ressources en eau, suivi et prévision des récoltes, gestion des forêts, aide à la mise en œuvre de la Politique Agricole Commune.
- **Protection de l'environnement :** définition des zones sensibles, suivi des évolutions, alerte aux pollutions, protection des paysages.
- **Risques naturels et technologiques majeurs :** définition et suivi des zones à risque, prévention de catastrophes, intervention en cas de sinistre, organisation des secours.

Les domaines d'application des SIG sont aussi nombreux que variés ? Nous pouvons en citer :

• Tourisme (gestion des infrastructures, itinéraires touristiques)

• Géomarketing (localisation des clients, analyse du site, présence de consommateurs potentiels d'un produit ou d'un service dans une région, Suivi d'expédition de paquets visualisés sur des cartes)

• Forêt (cartographie pour aménagement, gestion des coupes et sylviculture)

• Géologie (prospection minière)

• Biologie (études du déplacement des populations animales)

• Télecoms (implantation d'antennes pour les téléphones mobiles)

• Défense et sécurité.

2.11 Outils et logiciels pour les SIG

A l'heure actuelle, les SIG existent sur le marché sous forme de progiciels[5]. Les différents catalogues disponibles sur le marché recensent plus d'une soixantaine de progiciels, qu'ils tournent sur micro-ordinateurs, sur stations de travail ou sur ordinateurs centraux. Ils proposent des fonctions d'intérêt général, qu'il est nécessaire de compléter selon les besoins spécifiques aux domaines étudiés, par des développements supplémentaires. On peut décomposer ces logiciels en trois grandes familles:

2.11.1 SIG généralistes bureautiques

Ils ont pour vocation essentielle l'import de données externes et leur analyse pour donner des cartes à insérer dans des rapports ou des présentations. Ils permettent bien sûr la modification de données géométriques ou descriptives mais ils ne disposent pas d'outils d'assurance qualité perfectionnés pour saisir des Bases de Données complètes. Ils disposent d'outils de développement pour s'adapter à tout type d'application [web 2].

2.11.2 SIG généralistes de gestion

Ils disposent des mêmes capacités que les SIG bureautiques, sont fréquemment moins conviviaux, mais disposent d'outils de modélisation beaucoup plus puissants, qui vont mettre

[5] Progiciel : Un logiciel applicatif commercial "prêt-à-porter", standardisé et générique à l'opposer de Logiciel

des contraintes à la saisie et donc assurer une certaine qualité des données. Ces SIG vont également disposer de capacités client/serveur qui vont permettre à plusieurs personnes de travailler sur la même Base de Données à partir de postes informatiques distants. Ils disposent d'outils de développement pour s'adapter à tout type d'application. [web 2].

2.11.3 SIG métiers

Ces logiciels sont dès le départ très spécialisés, destinés à des métiers particuliers. Leur champ d'application est réduit mais ils sont souvent les seuls ou les meilleurs dans leur domaine. Ce sont néanmoins des SIG car ils possèdent les 5 fonctionnalités qui font les SIG : Affichage, Acquisition, Abstraction, Analyse, Archivage. Fréquemment, les éditeurs de logiciels commercialisent des modules additionnels qui transforment les SIG généralistes en SIG métiers [web 2].

La liste qui suit n'a pas la prétention d'être exhaustive. Les SIG généralistes mentionnés sont parmi les plus diffusés en France, en se limitant à des logiciels tournant sur PC. Ces progiciels sont soit payants soit libres, tout dépend de leurs fournisseurs :

MapInfo est un SIG généraliste bureautique typique. Il permet de sortir très facilement toutes sortes d'analyses thématiques. Il autorise l'utilisateur à ouvrir des fichiers EXCEL, à ouvrir et à modifier des fichiers ACCESS, à travailler sur des données ORACLE… de manière transparente. En revanche, ses possibilités de modélisation sont pauvres, il ne prévoit pas de travailler sur des données en client serveur, et le travail sur de grosses bases de données est difficile [web 2]. Auparavant, le langage de développement était spécial: Mapbasic, mais vu sa difficulté et sa non convivialité, il a été remplacé par le langage de programmation « Visuel basic » qui est plus rapide et conviviale doté par des API [9].

ArcView est un logiciel SIG (Système d'Information Géographique) complet permettant de visualiser, de gérer, de créer et d'analyser des données géographiques. ArcView permet de mieux comprendre le contexte géographique des données, dévoilant ainsi des relations et des propriétés sous de nouveaux angles. ArcView aide des milliers d'organisations à prendre des décisions plus pertinentes et à résoudre des problèmes plus rapidement [web 1]. ArcView est également un SIG généraliste bureautique, même si l'intégration de données externe est plus délicate, il est convivial mais ses possibilités de structurations sont restreintes et il nécessite des compléments pour partager une base de données [web 2].

GéoConcept est un progiciel à la frontière entre SIG bureautique et SIG de gestion. Il offre l'ouverture et la convivialité des premiers, et peut comme les seconds travailler en client serveur sur de grandes Bases de Données. Il propose pour les utilisateurs des solutions SIG, une nouvelle version du moteur GeoConcept Enterprise et l'accès aux

données de photographies aériennes ou satellites sur le monde entier avec la solution W@M ; Pour le géomarketing, une nouvelle version de GeoConcept Sales & Marketing avec la prise en charge de nouvelles méthodes d'analyse [web 4].

APIC4 est un SIG de gestion. L'intégration de données externes est lourde, en revanche, ce progiciel possède des possibilités de modélisation et de travail en groupe très étendues [9]. APIC4 est basé sur le moteur SIG APIC-Space complété par 4 modules applicatifs APIC-Edit(Saisir et mettre à jour les données), APIC-Explore (Rechercher et analyser les données), APIC-Visu (Définir la représentation des objets) et APIC-Compose (Générer des tracés cartographiques), conçus pour répondre à la majorité des besoins standards. Les points forts d'APIC, sont : son architecture client/serveur, une modélisation objet avec notion de relations simple et intuitive, une grande performance dans la gestion des gros volumes de données vecteur et raster (affichage et traitements), un langage de développement puissant et adapté à tous les métiers et une compatibilité directe et totale avec Oracle [web 5].

2.12 Les perspectives de développement des SIG

Les perspectives de développements dans le domaine des SIG, évoluent parallèlement avec les évolutions technologiques, économiques et sociales, qui s'articulent autour de trois grands axes : l'ouverture sur le monde via Internet, une meilleure intégration des outils d'aide à la décision, une grande capacité pour intégrer et modéliser des données de toute nature.

2.12.1 L'ouverture sur le monde

L'arrivé d'Internet, et sa porté au grand public a beaucoup valorisé l'ouverture des SIG au grand public. Les éditeurs des outils SIG explorent actuellement cette voie pour élargir le champ d'application des SIG, donc élargir la clientèle pour plus de bénéfice. Cependant, cette ouverture nécessitera évidemment de l'interopérabilité entre les SIG et le développement d'outils d'interrogation s'orientant vers les langages naturels pour s'adapter à toute catégorie d'utilisateurs [web 6].

A. L'ouverture sur l'Internet, et d'autres modes de type nomade

Outre la possibilité de consulter des données de SIG sur le Web, les SIG devront intégrer les technologies qui commencent à émerger dans le domaine de la connaissance. Il s'agit des Web services, de l'intégration du protocole SOAP[6]. Ils doivent aussi s'appuyer sur les technologies d'informations réparties et des systèmes

[6] SOAP : Ancien acronyme de *Simple Object Access Protocol* est un protocole de RPC orienté objet bâti sur XML

embarqués [6]. L'informatique nomade qui entre dans le cadre de l'information répartie, a changé radicalement la manière d'utiliser le système d'information géographique, en donnant la possibilité d'emmener le SIG sur le terrain et de communiquer directement avec le monde qui nous entoure. Les SIG doivent aussi pouvoir utiliser les systèmes embarqués traitant les données géo-référencés dans la même échelle de temps, où il est possible de localiser une personne, un véhicule ou tout autre matériel, afin de lister les différents évènements déclenchés.

B. L'interopérabilité [6]

La possibilité d'exploiter les données provenant d'un SIG dépend en partie des formats d'échange que ce dernier gère nativement ou génère pour l'exportation. Nous avons pu constater que de produits présentent de nombreux formats de fichier en exportation. Un consortium, l'OpenGis [7], réunit des éditeurs logiciels, des constructeurs, des organisations gouvernementales, des constructeurs informatiques et des producteurs de données géographiques (Spot Image, Landsat...). Cette organisation produit les spécifications d'architecture logicielle, de formats de fichiers, des balises spécifiques pour le Web, donnant de réelle capacité d'interopérabilité aux systèmes qui y souscrivent. D'autres organismes oeuvrent sur le même domaine ; EUROGI [8], en particulier rassemble des associations travaillant sur le domaine de l'information géographique de chacun des pays européens (Afigeo en France). La France a elle même produit une norme de présentation de données géographiques reprise par plusieurs des SIG présentés, il s'agit d'Edigeo[9].

C. Le développement des outils d'interrogations [6]

Le succès de l'utilisation des SIG et de leur intégration comme véritable système d'organisation dépendra en partie de leur capacité à rendre l'information facilement accessible. L'introduction de nouvelle interface, de nouveau langage, est une des pistes explorée. A titre d'exemple, le langage LVIS [10] permet à l'utilisateur de procéder à des analyses spatiales à l'aide d'icônes qui modélisent des concepts. Ce langage est destiné aux non informaticiens, il prend en compte la gestion des ambiguïtés.

2.12.2 Une meilleure intégration des outils d'aide à la décision

Le statut des SIG comme véritable système d'information d'une organisation, ne peut être acquis que, dans la mesure où il offre, au delà des tâches de gestion, les outils qui permettent la définition des stratégies pour l'aide à la prise de décision. Des travaux ont été réalisés dans ce domaine en associant des outils de modélisation, des systèmes multi-acteurs avec des SIG [11].

[7] http://www.opengeospatial.org/
[8] http://www.eurogi.org
[9] AFNOR NF-Z13-15

2.12.3 Une prise en compte des données de toute nature

La richesse d'un SIG repose sur la finesse de la modélisation d'une problématique selon le territoire. La capacité à intégrer des données d'origine de plus en plus large sera une des facteurs de réussite des SIG. Ces derniers touchent pratiquement tous les domaines d'où la nécessité de les adapter à tout type de donnée [6].

2.13 Conclusion

Les systèmes d'information géographique sont avant tous des systèmes d'information, avec une particularité qui est la nature de l'information traitée, dite géographique. L'information géographique est avant tout, une information, manipulée au sein d'un système pour apporter une aide à l'utilisateur (aide à la gestion ou à la décision). Cependant, elle possède cette particularité d'être « spatiale », et ceci nécessite un peu plus d'attention et de soin dans sa collecte, sa modélisation, sa conception et enfin sa présentation.

Nous avons vu aussi dans le présent chapitre, ce qui existe, comme fonctionnalités, composants, domaines d'application, mode de fonctionnement et les outils-logiciels pour les SIG, ainsi, ce qui est en cours de développement et ce qui est en perspectives d'avenir.

Chapitre 3 :

Les ontologies : Etat de l'art

"Entre ce que je pense, ce que je veux dire, ce que je crois dire, ce que je dis, ce que vous avez envie d'entendre, ce que vous croyez entendre, ce que vous entendez, ce que vous avez envie de comprendre, ce que vous croyez comprendre, ce que vous comprenez, il y a 10 possibilités qu'on ait des difficultés à communiquer. Mais essayons quand même..."

Bernard Werber, Encyclop´edie du savoir relatif et Absolu

3.1 Introduction

Aujourd'hui le partage sémantique entre les systèmes d'information est devenu un challenge dans les entreprises. Ce partage représente un vrai problème empêchant de faire interopérer et réutiliser ces systèmes à travers une architecture pratique. Depuis quelques années, l'usage des ontologies est de plus en plus grandissant pour résoudre le problème de partage sémantique.

Dans ce chapitre nous présentons un état de l'art sur l'histoire du mot « ontologie » depuis la définition d'Aristote jusqu'à celles des chercheurs en Intelligence Artificielle en arrivant au web sémantique. Ainsi, nous donnons les différentes définitions, les différents types, et composants d'une ontologie. Ensuite, nous décrivons les approches, méthodes et méthodologies de construction d'ontologie et nous terminons par la description d'une ontologie spatiale.

3.2 Origine et définitions des ontologies

Le mot ontologie trouve sa racine du grec, il est composé de deux mots : **onto** (le participe présent du verbe être) qui signifie l'étude de l'être en tant qu'être et **logos** qui signifie le discours. La définition d'Aristote du mot Ontologie (avec un grand O[10]) est « the science of being *qua* being » (la métaphysique d'Aristote), cette science fait partie de la philosophie, et traite de la description des entités du monde réel. Son but est de définir les catégories générales employées pour classifier toutes entités du monde (les êtres humains, les animaux, les objets, etc.). Au début des années 80, les chercheurs en intelligence artificielle ont empruntés le terme « Ontologie » au champ de la philosophie. Les ontologies sont devenues la définition des connaissances d'un domaine. Elles fournissent la possibilité de séparer les connaissances du domaine des connaissances opérationnelles [12].

Gruber en 1993 a proposé une définition intéressante des ontologies dans le domaine de l'intelligence artificielle : une ontologie est « *une spécification d'une conceptualisation pour aider des programmes et des humains à partager des connaissances*" [13]. Plus précisément : "*…les ontologies sont des spécifications explicites formelles d'une conceptualisation partagée "*. D'après Gruber, « *une conceptualisation* » est le résultat de la modélisation de phénomènes du domaine d'intérêt. Cette modélisation identifie les concepts et leurs relations décrivant ces phénomènes. *« Explicite »* signifie que les concepts et leurs relations sont typés, et les contraintes sur l'utilisation de ces types sont clairement expliquées. « *Formel* » se rapporte au fait que l'ontologie doit être compréhensible par la machine. *« Partagé »* reflète l'idée que

[10] 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.

l'ontologie doit capturer les connaissances consensuelles communément admises par l'ensemble de la communauté des acteurs du domaine [14].

Les ontologies peuvent être utilisées par des personnes, des bases de données et des applications qui ont besoins de partager des informations sur un domaine, elles incluent des définitions, des informations exploitables, des concepts élémentaires dans le domaine et leurs relations. Dans le domaine des systèmes de bases de données, les ontologies sont employées pour faciliter l'interopérabilité des sources d'information hétérogènes. L'ontologie est le schéma général organisant toutes les propriétés des entités décrites dans les schémas de base de données ou sources d'information. Chaque source d'information possède un médiateur qui traduit son schéma de base de données en éléments de l'ontologie. Ainsi, les utilisateurs peuvent questionner l'ontologie et obtenir un résultat intégrant toutes les données des sources d'information requises pour répondre à la requête [61]

Les techniques de recherche documentaire emploient des ontologies linguistiques telles que les thesaurus ou les langages d'indexation afin d'éviter les ambiguïtés sémantiques liées à l'usage des termes polysémiques ou synonymiques. Ainsi, le document et le contenu des requêtes sont représentés par des concepts (c.-à-d. le sens des termes) et non par les termes eux-mêmes (c.-à-d. l'ensemble des caractères). Avec cette technique il est possible d'améliorer la description du contenu des documents (et des requêtes), et également d'améliorer les performances du système de recherche d'information. Un des premiers systèmes opérationnels est le système Ontoseek, qui a utilisé la base terminologique de Wordnet [web 7] pour décrire les Pages Jaunes [15] [16].

3.3 Composants d'une ontologie

Une ontologie est un ensemble structuré sous la forme d'un graphe orienté qui contient, des nœuds (concepts) représentant le vocabulaire d'un domaine particulier et des arcs représentant les relations (ou rôles) nommées entre les concepts. A partir de cette structure, la sémantique de chaque mot est déduite par les relations que ce mot possède dans l'ontologie, ce qui permet de restreindre les interprétations possibles. Cependant pour être exploitable par une machine, une ontologie doit respecter certaines règles : être définie par une syntaxe formelle et une sémantique non ambiguë et permettre la déduction de nouvelles connaissances présentent implicitement dans l'ontologie [17].

Selon Gomez Pérez [18], les connaissances traduites par une ontologie sont véhiculées à l'aide de cinq éléments :

- **Concepts** : ils sont appelés aussi termes ou classes de l'ontologie, un concept est le constituant de la pensée (un principe, une idée, une notion abstraite) sémantiquement

évaluable et comminicable. Selon Gómez Pérez ces concepts peuvent être classifiés selon plusieurs dimensions : 1) *niveau d'abstraction* (concret ou abstrait) ; 2) *atomicité* (élémentaire ou composée) ; 3) *niveau de* réalité (réel ou fictif).

- **Relation** : Elles traduisent les associations existant entre les concepts du réel perçu (conçu), ces relations regroupent les associations suivantes ; 1) Sous-classe-de (spécialisation ou généralisation), 2) Partie-de (agrégation ou composition), 3) Associée-à, 4) Instance-de, Ces relations nous permettent d'apercevoir la structuration et l'interrelation des concepts, les uns par rapport aux autres [19].
- **Fonctions** : constituent des cas particuliers de relations, dans laquelle le $N^{ième}$ élément de la relation est défini de manière unique à partir des (n-1) éléments précédents.
- **Axiomes** : Constituent des assertions acceptées comme vraies, à propos des abstractions du domaine qui sont traduites par l'ontologie.
- **Instance** : Constituent la définition extensionnelle de l'ontologie, ces objets véhiculent les connaissance à propos du domaine du problème traité.

3.4 Différence entre ontologie et base de connaissance

Une ontologie est universelle *mais différente* de la base de connaissance qui elle serait individuée, relative, finalisée [20]. Une autre explication fournie par Farquhar en 1997 lors d'un forum de discussion sur l'ontologie. Ce dernier a proposé de poser quelques questions et il a affirmé que si "Plus la réponse à ces questions est positive, plus c'est ontologique". L'ensemble des questions proposé est :

"Est-ce que cela exprime la connaissance consensuelle d'une communauté de gens ? Est-ce que les gens l'utilisent comme une référence de termes définis avec précision? Est-ce que cela exprime la connaissance consensuelle d'une communauté d'agents ? Est-ce que le langage utilisé est suffisamment expressif pour que les gens puissent dire ce qu'ils veulent dire? Est-ce que cela peut être utilisé pour de multiples cas de résolution de problèmes? Est-ce que c'est stable? Est-ce que cela peut être utilisé pour résoudre une variété de différents types de problèmes? Est-ce que cela peut être utilisé comme point de départ pour construire demultiples types d'applications incluant une base de connaissances, un schéma de base de données ou un programme orienté-objet?".

L'opinion de Farquhar se fonde sur l'idée qu'il n'existerait pas de frontière bien définie entre ontologie et connaissance. Cela parait raisonnable si l'on considère *CYC*[11], dont la partie supérieure est certainement une ontologie mais dont l'ensemble apparaît comme une base de

[11] **Cyc :** est un projet d'intelligence artificielle (« IA ») qui cherche à développer une ontologie globale et une base de données de la connaissance générale, dans le but de permettre à des applications d'intelligence artificielle de raisonner d'une manière similaire à l'être humain [web 14]

connaissances. Toutefois, cette opinion induit en erreur –même si elle est partagée par de nombreux chercheurs en intelligence artificielle - puisqu'elle ne cherche pas à capturer une propriété essentielle de l'ontologie, à savoir que celle-ci s'attache aux concepts plutôt qu'au vocabulaire, et qu'elle porte sur ce qui existe dans le monde cible.

Riichiro [21] a répondu que nous avons besoin d'introduire la notion de relativité pour bien comprendre ce qu'est une ontologie. C'est-à-dire qu'une différenciation claire entre *"ontologie"* et *"base de connaissances"* devrait se faire à partir de son rôle, c'est-à-dire qu'une ontologie vous fournit un système de concepts qui sont utilisés pour construire une base de connaissances par-dessus ; par conséquent, une ontologie peut être une spécification de la conceptualisation du monde-cible que se fait l'ingénieur qui construit la base de connaissances, donc un méta-système d'une base de connaissances traditionnelle [21].

3.5 Différence entre ontologie et hiérarchie de classes en terme d'O.O

Au niveau supérieur, une méthodologie de développement d'une ontologie et celle d'une hiérarchie orientée-objet sont presque similaires. Cependant, au niveau inférieur, l'ontologie se concentre sur les aspects déclaratifs tandis que la hiérarchie orientée-objet se concentre sur les aspects reliés à la performance. Par conséquent, la différence essentielle entre les deux est que l'ontologie exploite la représentation déclarative, tandis que le paradigme orienté-objet est intrinsèquement procédural. Dans le paradigme orienté-objet, la signification d'une classe, d'une relation entre des classes, ainsi que les méthodes sont intégrées de façon procédurale et sont implicites, en revanche dans le paradigme ontologique, les descriptions sont faites de façon déclarative, ce qui permet au système de modifier son comportement en modifiant la connaissance qu'il possède [21].

3.6 Classification des ontologies

Les ontologies peuvent être classifiées selon plusieurs dimensions : [19], dans cette partie nous décrivons les plus connues dans la littérature et pour davantage de détails, le lecteur peut se référer à l'état de l'art de KEITA [22]

3.6.1 Selon l'objet de conceptualisation (le but de leur utilisation) : on peut distinguer selon Gomez Pérez, les types suivants :1) Ontologie de Représentation des connaissances; 2) Ontologie Supérieure/ Haut niveau; 3) Ontologie Générique ; 4) Ontologie de Domaine ; 5) Ontologie de Tâches; 6) Ontologie d'Application (figure 3.1).

- **Ontologie de Représentation des connaissances** : Elle décrit les concepts utilisés par les langages de représentation des ontologies. Elle qui définit des primitives de représentation utilisées pour formaliser la connaissance avec un paradigme donné. Les

exemples les plus représentatifs sont la "Frame Ontology" [13] et la "OKBC[12] Ontology", toutes les deux sont accessibles sur le serveur Ontolingua.

- **Ontologie Supérieure ou de Haut niveau** ("Top-level ou Upper-level ontologies"): Cette ontologie est une ontologie générale. Son sujet est l'étude des catégories des choses qui existent dans le monde, soit les concepts de haute abstraction tels que: les entités, les événements, les états, les processus, les actions, le temps, l'espace, les relations, les propriétés. L'ontologie de haut de niveau est fondée sur : la théorie de l'identité, la méréologie[13] et la théorie de la dépendance [19].
- **Ontologie Générique** : appelée, méta-ontologies, elle véhicule des connaissances génériques moins abstraites que celles véhiculées par l'ontologie de haut niveau, mais assez générales néanmoins pour être réutilisées à travers différents domaines [19].
- **Ontologie de Domaine** : Une ontologie de domaine décrit le vocabulaire ayant trait à un domaine générique (exemple : l'enseignement, la médecine…), notamment en spécialisant les concepts d'une ontologie de haut niveau [23]. Ces ontologies peuvent être réutilisées pour plusieurs applications qui touchent un domaine.
- **Ontologie de Tâches** : Une ontologie de tâche décrit le vocabulaire concernant une tâche générique (ex. : enseigner, diagnostiquer, planifier…), notamment en spécialisant les concepts d'une ontologie de haut niveau [23]. Certains auteurs emploient le nom « ontologie du domaine de la tâche » pour faire référence à ce type d'ontologie [24]. Ce type d'ontologies régit un ensemble de vocabulaires et de concepts qui décrit une structure de résolution des problèmes inhérente aux tâches et indépendante du domaine [25].
- **Ontologie d'Application** : Contrairement à l'ontologie de domaine, l'ontologie d'une application donnée est la plus spécifique, elle ne peut pas être réutilisée pour d'autres applications, elle sert à décrire des conceptualisations de domaine spécifique à l'application en question. L'ontologie d'application contient des concepts dépendants d'un domaine et d'une tâche particuliers, qui sont généralement subsumés par des concepts de ces deux ontologies. Ces concepts correspondent souvent aux rôles joués par les entités du domaine lors de l'exécution d'une certaine activité [23]. Il s'agit donc ici de mettre en relation les concepts d'un domaine et les concepts liés à une tâche particulière, de manière à en décrire l'exécution (ex. : apprendre les statistiques, effectuer des recherches dans le domaine de l'astronomie, etc.).

[12] Ontology Knowledge Base Connectivity

[13] Méréologie : La **méréologie** est une collection de systèmes formels axiomatiques qui traitent des relations entre la partie et le tout [web 14]

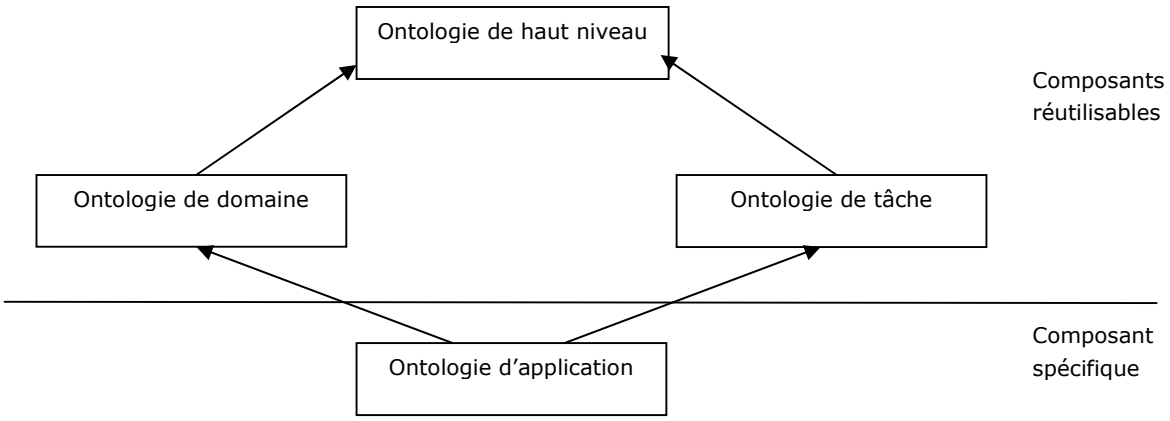


Figure 3.1 : Classification des ontologies Selon l'objet de conceptualisation

3.6.2 Selon le niveau de détail de l'ontologie

par rapport au niveau de détail utilisé lors de la conceptualisation de l'ontologie en fonction de l'objectif opérationnel envisagé pour l'ontologie, deux catégories au moins peuvent être identifiées :

- **Granularité fine** : correspondant à des ontologies très détaillées, possédant ainsi un vocabulaire plus riche capable d'assurer une description détaillée des concepts pertinents d'un domaine ou d'une tâche. Ce niveau de granularité peut s'avérer utile lorsqu'il s'agit d'établir un consensus entre les agents qui l'utiliseront [19].
- **Granularité large** : correspondant à un vocabulaire moins détaillé comme par exemple dans les scénarios d'utilisation spécifiques où les utilisateurs sont déjà préalablement d'accord à propos d'une conceptualisation sous-jacente [26], [27]. Les ontologies de haut niveau possèdent une granularité large, compte tenu que les concepts qu'elles traduisent sont normalement raffinés subséquemment dans d'autres ontologies de domaine ou d'application [26].

3.6.3 Selon le niveau de complétude

Le niveau de complétude a été abordé par [61] et [62], ce dernier propose une classification sur trois niveaux suivante :

- **Niveau Sémantique** : Tous les concepts (caractérisés par un terme/libellé) doivent respecter les quatre principes différentiels :

  - Communauté avec l'ancêtre;
  - Différence (spécification) par rapport à l'ancêtre;
  - Communauté avec les concepts frères (situés au même niveau);
  - Différence par rapport aux concepts frères (sinon il n'aurait pas lieu de le définir).

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

- **Niveau Référentiel** : Outre les caractéristiques énoncées au niveau précédent, les concepts référentiels (ou formels) se caractérisent par un terme/libellé dont la sémantique est définie par une extension d'objets. L'**engagement ontologique** spécifie les objets du domaine qui peuvent être associés au concept, conformément à sa signification formelle. Deux concepts formels seront identiques s'ils possèdent la même extension.
- **Niveau Opérationnel** : Outre les caractéristiques énoncées au niveau précédent, les concepts du niveau opérationnel ou computationnel sont caractérisés par les opérations qu'il est possible de leur appliquer pour générer des inférences (**engagement computationnel**). Deux concepts opérationnels sont identiques s'ils possèdent le même potentiel d'inférence.

3.6.4 Selon le niveau de formalisme

Par rapport au **niveau du formalisme de représentation** du langage utilisé pour rendre l'ontologie opérationnelle, [68] proposent une classification comprenant quatre **catégories** :

- **Informelles :** ontologies opérationnelles exprimées dans un langage naturel (sémantique ouverte) ;
- **Semi-informelles :** l'ontologie est exprimée dans une forme restreinte et structurée du langage naturel.
- **Semi-formelles :** l'ontologie est exprimée dans un langage artificiel défini formellement.
- **Formelles** : l'ontologie est exprimée dans un langage artificiel disposant d'une sémantique formelle.

3.6.5 Selon le niveau de complexité (Légères et Lourdes) [77]

Les ontologies légères (*lightweight ontologies)* incluent des concepts comprenant des propriétés et qui sont organisées en taxonomies avec des relations conceptuelles (e.g. *Yahoo! Directory*); certains auteurs considèrent les taxonomies comme des ontologies parce qu'elles fournissent des conceptualisations partagées pour des domaines donnés. Les ontologies lourdes (*heavyweight ontologies*) ajoutent aux ontologies légères des axiomes et des restrictions clarifiant le sens. Les ontologies lourdes modélisent un domaine de façon plus profonde avec plus de restrictions basées sur la sémantique du domaine. Ces ontologies

peuvent être modélisées avec des méthodes d'intelligence artificielle basées sur les "*frames*" et sur la logique du premier ordre (e.g. CyC 2 et Ontolingua 3 développés dans les années '90) ou avec la logique de description [28] utilisée pour développer le langage ontologique OWL du Web Sémantique (voir chapitre 4). Par contre les ontologies légères peuvent être modélisées à partir des modèles utilisés en génie logiciel: la modélisation UML[14] et la modélisation par diagramme Entité/Relation.

3.7 Méthodologies de construction d'ontologies

La conception d'ontologies est une tâche difficile nécessitant la mise en place de procédés élaborés afin d'extraire la connaissance d'un domaine, manipulable par les systèmes informatiques et interprétable par les êtres humains, elle relève plus du savoir-faire que de l'ingénierie. A l'heure actuelle, il n'existe pas de consensus à propos des meilleures pratiques à adopter lors du processus de construction, ni des normes techniques régissant le processus de développement des ontologies, bien que certaines contributions dans cette direction soient déjà disponibles.

La plupart des équipes de recherche dans le domaine travaillent de manière ad hoc. Bien que la plupart des méthodologies initient le processus de construction par l'identification, puis l'organisation et la structuration des concepts et des relations à représenter [29]. La construction d'une ontologie suppose certaines obligations qui découlent du choix d'utiliser certains concepts plutôt que d'autres pour représenter un phénomène. Ce sont les exigences ontologiques. [78]

Jusqu'en 1996, les premières ontologies ont été développées de façon complètement artisanale basée sur le savoir faire, sans suivre de méthode prédéfinie, citons à titre d'exemple, Enterprise Ontology, TOVE, MENELAS. Depuis 1998, c'était la naissance de cadres méthodologiques plus élaborés inspirés des méthodes de l'Ingénierie des Connaissances, par exemple : METHONTOLOGY et TERMINAE.

3.7.1 Enterprise Ontology

Elle est basée sur l'expérience du développement de l'ontologie *Enterprise Ontology* [web 8], qui repose sur l'identification de différentes étapes :

- Identification du POURQUOI de l'ontologie ;
- Construction de l'ontologie (identification des concepts clef, modélisation informelle, formalisation) et intégration d'ontologies existantes ;
- Evaluation et documentation de l'ontologie.

[14] UML : Unified Modeling Language

Cette méthode s'inspire du développement de SBCs[15]. Les étapes et sous-tâches sont décrites de façon abstraite. Les techniques à utiliser pour les sous-tâches ne sont pas précisées (ex : comment identifier les concepts clef ? Quel langage de formalisation utiliser ?). **[30]**

3.7.2 TOVE

Cette méthode est basée sur l'expérience du développement de l'ontologie du projet TOVE [web 9] (TOrento Virtual Enterprise). Elle aboutit à la construction d'un modèle logique de connaissance. L'ontologie est développée selon les étapes suivantes :

- Identification des scénarios (problèmes) dépendants d'une application, qui clarifient le domaine que l'on investit et les différentes applications dans lesquelles l'ontologie sera employée.
- Formulation de questions informelles (basées sur les scénarios) exprimées en langages naturel auxquelles l'ontologie doit permettre de répondre, ces questions et leurs réponses sont utilisées pour extraire les concepts principaux, leurs propriétés et les relations qui existent entre ces concepts.
- Spécification d'une terminologie à partir des termes Spécifier la terminologie de l'ontologie : cette étape consiste à représenter les termes identifiés dans l'étape précédente, en utilisant le formalisme de la logique du premier ordre. Les concepts seront représentés sous forme de constantes ou bien des variables, par ailleurs les propriétés et les relations seront représentées par des prédicats.
- Evaluation de la complétude de l'ontologie.

La méthode reste spécifiée de façon abstraite. Ni les différentes étapes ni les techniques ne sont décrites en détail. **[30]**

3.7.3 MENELAS [web 10]

Le projet MENELAS est un projet européen sur la compréhension de comptes rendus médicaux dans le domaine de la coronarographie ; l'ontologie devait constituer le support des graphes conceptuels, elle représente 1500 concepts et 500 relations. Les deux principaux objectifs de la méthode Ménélas sont:

- Offrir un meilleur compte et un meilleur accès à l'information médicale à travers les langues naturelles afin d'aider les médecins dans leur pratique quotidienne,
- Améliorer la coopération européenne par l'accès multilingue normalisé à nomenclatures médicales.

[15] SBCs : Systèmes à base de connaissances

3.7.4 METHONTOLOGY

Cette méthode a été développée par le groupe d'Ontologie au Laboratoire d'Intelligence Artificielle de l'Université polytechnique de Madrid. Elle vise la construction d'ontologies au "niveau connaissance" et repose sur :

- un processus de développement d'ontologies comportant des activités de gestion de projet (planification, assurance qualité), des activités orientées développement (spécification, conceptualisation, formalisation) et des activités de support (documentation).
- un cycle de vie des ontologies basé sur des prototypes évolutifs (figure 3.2). [30]

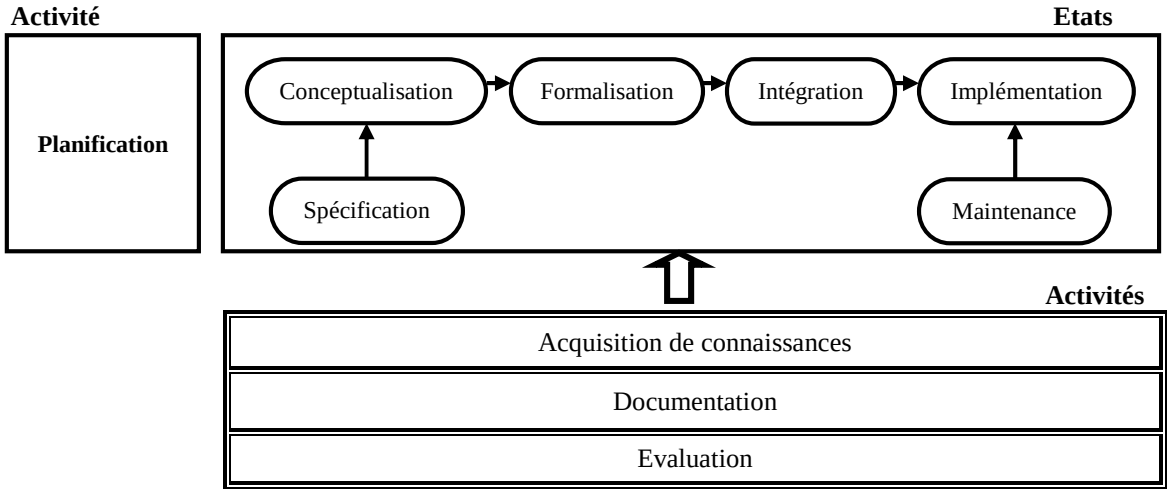


Figure 3.2 : Cycle de vie de METHONTOLOGY

METHONTOLOGY a été proposée pour la construction d'ontologie par la FIPA[16], qui favorise l'interopérabilité à travers les applications. Elle est spécifiée de façon très détaillée et a été utilisée pour construire plusieurs ontologies dont l'ontologie des ontologies : *Reference Ontology*. METHONTOLOGY inclut une méthode de ré-ingénierie pour résoudre certains des problèmes liés à la construction d'une ontologie par la réutilisation d'une autre ontologie. ODE et WebODE 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, OntoEdit.

3.7.5 TERMINAE [30]

Cette méthode a été développée au LIPN, à l'Université de Villetaneuse. S'insérant dans la problématique du groupe de recherche français TIA (Terminologie et IA), elle propose la construction d'ontologies à partir de textes en suivant quatre principales étapes :

[16] FIPA: Foundation for Intelligent Physical Agents

- Constitution d'un corpus (documents techniques, comptes rendus, livres de cours, etc.), à partir d'une analyse des besoins de l'application visée.
- *Etude linguistique*, pour identifier des termes et des relations lexicales, en utilisant des outils de traitement de la langue naturelle.
- Normalisation sémantique, conduisant à des concepts et des relations sémantiques définis dans un langage semi-formel.
- Formalisation et intégration des concepts au sein d'une BC formelle.

3.7.6 OntoSpec

OntoSpec permet d'élaborer des ontologies semi-informelles pouvant ensuite être représentées dans le langage formel choisi par le concepteur. Elle reprend ces méta-propriétés et incite le concepteur à utiliser certaines propriétés en particulier lors de l'élaboration de l'ontologie. Elle prend en amont un ensemble d'entités conceptuelles exprimées par des termes ainsi qu'un ensemble de définitions en langue naturelle.

3.7.7 OntoClean

La méthodologie **OntoClean** propose de typer les concepts d'une ontologie à l'aide de ces caractéristiques, puis de tester la cohérence de la hiérarchie des concepts en vérifiant que les contraintes induites par ces caractéristiques ne sont pas violées.

3.7.8 SENSUS

La méthode Sensus, proposée par l'équipe de Swartout , est utilisée pour construire le squelette d'une ontologie de domaine à partir d'une grande ontologie, l'ontologie Sensus. Elle est appliquée au traitement automatique du langage naturel et a été élaborée pour fournir une structure conceptuelle au développement de traducteurs (machine). Elle a surtout été utilisée dans le domaine de la défense et les ontologies créées à partir de cette méthode sont du domaine de la planification des campagnes militaires d'aviation et comprennent des ontologies sur les armes, sur le carburant, etc.

3.7.9 OnToKnowledge

On-to-Knowledge [31] est une méthodologie développée dans le cadre d'un projet dont les partenaires sont l'Institut AIFB[17] de l'Université de Kalsruhe[18], l'Université Libre d'Amsterdam, et la société British Telecom. OntoKnowledge 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

[17] AIFB :
[18] Kalsruhe :

d'évolution. OnToKnowledge 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 [32]

3.7.10 Ontology Development 101

Ontology Development 101 [33] 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 - Enumérer 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. [32]

3.7.11 EAR-O

Cette méthode qui a été développé au laboratoire MISC de l'université Mentouri Constantine[19], elle vise à construire une ontologie à partir une base de données relationnelle et son schéma Entité-association [34] (figure 3.3). Le passage du modèle classique vers l'ontologie du domaine s'effectue via le module d'extraction d'informations en appliquant des règles de transformation (voir chapitre 6), cette méthode a été généralisée par la suite pour supporter le modèle MADS[20] qui est un modèle conceptuel entité-association saptio-temporel (elle devient don M2O pour MADS vers Ontologie).

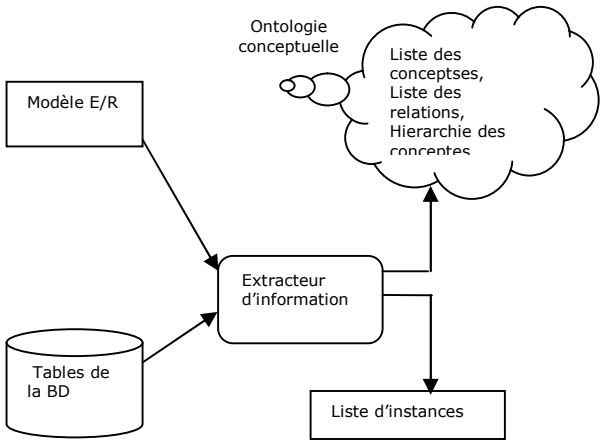


Figure 3.3 : Les étapes de la méthode EAR-O

[19] UMC : http://www.umc.edu.dz
[20] MADS :

EAR-O est basée sur trois grandes étapes, étant la transformation, la formalisation et la codification. L'étape cruciale dans le processus de développement de EAR-O c'est la transformation automatique qui se base sur un module d'extraction d'information, les deux autres étapes sont semi-automatique qui nécessitent l'intervention d'un expert dans le domaine pour enrichir le coté sémantique de l'ontologie.

3.8 Processus de construction d'une ontologie

Le processus de construction d'ontologies, appelé ingénierie ontologique, peut être décrit selon les principes qui le gouvernent, et les méthodologies et les outils qui le soutiennent [19]. Ce processus est une collaboration qui réunit des experts du domaine de connaissance, des ingénieurs de connaissances, à savoir les futurs utilisateurs de l'ontologie. Cette collaboration ne peut être fructueuse que si les objectifs du processus ont été clairement définis, ainsi que les besoins qui en découlent. Jusqu'à aujourd'hui aucun outil ou méthode ne permet de créer de façon complètement non supervisée des ontologies de bonne qualité, elle nécessite toujours l'intervention humaine pour les construire. Généralement le processus de construction d'ontologies est divisé en quatre étapes :

- **Spécification (Evaluation des besoins)**
  Le développement d'une ontologie commence par la définition du domaine et de la portée de celle-ci, cela est basée sur la réponse à certaines questions : quel est le domaine que l'ontologie va couvrir ? A quoi sert cette ontologie ? A quel type de questions les informations de l'ontologies doivent fournir des réponse ? Qui va utiliser et maintenir l'ontologie ? Etc.
- **Conceptualisation**
  La conceptualisation consiste à identifier et à structurer les connaissances d'un domaine, à partir des sources d'informations. L'acquisition de ces connaissances peut s'appuyer à la fois sur l'analyse de documents et sur l'interview avec des experts du domaine. Une fois les concepts sont identifiés par leurs termes et leurs sémantiques, l'ontologie peut être décrite dans un langage semi-formel (tables et graphes) à travers leurs propriétés, leurs instances connues et les relations qui les lient entre eux.
- **Formalisation (Ontologisation)**
  Une ontologie peut s'exprimer selon plusieurs degrés de formalisation allant des définitions les plus informelles en langage naturel aux expressions écrites en logique du premier ordre devant respecter une syntaxe et sémantique très stricte. Le degré de formalisation de l'ontologie va dépendre principalement des besoins. On peut considérer les quatre degrés cités au-dessus, Il est à retenir que les ontologies ont à être compréhensibles à la fois par les humaines et les ordinateurs **[18]**. Pour obtenir un bon équilibre entre la précision technique et la compréhensibilité il est important pour chaque définition technique de conserver une description informelle de la définition. Enfin, comme l'ontologie devra être exploitée par un ordinateur, il est nécessaire

qu'elle soit calculable. Et pour cela, il est nécessaire de l'implémenter dans un langage formel. [78]

- **Opérationnalisation**
  Transcription de l'ontologie dans un langage formel et opérationnel de représentation de connaissances du domaine adaptée à une application particulière.

3.9 Cycle de vie d'une ontologie

Puisque les ontologies sont destinées à être utilisées comme des composants logiciels dans des systèmes répondant à des objectifs opérationnels différents, leur développement doit s'appuyer sur les mêmes principes que ceux appliqués en génie logiciel. Ainsi, les ontologies doivent être considérées comme des objets techniques évolutifs et possédant un cycle de vie qui nécessite d'être précisé. Dans ce contexte, les activités liées aux ontologies sont, d'une part, des activités de gestion de projet (planification, contrôle, assurance qualité), et, d'autre part, des activités de développement (spécification, conceptualisation, formalisation) ; s'y ajoutent des activités transversales de support telles que l'évaluation, la documentation, la gestion de la configuration

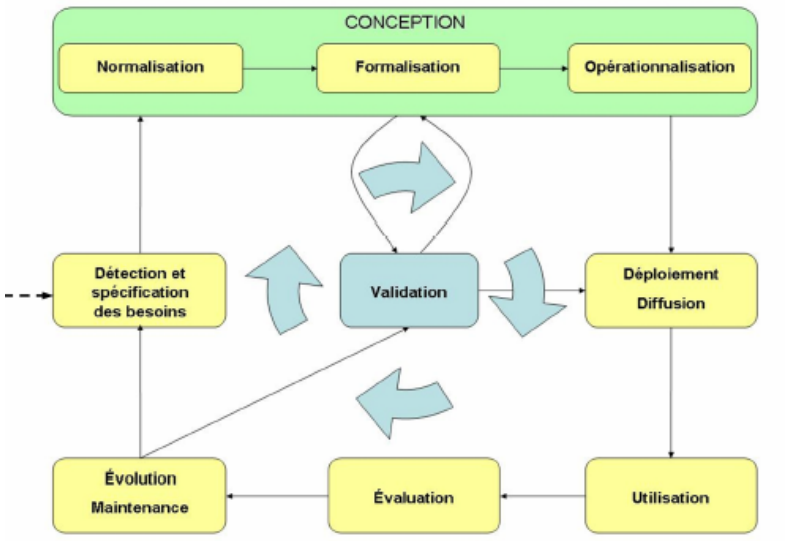


Figure 3.4 : Le cycle de vie d'une ontologie [37]

Un cycle de vie inspiré du génie logiciel est proposé dans [35] et [36], qui a été adapté à des besoins cité dans [29] suivant une vision du cycle de vie d'une ontologie (figure 3.3). Il comprend une étape initiale de détection et de spécification des besoins qui permet notamment de circonscrire précisément le domaine de connaissances, une étape de conception qui se subdivise en trois phases, une étape de déploiement et de diffusion, une étape d'utilisation, une étape, incontournable, d'évaluation, et enfin, une sixième étape consacrée à l'évolution et à la maintenance du modèle. Après chaque utilisation significative, l'ontologie et les besoins doivent être réévalués et l'ontologie peut être étendue et, si nécessaire, en partie

reconstruite. La validation du modèle de connaissances est au centre du processus et se fait de manière itérative. M. Fernandez (1997) insiste sur le fait que les activités de documentation et d'évaluation sont nécessaires à chaque étape du processus de construction, l'évaluation précoce permettant de limiter la propagation d'erreurs. Le processus de construction peut et doit être intégré au cycle de vie d'une ontologie comme indiqué en figure 3.4 [29].

3.11 Les ontologies spatiales

Les concepts qui composent une ontologie peuvent être de diverse nature, en particulier elles peuvent être des entités spatiales ; dans ce cas l'ontologie elle-même est dite *ontologie spatiale*. Les entités spatiales peuvent être particulièrement importantes dans une ontologie, en particulier celles décrivant les primitives géométriques qui sont décrite par des normes ISO/TC211, OGC, les technologies SIG, etc. et elles viennent enrichir l'aspect sémantique de l'ontologie. Les entités spatiales constituent généralement des objets de référence décrits par un ensemble d'attributs descriptifs de spatialité c'est-à-dire des attributs spatiaux et des relations spatiales. Les relations spatiales comprennent les relations topologiques d'inclusion, de superposition, de contiguïté, etc. ainsi que les relations de position relative (au-dessous, au-dessus, parallèle, perpendiculaire, etc.). Les attributs spatiaux incluent les dimensions des entités dans le système de référence : hauteur, aire, orientation, sens, etc., la forme des entités (lignes, point ou polygones) et le système de référence.

Les ontologies spatiales sont des ontologies dédiées à la description des concepts qui caractérisent l'espace comme le point, la ligne, etc... Ces ontologies sont typiquement élaborées par de grands organismes de normalisation. l'OpenGIS[21] propose par le biais de GML[22] un langage pour structurer et échanger des informations géographiques. En particulier, GML apporte un quasi-standard sur la définition des géométries des objets. Une composante temporelle est souvent nécessaire en complément pour la modélisation de l'information géographique, car les applications géographiques manient aussi très souvent des données temporelles, voire spatio-temporelles. Le modèle conceptuel MADS offre des facilités pour la modélisation des informations spatio-temporelles [39]. Les ontologies spatio-temporelles sont typiquement descriptives et peuvent s'appliquer à tout contenu nécessitant une modélisation localisée dans l'espace et le temps.

Comme toutes les ontologies, les ontologies spatiales ou géographiques peuvent être utilisées pour l'exploration, mais aussi l'extraction d'informations et au-delà pour l'interopération de SIG. Ces ontologies peuvent être de type thesaurus si elles sont limitées à la description de vocabulaire ou bien descriptives si elles incluent une description plus complexe ou

[21] www.opengis.net
[22] Geography Markup Language

sémantiquement plus riche de l'espace ou du domaine considéré. Des travaux récents tels que [38] montrent l'usage des ontologies pour résoudre l'hétérogénéité sémantique des SIG.

3.12 Ontologies et web sémantique

La démarche du Web sémantique consiste à ajouter des métadonnées aux ressources Web qui décrivent leurs contenus et leurs fonctionnalités, ces métadonnées doivent s'appuyer sur des ontologies afin de pouvoir être partagés et munies d'interprétations opérationnelles. Dans ce contexte du Web sémantique, les ontologies doivent trouver une place centrale puisqu'elles vont fournir le vocabulaire et les structures sémantiques, formelles ou non. Une ontologie peut être traitée par des systèmes d'information, utiliser des méthodes et des outils de systèmes à base de connaissances et être publiable sur le Web. La vision du Web sémantique cherche à ajouter au Web de la sémantique compréhensible pour des machines (méta information) en employant des ontologies pour pouvoir définir et organiser ce nouvel espace de méta information. Le Web sémantique cherche aussi à intégrer toutes les sources d'information sur le Web, permettant ainsi des recherches *intelligentes* et la réutilisation des structures et données. Le Web sémantique est une prolongation du Web qui permet aux ordinateurs et aux utilisateurs de travailler en collaboration [40].

Le Web sémantique intègre une communauté d'agents capables d'échanger des données et des services des sources différentes afin d'atteindre un but spécifique. Plus précisément, les services Web sont des composants logiciels réutilisables qui implémentent une fonctionnalité accessible sur le Web (par exemple, le service de réservation d'un hôtel). De plus, la description et le contenu des documents sont définis précisément grâce à une série de méta-données [41].

3.13 Conclusion

La notion d'ontologie est un élément clé pour une structuration sémantique de données portant sur un domaine particulier. Elle consiste en une spécification formelle et explicite des termes ainsi que des relations que ces termes entretiennent entre eux. C'est donc un vocabulaire formalisé de termes et de relations les liant, partagé par une communauté d'hommes ou de machines. Dans ce chapitre, nous avons donné un aperçu générale sur les ontologies : définitions, composants et méthodes de construction et dans le prochain chapitre nous allons présenter les langages de représentation et d'interrogation d'ontologies.

Chapitre 4 :

Les langages d’ontologies

4.1 Introduction

A sa création par Tim Berners Lee, au début des années 1990, le web était exclusivement destiné à partager des informations sous forme de pages html, affichables par un navigateur web, et généralement destinées à être consultés par un utilisateur humain. Très rapidement, on s'est rendu compte que cette conception du web était bien trop limitée, et ne permettait pas un réel partage du savoir : tout au plus cela permettait-il de présenter des connaissances, mais en aucun cas de les rendre directement utilisables et traitables [42].

Pour qu'un programme (logiciel) ou une machine puisse utiliser et échanger ces informations, il faut spécifier un modèle pour celles-ci. Différents langages, du simple XML au riche OWL, sont à l'heure actuelle utilisés pour modéliser l'information. Ces derniers ont été conçus pour répondre principalement aux besoins du web sémantique, qui est une nouvelle vision d'un Web non-ambiguë et compréhensible par une machine (à l'inverse du web classique) tout en gardant la compréhensibilité humaine. Ces langages permettent d'ajouter une sémantique plus ou moins riche à l'information et mènent à la création des ontologies qui peuvent être exploitées par des moteurs d'inférences et interrogées par des langages de requêtes [17].

Dans ce chapitre nous nous sommes intéressé que à la présentation des langages qui sont utilisés pour la construction et la manipulation d'ontologies et qui sont basés sur le langage XML, nous présentons aussi une partie détaillée sur la logique de description qui apporte et enrichi l'ontologie par l'aspect sémantique et donne la possibilité au moteur d'inférence de raisonner sur les classes et les individus.

4.2. Langages de construction d'ontologies

Dans cette partie, nous présentons différents langages pour créer des ontologies. Ces langages ont été conçus pour répondre principalement aux besoins du web sémantique, qui est une nouvelle vision d'un Web non-ambiguë et compréhensible par une machine (à l'inverse du web classique) tout en gardant la compréhensibilité humaine.

4.2.1 Présentation de XML : e**X**tended **M**arkup **L**anguage

XML[23] connait depuis ses débuts un succès indéniable. Défini dés son origine comme un métalangage facilitant l'élaboration de langages à balises spécialisés [42], il permet de décrire la structure arborescente de documents à l'aide d'un système de balises permettant de marquer les éléments qui composent la structure et les relations entre ces éléments. XML ne pose aucune contrainte sémantique sur la description des informations, il ne constitue donc

[23] Recommandation du W3C depuis le 10 février 1998

pas un langage de modélisation d'ontologies à lui seul [17]. Avec XML nous pouvons décrire une Personne Personne Benali âgée de 30 ans, qui habite 14 rue des martyres (figure 4.1), cet exemple correspond est compris pour une machine (ou un programme) comme une simple arborescence de chaînes de caractères.

```
<Personne id='Benali'>
<Adresse>14 rue des martyres</Adresse>
<Age>30</Age>
</Personne>
```

Figure 4.1 : Exemple de fichier XML simple

Le **XML Schéma** (XML-S) [web 11] est un outil de définition de grammaires caractérisant des arborescences de documents (notion de validité syntaxique). Avec les schémas XML, il est possible de contraindre la structure arborescente d'un document mais pas la sémantique des informations contenues dans ce document. [17]

4.2.1.1 Structure d'un document XML

Un document XML dispose de deux structures : une structure logique et une structure physique. La structure logique est conçue par le concepteur du document XML, et la structure physique est représenté sous la forme d'un fichier texte structuré en éléments, à l'aide de balises éventuellement imbriquées. En en-tête du document doit figurer un « prologue », une déclaration qui identifie le document comme un document XML. Ce prologue indique la version de XML employée, le codage de caractères, et si le document est associé à une DTD ou s'il est autonome. Il existe un élément particulier c'est l'élément « racine », encore appelé « élément document ». Cette racine doit contenir tous les autres éléments du document et ne peut apparaître qu'une fois dans un document XML. En conséquence, aucun autre élément ne contient la racine.

4.2.1.2. Propriétés et caractéristiques d'un document XML

Une source de données est un document XML si elle est « bien formée », c'est à dire si elle correspond parfaitement à la spécification de XML [43], un document XML bien formé est dit « valide » lorsqu'il est conforme à la déclaration de type de document (DTD ou XML Schema) qui l'accompagne.

Un document XML se caractérise par sa sensibilité à la casse, chaque élément doit commencer par une balise ouvrante et se termine par une balise fermante•

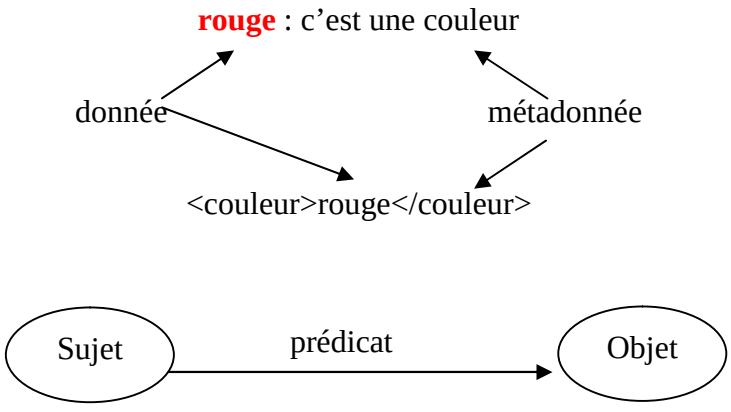
l'imbrication des éléments du document se fait sans chevauchement et la valeur d'un attribut doit être encadrée de guillemets, simples ou doubles.

Un document XML peut être transformé en utilisant des feuilles de style XSLT en un nouveau document qui est un fichier XML, évidemment, mais également tout autre format texte (pages html,fichier pdf, requête SQL, etc.). Enfin, XML propose différentes interfaces de programmation (API) qui facilitent la manipulation de documents par les processeurs XML. Les deux API principales sont Document Object Model (DOM), qui procède par construction et manipulation d'un arbre logique complet, et Simple API for XML (SAX), qui permet de débuter le traitement d'un document XML avant d'en connaître le contenu complet [42].

4.2.2 Présentation de RDF : **R**esource **D**escription **F**ramework (langages d'assertions)

Le RDF est un langage XML permettant de décrire des métadonnées (à propos de ressources) et facilitant leur traitement. Cette description est faite sous la forme d'un ensemble de triplets appelé un graphe RDF, ceci peut être illustré par un diagramme composé de noeuds et d'arcs dirigés, dans lequel chaque triplet est représenté par un lien noeud-arc-noeud (d'où le terme de "graphe" voir figure 4.2)

- Notion de méta-donnée : Donnée à propos d'une donnée ou une connaissance de plus haut niveau, utilisable par les machines.
  Exemple :



Sujet : la ressource que l'on définit
Prédicat : la propriété de la ressource, qui est une liaison étiquetée et orientée du sujet vers l'objet.
Objet : la valeur de la propriété pouvant être une autre ressource ou bien un littéral.

Figure 4.2 : Triplet RDF

4.2.2.1 Syntaxe de RDF

RDF n'est, en soi, qu'un modèle de graphes à arcs orientés et labellés, dont le label est le « prédicat » (ou « propriété »), le noeud de départ le « sujet », et le noeud cible « objet », mais on peut le décrire par la syntaxe XML, on parle alors de RDF/XML [24] qui sera appelé par la suite directement par RDF.
En prenant l'exemple précédent défini pour XML, nous pouvons définir avec RDF (figure 4.3) un graphe (figure 4.4) qu'un programme comprendra comme un concept Toto qui possède les propriétés adresse et age.

```
<rdf :RDF>
<rdf :Description about='Benali'>
<rdf :Property about='adresse'>
14 rue des martyres
</rdf :Property>
<rdf :Property about='age'>
30
</rdf :Property>
</rdf :Description>
</rdf :RDF>
```

Figure 4.3 : Représentation RDF/XML de Benali 30 ans qui habite au 14 rue des martyres

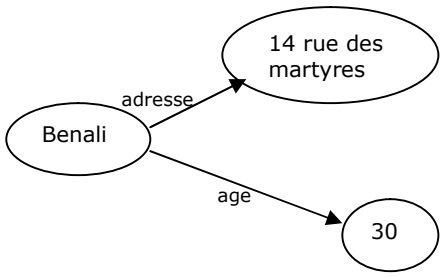


Figure 4.4 : Représentation graphique de Benali 30 ans qui habite au 14 rue des martyres

Si RDF fournit une capacité d'échange de connaissances, il ne permet pas à l'utilisateur de définir le vocabulaire des termes à utiliser, ni d'établir la sémantique des objets utilisés, Il est donc nécessaire, pour donner un sens aux informations stockées sous forme de triplets RDF, de se donner un vocabulaire, de définir la signification d'une propriété, ainsi que son type, son champ de valeurs, etc. C'est le rôle de RDF Schema, qui permet de créer des vocabulaires de métadonnées.

4.2.2.2 RDF Schema : RDFS

RDFS [web 12] est un langage permettant de définir des propriétés sémantiques pour les ressources par un schéma. Dans un schéma on peut définir de nouvelles ressources comme des spécialisations d'autres ressources. Les schémas contraignent aussi le contexte d'utilisation des ressources. Avec RDFS de nouvelles notions sémantiques apparaissent. La principale est la distinction entre une classe (concept d'une ontologie)

[24] Recommandation W3C depuis 10/02/2004

et une instance (individu d'une ontologie). Voici d'autres notions définies dans RDFS :

– La notion de classe (rdfs : Class) qui peut être rapprochée de la notion de concept d'une ontologie.

– La notion de sous-classe (rdfs : subClassOf) qui est une spécialisation d'une classe déjà définie.

– La notion de type (rdf : Type) : les instances d'une classe propriété et sous propriété (prédicat ou rôle de l'ontologie)

– Les notions de source (rdfs :domain) et de cible (rdfs :range) d'une propriété.

La hiérarchie de subsomption ou taxinomie peut être décrite en utilisant la propriété rdfs :subClassOf. Cette hiérachie définit les relations de spécialisation d'une classe Ressource par rapport à une classe Objet. Cette relation s'applique également aux relations par la propriété rdfs :SubPropertyOf.

Soit P une propriété de rdfs :range R et de rdfs :domain D, alors une propriété S est une rdfs :SubPropertyOf de P si : le rdfs :range de S est R ou une sous-classe de R et le rdfs :domain de S est D ou une sous-classe de D [17].

Nous avons vu que RDF et RDFS permettent de définir sous la forme d'un graphe de triplets des données ou des méta-données. Cependant, il est impossible de raisonner sur ces représentations car

la sémantique (hors subsomption) reste très limitée. C'est ce manque de sémantique que OWL comble par l'apport d'un vocabulaire plus riche [17].

4.2.3 Présentation OWL (**O**ntology **W**eb **L**anguage)

OWL [web 13] est un langage fondé sur la syntaxe RDF/XML et héritier des travaux de DAML+OIL [44][25]. OWL n'est pas simplement une extension de RDF, il introduit l'aspect sémantique lui manquant, comme les outils de comparaison de propriétés et de classes (identité, équivalence, contraire, cardinalité, symétrie, etc). Ainsi par ses primitives plus riches, OWL offre une capacité d'interprétation plus grande aux machines (programmes) que RDF et RDFS. Cette partie ne va pas reprendre tous les détails de OWL, mais uniquement les plus importants. Pour des explications exhaustives du rôle de chacun des éléments composant le vocabulaire d'OWL, il est recommandé de se reporter à la Recommandation du W3C « OWL Web Ontology Language Reference » [45].

[25] DAML+OIL: **D**arpa **A**gent **M**arkup **L**anguage et **O**ntology **I**nference **L**ayer. DAML est utilisé pour l'aspect description de connaissance et OIL pour le côté raisonnement, parmi les auteurs de ces travaux on peut citer Ian Horrocks et Tim Berners-Lee

Plus un outil est complet, plus il est, en général, complexe. C'est cet écueil qu'a voulu éviter le groupe de travail WebOnt du W3C en décomposant OWL en trois sous-langages offrant des capacités d'expression croissantes (figure 4.5) et, destinés à des communautés différentes d'utilisateurs : [42]

Les 3 niveaux d'OWL présentent une hiérarchie sur la validité des ontologies :
– une ontologie OWL Lite valide est également une ontologie OWL DL valide
– une ontologie OWL DL valide est également une ontologie OWL Full valide

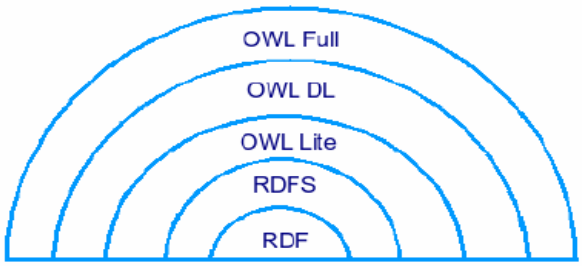


Figure 4.5 : Les sous-langages de OWL

4.2.3.1. OWL Lite est le sous langage de OWL le plus simple car son utilisation est contrainte, Il est destiné aux utilisateurs qui n'ont besoin que d'une hiérarchie de concepts simple. Le tableau 4.1, montre la liste des constructeurs proposés par OWL Lite.

Catégories	Constructeurs
RDF Schema	Class, rdfs :subClassOf, rdf :Property, rdfs :subPropertyOf, rdfs :domain, rdfs :range, Individual
(In) Equalité	equivalentClass, equivalentProperty, sameAs, differentFrom, AllDifferent, distinctMembers
Restrictions	onProperty, allValuesFrom, someValuesFrom
Cardinalités (0 ou 1)	minCardinality, maxCardinality, cardinality
Intersection	intersectionOf
Propriétés	SymmetricProperty,FunctionalProperty, ObjectProperty, DatatypeProperty, inverseOf, TransitiveProperty, InverseFunctionalProperty

Tableau 4.1: Liste des constructeurs proposés par OWL Lite

Le tableau 4.2 montre un exemple illustrant ces constructeurs sur une ontologie décrite en OWL Lite :

Catégories	Exemple
RDF Schema	Personne : subClassOf (Thing) Femme : subClassOf (Personne) Enfant : subClassOf (Personne) Fille : equivalentClass (intersectionOf(Femme, Enfant)) Parent : intersectionOf (Personne, Restriction (minCardinality(1), onProperty (aEnfant))) aParent : ObjectProperty (Enfant, Personne) aEnfant : inverseOf (aParent)
(In) Equalité	
Intersection	
Restrictions	
Cardinalités	
Propriétés	

Tableau 4.2: Exemple sur les constructeurs de OWL-Lite

4.2.3.2. OWL DL est plus complexe que OWL Lite, permettant une expressivité bien plus Importante, il est fondé sur la logique de description SHIN(D) [17] (d'où son nom OWL Description Logics). Malgré sa complexité OWL DL garantit la complétude des raisonnements (calculabilité des inférences) et leur décidabilité (leur calcul se fait dans un temps fini). Voici la liste des constructeurs ajoutés par OWL DL par rapport OWL Lite :

Catégories	Constructeurs
Axiome de Classe	oneOf (enumération), dataRange, disjointWith
Expressions booléennes	unionOf, complementOf
Cardinalités (0,n) minCardinality, Individu cible d'une propriété	minCardinality, maxCardinality, cardinality hasValue

Tableau 4.3 : Liste des constructeurs proposés par OWL DL

Le tableau 4.4 montre un exemple illustrant ces constructeurs sur une ontologie décrite en OWL DL :

Catégories	Constructeurs
Axiome de Classe	Gender : oneOf(Male, Female)
Expressions booléennes	Tante : intersectionOf (Femme , unionOf (aNeuve, aNiece))
Individu cible d'une propriété	Homme : intersectionOf (Personne, hasValue(sexe,Male))

Tableau 4.4 : Exemple sur les constructeurs de OWL-DL

4.2.3.3. OWL Full est la version la plus complexe d'OWL, mais également celle qui permet le plus haut niveau d'expressivité. Elle est destiné aux situations où il est plus important d'avoir un haut niveau de capacité de description. Son utilisation n'est contrainte que par le langage RDF, mais elle ne garantit pas la complétude et la décidabilité des calculs liés à l'ontologie. Cependant OWL Full offre des mécanismes intéressants, comme par exemple la possibilité d'étendre le vocabulaire par défaut de OWL [42].

Nous avons vu que OWL est un langage basé sur XML, qui nous permet de décrire de manière riche des informations. La prochaine partie présente les logiques de description qui forment la base théorique des langages d'ontologie comme OWL.

4.3. Les logiques de description

4.3.1 Introduction

Dans les dernières années, un courant de recherche très actif sur les systèmes à base de connaissances s'est développé et qui s'est nourri d'études effectuées sur la logique des prédicats, les réseaux sémantiques et les langages de frames, a donné naissance à une famille de langages de représentation appelés logiques de descriptions, ou encore logiques terminologiques qui ont été développées comme une extension des frames et des réseaux sémantiques, qui ne possédaient pas de sémantique formelle basée sur la logique. Deux principaux objectifs ont été fixés par Ces langages de représentation, le premier c'est de fournir un moyen pour décrire le monde et le deuxième les outils pour raisonner sur des connaissances.

Une logique de descriptions permet de représenter les connaissances d'un domaine d'application d'une manière formelle et structurée à l'aide de ''descriptions'' qui peuvent être des concepts (classes d'individus), de rôles (relations binaires entre classes) et d'individus. À l'instar de la logique classique, une sémantique formelle définie en logique du premier ordre est associée aux concepts, aux rôles et aux individus par l'intermédiaire d'une interprétation (**voir paragraphe Interprétation**). La relation de subsomption permet d'organiser les concepts et les rôles en hiérarchies sur lesquelles opèrent les processus de classification et d'instanciation, ces dernières sont les opérations qui sont alors à la base du raisonnement sur les descriptions, ou raisonnement terminologique. La classification s'applique aux concepts, le cas échéant aux rôles, et permet de déterminer la position d'un concept et d'un rôle dans leurs hiérarchies respectives, tandis que l'instanciation permet de retrouver les concepts dont un individu est susceptible d'être une instance[26].

Les logiques de description sont utilisées pour de nombreuses applications, telles que le web sémantique (représentation d'ontologies, et recherche d'information basée sur la logique), médecine/bioinformatique (représenter et gérer des ontologies biomédicales), Ingénierie de processus (représenter des descriptions de service) Ingénierie de la connaissance (représenter des ontologies) et l'Ingénierie logicielle (représenter la sémantique des diagrammes de classe UML) [web 14]

4.3.2 Petit historique des logiques de description

Le calcul des prédicats du premier ordre, ou calcul des relations, ou logique du premier ordre, ou tout simplement calcul des prédicats est une formalisation du langage des mathématiques proposée par les logiciens de la fin du XIX^{e} siècle et du début du XX^{e} siècle [web 14] et comme ses prédécesseurs, les schémas (ou frames) et les réseaux sémantiques, les logiques de description (LD) ont pour but d'obtenir une représentation plus proche de la représentation humaines que les formules de la logique des prédicats. Néanmoins, toute formule écrite en LD, peut être réécrite en formule de la logique des prédicats (LP). Un concept atomique ***A*** correspond à un prédicat unaire ***A***(x), un rôle ***R*** à un prédicat binaire ***R***(x,y), un individu correspond à une constante et un concept composé à une formule à une variable libre[27] ***C***(x) [17].

Les premiers travaux sur les LD commencèrent au début des années 1980 avec des systèmes à base de connaissances tels que KL-ONE, BACK et LOOM [22]. Ces premières implantations résolvent des problèmes d'inférence en temps souvent polynomial, par le biais d'une catégorie d'algorithmes de vérification de subsomption de type normalisation/comparaison.

[26] Au contraire des langages orientés objets, l'instanciation désigne l'opération qui permet d'engendrer des instances à partir d'une classe.

[27] Lorsqu'une variable ***x*** appartient à une sous-formule précédée d'un quantificateur, $\forall x$ ou $\exists x$, elle est dite **liée** par ce quantificateur. Si une variable n'est liée par aucun quantificateur, elle est **libre** [web 14].

Ces algorithmes ne s'appliquent qu'à des LD peu expressives, sans quoi ils sont incomplets, c'est-à-dire qu'ils sont incapables de prouver certaines formules vraies.

Dans les années 1990, une nouvelle classe d'algorithmes est apparue : les algorithmes de vérification de satisfiabilité à base de tableaux (tableau-based algorithms). Ces derniers raisonnent sur des LD dites expressives ou très expressives, mais en temps exponentiel. Cependant, en pratique, le comportement des algorithmes est souvent acceptable. L'expressivité accrue a ouvert la porte à de nouvelles applications telles que le Web sémantique. Le terme logiques de description expressives (LDE) désigne l'ensemble des LD qui ont émergé pendant cette période [22].

4.3.3 Rapport avec la logique du premier ordre

Les logiques de description sont une famille de langages de représentation de connaissance qui peuvent être utilisés pour représenter la connaissance terminologique d'un domaine d'application d'une manière formelle et structurée. Le nom de logique de description se rapporte, d'une part à la description de concepts utilisée pour décrire un domaine et d'autre part à la sémantique basée sur la logique qui peut être donnée par une transcription en logique des prédicats du premier ordre [web 14].

Un langage de description est généralement une variante de la logique du premier ordre, cette dernière est une logique plus expressive que la plupart des logiques propositionnelles, cette expressivité s'exprime par le remplacement des variables propositionnelles par des formules plus élaborées décrivant les propriétés de valeurs dans un domaine d'intérêt ce qui nous aboutit à la *logique du premier ordre*, ou *calcul des prédicats*.

Une correspondance existe entre la logique $\mathcal{AL}$ et la logique de premier ordre [22]. Un concept atomique **A** correspond à un prédicat unaire $\phi_A(x)$, un rôle **R** à un prédicat binaire $\phi_R(x,y)$, un individu correspond à une constante et un concept composé à une formule à une variable libre $\phi_C(x)$. Le tableau 4.5, illustre la façon de traduire les constructeurs d'$\mathcal{AL}$ en leur équivalent en logique des prédicats.

$\phi_{\neg C}(x)$	$= \neg\phi_C(x)$
$\phi_C(x)$	$= \phi_C(x) \wedge \phi_D(x)$
$\phi_C(x)$	$= \phi_C(x) \vee \phi_D(x)$
$\phi_{\exists R.C}(y)$	$= \exists x, R(x,y) \wedge \phi_C(x)$
$\phi_{\forall R.C}(y)$	$= \forall x. R(x,y) \rightarrow \phi_C(x)$

Tableau 4.5: Correspondance entre les constructeurs d'$\mathcal{AL}$ et la logique des prédicats

4.3.4 Formalisme de Description

La modélisation des connaissances d'un domaine avec les LDs se réalise en deux niveaux. Le premier, le niveau terminologique ou TBox, décrit les connaissances générales d'un domaine alors que le second, le niveau Assertionnel ou ABox, représente une instanciation spécifique. Une TBox comprend la définition des concepts et des rôles, alors qu'une ABox décrit les individus[28] et l'ensemble d'assertions qui portent sur ces individus. Plusieurs ABox peuvent être associés à une même TBox ; chacune représente une configuration constituée d'individus, et utilise les concepts et rôles de la TBox pour l'exprimer [46]. Par exemple : *Tout employé est une personne* (Employe $\subseteq$ Personne) sera définie dans la TBox, alors que *Ahmed est un employé* (Ahmed : Employe) sera définie dans la ABox.

4.3.4.1 Le niveau terminologique (TBox ou boite terminologique)

Le niveau terminologique (Terminology Box, ou Taxonomy Box) représente la composante intensionnelle ou la connaissance en intention du monde, il contient la modélisation du monde en termes de concepts, de leurs propriétés, et des relations entre les concepts.

- **Définition de quelques notions (concepts) de base**

- Entités atomiques

Les concepts et les rôles atomiques constituent les entités élémentaires d'une TBox. Les noms débutant par une lettre majuscule désignent les concepts, alors que ceux débutant par une lettre minuscule dénomment les rôles (par exemple : les concepts *Femelle*, *Mâle*, *Homme* et *Femme*, et le rôle *relationParentEnfant*).

- Concepts et rôles atomiques prédéfinis

La convention ontologique des logiques de description représente les concepts comme des ensembles d'individus définis en intention. Formellement, Δ représente l'ensemble des individus d'un domaine concret, donc un concept est un sous-ensemble de Δ. Il en découle que deux concepts constitués des mêmes individus sont égaux, ou du moins équivalents.

Le concept constitué de tous les individus de Δ s'appelle Top, et est noté $\top$. Le concept qui n'est constitué d'aucun individu s'appelle Bottom, et est noté $\bot$. Cette convention ontologique définit de la même façon le rôle $\top_R$ et le rôle $\bot_R$.

[28] Un individu est une assertion de concepts identifié par un nom unique et pouvant avoir des rôles avec d'autres individus.

- Entités composées

Les concepts et rôles atomiques peuvent être combinés au moyen de constructeurs pour former respectivement des concepts et des rôles composés. Par exemple, le concept composé *Mâle*∩*Femelle* résulte de l'application du constructeur ∩ aux concepts atomiques *Mâle* et *Femelle*. Le concept *Mâle*∩*Femelle* s'interprète comme l'ensemble des individus qui appartiennent aux concepts *Mâle* et *Femelle*.

- La notation

Dans cette partie de ce chapitre, nous adoptons la notation suivante : R dénote un rôle, C,D des concepts composés et A,B des concepts atomiques.

- L'interprétation

Expliciter formellement la sémantique d'une TBox, requiert de définir préalablement la notion d'interprétation. Une interprétation $\mathcal{I}$ se compose d'un domaine Δ^I d'interprétation et d'une fonction d'interprétation $.^I$. Le domaine d'interprétation consiste en un ensemble d'individus. La fonction d'interprétation assigne à chaque concept atomique **A** un ensemble tel que $A^I \subseteq \Delta^I$, et à chaque rôle atomique **R** une relation binaire $R^I \subseteq \Delta^I \times \Delta^I$

L'interprétation dans un monde fermé (Closed-World Assumption) est l'interprétation la plus courante, on la retrouve notamment dans les bases de données. Cette interprétation suppose qu'il n'existe pas d'instance, ni de relation, autre que celles définies au niveau Assertionnel, et donc une information dont on ignore l'existence est désignée comme fausse.

Dans les logiques de description, l'interprétation des connaissances respecte l'hypothèse du monde ouvert (Open-World Assumption). Par opposition au monde fermé, le monde ouvert suppose qu'il peut exister des instances ou des relations qui ne sont pas définies dans la Abox, ainsi une information dont on ignore l'existence n'est pas forcément fausse, elle est inconnue [17].

- **Syntaxe (Terminologie)**

Les éléments importants du domaine sont décrits par des expressions établies à partir de concepts primitifs (relation unaire) et des rôles primitifs (relations binaires).

Des constructeurs permettent de construire des concepts composés à partir d'autres concepts, l'ensemble de ces constructeurs est celui qui définit le langage $\mathcal{AL}$ (Voir paragraphe La famille $\mathcal{AL}$) :

Le concept TOP ($\top$) représente le concept le plus général ou universel

Le concept vide ($\bot$) c'est le plus spécifique

La négation atomique de concept : $\neg C$

L'intersection de concepts : $C \cap D$

la quantification universelle de rôle : $\forall r.C$
La quantification existentielle limitée de rôle : $\exists r.C$

Les DL prédéfinissent également la subsomption (noté $\subseteq$) qui est une relation transitive hiérarchique entre des concepts. Cette notion est proche de la relation "est impliqué par" en logique classique, de la relation "contient" en logique ensembliste, ou encore la notion d'héritage" en conception objet.

- **Sémantique (Formel)**

Une TBox contient des axiomes terminologiques de la forme $C \subseteq D$ ou $C \equiv D$. La première sert à énoncer des relations d'équivalence entre concepts, alors que la seconde permet d'exprimer des relations d'inclusion. Une interprétation $\mathcal{I}$ satisfait un axiome $C \subseteq D$ si et seulement si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Une interprétation $\mathcal{I}$ satisfait un axiome $C \equiv D$ si et seulement si $C^{\mathcal{I}} \equiv D^{\mathcal{I}}$.
NB : Une interprétation satisfait une TBox (est un modèle[29] de TBox) si et seulement si l'interprétation satisfait tous les axiomes de la TBox.
Le tableau 4.6 montre la sémantique des différents constructeurs utilisés dans $\mathcal{AL}$

$T^{\mathcal{I}}$	$= \Delta^{\mathcal{I}}$
$\bot^{\mathcal{I}}$	$= \phi$
$(\neg A)^{\mathcal{I}}$	$= \Delta^{\mathcal{I}} \setminus T^{\mathcal{I}}$
$(C \cap D)^{\mathcal{I}}$	$= C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$(\forall R,C)^{\mathcal{I}}$	$= \{ a \in \Delta^{\mathcal{I}} / \forall b,(a,b) \in \mathcal{R}^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}} \}$
$(\exists R,T)^{\mathcal{I}}$	$= \{a \in \Delta^{\mathcal{I}} / \exists b,(a,b) \in \mathcal{R}^{\mathcal{I}} \}$

Tableau 4.6 : La sémantique formelle d'$\mathcal{AL}$

4.3.4.2 Le niveau Assertionnel (ABox ou boite d'assertions)

Une ABox (Assertional Box) représente la composante extensionnelle ou la connaissance en extension, elle contient un ensemble d'assertions sur les individus, des assertions d'appartenance et des assertions de rôle. Chaque ABox doit être associée à une TBox, car les assertions s'expriment en terme de concepts et de rôles de la TBox.

Une ABox désigne des individus caractérisés par des assertions d'individus nommés [46]. L'exemple du tableau 4.7 comprend les individus nommés suivants : Pierre et Sophie.

[29] On appellera modèle de la logique de description, une interprétation (du niveau terminologique étendue aux individus) telle que les faits du niveau assertionnel soient vérifiés dans cette interprétation **[3LD.4]**.

TBox	ABox
Sexe : {Male, Femelle} Humain sexe(Humain, Sexe) Parent ⊆ Humain ∩ aEnfantHumain Homme ⊆ Humain ∩ sexe.Male Femme ⊆ Humain ∩ sexe.Femelle Pere ⊆ Homme ∩ Parent Mere ⊆ Femme ∩ Parent Parent ≡ Pere Mere estFrereDe(Homme,Humain)$^+$ estSoeurDe(Femme,Humain)$^+$ estFrereDe(Homme, Femme)- ≡ estSoeurDe(Femme,Homme) Oncle ⊆ Homme ∩ ∃estFrereDe.Parent Tante ⊆ Homme ∩ ∃estSoeurDe.Parent	Pierre (Humain) Sophie (Femme) Homme (Jean) Mere (Marie) Femme (Catherine) aEnfant(Marie, Catherine) sexe(Pierre, Male) aEnfant(Sophie, Pierre) estSoeurDe(Sophie, Jean)

Tableau 4.7 : Une base de connaissances composée d'une TBox et d'une ABox

- Chaque assertion d'appartenance d'une ABox (notée C(a)), déclare que pour cette ABox, il existe un individu nommé **a**, membre du concept C de la TBox associée. Une interprétation satisfait une assertion d'appartenance C(a) si et seulement si $a^I \in C^I$.
- Une assertion de rôle, de la forme R(a, b) indique que pour cette ABox, il existe un individu nommé **a** qui est en relation avec un individu nommé **b** par le rôle **R** (défini dans la TBox associée), tel que **a** fait partie du domaine de R et b fait partie de l'image. Une interprétation satisfait une assertion de rôle R(a, b) si et seulement si $(a^I, b^I) \in R^I$.

NB : Finalement, une interprétation $\mathcal{I}$ satisfait une ABox A (est un modèle d'ABox) si et seulement si $\mathcal{I}$ satisfait toutes les assertions de A.

4.3.5 L'inférence

L'inférence s'effectue au niveau terminologique ou factuel, la figure 4.6 montre la structure générale d'un système LD.

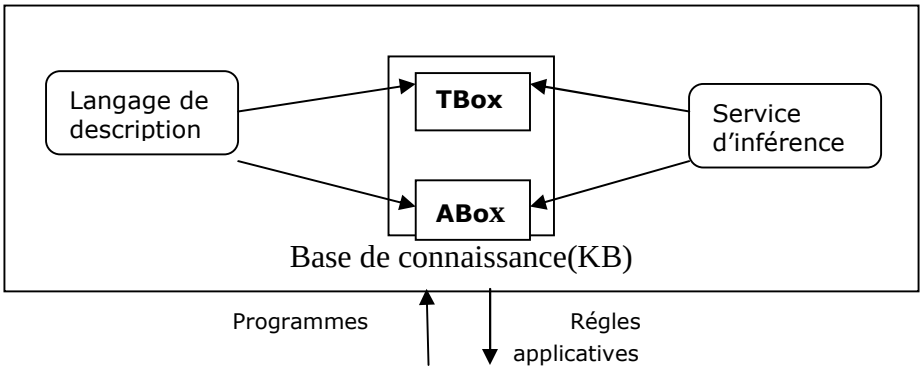


Figure 4.6 : Structure générale de système LD

4.3.5.1 L'inférence au niveau terminologique

Avoir une base de connaissances, c'est bien. Pouvoir raisonner dessus, c'est encore mieux [47]. Raisonner sur la TBox, c'est essentiellement tester la subsomption entre deux concepts, mais il y a quatre principaux problèmes d'inférence qui se présentent [22]:

- **Satisfiabilité** : Trouver tous les concepts insatisfiables (par exemple : $C \cap \neg C$), Un concept **C** d'une terminologie $\mathcal{T}$ est satisfiable si et seulement existe un modèle $\mathcal{I}$ de $\mathcal{T}$ tel que $C^{\mathcal{I}} \neq \varnothing$
- **Subsomption** : Calculer la hiérarchie de subsomption ou taxonomie des concepts, un concept C est subsumé par un concept D pour une terminologie si et seulement si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ pour tout modèle $\mathcal{I}$ de $\mathcal{T}$.
- **Équivalence** : Trouver les concepts équivalents, un concept C est équivalent à un concept D pour une terminologie si et seulement si $C^{\mathcal{I}} \equiv D^{\mathcal{I}}$ pour chaque modèle $\mathcal{I}$ de $\mathcal{T}$.
- **Incompatibilité (Disjonction)** : Trouver les concepts disjoints, deux concepts C et D sont incompatibles (disjoints) si seulement si $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \varnothing$ pour toute interprétation $\mathcal{I}$.

Les moteurs d'inférence actuels tirent généralement profit du fait que les quatre types de problèmes d'inférence peuvent être réduits à des problèmes de subsomption ou à des problèmes de satisfiabilité. Les figures 4.7 et 4.8 illustrent cette propriété qui implique, que les moteurs d'inférence des LDs ne nécessitent souvent qu'un seul algorithme d'inférence pour raisonner au niveau terminologique. D'ailleurs, les deux grandes classes d'algorithmes de raisonnement pour les LDs (algorithmes de subsomption de type normalisation/comparaison et algorithmes de vérification de satisfiabilité à base de tableaux) correspondent aux façons de réduire respectivement des problèmes d'inférence à des problèmes de subsomption et de satisfiabilité [22].

C est insatisfiable	$\Leftrightarrow$	C est subsumé par $\perp$
C et D sont équivalents	$\Leftrightarrow$	C est subsumé par D, et D par C
C et D sont disjoints	$\Leftrightarrow$	$C \cap D$ est subsumé par $\perp$

Figure 4.7 : Réduction des problèmes d'inférence d'une TBox à des problèmes de subsomption

C est subsumé par D	$\Leftrightarrow$	$C \cap \neg D$ est insatisfiable
C et D sont équivalents	$\Leftrightarrow$	$C \cap \neg D$ $\neg C \cap D$ sont insatisfiables
C et D sont disjoints	$\Leftrightarrow$	$C \cap D$ est insatisfiable

Figure 4.8 : Réduction des problèmes d'inférence d'une TBox à des problèmes de satisfiabilité

Et finalement, la subsomption peut être réduite à un problème de satisfiabilité et réciproquement :

$C \cap \neg D$ est non satisfiable $\Leftrightarrow C \subseteq D$ $\neg C \subseteq \perp \Rightarrow C$ est satisfiable

4.3.5.2 L'inférence au niveau Assertionnel

Le niveau assertionnel est défini par un ensemble d'individus désignant des objets nécessairement différents dans toute interprétation et d'instances de concepts et de relations vérifiées par ces individus. Les principaux problèmes d'inférence qu'ils peuvent être posés sont :

- **Cohérence de la ABox** : Les assertions définies restent cohérentes avec la TBox. Une ABox **A** est cohérente par rapport à une TBox **T** si et seulement s'il existe un modèle $\mathcal{I}$ de A et $\mathcal{T}$.
- **Vérification d'instance** : Vérifier que chaque instance respecte la définition de son concept, autrement dit, c'est vérifier par inférence si une assertion C(a) est vraie pour tout modèle $\mathcal{I}$ d'une ABox **A** et d'une TBox $\mathcal{T}$.
- **Vérification de rôle** : Vérifier si les rôles de l'instance sont correctement utilisés, autrement dit, c'est vérifier par inférence si une assertion R(a, b) est vraie pour tout modèle $\mathcal{I}$ d'une ABox **A** et d'une TBox $\mathcal{T}$.
- **Problème de récupération** : Trouver le concept le plus spécifique pour chaque instance. Pour une ABox **A**, un concept C d'une terminologie $\mathcal{T}$, infère les individus $a^{\mathcal{I}}_1 \ldots a^{\mathcal{I}}_n \in C^{\mathcal{I}}$ pour tout modèle $\mathcal{I}$ de $\mathcal{T}$.

  Il existe de très nombreux moteurs d'inférences (Racer, Pellet, FaCT, FaCT++, Flora, Jena, Jess, Bossam...) pour raisonner sur des logiques de description. La plupart acceptent en entrée des fichiers OWL. Par exemple, Racer le moteur d'inférence sans doute le plus connu, commercialisé par *Racer Systems GmbH & Co. KG*, fondé en 2004 par des chercheurs qui travaillaient à l'université de Hambourg, travaille sur les ontologies modélisées par son langage, mais accepte des ontologies décrites en RDF ou OWL, ces dernières étant traduites vers le langage utilisé par Racer. Ce moteur d'inférence possède également son propre langage de requête nRQL (new Racerpro query Language) pour interroger les ontologies sur la ABox et la TBox [17].

4.3.6 La famille des langages LD : Les extensions d'$\mathcal{AL}$ (*Attributive Language*)

Le nom de chaque logique de description permet d'identifier, les constructeurs et rôles utilisables par celle-ci. Chaque logique de description se base au minimum sur une

logique minimale nommée $\mathcal{AL}$ [30] (Attribu tive Language) qui a été introduite par Schmidt-Schauß et Smolka en 1991. Cette logique est minimale, dans le sens où elle est la moins expressive, c'est-à-dire qu'elle propose le moins nombre de constructeurs et qui forme le préfixe de nom de toutes les logiques de description, par exemple les lettres suivantes qui sont dans les crochets forment les extensions de $\mathcal{AL}$ *[U] [E] [C] [N] [R].* Certains de ces langages sont inclus dans d'autres, par exemple $\mathcal{ALUE} \equiv \mathcal{ALC}$ et $\mathcal{ALE} \subseteq \mathcal{ALR}$ [47].

Il y a une possibilité de construire 32 langages différents à partir de $\mathcal{AL}$, enrichi par les 5 constructeurs possibles (tableau 4.9) [47]. Par exemple, la logique $\mathcal{AL}$, enrichie de l'union ($\mathcal{U}$) et de la quantification existentielle complète ($\mathcal{E}$), se nomme $\mathcal{ALUE}$. La différence entre $\mathcal{AL}$ et $\mathcal{ALC}$ vient du constructeur de négation complète (C). Comme il est noté au-dessus que l'appellation $\mathcal{ALC}$ équivaut à $\mathcal{ALUE}$, puisque l'union et la quantification existentielle complète s'expriment par la négation complète et inversement, car $\mathbf{C \cup D \Leftrightarrow \neg(\neg C \cap \neg D)}$ **et** $\mathbf{\exists r.C \Leftrightarrow \neg\forall r.\neg C}$ [22].

A partir de $\mathcal{AL}$ ou $\mathcal{ALC}$ (Voir Tableau 4.8 la différence entre $\mathcal{AL}$ et $\mathcal{ALC}$), il est possible d'obtenir une logique de description étendue par l'ajout de constructeurs de concepts ou de rôles. Il existe trois façons d'étendre $\mathcal{ALC}$: la première est d'ajouter des constructeurs de concepts, la seconde d'ajouter des constructeurs de rôles et enfin d'énoncer des contraintes sur l'interprétation des rôles [22].

Syntaxe des constructeurs	$\mathcal{AL}$	$\mathcal{ALC}$
La restriction existentielle complète		$\exists$ r.C
La restriction existentielle limitée	$\exists$ r. $\top$	
La restriction universelle	$\forall$ r.C	
Le concept Atomique	C	
Le concept universel	$\top$	
Le concept le plus spécifique	$\bot$	
La négation atomique	$\neg$C	
La conjonction	C $\cap$ A	

Tableau 4.8 : Différence entre $\mathcal{AL}$ et $\mathcal{ALC}$

Il est également possible d'étendre une logique de description par ajout de contraintes sur l'interprétation des rôles. Le rôle transitif R^+ en est un exemple. Il permet d'exprimer des rôles transitifs tels que **estFrereDe** (si **A** et **B** sont des **Hommes**, que **A estFrereDe B** alors **B estFrereDe A**). La spécification d'un ensemble de rôles transitifs se nomme $\mathcal{NR}_+$.

[30] Le langage AL s'appuie sur les langages FL et FL-, qui sont les langages pour lesquels ont été établis les premiers résultats théoriques sur les logiques de descriptions [48]

Enfin la lettre S désigne la logique $\mathcal{ALC}$ additionnée de R^+, OWL fait partie de cette famille de LD, puisqu'il est basé sur la logique de description $\mathcal{SHOIN(D)}$ [17].

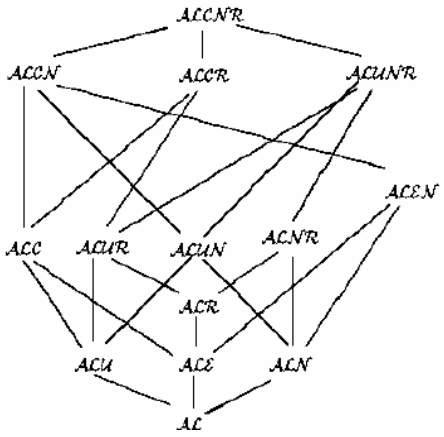


Figure 4.9 : Treillis d'inclusion des langages $\mathcal{AL}$

Extensions de $\mathcal{AL}$ $\mathcal{AL}$ = { ⊤, ⊥, ¬A, C ∩ D, ∀r.C, ∃ r}	**Description**
$\mathcal{ALC}$ = $\mathcal{AL}$ ∪ {¬C}	Négation de concepts primitifs ou définis, qui est notée par (¬C)
$\mathcal{ALU}$ = $\mathcal{AL}$ ∪ {C ∪ D}	Disjonction de concepts, qui est notée par (C ∪D) - Notons les équivalences : ⊤ ≡ C ∪ ¬C et C ∪ D ≡ ¬(¬C ∪ ¬D)
$\mathcal{ALE}$ = $\mathcal{AL}$ ∪ {∃r.C}	Quantification existentielle typée, qui est notée par (∃r.C) - La quantification existentielle typée ∃r.C introduit un rôle r de co-domaine C et impose l'existence d'au moins un couple d'individus (x,y) en relation par l'intermédiaire du rôle r, où C est le type de y. - Notons les équivalences ∃ r ≡ ∃r.⊤ et ∃r.C ≡ ¬(∀r.¬C).
$\mathcal{ALN}$ = $\mathcal{AL}$ ∪ {≥n r, ≤n r}	Cardinalité sur les rôles, qui est notée par (≥n r) et (≤n r) - Les constructeurs ≥n r et ≤n r fixent la cardinalité (nombre de valeurs élémentaires) minimale et maximale du rôle auquel ils sont associés. En particulier, la construction (∃ r) est équivalente à la construction (≥1 r).
$\mathcal{ALR}$ = $\mathcal{AL}$ ∪ {r1 ∩ r2}	Conjonction de rôles, qui est notée par (r1 ∩ r2), les rôles r1 et r2 étant primitifs.

Tableau 4.9 : Extensions de $\mathcal{AL}$ [48]

4.3.7 Ontologies et Logiques de description

Les LD utilisent une approche ontologique, c'est-à-dire que pour décrire les individus d'un domaine, elles requièrent tout d'abord la définition des catégories générales d'individus et les relations logiques que les individus ou catégories peuvent entretenir entre eux. Cette approche ontologique est naturelle pour le raisonnement puisque même si

la majorité des interactions se déroulent au niveau des individus, la plus grande partie du raisonnement se produit au niveau des catégories [49].

Comme nous allons le voir dans le paragraphe des langages d'ontologies, OWL est considéré comme un standard de langage d'ontologie de W3C[31] basé sur les logiques de description, les axiomes et les constructeurs d'OWL sont restreints pour que le raisonnement soit décidable.
La Sémantique Web s'organise dans une architecture en couche, XML produit la couche de transport syntaxique, RDF(S) produit un langage relationnel basique et des primitives ontologiques simples et bien sûr OWL produit un langage d'ontologie puissant mais restant décidable.

4.3.8 Complexité des logiques de description [46]

Ils existent de nombreuses logiques de descriptions qui se distinguent par les constructeurs qu'elles proposent. Plus une logique de description est expressive, plus sa complexité est élevée. Par contre, une LD trop peu expressive ne permet pas de représenter des connaissances de domaines complexes. Par exemple la logique $\mathcal{AL}$ est la moins expressive, parce que qu'elle propose le moins de nombre de constructeurs.
Les logiques que nous avons vu jusqu'à maintenant sont toutes décidables, mais de complexité différente. L'intérêt de raisonner avec le minimum d'extensions nécessaire est de limiter la complexité, ce qui a son importance lorsqu'on raisonne sur de gros volumes d'informations.
La complexité est calculée en fonction des extensions choisies ; le tableau 3 présente la complexité des relations de non-satisfiabilité et de subsomption dans les logiques de descriptions de la famille $\mathcal{AL}$. Pour fixer les idées, cette complexité est polynomiale pour $\mathcal{AL}$ et $\mathcal{ALN}$, mais devient NP (**N**on-déterministe **P**olynomiale) ou co-NP lors de l'ajout de certaines extensions, et devient rapidement **PSPACE**, c'est-à-dire décidable par un algorithme déterministe en espace polynomial par rapport à la taille de son instance, lorsque plusieurs extensions sont ajoutées [50].

La notion d'expressivité définie dans cette section se réfère à la facilité qu'a un système à représenter, d'une part, des données et, d'autre part, des requêtes. Cependant, un compromis doit généralement être fait entre expressivité et efficacité, puisque l'une est souvent incompatible avec l'autre. Les applications n'étant pas a priori figées, une certaine

[31] Le **World Wide Web Consortium**, abrégé par le sigle **W3C**, est un organisme de standardisation à but non lucratif, fondé en octobre 1994 comme un consortium chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML, XHTML, XML, RDF, CSS, PNG, SVG et SOAP. Le W3C n'émet pas des normes au sens européen, mais des recommandations à valeur de standards industriels.

flexibilité est souhaitée afin de pouvoir satisfaire l'une ou l'autre de ces contraintes selon les besoins. La structure des documents du Web sémantique est basée sur XML (Voir paragraphe XML), l'une de ses limites en terme d'expressivité réside dans le fait que, par sa structure même, il impose la notion d'arbre [50]

Complexité	Logique de description
P	$\mathcal{AL}$, $\mathcal{ALN}$
NP	$\mathcal{ALE}$
Pspace	$\mathcal{ALL}$, $\mathcal{ALEN}$
ExpTime	$\mathcal{SH}$ $\mathcal{I}$ $\mathcal{Q}$,$\mathcal{SHOQ}$
NExpTime	…

Tableau 4.10 : Complexité de la vérification de la subsomption et de la satisfiabilité en fonction de l'expressivité des LDs ($P \subseteq NP \subseteq PSpace \subseteq ExpTime \subseteq NExpTime$)

Les classes de complexité énumérées dans le tableau 3 proviennent de la théorie de la complexité informatique, voici une définition de ces classes [46]:

- **P**: La classe des problèmes de décision (un problème de décision prend en entrée un énoncé de problème et produit en sortie une réponse positive ou négative: oui ou non). qui requièrent un temps polynomial par rapport à la taille du problème pour obtenir une solution avec une machine de Turing[32] déterministe.
- **NP**: La classe des problèmes qui nécessitent un temps polynomial pour trouver une solution avec une machine de Turing non déterministe.
- **PSpace** : La classe des problèmes de décision qui requièrent une quantité de mémoire polynomiale pour résoudre un problème avec une machine de Turing déterministe ou non déterministe.
- **ExpTime** : La classe des problèmes de décision solvables par une machine de Turing déterministe en un temps $\Phi(2^{P(n)})$ où $P(n)$ est une fonction polynomiale de n, la taille du problème.
- **NExpTime** : La classe des problèmes de décision solvables par une machine de Turing non-déterministe en un temps $\Phi(2^{P(n)})$ où $P(n)$ est une fonction polynomiale de n, la taille du problème.

4.3.9 Relation avec d'autres formalismes

Les logiques de description ont été conçues à partir des réseaux sémantiques de Quillian [51] qui sont des graphes orientés étiquetés auxquels on associe des concepts aux nœuds et des relations aux arcs, et de la sémantique des Frames de Minsky [52] où l'on a

[32] Une **machine de Turing** est un modèle abstrait du fonctionnement des appareils mécaniques de calcul, tel un ordinateur et sa mémoire, créé par Alan Turing en vue de donner une définition précise au concept d'algorithme ou « procédure mécanique ». Ce modèle est toujours largement utilisé en informatique théorique, en particulier pour résoudre les problèmes de complexité algorithmique et de calculabilité [web 14]

des concepts représentés par des Frames (cadres) qui sont caractérisés par un certain nombre d'attributs (appelés aussi slots) qui contiennent de l'information sur leur contenu. Quant aux graphes conceptuels, ils se caractérisent par leur interprétation en terme de fragment de la logique du premier ordre et l'équivalence entre spécialisation de graphes et implication de formules. Cependant l'expérience montre qu'il vaut mieux travailler directement avec des logiques non classiques (telles que les logiques modales) plutôt que de les retranscrire en termes de logique de premier ordre. C'est dans cette démarche que se situe la famille des logiques de description [53].

L'avantage ou l'intérêt des approches basées sur la logique telles que les LDs, c'est bien qu'elles adoptes une approche généraliste ; généralement une LD est une variante de la logique du premier ordre, le raisonnement aussi revient à tester ou à calculer des conséquences logiques. Leurs inconvénients, c'est l'éloignement des domaines d'application et la difficulté d'adaptation pour modéliser les connaissances d'un point de vue Humain. Par contre, les approches non-basées sur la logique telles que celles basées sur des structures de réseaux (les réseaux sémantiques de Quillian et les Frames de Minsky), ont eu plus de succès à cause de nombreux langages qui ont été proposés (avec des formalismes pensés pour des domaines d'application spécifiques) avec des procédures ad hoc pour effectuer le raisonnement.

Les limites des approches basées sur des structures réseaux, c'est bien le manque de caractérisation sémantique précise et le comportement parfois différent pour le raisonnement. Les premiers travaux qui ont donné une sémantique formelle pour les principales caractéristiques ont montré qu'elles sont basées sur un fragment de la logique du premier ordre (ajout de caractéristiques spécifiques $\Rightarrow$ différents fragments $\Rightarrow$ différentes classes de problèmes et différentes classes de complexité pour les algorithmes de raisonnement).

Les logiques de Description, sont considérées comme le fruit de l'évolution des travaux pour donner une sémantique formelle et mieux classifier les précédentes approches [54].

4.3.10 Les moteurs d'inférences [17]

L'inférence est l'opération qui consiste à admettre une proposition en raison de son lien avec une proposition préalable tenue pour vraie. C'est un terme général dont les mots raisonnement, déduction, induction, etc., sont des cas spéciaux. Le moteur d'inférence est le programme qui réalise les déductions logiques d'un système expert à partir d'une base de connaissances (faits) et d'une base de règles [55].

Une fois modélisée et chargée par un moteur d'inférence, une ontologie peut être interrogée par des langes de requêtes. Suivant les moteurs d'inférence employés le choix du langage de

requêtes est restreint. En effet ces langages interrogent l'ontologie en passant par le moteur d'inférence qui se charge pour d'effectuer les calculs d'inférences sur la TBox et la ABox. La plupart de ces moteurs (tableau 4.11) sont conçus pour raisonner sur les logiques de descriptions.

Actuellement, plusieurs moteurs d'inférences gratuits ou commerciaux tels que Racer [56] [web 15], Pellet [web 16], Fact [web 17], Fact++[33], Surnia, F−OWL et Howlet sont disponibles. La plupart de ces moteurs sont conçus pour raisonner sur les logiques de description, mais acceptent en entrée des fichiers OWL. Certains moteurs d'inférence ne peuvent raisonner qu'au niveau terminologique (c'est−à−dire au niveau des concepts et des propriétés) alors que des moteurs comme Pellet et Racer permettent de raisonner aussi sur les instances de concepts.

Dans cette partie nous présenterons en détail, seulement les deux moteurs d'inférence **Pellet** et **Racer** qui sont les plus permettant le raisonnement sur la ABox et la TBox et exploitent des ontologies possédant un niveau d'expressivité en logique de description et en OWL satisfaisant.

Moteur	Racer	Pellet	FaCT	FaCT++	Surina	Hollet	F-OWL
DL **Implantation** **Inférence** **API java** **OWL** **Décidébalité**	$\mathcal{SH}\,\mathcal{I}\,\mathcal{Q}$ c++ TBox/Abox Oui OWL DL Oui(<owl-Lite)	$\mathcal{SH}\,\mathcal{IN},\mathcal{SHON}$ java TBox/Abox natif OWL DL Oui(OWL- Lite)	$\mathcal{SHIO},\mathcal{SHF}$ Python Common lisp oui OWL DL oui	Logiue,Prédicat C++ TBox Oui OWL- Lite Oui	Logiue,Prédicat Python TBox/Abox Non OWL Full non	Logique Prédicat TBox/Abox Oui OWL DL non	$\mathcal{SHIQ}$ java TBox/Abox Oui OWL Full non

Tableau 4.11 : Tableau comparatif (d'après [57]) des principaux moteurs d'inférence pour les DL expressives.

4.3.10.1 Racer

Le système Racer (Renamed ABox and Concept Expression Reasoner ou Raisonneur d'expression de concepts et de ABox renommées) est le moteur d'inférence sans doute le plus connu et l'un des plus utilisés dans le domaine pour ces performances et sa stabilité. Il est commercialisé par Racer Systems GmbH & Co. KG. Racer se présente sous la forme d'un serveur qui peut être accédé par le protocole TCP ou http. Considéré aujourd'hui comme le plus intéressant, c'est un système de représentation de connaissances, qui implémente un calcul de tableaux hautement optimisé pour une logique de description très expressive. Racer travaille sur les ontologies modélisées

[33] Fast Classification of Terminologies, implémenté en C++

par son langage, mais il peut interpréter des documents OWL et offre des services de raisonnement aussi bien pour le niveau terminologique la TBox de l'ontologie, que pour le niveau assertionnel la ABox. Il permet aussi de vérifier la consistance et subsomption des concepts et d'autre tests plus élaborés sur les instances, les rôles et les restrictions. Racer possède également son propre langage de requête nRQL[34] pour interroger les ontologies Il est très simple à utiliser et possède une interface DIG[35] [web 18] ce qui permet de l'interfacer très facilement. Il est gratuit pour la recherche mais n'est malheureusement pas en open source.

4.3.10.2 Pellet

Le moteur Pellet est beaucoup plus récent, il a été en 2003 le premier raisonneur OWL-DL sûr et complet. C'est l'un des projets du MINDSWAP Group, un groupe de recherche sur le Web sémantique de l'université du Maryland. Il est disponible en OpenSource et offre des évolutions fréquentes. Pellet travaille sur des ontologies décrites en RDF ou OWL et permet les requêtes avec RDQL[36] et SPARQL[37] sur la ABox et la TBox [55].

4.3.11 Langages d'interrogation d'ontologies [17]

Dans cette partie, nous présentons les trois langages RDQL, SPARQL et nRQL. Ces trois langages sont des langages d'interrogation basés principalement sur la reconnaissance de graphe RDF. Le principe est de décrire un graphe RDF à l'aide de variables. Les résultats d'une requête étant les valeurs des variables pour lesquelles il existe un graphe RDF dans l'ontologie correspondant au graphe défini par la requête. RDQL et SPARQL sont les langages d'interrogation utilisables sur Pellet et nRQL est le langage d'interrogation de Racer.

4.3.11.1 RDQL est un langage d'interrogation de données définies en RDF. Ce langage n'est pas standardisé, il existe de nombreuses implémentations bien que la soumission W3C définit une base commune. Sa syntaxe est très proche de SQL :

```
SELECT variable [, variable]*
FROM documents rdf [, documents rdf]*
WHERE modele de triplets
AND restrictions booléennes
USING définition des raccourcis
```

[34] new Racerpro query Language
[35] est une spécification d'interface communément utilisée par les raisonneurs comme Racer.
[36] RDF Data Query Language
[37] Stands for Protocol And RDF Query Language

- La clause **SELECT** définit la liste des variables que l'on désire obtenir. Une variable est composée de caractères alphanumériques avec le ' ' et commence par un ' ?'. Les variables d'une requête sont séparées par un espace (et/ou) une virgule. La valeur * signifie que l'on désire obtenir l'ensemble des variables utilisées dans la requête.
- La clause **FROM** définit l'emplacement des documents RDF utilisés pour la requête.
- La clause **WHERE** définit les triplet RDF (sujet - prédicat - objet), les éléments de ce triplet sont décrits soit par les valeurs de l'ontologie interrogée soit par des variables. Plusieurs triplets peuvent être définis dans une même clause WHERE, cette succession de triplets indiquant la conjonction de chacun des termes.
- La clause **AND** définit les restrictions booléennes de la requête. Une restriction booléenne est constituée de valeurs ou variables composé à l'aide d'opérateurs :
    - opérations numériques : +,-, ff, /
    - conjonction, disjonction : kk,&&
    - comparateurs sur les numériques : <,>,<=,>=,==
    - comparateurs sur les chaˆýnes de caractères : EQ pour l'égalité, NE pour la différence.
- La clause **USING** permet l'utilisation de simplification de nommage à d'alias WHERE (?email, <http ://www.w3.org/2001/vcard-rdf/3.0/EMAIL>, "Dujardit@lifl.fr")
WHERE (?email, vcard :EMAIL, "Dujardit@lifl.fr")
USING vcard FOR <http ://www.w3.org/2001/vcard-rdf/3.0/>

Voici un exemple de requête RDQL, qui sélectionne l'âge d'une personne dans le document annuaire.rdf sachant que la personne possède un âge inférieur ou égal à 50 ans, que son nom est Dujardin et son adresse email est Dujardit@lifl.fr.

```
SELECT ?name ?email, ?age
FROM <http ://www.lifl.fr/ANNUAIRE/annuaire.rdf>
WHERE ( ?email, vcard :EMAIL, ''Dujardit@lifl.fr'')
AND ( (10 + ?age)*2 <= 60/2) && ( ?name EQ ''Dujardin'') )
WHERE ( ?email, vcard :EMAIL, ''Dujardit@lifl.fr'')
USING vcard FOR <http ://www.w3.org/2001/vcard-rdf/3.0/>
```

4.3.11.2 SPARQL est une amélioration de RDQL, il est plus spécifiquement défini par le groupe de travail RDF Data Access, est à la fois un langage et un protocole. Le protocole va notamment permettre à un client Web de consulter au travers d'une requête SPARQL. Le langage permet tout particulièrement, l'interrogation de descriptions RDF. A l'image de RDQL (dont il est une extension mais y ajoute notamment les opérateurs UNION et OPTIONAL dans la clause WHERE), SPARQL exploite, en premier lieu, RDF au travers de la notion de triplets ou d'ensembles de triplets. La réponse à la requête posée va ainsi correspondre à la restitution du sous-graphe RDF satisfaisant le filtre exprimé au travers d'opérations de mise en

correspondance de patterns de graphe (possiblement optionnels) et d'opérations basées sur les connecteurs logiques (conjonction, disjonction).
Forme des requêtes SPARQL :

PREFIX indique l'adresse (espace de noms) d'un schéma pouvant être exploité ensuite dans la construction de la requête .
SELECT…[FROM]…WHERE retourne les ressources qui sont associées aux variables liées dans la clause WHERE

CONSTRUCT engendre un nouveau graphe qui complète le graphe interrogé (ancêtre, tante, oncle etc dans un graphe contenant des informations généalogiques). CONSTRUCT va donc permettre d'exploiter des requêtes dites constructives (SELECT exploite, à l'inverse, des requêtes dites interrogatives)
UNION graphes alternatifs (correspond à au au moins un des graphes précisés)
FILTER rajouter des conditions devant être satisfaites
DESCRIBE retourne une description des ressources satisfaisant la requête
OPTIONAL pattern(s) de graphe optionnel(s)
ASK évalue si la requête va retourner un ensemble de ressources ou bien l'ensemble vide

4.3.11.3 nRQL est le langage d'interrogation de Racer. Comme RDQL et SPARQL, nRQL est basé sur la recherche de graphes RDF. Sa syntaxe est proche des deux autres, sauf pour sa notation préfixée des opérateurs. Ci-dessous, nous présentons deux requêtes cherchant tous les oncles (variable z écrite ?z ou $?z) dans l'ontologie myOntology. La première requête est en RDQL (la version SPARQL est identique) et la deuxième en nRQL.

```
SELECT ?z from < myOntology >
WHERE ( ?x < aEnfant >?y),( ?z < estFrereDe >?x)
RETRIEVE ($ ?z)
(AND ( $ ?x $ ?y |myOntology#aEnfant|)
($ ?z $ ?x |myOntology#estFrereDe|))
```

4.3.12 Outils disponibles pour les ontologies [42]

4.3.12.1 Editeur d'ontologies Protégé [29]

- **Oiled :** (Oil Editor) est un éditeur d'ontologies utilisant le formalisme DAML+OIL et les Logiques de Description. Il est essentiellement dédié à la construction d'ontologies dont on peut ensuite tester la cohérence à l'aide de

FACT, un moteur d'inférences bâti sur OIL. Il permet l'export d'ontologies sous les formats RDF, DAML+OIL, OWL et d'autres langages moins consensuels comme SHIQ.

- **OntoEdit :** (Ontology Editor) est un environnement de construction d'ontologies. Il a été développé par la compagnie Ontoprise. Il permet l'édition des hiérarchies de concepts et de relations dans le cadre de la logique des frames, ainsi que l'expression d'axiomes algébriques. Des outils graphiques dédiés à la visualisation d'ontologies sont inclus dans l'environnement. OntoEdit intègre, dans sa version commerciale, un serveur destiné à l'édition d'une ontologie par plusieurs utilisateurs ainsi qu'un plug-in permettant le test de la cohérence d'une ontologie. OntoEdit gère de nombreux formats de représentation de connaissances dont OWL, RDFS et FLogic.
- **Ontolingua :** développé au Knowledge Systems Laboratory de l'Université de Stanford, est un serveur d'édition d'ontologies permettant la construction collaborative d'ontologies. Une ontologie y est directement exprimée dans un formalisme également nommé ONTOLINGUA.
- **WebOnto**, développé au Knowledge Media Institute de l'Open University, offre une interface graphique d'édition collaborative, de tests et de parcours d'ontologies sur le Web. Le modèle de connaissance utilisé est celui du langage OCML (Operational Conceptual Modeling Language), un langage à base de Frames.
- **Protégé2000** : est une interface modulaire, développée au Stanford Medical Informatics de l'Université de Stanford, permettant l'édition, la visualisation, le contrôle (vérification des contraintes) d'ontologies. Le modèle de connaissances de Protégé2000 est issu du modèle des frames et contient des classes (concepts), des slots (propriétés) et des facettes (valeurs des propriétés et contraintes), ainsi que des instances des classes et des propriétés. Protégé2000 autorise la définition de métaclasses, dont les instances sont des classes, ce qui permet de créer son propre modèle de connaissances avant de bâtir une ontologie. De nombreux plug-in sont disponibles ou peuvent être créés par l'utilisateur. Parmi ceux-ci, citons le plug-in permettant d'utiliser le langage OWL et les plug-ins de visualisation tels que Graphviz et Ontoviz.

4.3.12.2 Framework Jena

Jena est un cadre de travail java open source permettant de construire des applications de Web sémantique. Il fournit un environnement de programmation pour RDF, RDFS et OWL, ainsi qu'un moteur d'inférence basé sur les règles. Le cadre de travail inclut :

- API RDF

- API OWL
- Lecture et écriture RDF en RDF/XML et N-Triples
- Stockage en mémoire
- RDQL un langage d'interrogation du RDF

Le sous-système d'inférence de Jena est conçu pour permettre à certains moteurs d'inférence ou raisonneurs d'être connectés à Jena. Le terme inférence est utilisé pour se référer au processus abstrait de dérivation d'informations supplémentaires, et le terme raisonneur pour faire référence à un code objet spécifique qui effectue cette tache.

4.4 Conclusion

Dans ce chapitre nous avons présenté les langages de spécification d'ontologies qui sont destinés beaucoup plus au domaine du web sémantique, nous avons vu que XML est la base syntaxique de tous ces langages et RDF s'intéresse à des assertions sur les relations entre objets, tandis que OWL s'intéresse à décrire les classes de ces objets. Il s'agit d'un découpage assez naturel, entre connaissances factuelles et les connaissances ontologiques. Ensuite nous avons vu qu'il y a des outils permettant de construire, d'inférer et d'interroger une ontologie et nous avons présenté les deux moteurs d'inférence Racer et Pellet qui sont les plus connus et les plus utilisés pour inférer une ontologie.

Le développement des applications basées sur les ontologies est une tâche très lourde et qui nécessite préalablement des infrastructures ou bien des plateformes déjà construites et qui sont prêtes à l'emploi, pour cela nous avons présenté une panoplie d'éditeurs pour exploiter des ontologies (création, vérification, interrogation et visualisation) ainsi le **framework** Jena qui est un cadre de travail java open source permettant de construire des applications de Web sémantique basées sur des ontologies.

Chapitre 5 :

Les Systèmes d'aide à la décision

« Quelle est la voie ? demanda le disciple.
La perception aiguë de l'évidence des choses », dit le maître zen.
Brunel H, *Contes zen*, 2000

5.1 Introduction

Généralement, dès que l'on se trouve placé dans une situation où diverses actions sont envisageables, il convient de décider de celle qu'il est nécessaire de choisir. C'est en ce sens que le terme de « décision » est utilisé. Dans la vie quotidienne, nos décisions sont souvent prises sur la base d'intuitions ou d'expériences passées. Ce type de technique ne peut s'appliquer qu'à des problèmes habitués ou familiers. Lorsque nous sommes confrontés à des nouvelles situations, la tâche de prise décision devient beaucoup plus difficile, donc il devient indispensable d'utiliser des systèmes d'aide à la décision [58].

Un système d'aide à la décision (SAD) est avant tout un système informatique incluant une base de données (base de connaissances, ontologies, etc.), un ensemble de procédures et de modèles, permettant l'exploitation des données et apportant les éléments de réponse aux questions posées par l'utilisateur au travers d'une interface avec le système [59].

Dans ce chapitre nous introduisons quelque notions de base de la théorie de la décision, définitions, classification et processus de décision ; puis nous entamons l'aspect système de prise de décision, ses types, composants et architectures ainsi nous abordons l'exemple des SAD intelligents en montrant l'apport des ces derniers par rapport ceux qui sont classiques.

Pas mal de SAD sont organisés autour d'un système d'information géographique (SIG) permettant la gestion et l'analyse d'informations géographiques (figure 5.1). Dans ce chapitre nous présentons une démonstration d'une étude de cas d'un SAD pour la gestion des panneaux publicitaires basé sur une ontologie urbaine.

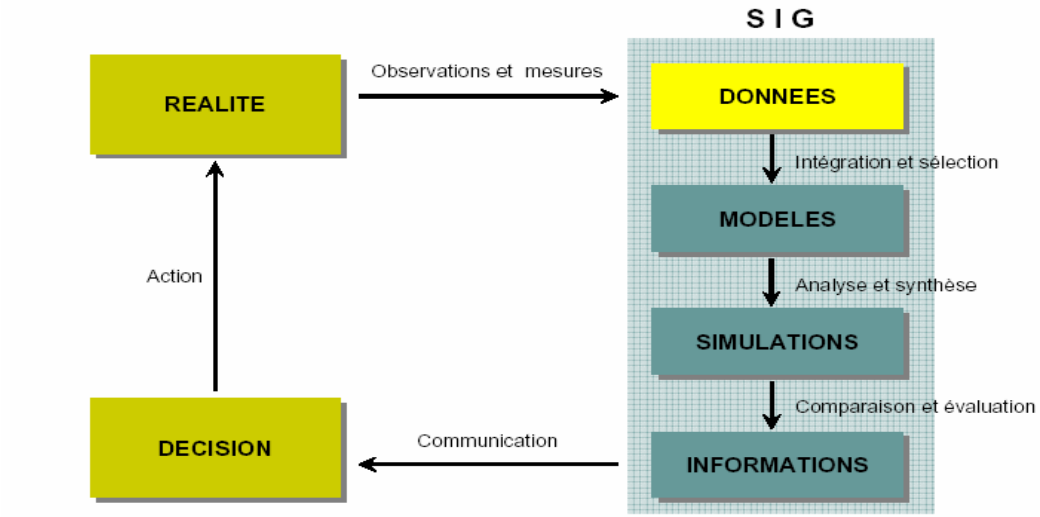


Figure 5.1 : SIG et SAD

5.2. Théorie de la décision

La théorie de la décision modélise le comportement d'un agent face à des situations de choix. Cette discipline vise non seulement les individus, mais également les organisations et les systèmes dans leur diversité.

5.2.1 Définitions d'une décision

Le terme de décision a plusieurs définitions .Il est assimilé à un acte, une action ou à un processus de résolution de problème. Une décision est toujours prise pour faire face à une difficulté ou répondre à une modification de l'environnement, c'est-à-dire pour résoudre un problème qui se pose à l'organisation ou à l'individu. Le mot **décision** désigne à la fois le produit de la réflexion et le processus de réflexion ou de calcul lui même. Le concept de décision ne peut être complètement séparé de celui de processus de décision. C'est l'ensemble des temps forts du déroulement d'un processus de décision (son aboutissement en est un) qui détermine la décision finale [web 19].

En algorithmique, un **problème de décision** est une question mathématiquement définie portant sur des paramètres donnés sous forme manipulable informatiquement, et demandant une réponse par oui ou non. Un problème de décision peut être indécidable s'il n'existe aucun programme informatique qui permette de le résoudre (sans restriction de mémoire ou temps), ce que l'on formalise par l'impossibilité de répondre au problème à l'aide d'une machine de Turing [web 20].

En programmation, on peut se retrouver dans des situations critiques comme le déclenchement d'une exception[38], qui nécessite une prise de décision sur quelle méthode ou sous programme à exécuter pour que le programme ne s'interrompe pas.

5.2.2 Prise de décision [web 20]

La prise de décision est un processus cognitif complexe visant à la sélection d'un type d'action parmi différentes alternatives. La théorie étudiant la prise de décision est la théorie de la décision. La prise de décision *concerne* tout organisme vivant. Elle intéresse chaque individu et chaque groupe. Il s'agit d'une méthode de raisonnement pouvant s'appuyer sur des arguments rationnels et/ou irrationnels. On peut distinguer différents modes de décision : **Autoritaire :** un seul membre prend la décision, **Majoritaire :** l'ensemble des membres est d'accord, **Minoritaire :** un sous groupe prend la décision pour tous et **Unanimité :** un vote à l'ensemble du groupe [web 20].

[38] Mécanisme d'exception : voir langages ADA, JAVA, etc.

5.2.3 Classification des décisions [web 21]

H.A. Simon[39] après avoir proposé le modèle descriptif du processus de prise de décision, s'est attaché à préciser le rôle des déterminants psychologiques et comportement dans la prise de décision. D'où la classification des décisions illustrée par le schéma de la figure 5.1.

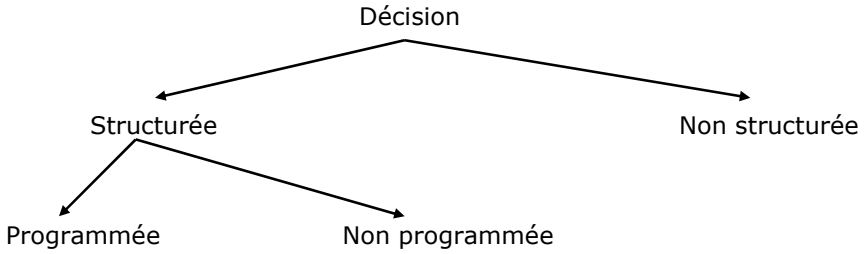


Figure 5.2 : Classification de décisions

- **Les décisions programmées**

Les décisions sont programmées lorsqu'elles sont répétitives et routinières, et que l'on a établi une procédure déterminée pour les prendre, de façon à ne pas avoir à les reconsidérer chaque fois qu'elles se présentent.
Ces décisions relèvent d'un comportement stimuli-réponse. Leur intégration se fait par l'apprentissage, l'habitude ou la formalisation du comportement. Il en découle de modèles très formalisés de prise de décision; du type:

Information ⟶ modèle ⟶ décision

- **Les décisions non-programmées**

si la décision à prendre est non programmée, elle répond donc à un événement nouveau. Les acteurs de la conduite doivent alors se rabattre sur toute la capacité générale dont ils disposent pour mener des actions intelligentes, adaptables et orientées vers la résolution de problème. Il est évident que ce genre de décision est le plus coûteux en temps et en financement [web 19]

- **Les décisions structurées**

Les décisions programmées sont des décisions structurées. Mais toutes les décisions structurées ne peuvent être programmées. Nous nous intéressons ici plus particulièrement aux décisions structurées non programmées. Pour ce type de

[39] Herbert A Simon a reçu le prix Nobel d'économie en 1978. Il fait partie du Département de psychologie de l'Université de Carnégie-Mellon, où il poursuit une série d'études sur le processus de prise de décisions à travers des recherches sur la cognition chez l'être humain.

décisions nous ne disposons pas de règle formelle précisant quelle est la décision à prendre face à un problème qui a été identifié. Mais dans ce cas le décideur peut faire appel à des techniques d'aide à décision et à des procédures lui précisant quelles informations il doit rechercher, quels outils il employer, quels interlocuteurs il peut contacter pour trouver une solution au problème.

- **Les décisions non structurées**

Les décisions non structurées se rencontrent lorsque le décideur se trouve dans des situations ou il n'existe pas de plan d'action préalablement défini lui permettant de prendre une décision. Dans ce cas, le plus souvent il doit prendre en compte de multiples critères, dont il ne dispose pas à priori, et qui ne sont pas clairement définis. Ces critères sont par ailleurs souvent implicites, et qualitatifs. Face à de telles situations le décideur ne peut faire appel qu'à son imagination, son intuition et son expérience. Les raisonnements s'appuient sur l'analogie des situations, on raisonne par l'absurde, on approche de la solution par approximations successives.

5.2.4 Les différents types (niveaux) de décision

On distingue traditionnellement trois grands types de décisions qui doivent être prises dans une entreprise (figure 5.3) : [web 22]

Figure 5.3 : Pyramide de types de décision

- **Les décisions stratégiques**

Le niveau Stratégique concerne les décisions prises dans un horizon à long terme. Elles sont orientées vers la réalisation d'un **but général** souvent à **long terme** dont certains aspects sont souvent **malaisés à définir** et surtout à quantifier. Ces décisions stratégiques engagent l'entreprise sur une longue période puisqu'elles conditionnent la manière dont l'entreprise va se positionner sur un marché de manière à retirer le maximum de profit des ressources qu'elle mobilise. On cherche alors à répondre à la question essentielle de l'entreprise qui est « quoi produire ?» et son corollaire qui est « quels moyens mettre en œuvre de manière efficace pour produire ? ». En définitive, il s'agit de définir la manière

dont l'entreprise va s'insérer dans son environnement. Les modèles analytiques sont souvent utilisés dans ce niveau de décision.

- **Les décisions administratives ou tactiques**

Le niveau Tactique concerne les décisions prises dans un horizon à moyen terme. Il décrit le plan de production global et permet de définir comment les ressources de l'entreprise doivent être utilisées pour parvenir à réaliser les objectifs définis dans le cadre des décisions stratégiques. La simulation est très utilisée à ce niveau du fait de la complexité des systèmes.

- **Les décisions opérationnelles**

Le niveau Opérationnel concerne les décisions prises dans un horizon à court terme. Il s'agit de problèmes bien définis, assez aisément quantifiables, souvent de nature technique. Ces décisions s'appliquent dans le cadre de la gestion courante de l'entreprise et concerne l'utilisation optimale des ressources allouée dans le cadre du processus productif de l'entreprise (gestion de stocks, planification et optimisation hebdomadaire de la production, séquencement de tâches, etc.).

5.2.5 Processus de décision [62] [63]

Herbert Simon a proposé vers la fin des années cinquante, notamment en 1957, puis dans plusieurs articles et ouvrages des années soixante et soixante-dix [64], un schéma de la prise de décision suffisamment général pour pouvoir être reconnu comme un véritable modèle canonique de la décision. Ce modèle distingue quatre phases dans le processus de décision (figure 5.4):

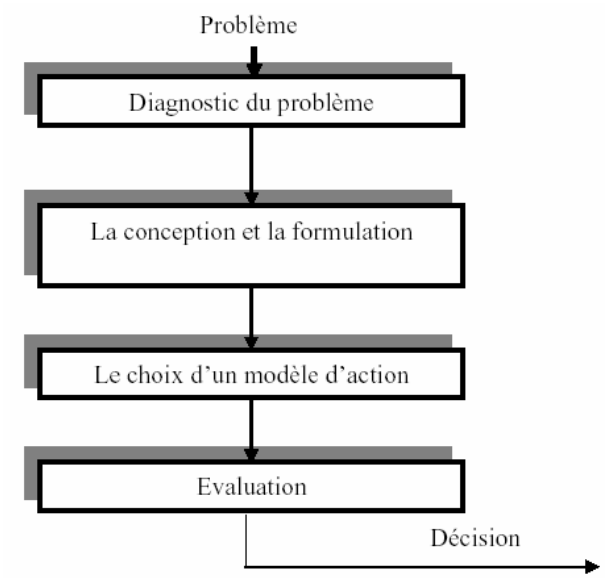


Figure 5.4 : Les étapes de la décision

- **Diagnostic** d'un problème et d'exploration / reconnaissance des conditions dans lesquelles il se pose (phase d'identification du problème pour modélisation /Analyse)

- **Conception et formulation** (phase de modélisation proprement dite) Cette étape du processus de décision conduit le décideur à recenser toutes les solutions envisageables pour résoudre le problème.
- **Choix d'un mode d'action particulier parmi les actions possibles :** c'est la phase de sélection (phase d'élaboration de la solution : gestion des scénarios) dans la quelle, le décideur choisit entre les différentes actions qu'il a conçu et d'identifier pendant la phase de conception.
- **Evaluation,** cette étape vise à confirmer le choix effectué ou à remettre en question le processus de décision en réactivant l'une des trois phases précédentes. Après la phase d'évaluation, la décision retenue est concrétisée sous la forme de programmes d'actions, diffusés auprès des personnes et services concernés. L'application et les effets de la décision pourront être contrôlés. Ce contrôle confirmera ou infirmera le bien fondé de la décision. [63]

5.2.6 L'aide à la décision [63]

L'aide à la décision est définie comme étant l'activité de celui (homme d'étude) qui, prenant appui sur des modèles clairement explicités et plus ou moins complètement formalisés, cherche à aboutir des éléments de réponse aux questions que se pose un intervenant (décideur) [62]. On peut classer les différentes situations en quatre catégories selon un degré d'incertitude croissant.

- **L'aide à la décision en univers certain**

En univers certain, le décideur a une connaissance parfaite des différents paramètres de la décision. Il peut ainsi prévoir les conséquences de ses choix. Certaines techniques d'aide à la décision pourront néanmoins être utilisées pour évaluer les conséquences des différents choix.

- **L'aide à la décision dans un univers aléatoire**

En univers aléatoire, le décideur peut associer une probabilité à chaque éventualité de la décision. **Le calcul des probabilités** (espérance mathématique), **des statistiques** (variance, écart type pour apprécier les risques), et **la technique des arbres de décisions** (intéressante lorsque l'on veut étudier les conséquences d'une série de décisions successives) pourront l'assister dans le processus conduisant au choix final.

- **L'aide à la décision en univers incertain**

En univers incertain, le décideur n'a pas suffisamment d'informations pour connaître ou prévoir les différents événements liés à la décision. Dans de telles situations, il peut faire appel à certains critères de **la théorie des jeux**. C'est un instrument de recherche qui permet l'analyse des décisions des agents économiques. Les critères du minimax et du maximax sont généralement retenus.

- Si le décideur est optimiste, il privilégiera le choix pour lequel le maximum espéré est le plus élevé (**maximax**)
- Si le décideur est pessimiste, il privilégiera la solution pour laquelle le gain minimum espéré est le plus élevé (**minimax**)

Exemple : le décideur peut choisir entre deux solutions (1) et (2)

	Gain minimum espéré	**Gain maximum espéré**
(1)	70	300
(2)	150	250

Il retiendra la solution (1) d'après le critère du maximax, et la solution (2) d'après le critère du minimax.

- L'aide à la décision en univers conflictuel

En univers conflictuel, tous les événements dépendent d'intervenants par nature hostiles. Les décisions peuvent en effet concernaient plusieurs agents. La théorie des jeux peut une nouvelle fois permettre au décideur d'analyser une décision dans une situation où plusieurs agents économiques interagissent. Chacun devra tenir compte des actions des autres joueurs pour prendre une décision.

5.3 Systèmes d'aide à la décision

5.3.1 Définition

Un Système Aide à la Décision (SAD) est un système **Homme-Machine** qui, à travers un dialogue permet à un décideur d'amplifier son raisonnement dans l'identification et la résolution de problèmes mal structurés. Un SAD, est composé de base(s) de données, de modèles et d'outils spécialisés permettant de gérer, d'analyser et de comparer des données. Il s'appuie sur un ensemble d'outils : d'intelligence artificielle pour la manipulation des données, statistiques pour l'exploration, le filtrage et l'analyse, et graphiques pour la représentation des informations.

5.3.2 Système d'information ou interactif d'aide à la décision

Le mot ou les initiales S.I.A.D peuvent désigner à la fois un **S**ystème d'**I**nformation d'**A**ide à la **D**écision ou un **S**ystème **I**nteractif d'**A**ide à la **D**écision, mais les deux se complètent pour accomplir la tâche de la prise de décision dans un système. Dans ce paragraphe nous définissons un **SIAD** qui est système d'information conçu spécialement pour la prise de décision, et qui est destiné plus particulièrement aux dirigeants (décideurs) d'entreprise. Le système d'aide à la décision est un des éléments du système d'information de

gestion. Il se distingue du système d'information pour dirigeants, dans la mesure où sa fonction première est de fournir non seulement l'information, mais les outils d'analyse nécessaires à la prise de décision. Ainsi, il est habituellement constitué de programmes, d'une ou de plusieurs bases de données, internes ou externes, et d'une base de connaissances. Il fonctionne avec un langage et un programme de modélisation qui permettent aux dirigeants d'étudier différentes hypothèses en matière de planification et d'en évaluer les conséquences [web 23].

Un **SIAD** est aussi un système interactif, flexible, adaptable et spécifiquement développé pour aider la résolution d'un problème de décision en améliorant le processus de la prise de décision. Il est composé de programmes interactifs, c'est-à-dire qu'à tout moment, l'utilisateur peut interroger son système qui va lui proposer des solutions. Un SIAD est généralement caractérisé par quatre éléments [63]: sa technologie (monoposte, réseaux, outils TIC, etc.), le type de décisions à traiter (simples ou complexes), le langage ((le langage de communication du SIAD est celui dit de quatrième génération de type non procédural, c'est à dire que l'utilisateur exprime à l'aide de mots clés ce qu'il souhaite obtenir), et la convivialité (celle-ci est déterminée par l'interactivité, la puissance, l'accessibilité et la flexibilité du SIAD).
Par convention la définition d'un SIAD peut être précisée de la manière suivante : un SIAD est "un système d'information interactif, flexible et adaptable, développé spécialement pour aider à la solution de problèmes de management peu ou non structurés. Il utilise des données, fournit une interface simple et autorise le manipulateur à exprimer ses propres opinions" [65].

5.3.3 Composants d'un SIAD

Un SIAD est généralement composé de trois types de modules de base, étant : La base de données, La base de modèles et Une interface de dialogue :

- **Interface de dialogue**

  Le module du dialogue manipule des représentations, suivant les types de ces représentations, l'utilisateur est invité à réagir sur des graphiques, du texte, etc. Le module de dialogue étant le lien entre l'interactivité et le contrôle, c'est un des facteurs les plus importants pour l'évaluation d'un SAD.

- **Base de données**

  Dans un SIAD, le module de base de données ne se contente pas de stocker passivement les informations, mais grâce au SGBD[40], les données participent activement au fonctionnement d'un SIAD.

[40] SGBD : Système de gestion de base de données

Le rôle de la base de données étant important dans un SIAD, la question de sa réalisation est donc un point fondamental dans le processus de sa conception, et l'accès à la BD doit être assuré par des interfaces graphiques (masques de saisie et états de sortie).

- **Base de modèles**

Les modèles utilisés dans un SIAD ont des tâches spécifiques d'aide à la décision. Ce sera suivant le cas, un tri, un choix, un classement, un calcul, etc. donc toute opération qui tend à organiser de l'information, conduire à des recommandations et/ou à des choix d'action.
Plusieurs modèles peuvent cohabiter dans un seul SIAD. Leur stockage s'effectue dans une base de modèles. On peut alors introduire la notion de Système de Gestion de Bases de Modèles (SGBM).

5.3.4 Architectures des SIAD

Le rôle de l'architecture d'un SIAD consiste à relier de toutes les façons efficaces possibles ses trois modules :

- **Architecture centralisée**

Dans cette architecture, chaque modèle relève d'un unique module de dialogue et communique avec une seule base de données (figure 5.5).

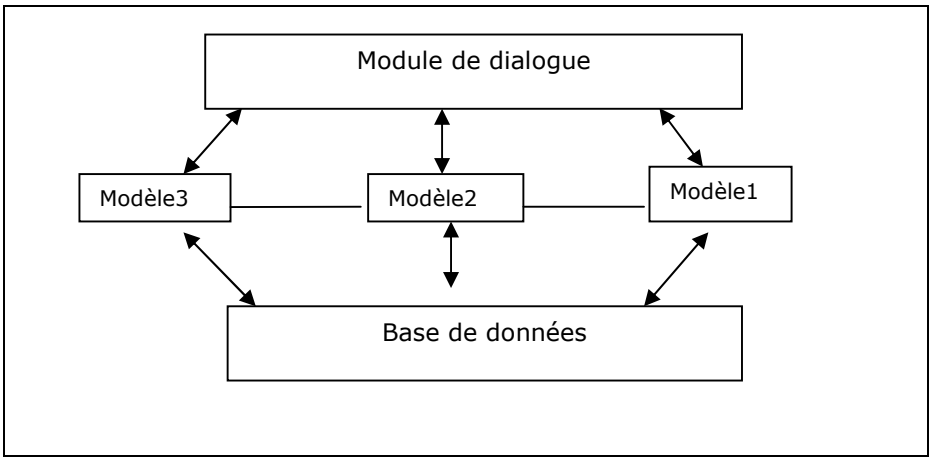


Figure 5.5 : Architecture centralisée

Dans ce cas, l'intégration des différents modèles est excellente et le partage d'une base de données facilite les échanges d'informations entre les modèles. Le contrôle se fait via le module de dialogue. Le principal inconvénient est le manque de souplesse quand on veut introduire un nouveau modèle.

- **Architecture en réseau**

Dans cette architecture (figure 5.6), Chaque modèle possède sa base de données, son module de dialogue ainsi que des modules d'intégration, autrement dit, chaque modèle est ses bases satellites forment un complexe différencié. Le principal avantage de cette

architecture est la grande modularité qui assure que les modifications à l'intérieur d'un complexe ne retentissent pas sur les autres.

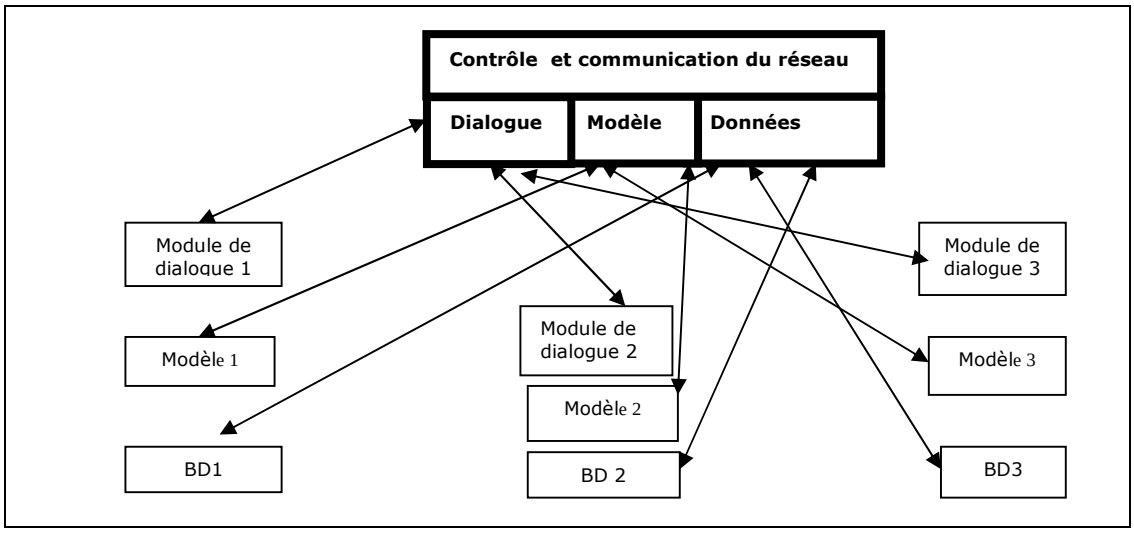


Figure 5.6 : Architecture en réseau

- **Architecture hiérarchisée**

L'architecture hiérarchisée se rapproche du système centralisé. La principale différence c'est que le module de dialogue est divisé, tandis que le module « base de données » est muni d'une couche supplémentaire, cette couche est destinée à remédier au principal défaut de l'architecture centralisée en permettant des adaptations plus faciles (ajout et suppression de modèles) (figure 5.7)

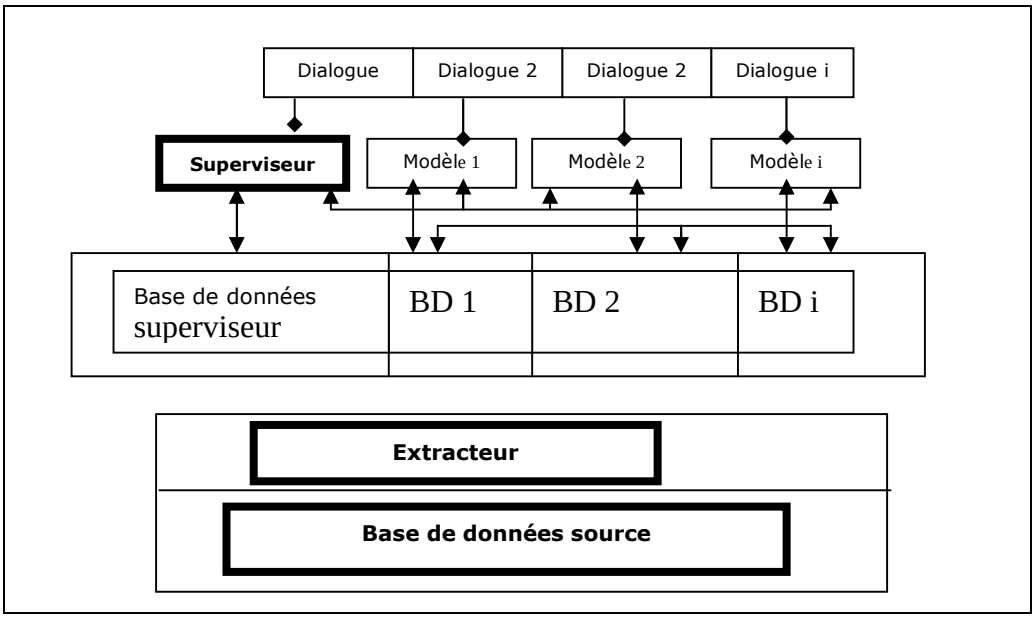


Figure 5.7 : Architecture hiérarchisée

5.4 Systèmes interactifs d'aide à la décision intelligents

Les applications modernes de l'informatique ont conduit à un usage généralisé des représentations de connaissances dans des contextes variés, ces représentations s'appuient sur des formalismes de représentation de connaissances, qui déterminent à la fois les types de connaissances qui peuvent être représentées et les mécanismes de raisonnement sur ces connaissances. La gestion de connaissances se propose ainsi de repérer, formaliser, partager et valoriser les connaissances de l'organisation et en particulier celles qui revêtent un caractère stratégique et décisionnel. La notion d'intelligence artificielle a rendu les systèmes d'aide à la décision plus pratique, plus efficace et plus utiles, pour assister les décideurs d'une organisation dans leurs tâches.

Les SIAD intelligents sont des SIAD qui incluent un système expert, c'est-à-dire un moteur d'inférence et une base de connaissances et de règles les rendant capables de simuler des raisonnements (via les axiomes de la théorie de la décision) et les prises de décision des experts humains dans des domaines particuliers. On distingue encore les EIS (Executive Information Systems) réservés aux dirigeants pour le contrôle et le pilotage de l'entreprise (niveau stratégique) et les GDSS (Group Decision Support Systems) qui traitent des problèmes de décision mettant en jeu une multiplicité de participants qui interagissent de façon plus ou moins coopérative, dans un processus de décision plus ou moins distribué et/ou hiérarchique [66].

5.5 Systèmes coopératifs d'aide à la décision (SCAD)

L'évolution de l'informatique distribuée a permis l'émergence de plusieurs disciplines ainsi que des systèmes supportant la notion de décision, nous pouvons en citer : Groupware, Les outils de Workflow, Systèmes Multi Agents, Systèmes à base de connaissance Coopératifs. Pour qu'un SIAD puisse supporter la notion d'une décision coopérative, il en est nécessaire d'intégrer une dimension de coopération. L'éclatement géographique et temporel du processus de décision, implique une coopération entre les divers acteurs. La coopération peut être située à différents niveaux lors de la réalisation de systèmes coopératifs : [67]

- Coopération homme/machine : pour les concepteurs de systèmes, le problème consiste à doter la machine de capacités supplémentaires afin de guider l'utilisateur dans son processus de résolution de problèmes. L'idée se résume ici à concevoir des interfaces adaptatives intelligentes.
- Système supportant des tâches coopératives : le problème consiste ici à développer des systèmes capables de supporter des tâches collectives impliquant une coopération entre différents acteurs.

- Système coopératif : Le système en lui-même possède des fonctionnalités capables de supporter la coopération.

Gachet (2002) définit une nouvelle vision pour les SIAD distribués. Et il propose la définition suivante : un SIAD distribué est une collection de services et de dispositifs qui sont organisés dans un réseau dynamique, non prédictible, peu stable de matériels et d'entités logicielles travaillant coopérativement pour un but commun d'aide à la décision.

P. Zaraté [67] a défini une architecture d'outils ou plutôt une plateforme d'outils capables de supporter la décision coopérative. Cette plateforme est composée de quatre outils de base qui sont :

- Outil de communications interpersonnelles,
- Outil de gestion des tâches,
- Outil de capitalisation des connaissances,
- Outil d'Interaction Homme / Machine.

5.6 Systèmes d'aide à la décision Spatiale

Les Systèmes Spatiaux d'Aide à la Décision (SSAD), encore appelés Systèmes d'Aide à la Décision Spatiale, ce sont des systèmes informatiques, basés sur des Systèmes de Gestion de Base de Données, diffusant l'information géographique via des interfaces conviviales et ergonomiques, et permettant leur utilisation par un grand nombre d'utilisateurs surtout non spécialisés. Le but principal d'un SSAD étant d'améliorer l'efficacité de la prise de décision en incorporant des jugements du décideur et des programmes sur ordinateur dans le processus décisionnel. Le SSAD ne doit pas remplacer le jugement de l'utilisateur, mais doit l'aider à prendre de « meilleures » décisions. Un SSAD contient typiquement 3 composants : un SGBD (Système de Gestion de Bases de Données) ainsi qu'une base de données géographique, un SGBM (Système de Gestion à Base de Modèles) ainsi qu'un modèle de base, et un SGGD (Système de Gestion et de Génération de Dialogues), ce dernier contient les mécanismes permettant aux données et à l'information de rentrer et de sortir du système.

Les SSAD ont évolué en parallèle avec les SAD. Un SSAD peut être défini comme un système interactif et informatisé conçu pour soutenir un utilisateur ou un groupe d'utilisateurs en réalisant une efficacité plus élevée de prise de décision tout en résolvant un problème spatial de décision semi-structuré. Un SSAD a les caractéristiques suivantes [web 24]:

- une conception explicite pour résoudre des problèmes mal structurés.
- une interface utilisateur puissante et facile à utiliser.
- une capacité de combiner les modèles analytiques avec des données avec souplesse.
- une capacité d'explorer l'espace de solution par des solutions de rechange.

- la possibilité de soutenir une variété de modèles de prise de décision.
- la possibilité de permettre la résolution des problèmes interactifs et récursifs.

De plus il existe quatre possibilités et fonctions de distinction d'un SSAD qui doit [web 24]:

- fournir des mécanismes pour l'entrée des données spatiales.
- permettre la représentation les relations spatiales et des structures.
- inclure les techniques analytiques de l'analyse spatiale.
- fournir le rendement dans une variété de formes spatiales, comme sous forme de cartes.

5.5. Exemple d'un SAD pour la gestion des panneaux publicitaires [70]

La recherche d'information et l'exploration des bases de données, pour trouver une réponse ou prendre une décision, face à des situations critiques demandant une solution optimale, dans le sens de minimisation du coût et de temps d'exécution, devient la préoccupation primordiale des SIAD, notamment dans des situations ambiguës en terme de signification et de compréhension des connaissances ainsi le choix multiples des solutions intermédiaires, une réponse prometteuse à ces objectifs est le développement d'ontologie. Les ontologies occupent aujourd'hui une place pivot dans de nombreux domaines, de l'intelligence artificielle au Web sémantique, du génie logiciel à l'informatique décisionnelle. Les ontologies permettent la spécification de connaissances agréées par une communauté de personnes et partageables sur le Web, ce partage nécessite la représentation de la sémantique des informations afin de les rendre compréhensibles à une communauté d'utilisateurs relativement à un domaine ou à une activité. La création de l'ontologie n'est pas un objectif en soi mais elle est généralement employée pour raisonner à propos des objets du domaine concerné après avoir une modélisation d'un certain ensemble de connaissances relevant du domaine en question. L'un des aspects les plus importants de l'utilisation des ontologies renvoie à leur caractère évolutif, et cela tant pour les ontologies du Web sémantique que pour celles utilisées dans un système d'aide a la décision.

A la suite des problèmes rencontrés dans le service de la voirie de la ville de Constantine, tels que l'intervention lors des incendies, l'organisation du transport, le trafic important notamment à cause des travaux commencés par l'Algérienne des Eaux, ainsi que les travaux liés au tramway. Une ontologie pourra être utile à la prise de décision en facilitant la recherche d'information et l'accès aux diverses sources d'informations. L'ontologie répondra à chaque type de problème rencontré, par exemple face à un incendie elle indiquera le chemin le plus court pour l'intervention des agents de la protection civile ainsi la localisation des bouches d'incendies les plus proches à l'endroit de l'incendie.

Quant au service de la gestion du parc des panneaux publicitaires, le choix du panneau sur lequel l'annonce doit être affichée implique une prise décision capitale en marketing, dans la mesure où la localisation et l'adaptation des contraintes de la requête et les caractéristiques des panneaux (type, mode, taille, etc.) doivent être prise en considération pour prendre décision adéquate. Non seulement ces panneaux sont utilisés pour des raisons publicitaires mais aussi pour la prévention des risques sur les routes, par exemple lors un accident routier, le service du transport qui contrôle la circulation sur les routes, lance deux requêtes aux :

- Service de gestion des panneaux, pour qu'il informe les utilisateurs de la route en leur affichant un message sur les panneaux se trouvant sur les chemins amenant à la route en question pour qu'ils puissent prendre leurs précautions.
- Service d'intervention pour prendre les mesures possibles et régler le problème en libérant la route.

Dans cette étude de cas nous nous sommes intéressés qu'à la construction d'une ontologie urbaine du domaine, modélisant les informations et les tâches du service de la voirie, c'est à dire de constituer un vocabulaire formalisé regroupant le domaine de la voirie, et l'ensemble des concepts ainsi leurs relations, qui est résumé par les deux étapes suivantes:

-l'identification des concepts du domaine en se basant sur une terminologie normalisée.

-l'organisation des concepts pour définir les relations entre eux.

La construction de l'ontologie est effectuée en suivant la démarche de la méthodologie de construction d'ontologie METHONTOLOGY (chapitre 3)

Dans cette partie de ce chapitre nous présentons que l'architecture du système proposé (figure 5.8) et pour plus de détails sur la construction de l'ontologie, vous vous référer à l'annexe A de cette thèse.

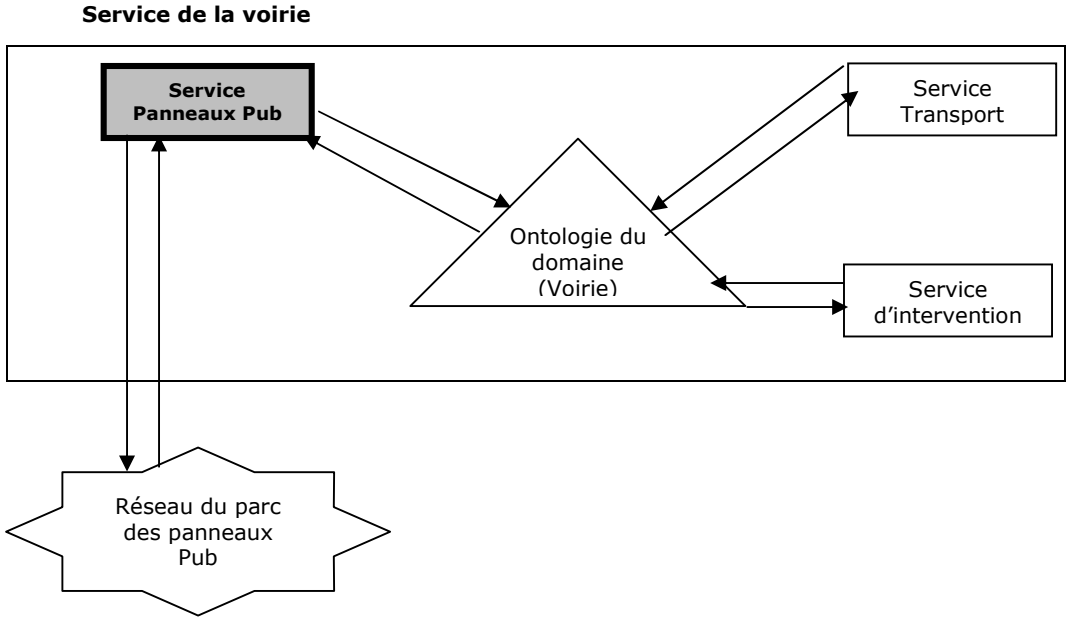


Figure 5.8 : Système de gestion des panneaux publicitaires

5.7. Conclusion

Dans le présent chapitre, nous avons donné une vue globale sur les principes de la théorie de la décision, ainsi nous avons représenté les différentes étapes du processus de décision, et la définition d'un système d'aide à la décision (SAD), ainsi qu'une panoplie de système interactif d'aide à la décision (SIAD) avec leurs différentes architectures. Nous avons montré que les SIAD intègrent de plus en plus de nouvelles fonctionnalités pour en arriver aux Systèmes Coopératifs d'Aide à la Décision proposant une gestion de la coopération entre divers décideurs. Une expérience d'un SAD pour la gestion des panneaux publicitaires basé sur une ontologie a été présentée à la fin de ce chapitre pour montrer l'apport des ontologies pour l'amélioration du processus de la prise de décision.

Chapitre 6 :

Architecture du système

6.1 Introduction

L'objectif primordial pour le développement d'ontologies, c'est de les rendre partageables pour une communauté d'utilisateurs ou de systèmes. Elles sont construites sur la base d'un ensemble communs de concepts fondamentaux et sont utilisables à travers de multiples domaines d'application. Ceci implique l'établissement d'une combinaison des ontologies développées séparément pour permettre la communication et résoudre les conflits d'hétérogénéité sémantique entre les systèmes d'information qui doivent partager des informations basées sur des concepts communs.

Dans ce dernier chapitre nous présentons une description de l'architecture générale de notre système, nous commençons par l'introduction des notions de l'interopérabilité et la réutilisation dans les SI, nous présentons aussi le processus de mise en correspondance (matching) et les mesures de similarités utilisées pour détecter les concepts homologues, nous venons par la suite à la description détaillées du système proposé et ses différents modules, notamment le module d'annotation sémantique qui est un module supplémentaire et complémentaire en même temps pour compléter le processus de création de l'ontologie issus du processus de transformation du modèle E/A et le modèle relationnel. Pour cela l'ensemble de règles est illustré en détail dans ce chapitre ainsi notre méthode de mise en correspondance que nous avons proposé, se caractérisant par une comparaison bidirectionnelle entre deux ontologies.

6.2 Interopérabilité

L'interopérabilité se traduit par la possibilité pour les systèmes d'information de participer à une activité dont les données et l'exécution de tâches sont réparties entre ces systèmes. Ainsi, il est essentiel d'identifier quel service de quel système est en charge de réaliser l'opération demandée et après rendre les résultats aux demandeurs. Bien que les systèmes d'information soient construits pour des besoins spécifiques, ils devraient assurer un certain potentiel pour être interopérables dans l'architecture de l'entreprise [69].

Les ontologies peuvent être utilisées pour la communication entre les personnes ou les machines. Dans le premier cas, les ontologies informelles peuvent servir pour préserver la cohérence et éliminer la confusion terminologique concernant les termes utilisés. En revanche, la communication entre les machines est exprimée en terme d'interopérabilité des systèmes [71]. Les ontologies formelles contribuent essentiellement à résoudre le problème d'hétérogénéité sémantique entre les applications. L'utilisation des ontologies permet d'exprimer explicitement la sémantique inhérente aux applications, dans ce cas-là, les systèmes peuvent coopérer sans avoir aucune confusion [69], en ce qui concerne le domaine de coopération, on peut utiliser des ontologies informelles, sans avoir un contenu sémantique

et cela dans les cas des systèmes travaillant dans le même domaine mais qu'ils ont un besoin d'une coopération pour accomplir certaines tâches.

6.3 Réutilisation

La réutilisation, par définition, concerne l'adaptation d'une application ou bien un module pour être utilisé dans une autre situation [69]. En terme de programmation cette réutilisation est essentiellement basée sur le principe de la généricité pour éviter au développeur de réécrire le code. Des langages de programmation avec des frameworks adaptés ont vu le jour pour atteindre cet objectif. La réutilisation, dans les systèmes peut être exprimée avec des ontologies qui peuvent être conçues et utilisées pour résoudre plusieurs problèmes en même temps en tenant compte de la sémantique des systèmes.

Une deuxième catégorie de la réutilisation des ontologies, est orientée mode de développement en tenant compte des ontologies durant le développement des systèmes et non durant l'exécution. Par exemple dans le cadre du génie logiciel, les ontologies peuvent aider dans le processus de la réutilisation qui dépend essentiellement du partage de la conceptualisation. Par conséquence la réutilisation peut réduire le temps et le coût de développement des systèmes, par contre le manque de la réutilisation est reconnu comme étant l'un des faiblesses des techniques de développement des logiciels [72].

6.4 Transformation

La transformation des schémas de base de données vers d'autres formalismes de représentation de connaissances plus standard et sémantiquement riches, tels que les schémas XML et les schémas d'Ontologies, ont coulé beaucoup d'ancres ces dernières années, vu les exigences des systèmes d'informations actuels pour répondre à des besoins de coopérations et d'interopérabilité, donc la solution est de migrer vers des systèmes robustes supportant ces types de formalisme. Plusieurs travaux de recherches ont été faits pour résoudre ce problème, notamment les outils ayant été développés et qui sont opérationnels. Dans cette partie nous citons trois, étant DataMaster[41], KAON2[42] [43], et RDBToOnto[44]

DataMaster est un plug-in de Protégé spécialisé dans l'import des structures et des données de bases de données relationnelles [79] qui propose une méthode d'import des tables de la base de données relationnelle comme des concepts OWL. Cet outil n'est pas dimensionné pour effectuer une restructuration profonde du modèle et des données sources, il permet néanmoins de migrer les données vers une description dans une ontologie. En utilisant DataMaster,

[41] http://protegewiki.stanford.edu/index.php/DataMaster
[42] http://kaon2.semanticweb.org/
[43] http://kaon2.semanticweb.org/#documentation
[44] http://www.tao-project.eu/researchanddevelopment/demosanddownloads/RDBToOnto.html

chaque table de la base de données sera convertie en un concept et chaque ligne de la table sera convertie en une instance du concept correspondant. Les valeurs des attributs seront instanciées avec les valeurs des champs correspondants de la table [80].

KAON2 est une plateforme de construction d'ontologies développée à l'université de Karlsruhe. Cette plateforme est en fait équipée d'une fonctionnalité de mise en correspondance entre bases de données et ontologies. KAON2 permet de fournir une « vue » sous forme d'ontologie (ce que ses concepteurs appellent une « ontologie virtuelle »), alimentée à la volée par des instances extraites d'une base de données. Les procédés de mapping proposés présupposent l'utilisation de structures ontologiques simples, sans axiomes, quasi-calquées sur les schémas relationnels sources [80].

RDBToOnto [81] est un outil dédié à la conversion de bases de données en ontologies, développé dans le cadre du projet européen TAO (Transitioning Applications to Ontologies). Le développement de RDBToOnto est orienté vers la reconnaissance de structures de catégorisation en analysant conjointement le schéma de la base et les données stockées. Ainsi, le convertisseur RTAXON, central dans cet outil, implémente une méthode générique de reconnaissance des attributs de catégorisation, se basant d'une part sur le nom de l'attribut, d'autre part sur la redondance dans les extensions des attributs à l'aide d'une méthode basée sur l'entropie [82]. Le paramétrage de l'outil en permet d'obtenir un second niveau de profondeur hiérarchique, en détectant les relations de subsomption où chaque table est convertie en une classe et chaque ligne de la table donne lieu à la création d'une sous-classe[80].

Le module de transformation intégré dans notre architecture (figure 6.3) que nous avons développés à propos de cette thèse, il se base sur un jeu de règles de transformations [83], ces règles sont décrites dans les tableaux (6-1,6-2 et 6-3 paragraphe 6.6.1) faisant une migration du paire (schéma entité/association, schéma relationnel).

6.5 Annotation sémantique :

Le processus de transformation décrit dans le paragraphe 6.7 qui permet d'une construction automatique d'ontologies à partir des schémas E/A et BD, est généralement basée sur le texte proprement dit, et le domaine décrit est circonscrit au contenu du texte. Afin de concevoir des ontologies sémantiquement plus riches, nous proposons d'ajouter un module d'annotation sémantique (figure 6.3) pour enrichir l'ontologie obtenue après transformation, Cette étape joue un rôle crucial pour améliorer les résultats du matching tout en donnant plus de sémantique aux termes des ontologies. Elle s'effectue d'une manière manuelle ou semi-

automatique par l'intervention d'un expert du domaine ; généralement les systèmes d'annotation automatique existants sont basés essentiellement sur la description syntaxiques.
Le mot annotation s'exprime par le fait d'ajouter une connaissance sur une ressource. L'annotation sémantique se définit comme le processus qui fixe l'interprétation d'un document en lui associant une sémantique formelle et explicite. Si cette interprétation s'exprime en termes ontologiques, nous parlons d'une interprétation ontologique. L'annotation sémantique a pour objectif de formaliser l'interprétation qui peut être faite des textes sous la forme de méta-données attachées aux textes ou à certains de leurs segments[73].

La description lexicale et structurelle ne permet pas de garantir une vraie interopérabilité entre les systèmes, pour cela, nous devrons définir la sémantique explicite pour les SI et une dimension de coopération pour l'analyse, le développement et l'utilisation des objets sémantiquement enrichis [69]. C'est dans cette optique là, qu'un module d'annotation a été intégré dans l'architecture du système proposée pour permettre le partage de la sémantique et pour assurer réellement l'interopérabilité entre ces systèmes.

6.6 Matching

De l'intégration des schémas de bases de données jusqu'à l'alignement (matching) d'ontologies, la problématique qui a suscité le plus de points ardus à résoudre ces dernières années, est la recherche des correspondances (entre schémas de bases de données, schémas ou documents XML ou encore entre ontologies). Différentes solutions existent pour identifier et associer les concepts communs des différentes sources en utilisant une ontologie. Trois approches peuvent être adoptées [74]: l'approche globale, l'approche multiple et l'approche hybride [75].

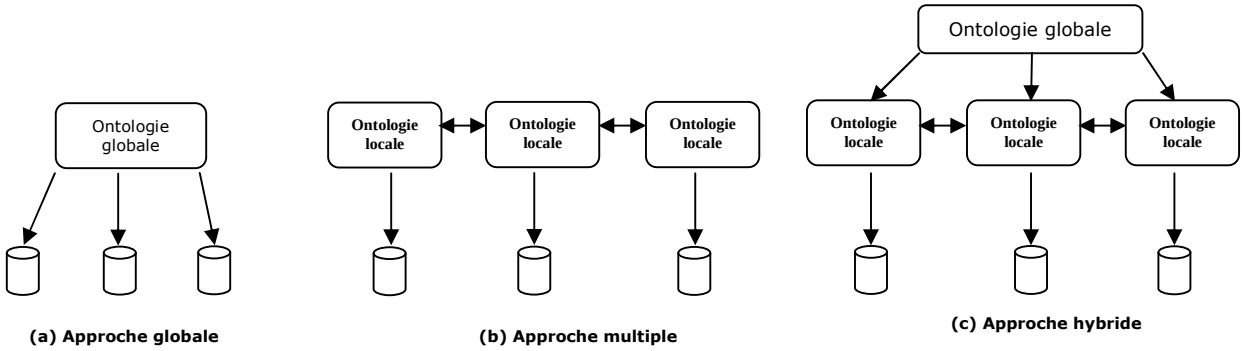


Figure 6.1 : Trois approches pour gérer l'hétérogénéité dans les SI

Dans la première approche, une seule ontologie globale est définie (figure 6.1 (a)). Chaque source est reliée à cette ontologie globale et la similarité sémantique peut être évaluée en vérifiant que les éléments des sources sont reliés au même concept de l'ontologie. Dans la seconde approche (figure 6.1 (b)), (l'approche multiple) une ontologie locale est définie pour

chaque source. Il n'existe pas de vocabulaire commun et un matching (Alignement) entre les ontologies locales est nécessaire pour trouver les correspondances entre termes égaux ou similaires. L'approche hybride (figure 6.1 (c)) mêle les deux solutions précédentes où chaque source a sa propre ontologie définie à partir d'une ontologie globale (ou d'un vocabulaire commun). Les ontologies locales sont ainsi plus facilement comparables et l'ajout de nouvelles sources est aisément supporté. C'est l'approche notamment suivie par [76].
L'alignement d'ontologies représente un grand intérêt dans le domaine de la gestion des connaissances hétérogènes. Le problème de l'alignement de deux ontologies revient à trouver une correspondance entre leurs entités qui sont sémantiquement similaires [87]. Plusieurs méthodes ont été proposées exploitant différents formats d'ontologies [86] [88] :

- **méthode terminologique** : compare les labels des entités. Elle est décomposée en approches purement syntaxiques et celles utilisant un lexique. L'approche syntaxique effectue la correspondance à travers les mesures de dissimilarité des chaînes. Tandis que, l'approche lexicale effectue la correspondance à travers les relations lexicales (exemple : synonymie, hyponymie, etc.) ;
- **méthode de comparaison des structures internes** : compare les structures internes des entités (exemple : intervalle de valeur, cardinalité d'attributs, etc.). Cette méthode utilise différents outils pour proposer des alignements basés sur la correspondance entre les chaînes de caractères représentant les concepts ;
- **méthode de comparaison des structures externes** : compare les relations d'entités avec d'autres entités. Elle reposent sur l'idée que si deux noeuds sont alignés, alors leurs ancêtres et leurs enfants doivent s'aligner mutuellement entre eux (cette hypothèse permettant à la fois de tester la cohérence des propositions d'appariements ainsi que d'en proposer de nouveaux);
- **méthode de comparaison des instances** : compare les extensions des entités (instances des classes);
- **méthode sémantique** : comparent les interprétations (ou plus exactement les modèles) des entités.

Ces méthodes peuvent être classées en deux grandes catégories, celle dite intentionnelle (sur les schémas) et l'autre extensionnelle (sur les instances). La première catégorie comporte les techniques terminologiques et les techniques structurelles. Les techniques terminologiques se basent sur les techniques syntaxiques et linguistiques, par contre les techniques structurelles sont basées sur la comparaison des structures internes et la comparaison des structures externes des entités (classe) des deux ontologies. La deuxième catégorie consiste à comparer les instances des classes. Dans cette thèse nous nous sommes intéressés aux techniques terminologiques syntaxiques et les techniques de comparaison des structures internes.

La classification de Shvaiko et Euzenat [90] qui est basés sur la première catégorie, peut distinguer les approches terminologiques, structurelles et sémantiques (figure 6.2). Les deux premières approches se basent par exemple sur les ontologies et les thésaurus, Alors que la dernière approche exige un raisonnement sémantique.

Cette classification peut être décomposer en catégories: les approches terminologiques qui peuvent être basée sur les chaînes de caractères ou être basées sur l'interprétation des termes en tant qu'objets linguistiques. La catégorie des méthodes structurelles est divisée en deux types de méthodes : celles qui considèrent la structure interne des entités (par exemple, les attributs et leurs types) et celles qui considèrent les relations (équivalence et subsomption).

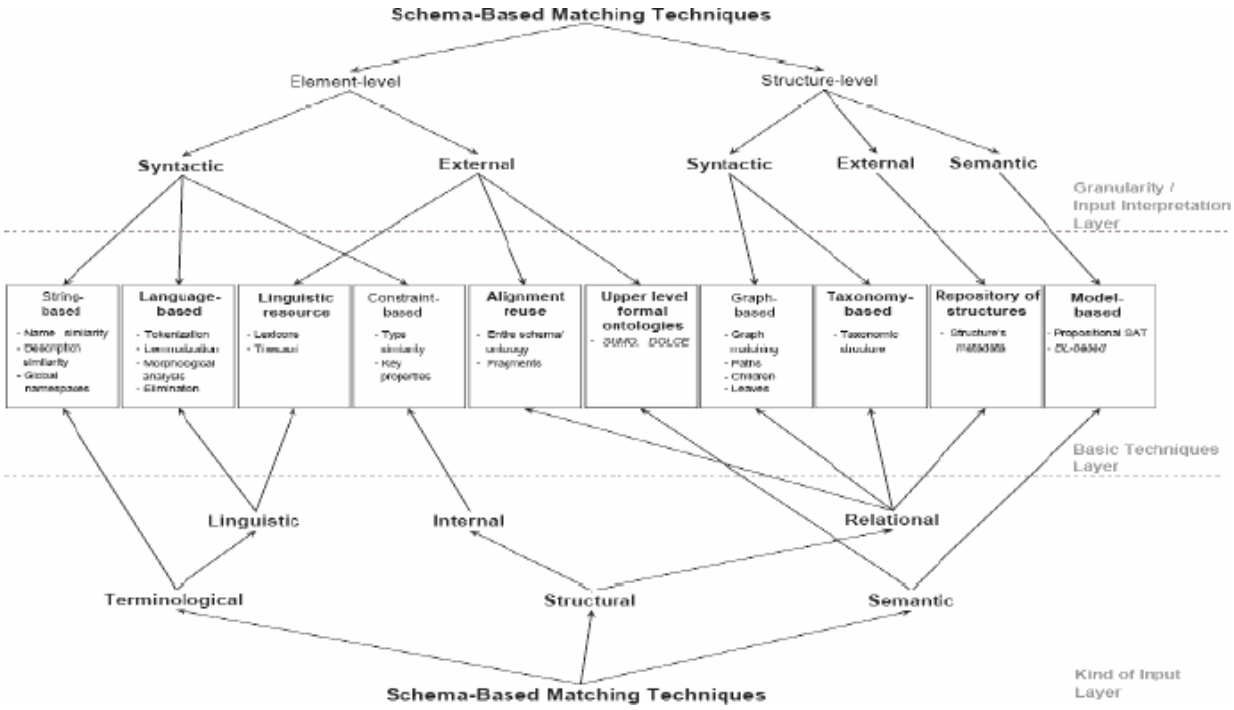


Figure 6.2 : La classification de Shvaiko et Euzenat des approches de matching basée-schéma

D'une manière générale, l'alignement vise à identifier un ensemble d'appariements entre éléments issus de deux représentations structurées distinctes. Parmi les méthodes s'appuyant sur les descriptions intensionnelles (basée-schéma), on trouve les travaux de [91] et [90] qui ont distingué les méthodes terminologiques basées sur les descriptions textuelles des approches structurelles.

6.6.1 Techniques terminologiques

Les méthodes terminologiques sont appliquées en priorité, elles s'appuient sur la comparaison des chaînes de caractères afin d'en déduire une similarité (ou dissimilarité) ou une relation d'hyperonomie/hyponymie. Les techniques terminologiques sont appliquées en priorité. Elles permettent d'exploiter toute la richesse des labels des concepts, en particulier

dans les domaines où les homonymes sont rares et où les labels des concepts sont précis et souvent composés de plusieurs mots. Mais si les noms d'ontologies sont exprimées dans différents langages naturels, ces mesures ne seront pas les plus utiles. Le moyen le plus simple pour comparer deux chaînes de caractères consiste à comparer leur structure : plus elles partageront de caractères ou mots en commun, plus elles seront similaires. En OWL, chaque ressource est identifiée par un URI, mais peut également avoir des propriétés d'annotation attachées. Parmi celles-ci, la propriété rdfs :label est généralement utilisée pour fournir le nom usuel de la ressource.

Les méthodes s'appuyant sur la comparaison des chaîne de caractères sont appelées méthodes **terminologiques syntaxiques**. D'un autre coté, les méthodes terminologiques linguistiques comparent deux chaînes de caractères en ayant recours à une base de données lexicale sous forme de réseau sémantique. A partir de ce type de ressources terminologiques, ces méthodes permettent de calculer une similarité ou de déduire la relation qu'entretiennent deux termes. [92]

Les méthodes syntaxiques prennent en compte seulement la ressemblance physique de chaînes de caractères. Cependant, il existe dans toutes les langues des synonymes pour désigner une même entité, et l'utilisation de similarités syntaxiques sur chaînes de caractères ne permet pas de repérer de proximité dans ce cas. Afin de dépasser ces limites, d'autres méthodes basées sur des connaissances linguistiques existent.

- **Techniques terminologiques syntaxiques**

  Les techniques considérant une chaîne de caractères comme une séquence de caractères seront plus adaptées pour comparer les termes simples (identifiant, labels). Les méthodes syntaxiques permettent seulement de quantifier la ressemblance entre deux chaînes de caractères et sont, par conséquent, limitées à la découverte d'appariements basés sur une relation d'équivalence. Le moyen le plus simple de comparaison de deux chaînes de caractères est de vérifier leur identité. La similarité sera alors binaire, c.-à-d. égale à 1 lorsque $A = B$ et 0 lorsque $A \neq B$.

  Un modèle simple pour comparer deux séquences de caractères est d'ignorer l'ordre d'apparition des lettres dans la séquence. Dans ce cas, la comparaison de deux chaînes a et b, représentées respectivement par les ensembles de caractères A et B, consiste à utiliser une mesure de similarité ou une distance ensembliste.

- **Techniques terminologiques linguistiques**

  Les techniques linguistiques s'appuient sur des connaissances de la langue analysée et/ou sur des bases de données lexicales qui recensent les divers liens sémantiques qu'entretiennent les termes (mots ou groupes de mots) d'une langue. Un exemple de base

de données lexicale est Wordnet développée par l'université de Princeton. Les éléments de base de Wordnet sont des ensembles de termes synonymes appelés synsets. Chaque synset est associé à une définition et à un ensemble de synsets avec lesquels il est en relation. Les principales relations prises en compte pour les synsets de type nom ou groupe nominal sont : l'hyperonymie[45], l'hyponymie, la meronymie[46] et l'holonymie. [92]

6.6.2 Techniques structurelle

Shvaiko et Euzenat [90] distinguent deux types de méthodes structurelles : celles basées sur la structure interne d'une entité et celles basées sur la structure externe. La structure interne d'une entité représente les attributs possédés par l'entité (attributs de type simple, cardinalités, restrictions). La structure externe représente les relations qu'entretient l'entité avec d'autres. Au niveau de la structure externe, on distinguera les méthodes basées sur la relation d'ordre, des méthodes basées sur tous les autres types de relations. Dans notre étude nous prenons en considération que le premier cas, où la comparaison s'effectue sur les deux ensembles d'attributs constituant les deux concepts alignés, généralement ce type de comparaison se base sur des opérations ensemblistes telles que l'union et l'intersection.

6.6.3 Mesures de similarité

D'une façon formelle, l'alignement est défini par la fonction **map** comme suit :

$$map : O_1 \rightarrow O_2 \quad \text{tel que } map(e_1)=e_2 \quad \text{si } sim(e_1,e_2)>=t$$

Formule 6.1 : Définition de l'alignement

Où O_1 et O_2 sont les deux ontologies à aligner, **t** désigne un seuil minimal de similarité appartenant à l'intervalle [0,1], $e_1 \in O_1$ et $e_2 \in O_2$. e_1 et e_2 représentent les entités (concepts ou relations) au niveau des deux ontologies. Le seuil **t** indique le niveau minimum pour que deux entités soient similaires.

L'alignement des deux ontologies O_1 et O_2 revient à déterminer la correspondance entre les différentes entités ontologiques par catégorie (type). Toutes les méthodes d'alignement déterminent des correspondances entre les entités ontologiques en utilisant des mesures de similarité [88]. Les mesures de similarité ou de distance (dissimilarité) permettent d'évaluer la ressemblance ou l' éloignement, entre deux éléments (ou individus) .

[45] Un hyperonyme est une catégorie générale regroupant des sous-catégories. Par exemple, armi les coiffures nous pouvons distinguer les chapeaux et les couronnes. « coiffure » est un hyperonyme de « chapeau » et de « couronne ». [web 14]

[46] La méronymie est une relation partitive hiérarchisée : une relation de partie à tout. Un méronyme A d'un mot B est un mot dont le signifié désigne une sous-partie du signifié de B. La relation inverse est l'holonymie. Par exemple, bras est un méronyme de corps, de même que toit est un méronyme de maison. [web 14]

Comme nous avons déjà dit dans cette thèse que l'approche adoptée est celle qui prenne en considération la description textuelle des entités, que ce soit au niveau terminologique qu'au niveau structurelle. Donc en fonction du modèle de similarité utilisé, une chaîne de caractères **A** peut être modélisée de différentes façons :

- Une séquence de caractères notée A = (a_1; …; a_n) où a_i sont des lettres (ou symboles).
- Une séquence de sous-chaînes de caractères (appelées mots) séparées par des délimiteurs (espaces, tirets bas, ponctuation). Dans ce cas, une chaîne de caractères sera représentée par une séquence de mots notée:
A = (A_1; ...; A_m) où A_j est une chaîne de caractères.

Plusieurs mesures de similarité sont définies en fonction des domaines et contextes dans lesquels sont utilisés et les plus couramment utilisées sont la distance de Hamming ou la similarité de Jaccard. La mesure de Jaccard [93] est calculée par la formule :

$$s_{jaccard}(e_1, e_2) = \frac{|A \cap B|}{|A \cup B|}$$

Formule 6.2: Mesure de similarité de Jaccard

La distance de Hamming est basée sur le complément à 1 de la mesure de Jaccard pour l'estimation de la distance entre deux termes, cette distance peut être utilisée pour mesurer aussi la dissimilarité entre deux ontologies :

Soient O_1 et O_2 deux ontologies et L(·) une fonction retournant les noms des entités dans une ontologie, la distance de Hamming sur les noms de classe est caractérisée par :

$$\delta_{hamming}(O_1, O_2) = 1 - \frac{|L(O_1) \cap L(O_2)|}{|L(O_1) \cup L(O_2)|}$$

Formule 6.3: Mesure de distance de Hamming

Ces deux mesures ne prennent pas en compte l'ordre d'apparition des lettres dans la chaîne de caractères, par contre, si l'on prend en compte cette contrainte, des mesures adaptées dites distances d'édition. Une distance d'édition mesure le coût minimal associé à une séquence d'opérations élémentaires permettant de passer d'une chaîne A à une chaîne B. L'ensemble Op des opérations élémentaires est constitué de :

- Ajout(A, x), l'ajout d'un caractère x dans A.
- Supp(A), la suppression d'un caractère de A.
- Subst(A,x,y), la substitution dans A du caractère x par le caractère y.

Un modèle de coût, noté coût : Op→R, associe une valeur réelle à chacune des opérations o∈Op. Soit $T_{A\to B,}$ l'ensemble des séquences d'opérations s_i = (op_1;...;op_n) (avec $op_x \in$ Op) permettant de passer d'une chaîne A à une chaîne B. La distance d'édition δ entre les chaînes de caractères A et B, est définie par :

$$\delta(A,B) = \min_{Si} \in T_{A\to B} \sum_{op_x \in s_i} coût(op_x)$$

Formule 6.4 : Mesure de distance d'édition

La mesure de Jaro [Jar89] est également utilisée pour comparer deux séquences de caractères. Cette mesure est basée sur le nombre et l'ordre des caractères communs à deux chaînes. Etant donné A = a1…an et B = b1…bm, deux chaînes de caractères, un caractère ai de A sera commun à un caractère bj de B si ai = bj et si i-H <= j <= i + H, avec :

$$H = \frac{\max(\ card\ (A), card\ (B))}{2}$$

Soit A' et B' les chaînes contenant respectivement les caractères de A communs à B et les caractères de B communs à A. Une transposition est définie comme les positions i telles que a'i≠ b'i. $t_{A',B'}$ représente la moitié du nombre de transpositions entre les chaînes A et B. donc la similarité de Jaro entre deux chaînes de caractères A et B est définie par :

$$s_{Jaro}(\mathrm{A},\mathrm{B}) = \frac{1}{3}\left(\frac{card(\mathrm{A}')}{card(\mathrm{A})} + \frac{card(\mathrm{A}')}{card(\mathrm{B})} + \frac{card(\mathrm{A}') - t_{A',B'}}{card(\mathrm{A}')}\right)$$

Formule 6.5 : Mesure de Jaro

Les chaînes A' et B' contiennent les mêmes caractères mais dans des positions qui peuvent être différentes. Comme card(A') = card(B'), ces cardinalités peuvent être interchangées dans l'équation 3.4.1.

Une variante de la similarité de Jaro est celle de Jaro-Winkler [94] qui utilise également la taille du plus grand préfixe commun aux chaînes A et B, noté P.

$$s_{JaroWinkler}(A,B) = s_{Jaro}(A,B) + \frac{\max(4,P)}{10}.(1 - s_{Jaro}(A,B))$$

Formule 6.5 : Mesure de Jaro

Les mesures de similarités que nous avons vu jusqu'à présent sont utilisées sur des concepts isolés mais nous pouvons les utiliser pour estimer la ressemblance au niveau structurelle, parce que en effet la comparaison des structures des concepts revient entre autre à une comparaison de chaque attribut du concept source avec les attributs du concept cible et comme nous savons très bien qu'un attribut est définie par un nom (label) et un type de données (String , Integer, etc.)

6.7 Description du système

L'architecture proposée est constituée de trois grands modules, étant la transformation, l'enrichissement (facultatif) ainsi un module de matching (ou mise en correspondance) entre les ontologies résultats du processus de transformation. A propos de ce dernier module, nous avons proposé une nouvelle méthode de mise en correspondance bidirectionnelle pour détecter et extraire les concepts homologues, constituant par la suite une ontologie globale du domaine partageable entre les différents SI, et cela pour unifier par la suite leur univers de discours. Les concepts homologues sont utilisés pour formuler des requêtes et établir des réponses compréhensibles dans le contexte, là où elles sont utilisées. La méthode proposée se caractérise par une vérification bidirectionnelle assurant la contrainte d'unicité des concepts homologues ainsi l'optimisation des résultats obtenus. Trois types de critères sont utilisés au cours du processus de matching, pour mesurer la similarité entre les concepts de deux ontologies, étant la mesure lexicale, mesure structurelle et mesure sémantique.

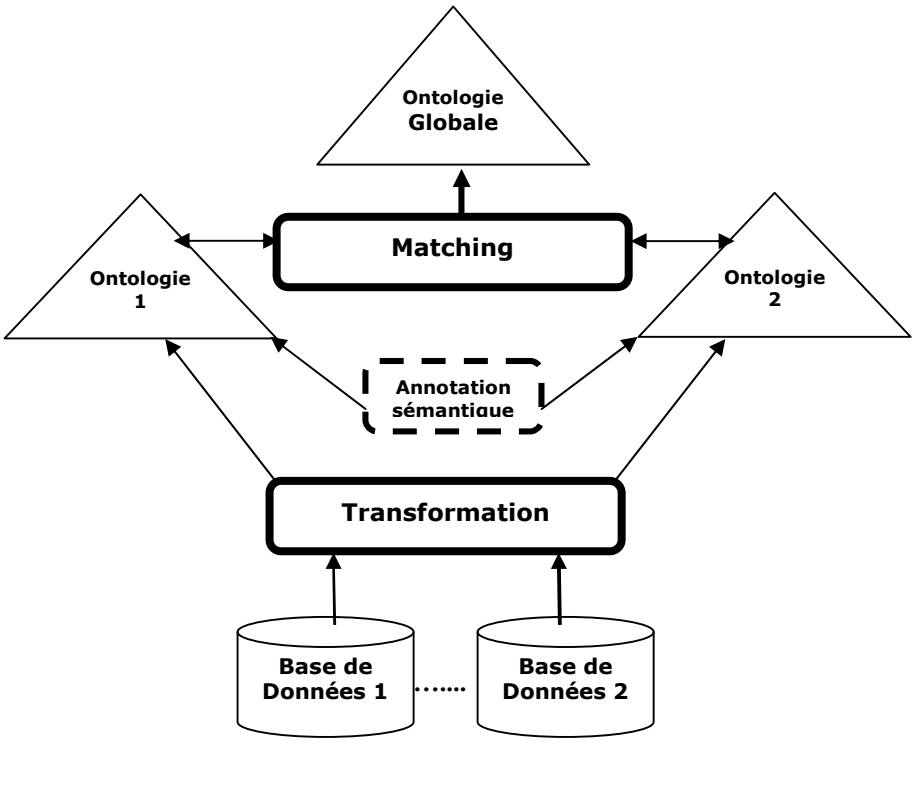


Figure 6.3 : Architecture du système

6.7.1 Module de transformation

Ce module est basé sur un ensemble de règles de transformation (E/A, Relationnel) vers une ontologie. Avant de présenter ce jeu de règles, nous définissons tout d'abord la structure du modèle E/A telle qu'il est présenté dans [84] :

Un modèle E/A est un quintuple S= (L_s , isa_s , att_s , rel_s , $card_s$) avec :

- L_s est un ensemble finit décomposé en un ensemble E_s d'entités, un ensemble A_s d'attributs, un ensemble U_s de rôles, un ensemble R_s de relations, et un ensemble de domaines.
- $isa_s \subseteq E_s \times E_s$ c'est une relation binaires qui représentes la notion de sous-type.
- att_s c'est une fonction qui représente toute entité $E \in E_s$ sous la forme suivante : $att_s(E)$= […, A : D, …]
- rel_s est une fonction qui représente toute relation $R \in R_s$ sous la forme suivante : $rel_s(R)$= […, U : E,…] à condition que chaque rôle est affecté à une, et une seule relation.
- $card_s$ est une fonction de $E_s \times R_s \times U_s$ *à* $N_0 \times (N_1 \cup \{\infty\})$ où N_0, $N1$ deux ensembles finis non négatifs, pour toute rel_s(R)= [U_1 :E_1, ….,U_k :E_k] il existe une $card_s(E_k, R, U_k)$

Nous avons proposé trois types de transformation selon le type de la liste à générer ou selon le type d'entrée à transformer:

6.7.1.1 1er Ensemble de règles : Génération de la liste des concepts

L'ensemble des règles, est défini dans le tableau 6.1 et le résultat de l'application de ces règles, est une liste sous la forme suivante :

Concept	Liste d'attributs	Sous-concept

Au niveau du modèle E/A	Au niveau de l'ontologie conceptuel
Pour toute entité $E \in E_s$	E sera un concept Ajouter une ligne dans la liste avec l'identifiant E comme nouveau concept
Pour tout attribut $A \in A_s$ / $att_s(E)$ = […, A:D,...]	A sera une propriété du concept E Mettre à jour la colonne d'attribut du concept qui porte comme nom l'identifiant E par la propriété E-A avec D le domaine du A **NB** On rajoute l'identifiant E au nom de la propriété, puisqu'au niveau de l'ontologie une propriété ne peut pas être utilisée par deux concepts
Pour tout couple $E_1, E_2 \in E_s$/ E_1 isa_s E_2	Le concept E_1 sera un sous-concept du concept E_2 Mettre à jour la colonne « sous-concept » du concept qui porte comme nom, l'identifiant E_2 par l'identifiant E_1 **NB** Cette colonne va nous servir par la suite pour créer la représentation hiérarchique de l'ontologie.
Pour toute E_1, E_2 , …, $E_i \in E_s$/ E_i isa_s E	Ajouter pour toute entité E_i une ligne dans la liste. Dans la colonne liste d'attributs, ajouter tous les attributs du E avec bien sûr le nomE_i-A_j avec $A_j \in A_s$ / $att_s(E)$ = […, A_j:D,...]

Tableau 6.1 : Génération de la liste des concepts

6.7.1.2 2ème Ensemble de règles : Génération de la liste des relations

L'ensemble des règles est défini dans le tableau 6.2 et le résultat de de l'application de ces règles est une liste sous la forme suivante :

Relation	Concept source	Concept cible	Cardinalité source	Cardinalité cible	Relation inverse

- **Hypothèses :**
  1. Les relations sont binaires, c.à.d. chaque relation relie au plus deux entités : $rel_s(R)= [U_1 :E_1, U_2 :E_2]$
  2. Les attributs des associations dont le type de relation est **n.m** ne sont pas pris en considération.

Au niveau du modèle E/A	Au niveau de l'ontologie conceptuel
Pour tout rôle $U_i \in U_s$ / $rel_s(R)= [U_1 :E_1, U_2 :E_2]$	U_i sera une relation • Ajouter une ligne dans la liste avec l'identifiant U_i comme nouvelle relation. • Mettre à jour la colonne concept-source par l'identifiant E_1 • Mettre à jour la colonne concept-cible par l'identifiant E_2 • Mettre à jour la colonne relation inverse par l'identifiant U_2
Pour toute cardinalité $card_s(E_1 ,R ,U_1)$ et $card_s(E_2 ,R ,U_2)$ / $rel_s(R)= [U_1 :E_1,U_2:E_2]$	si $U_i \in$ dans la colonne relation alors mettre à jour la colonne cardinalité source par $card_s(E_1 ,R ,U_1)$ sinon /* U_i est une relation inverse*/ compare U_i avec tous les éléments de la colonne « relation inverse » s'il existe mettre à jour la colonne cardinalité cible par $card_s(E_2 ,R ,U_2)$
Pour tout attribut $A \in A_s$ / $att_s(R) = [\ldots, A:D,\ldots]$ Avec $R(E_1,E_2)$, $card_s(E_1 ,R ,U_1)$ et $card_s(E_2 ,R ,U_2)$	• si le type de relation est 1:n alors si $card_s(E_1 ,R ,U_1) = 1.1$ alors Mettre à jour la colonne d'attribut du concept qui porte comme nom l'identifiant E_1 par la propriété E_2-A avec D le domaine du A sinon /* $(E_2 ,R ,U_2)= 1.1$*/ Mettre à jour la colonne d'attribut du concept qui porte comme nom l'identifiant E_2 par la propriété E_2-A avec D le domaine du A • si le type de relation est 1:1 alors Mettre à jour la colonne d'attribut du concept qui porte comme nom l'identifiant E_1 ou E_2 par la propriété E_1-A, E_2-A respectivement avec D le domaine du A

Tableau 6.2 : Génération de la liste des relations

6.7.1.3 3ème **Ensemble de règles** : Génération de la liste d'instances

L'ensemble de règles est définit dans le tableau 6.3 et le résultat de l'application de ces règles c'est une liste de la forme suivante :

concept	Liste d'instances
N	NA_1 | NA_2 | ... | En-relation-avec A_k | ... | NA_i

- **Hypothèse** : dans le cas des associations pourvus d'au moins un attribut, nous avons traité que le type de relation **1.1** et le type **1.n**, le type de relation **n.m** n'est pas pris en considération c.à.d. dans le MPD vous ne trouverez pas des tables qui représentant des associations.

Au niveau du schéma extensionnel de la BD	Au niveau de l'ontologie conceptuelle
Pour chaque nom *N* d'une table	*N* sera le nom du concept Ajouter une ligne dans la liste avec l'identifiant *N* comme nouveau concept
Pour tous les attributs de la table *N*	• Ajouter le préfixe *N*- • Comparer les attributs de la table *N* avec les propriétés du concept qui port comme nom l'identifiant *N* • Pour tous les attributs n'appartiennent pas à l'ensemble des propriétés du concept *N* remplacer le préfixe *N*- par « en-relation-avec- »
Pour toute table *N* modifiée	Mettre à jour la colonne « liste d'instance » de la ligne correspondante au nom du concept *N* par la table *N* modifiée

Tableau 6.3 : Génération de la liste d'instance

6.7.2 Module de matching

Le Matching des schémas est une technique qui effectue la découverte de correspondances sémantiques entre les éléments et les attributs des schémas. Le Matching est donc, une opération qui prend par exemple deux schémas de données en entrée et retourne à la fin les valeurs de similarités sémantiques entre les éléments des schémas [85].

Plusieurs travaux ont été réalisés afin de fournir des algorithmes de Matching gérant les correspondances ou incompatibilités des schémas. Notre approche se base et se caractérise

par une vérification bidirectionnelle afin de diminuer le nombre des correspondances de similarités faibles et ambiguës ainsi d'augmenter la qualité des résultats obtenus.

L'algorithme proposé se décompose en quatre étapes. La première étant la normalisation (Prétraitement), la seconde est la vérification syntaxique, la troisième étant la vérification structurelle et la quatrième c'est la vérification bidirectionnelle qui s'utilise en complément de la deuxième et la troisième étape. L'ordonnancement de cet algorithme n'est est pas séquentiel et l'ordre de l'exécution des étapes 2 et 3 est au choix de l'utilisateur qui décide par quel étape veut commencer. Les notations utilisées dans les algorithmes développés sont résumées dans le tableau 6.4.

Les symboles utilisés dans l'algorithme sont les suivants :
- O_1, O_2 : les deux ontologies à aligner en format OWL. - n est le nombre de concepts de l'ontologie O_1 - m représente le nombre de concepts de l'ontologie O_2 - N_{1i} : le nœud de l'ontologie O_1 $\{\forall i \in [1..n] / N_{1i} \in O_1\}$ - N_{2j} : le nœud de l'ontologie O_2 $\{\forall j \in [1..m] / N_{2j} \in O_2\}$ - SimSyn : la valeur de similarité syntaxique - SimStr : la valeur de similarité structurelle - MSsyn : Matrice de la similarité syntaxique de taille (n,m). - MSstr : Matrice de la similarité structurelle de taille (n,m).
Tableau 6.4 : Notations utilisées dans l'algorithme

Dans le Tableau 6.5, nous avons esquissé les primitives les plus importantes qui permettent de lire et de parcourir des ontologies OWL. Le package « com. hp.hpl.jena.ontology » fournit un ensemble de classes et méthodes pour manipuler et accéder à un tel type d'ontologie.

Rôle	Instruction
Création d'un modèle d'ontologie **m**.	OntModel m=ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM,null);
Chargement et création d'une copie de l'ontologie.	m.getDocument Manager. addAltEntry()
Lecture de l'ontologie avec la méthode read.	m.read ( );
Lister les classes de l'ontologie. Prendre toutes les classes de l'otologie	for (Iterator i= m.listClasses ( ) ; i.hasNext()); {OntoClass cls= (OntClass) i.next();

Nom local de la classe par la méthode	getLocalName () System.out.println(cls.getLocalName);
Lister les super classes de cls.	for (Iterator i =cls. listSuper.Classes( ) ; i .haslNext()) { Sysem.out.print(((OntClass)i.next()).getLocalName() +" "); }
Lister les sous classes de la classe cls.	for (Iterator i =cls. listSub. Classes( ) ; i .haslNext()) { Sysem.out.print(((OntClass)i.next()).getLocalName() +" ");}

Tableau 6.5 : Primitives d'exploitation d'une ontologie OWL

L'algorithme d'alignement proposé (figure 6.5), prend en entrée deux ontologies en format OWL, une ontologie dite source (O_1) et une ontologie dite cible (O_2), et fournit en sortie un fichier XML (figure 6.4). L'algorithme fonctionne en deux grandes étapes non successives. La première étape, implémentée par le biais de la fonction FuncSyn() qui permet de calculer la similarité syntaxique entre les deux ontologies et la deuxième étape implémentée par la fonction FuncSruct(), permettant de calculer la similarité structurelle. Nous pouvons ajouter une troisième étape pour mesurer la sémantique entre les deux ontologies alignées. Mais ce type de problème n'a pas été abordé dans cette thèse, vu le contenu des ontologies considérées qui sont issus d'une transformation simple sans prendre en considération l'aspect sémantique, sauf si ces ontologies ont subi une étape d'annotation sémantique (figure 6.3).

```
< concept1>N1</concept1>
< concept2>N2</concept2>
<similarité>sim</similarité>
```

Figure 6.4 : Structure du fichier XML contenant le résultat du matching

Dans cet algorithme nous considérons que les alignements dans lesquels les relations sont d'équivalence (≡) ou de subsomption (⊆,⊇) entre entités nommées de chaque ontologies (alignements simples).

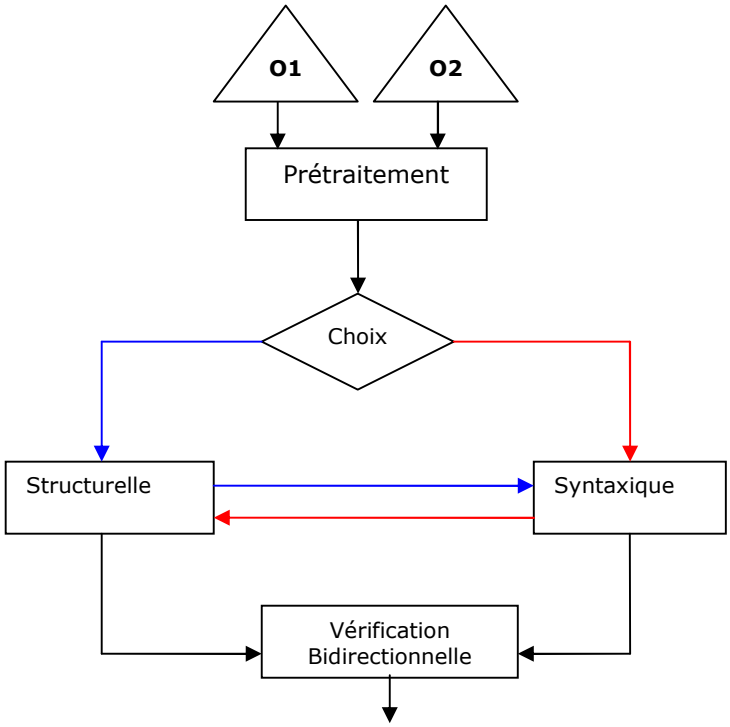


Figure 6.5 : Organigramme de l'algorithme

6.7.2.1 Etape 1 : Normalisation (Prétraitement)

Cette étape a pour objectif de réaliser un prétraitement sur les schémas d'ontologie. Elle prend comme entrée une ontologie OWL et l'analyser grâce au parseur des schémas d'ontologies pour simplifier le contenue des chaînes de caractères représentant les concepts afin de les ramener à des formats équivalents [95]. Dans cette étape de normalisation terminologique, nous effectuons plusieurs opérations successives :

- La normalisation de la casse (majuscule/minuscule) : consiste à convertir chaque lettre de la chaîne en minuscule (ou majuscule);
- supprimer les espaces et les remplacer par le caractère « - » pour chaque bloc de blancs;
- Supprimer les mots de liaisons pour ne garder que les noms (e.g. suppression de « l' », « le », « la », etc.) ;
- Remplacer les accents, cédilles, etc., par des caractères standards de l'alphabet et ce pour corriger les erreurs d'orthographe en minimisant les problèmes d'ambiguïtés.

6.7.2.2 Etape 2 : Vérification syntaxique

La vérification syntaxique s'effectue à travers la fonction FuncSyn(), qui prend comme paramètres d'entrée les deux ontologie O_1, O_2 et le seuil **t1** fixé par l'utilisateur. Elle fournit à la fin de son exécution une matrice MSsyn (figure 6.6) contenant les différentes

valeurs de similarité syntaxique calculées en appliquant la fonction mesureSyntaxique (N_1, N_2). Cette fonction prend plusieurs formes et variantes, elle exécute à la fois une seule mesure de similarité qui est sélectionnée par l'utilisateur via une interface graphique (voir Annexe B). Les mesures de similarités que nous avons utilisé sont celles définies dans la section 6.6.3

	N_{21}	N_{22}	.....	N_{2m}
N_{11}	SimSyn			
N_{12}				
⋮				
N_{1n}				

Figure 6.6 : Structure de la matrice de similarité

Fonction : FuncSyn().
Données :
1) O_1 et O_2 : deux ontologies à aligner
2) t1 : seuil de similarité syntaxique
Résultat : MSsyn : Matrice de similarité syntaxique
début
pour chaque ($N_{1i} \in O_1$) **faire** /* parcours des nœuds de l'ontologie O_1 */
pour chaque ($N_{2j} \in O_2$) **faire** /* parcours des nœuds de l'ontologie O_2 */
SimSyn ← mesureSyntaxique (N_{1i},N_{2j})
si SimSyn >= t_1 alors
MSsyn(i , j) ← SimSyn
retourner (MSsyn)
fin

6.7.2.3 Etape 3 : Vérification structurelle

Dans cette étape de vérification, nous appliquons une mesure de similarité structurelle qui s'exprime par la fonction mesureStructure (N_1, N_2). Cette mesure est utilisée à travers la fonction FuncStuct() qui prend en entrée les deux ontologie O_1, O_2 et le seuil **t2** qui est toujours fixé par l'utilisateur. Elle fournit à la fin de son exécution une matrice MSstr ayant

une structure similaire à la matrice de la figure 6.6 mais contenant des valeurs concernant des mesures de similarité structurelle.

Fonction : FuncStuct().
Données :

1) O_1 et O_2 : deux ontologies à aligner
2) t_2 : seuil de similarité structurelle
Résultat : MSstr: Matrice de similarité lexicale

début

pour chaque ($N_{1i} \in O_1$) **faire** /* parcours des nœuds de l'ontologie O_1 */

pour chaque ($N_{2j} \in O_2$) **faire** /* parcours des nœuds de l'ontologie O_2 */

SimStr = mesureStructure(N_{1i},N_{2j})

si SimStr >= t_2 alors

MSstr(i , j) ← SimStr

retourner(MSstr)

fin

La fonction mesureStructure (N_1, N_2) compare la structure interne des deux concepts N_1 et N_2 en vérifiant le nombre, les noms des attributs et leur types de données. La vérification sur les noms fait appel à la fonction mesureSyntaxique (n_1 ,n_2) mais cette fois-ci prend comme paramètres les deux chaînes de caractères dénotant les noms des deux attributs.

6.7.2.4 Vérification bidirectionnelle

La vérification bidirectionnelle, consiste à faire un appariement dans les deux sens des deux ontologies en question, le but de cette étape est d'éliminer l'ambiguïté au niveau des candidats trouvés (concepts homologues). La figure 6.7, montre que le concept 24 de l'ontologie cible (O_2) est un homologue de deux concepts 14 et 15 de l'ontologie source (O_1). Si on prend en considération les relations de type équivalence (≡), cela veut dire que les deux concepts 14 et 15 sont les mêmes c'est-à-dire ils représente la même entité physique, chose qui se contredit avec la définition de l'ontologie qui n'accepte pas la redondance de définition.

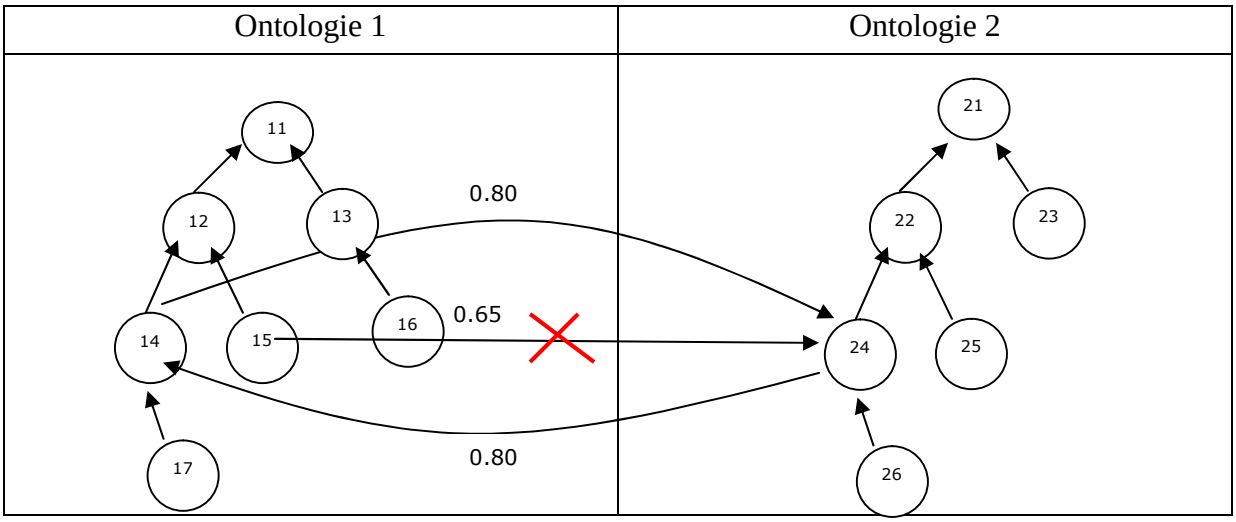


Figure 6.7 : Vérification bidirectionnelle

6.8 Conclusion

La conception et la création d'une ontologie, n'est pas vraiment un objectif en soit (ou principal) mais plutôt intermédiaire, dans notre cas l'ontologie est développée pour assurer l'interopérabilité entre les systèmes d'information en question. L'architecture proposée est focalisée autour d'une ontologie globale du domaine qui résulte d'une opération de mise en correspondance (matching) entre deux ontologies locales issues en appliquant un jeu de règles de transformation sur le modèle conceptuel E/A et le modèle relationnel des bases de données.

L'objectif ici est d'unifier l'univers de discours entre les différents SI partageant d'une manière ou d'une autre certaines ressources de données et travaillent dans des domaines proches ou identiques.

La méthode de matching proposée se caractérise par une vérification bidirectionnelle en se basant sur une vérification terminologique-syntaxiaue et une autre structurelle interne, cette méthode peut supporter d'autres étapes de vérification telle que linguistique et structurelle-externe, le but ici c'est pas de décrire une méthode robuste de matching mais de montrer sa faisabilité dans notre système.

Conclusion générale et Perspectives

Conclusion et perspectives

La conception et le développement des systèmes n'est pas une tâche aussi facile que le mot conception, un tel travail nécessite une collaboration d'un ou plusieurs groupes de personnes dans des domaines pluridisciplinaire. Dans cette thèse, nous avons essayé de décrire un cadre général d'un futur système capable de gérer l'interopérabilité entre différents systèmes d'information (SI) notamment les SI de type spatial, ces derniers se caractérisent par la diversité de la nature et le grand volume de données utilisés, vu les besoins des applications qu'ils les utilisent.

Les modèles conceptuels pour les systèmes d'informations classiques que ce soit de type de données simple ou spatial n'ont pas vraiment une grande différence et ça reste toujours en terme d'entité et relation (association), mise à part le type des attributs constituant les entités.

Le besoin de migrer vers des solutions plus efficaces et robustes, nous amené à la réflexion sur la création des mécanismes de transformation de l'infrastructure conceptuelle de ces systèmes tout en gardant leur principe de fonctionnement et tout en minimisant la perte d'information.

Dans cette thèse nous avons décrit un cadre général de transformation qui se base sur un jeu de règles s'appliquant sur un modèle conceptuel Entité-Association et un modèle relationnel d'une base de données. L'application des ces règles sur un système classique assure une migration vers un modèle ontologique. Le choix de cette destination s'explique par les avantages et les points forts apportés par la notion d'ontologie, notamment la puissance de sa structure pouvant supporter l'hétérogénéité de différents points de vu. Notre idée ne s'arrête pas à ce stade de transformation mais elle va plus loin. Les ontologies indépendantes qui sont issues de cette transformation ne peuvent pas gérer le problème de l'interopérabilité, donc il faut les doter par un mécanisme pouvant assurer certaine coopération entre ces ontologies. La mise en correspondance ou le matching semble la solution la plus efficace pour aligner des ontologies afin d'unifier leurs univers de discours. De sa nature l'ontologie est définie comme une spécification d'une conceptualisation partagée, donc elle peut spécifier la conception ayant été faite par les SI classiques et les réunir sous un seul modèle conceptuel.

Nous sommes convaincu que le travail coopératif entre experts de différents domaines est décisif pour la construction et la maintenance des ontologies via la clarification et la catégorisation des termes du domaine. Toutefois, les experts sont limités par des contraintes de mobilité ou de temps, mais aussi par des paradigmes difficiles à changer [ ]. Par conséquent, notre système peut aider les experts à prendre des décisions pour mettre un consensus sur le vocabulaire et la manière de gérer les choses entre les différents acteurs.

Les perspectives envisagées pour améliorer ce travail sont multiples, elles concernent deux points essentiels, étant l'enrichissement automatique de la sémantique de l'ontologie et la création d'une plate forme basée sur un système multi-agents.

Dans un système multi-agents ouvert, les agents sont hétérogènes, c'est-à-dire qu'ils n'ont pas forcément été conçus au même moment, ni par les même organisations. Ainsi, pour qu'un agent puisse découvrir et exploiter les services disponibles, il doit être capable de communiquer avec d'autres agents, c'est-à-dire interopérer[17]. C'est dans ce contexte que la nécessité d'introduire la notion d'ontologie pour améliorer l'interopérabilité dans un système ouvert ; donc nous proposons que chaque agent possède sa propre ontologie, qui l'utilise pour décrire formellement ses connaissances tout en assurant la transformation et l'enrichissement sémantique. Cette ontologie doit lui permettre, de découvrir et sélectionner les services proposés par les agents du système. Un autre type d'agent peut être intégré dans cette nouvelle infrastructure, il s'agit d'un agent coordinateur qui s'occupe de l'opération de matching.

Annexe

Annexe A : Système de gestion des panneaux publicitaires

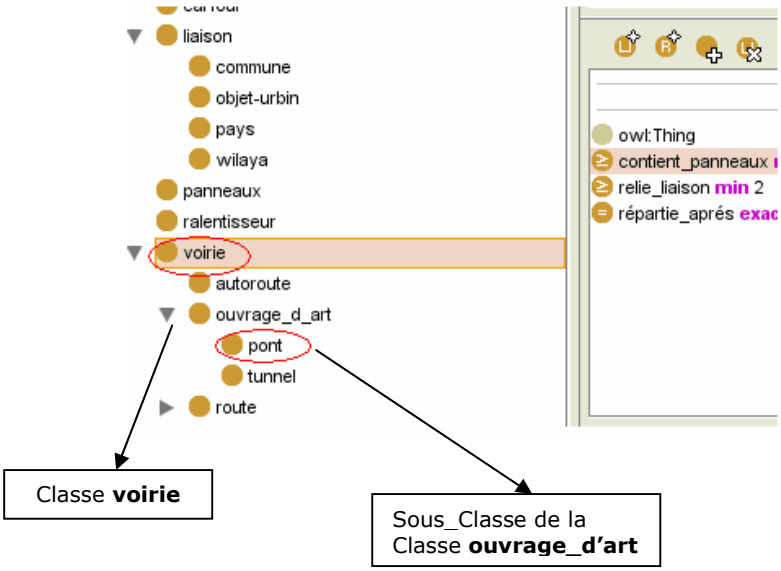


Figure A.1 : Hiérarchie des classes (Service de la voirie)

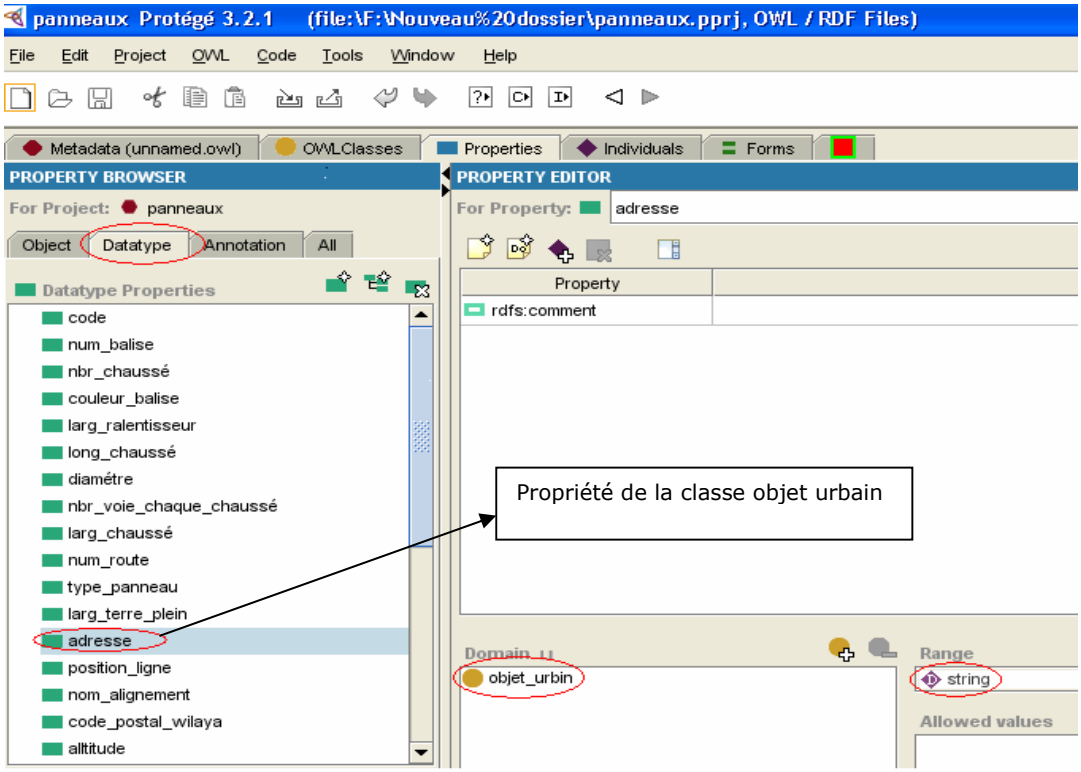


Figure A.2 : Les propriétés des classes

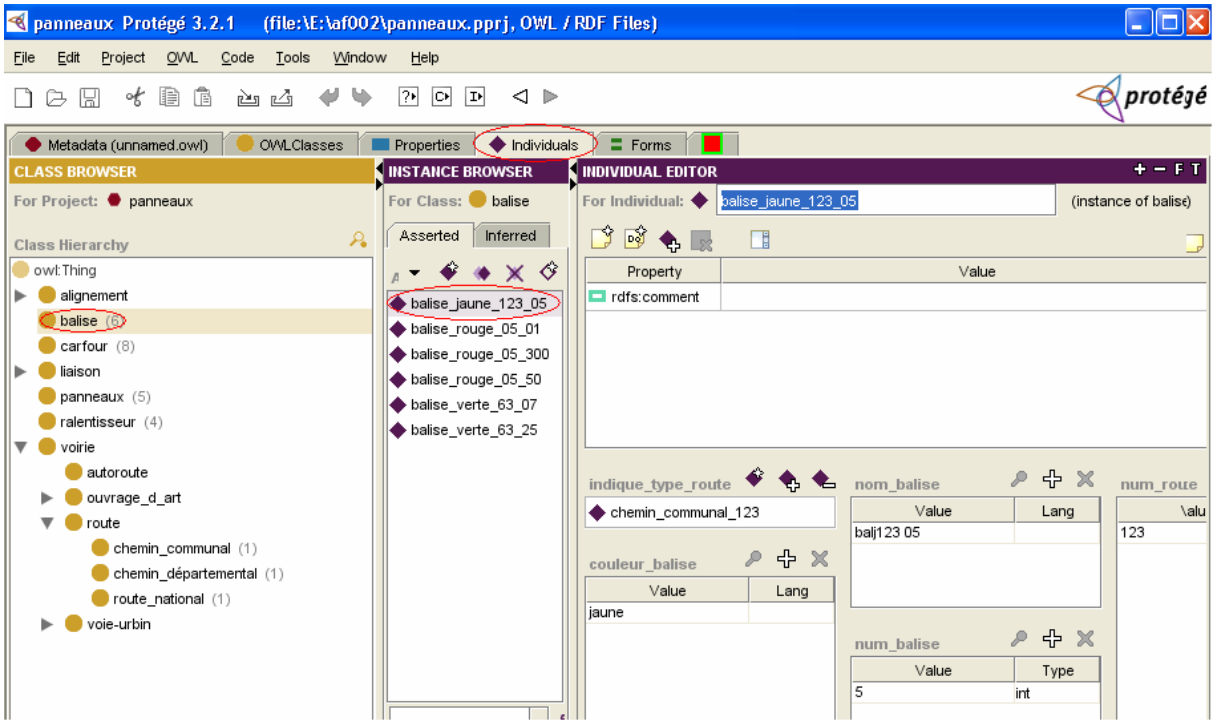


Figure A.3 : Instanciation

Annexe B : Techniques et langages utilisés

C.1 Eclipse

Eclipse est un environnement de développement intégré (Integrated Development Environment), développé par IBM dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques.

Eclipse utilise énormément le concept de modules nommés "plug-ins" dans son architecture. D'ailleurs, hormis le noyau de la plate-forme nommé "Runtime", tout le reste de la plate-forme est développé sous la forme de plug-ins. Ce concept permet de fournir un mécanisme pour l'extension de la plate-forme et ainsi fournir la possibilité à des tiers de développer des fonctionnalités qui ne sont pas fournies en standard par Eclipse.

Les principaux modules fournis en standard avec Eclipse concernent Java mais des modules sont en cours de développement pour d'autres langages notamment C++, Cobol, mais aussi pour d'autres aspects du développement (base de données, conception avec UML, ...). Ils sont tous développés en Java soit par le projet Eclipse soit par des tiers commerciaux ou en open source.

Annexe

Les modules agissent sur des fichiers qui sont inclus dans l'espace de travail (Workspace). L'espace de travail regroupe les projets qui contiennent une arborescence de fichiers.
Bien que développé en Java, les performances à l'exécution d'Eclipse sont très bonnes car il n'utilise pas Swing pour l'interface homme-machine mais un toolkit particulier nommé SWT associé à la bibliothèque JFace. SWT (Standard Widget Toolkit) est développé en Java par IBM en utilisant au maximum les composants natifs fournis par le système d'exploitation sous jacent. JFace utilise SWT et propose une API pour faciliter le développement d'interfaces graphiques.
Eclipse ne peut donc fonctionner que sur les plate-formes pour lesquelles SWT a été porté. Ainsi, Eclipse 1.0 fonctionne uniquement sur les plate-formes Windows 98/NT/2000/XP et Linux.

SWT et JFace sont utilisés par Eclipse pour développer le plan de travail (Workbench) qui organise la structure de la plate-forme et les intéractions entre les outils et l'utilisateur. Cette structure repose sur trois concepts : la perspective, la vue et l'éditeur. La perspective regroupe des vues et des éditeurs pour offrir une vision particulière des développements.

Eclipse possède de nombreux points forts qui sont à l'origine de son énorme succès dont les principaux sont :

- Une plate-forme ouverte pour le développement d'applications et extensible grâce à un mécanisme de plug-ins.
- Plusieurs versions d'un même plug-in peuvent cohabiter sur une même plate-forme.
- Un support multi langage grâce à des plug-ins dédiés : Cobol, C, PHP, C# , ...
- Support de plusieurs plate-formes d'exécution : Windows, Linux, Mac OS X, ...
- Malgré son écriture en Java, Eclipse est très rapide à l'exécution grâce à l'utilisation de la bibliothèque SWT.
- Une ergonomie entièrement configurable qui propose selon les activités à réaliser différentes « perspectives ».
- La construction incrémentale des projets Java grâce à son propre compilateur qui permet en plus de le code même avec des erreurs, de générer des messages d'erreurs personnalisés, de sélectionner la cible (java 1.3 ou 1.4) et de mettre en oeuvre le scrapbook (permet des tests de code à la volée).
- Une exécution des applications dans une JVM dédiée sélectionnable avec possibilité d'utiliser un débogueur complet (points d'arrêts conditionnels, visualiser et modifier des variables, évaluation d'expression dans le contexte d'exécution, changement du code à chaud avec l'utilisation d'une JVM1.4, ...).
- Le gestionnaire de mise à jour permet de télécharger de nouveaux plug-ins ou nouvelles versions d'un plug-in déjà installées à partir de sites web dédiés (Eclipse 2.0).

C.2 Jena

Le Framework Jena est une API Java permettant de lire et de manipuler des ontologies décrites en documents OWL, RDF(S),OIL+DAML et d'y appliquer certains mécanismes d'inférences .Jena est diffusé en code source libre.

C.3 Intégration du Framework Jena dans Eclipse

Pour intégrer l'API Jena dans l'IDE Eclipse, nous avons suivi les étapes suivantes :

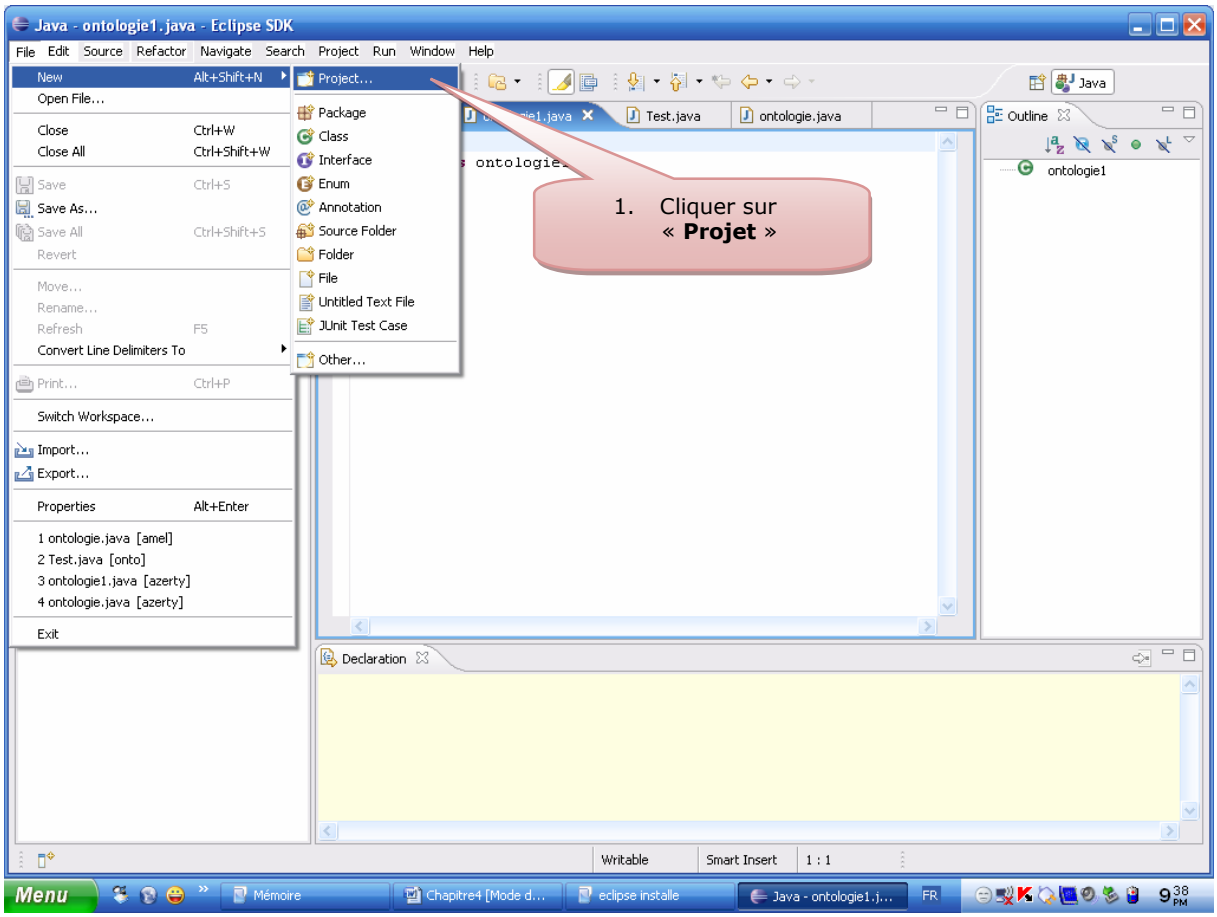


Figure B.1 : Création d'un nouveau projet (Etape 1)

Annexe

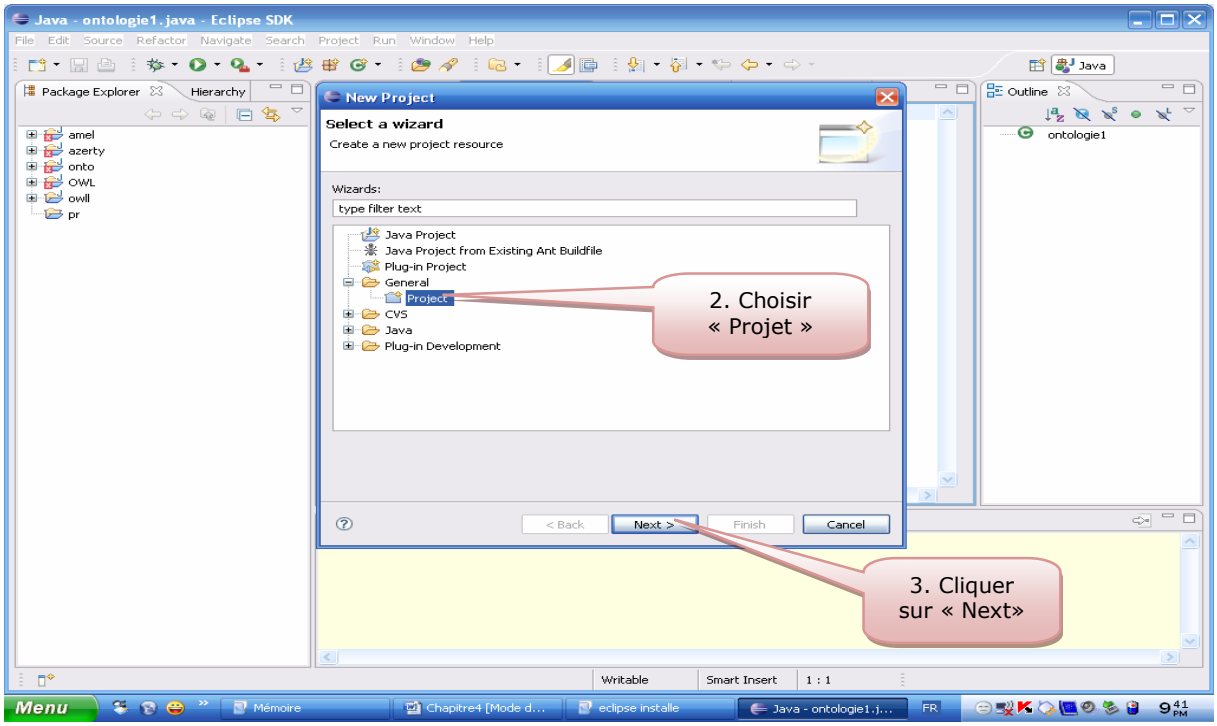


Figure B.2 : Création d'un nouveau projet (Etape2)

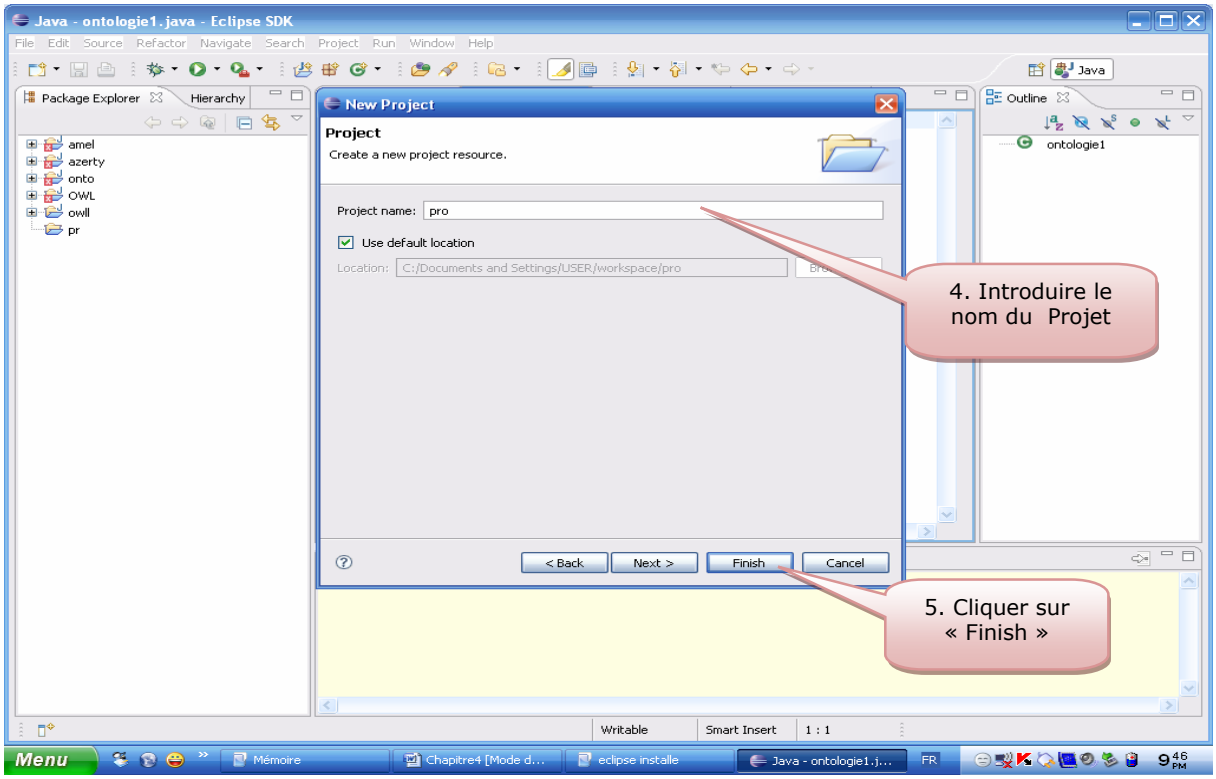


Figure B.3 : Création d'un nouveau projet (Etape 3)

Annexe

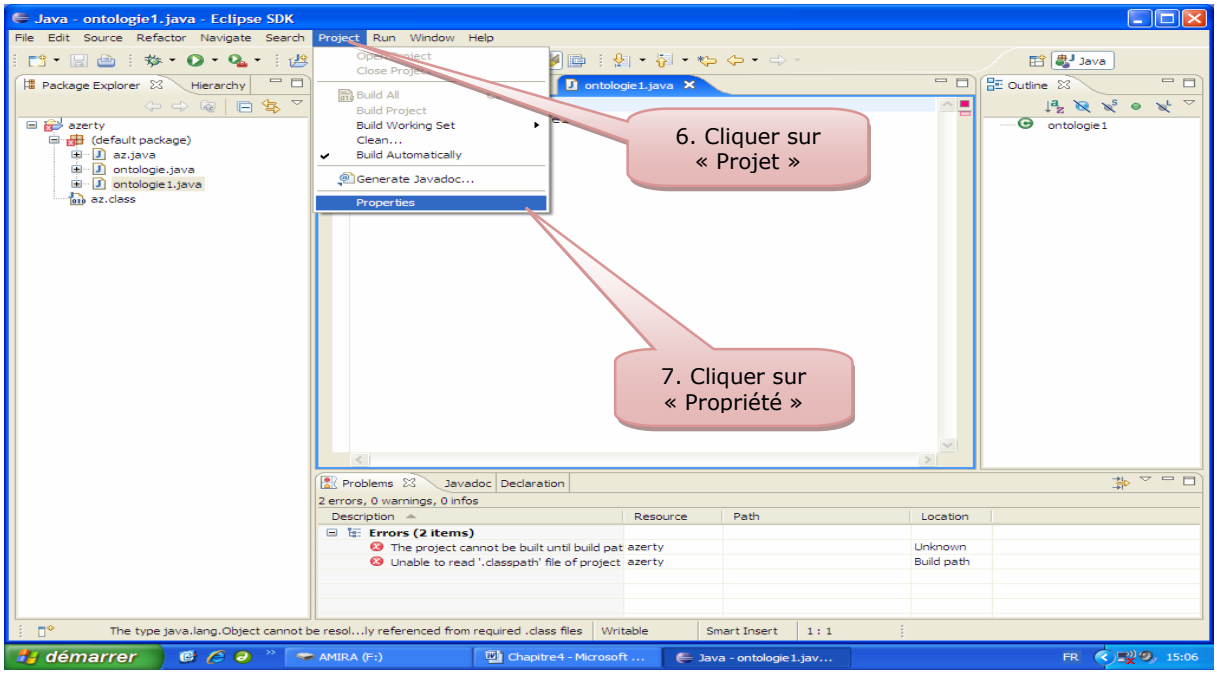


Figure B.4 : Importation de l'API Jena (Etape 1)

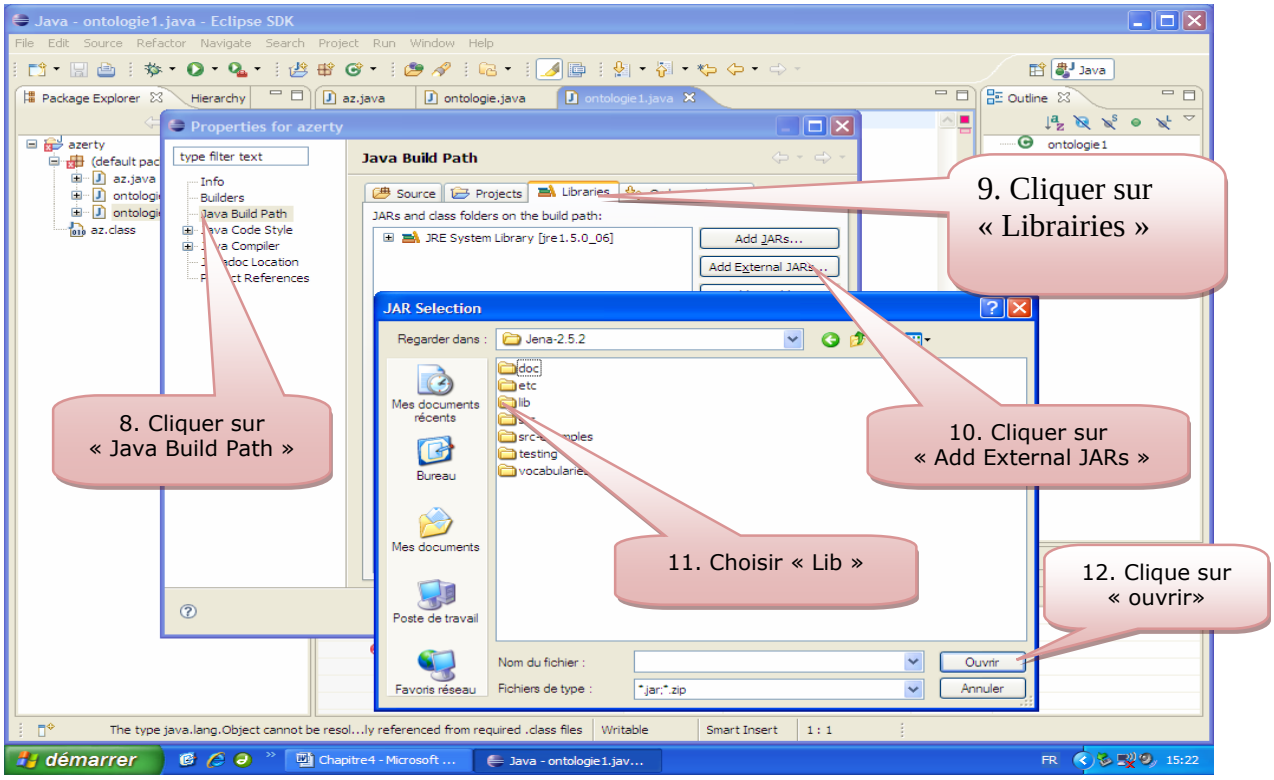


Figure B.5 : Importation de l'API Jena (Etape 2)

Annexe

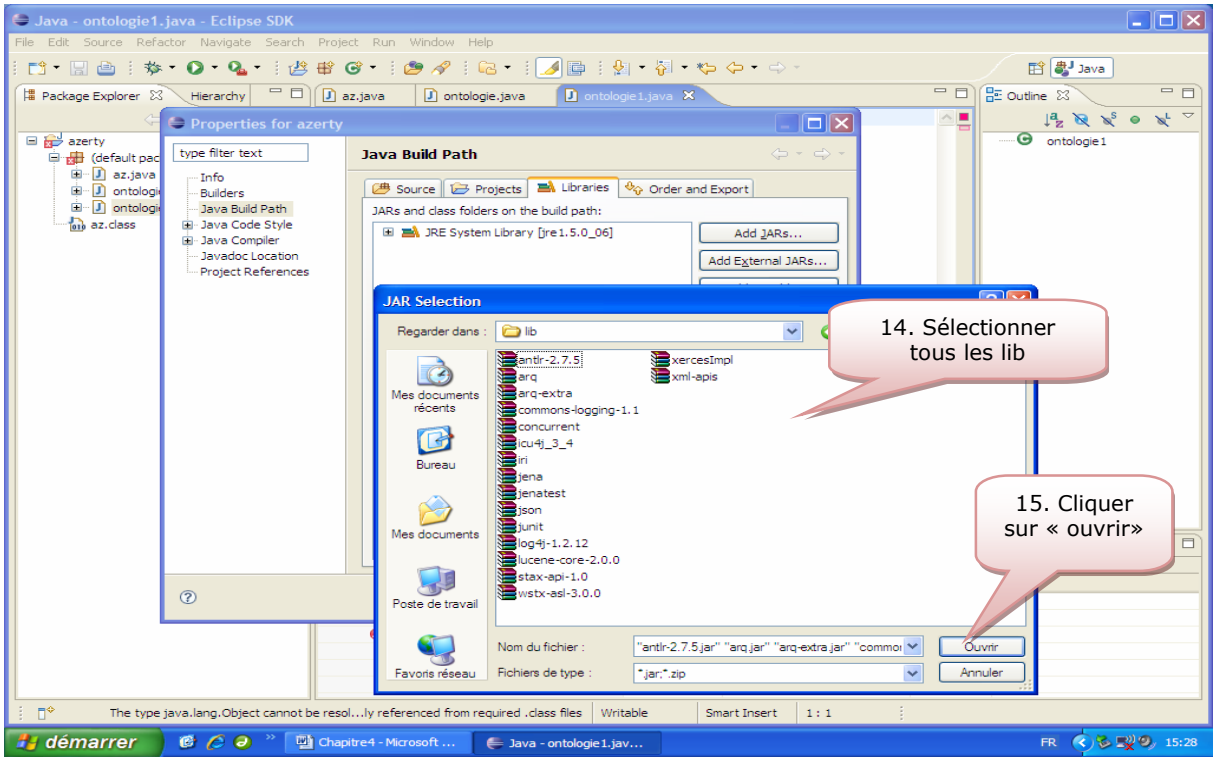


Figure B.6 : Importation de l'API Jena (Etape 3)

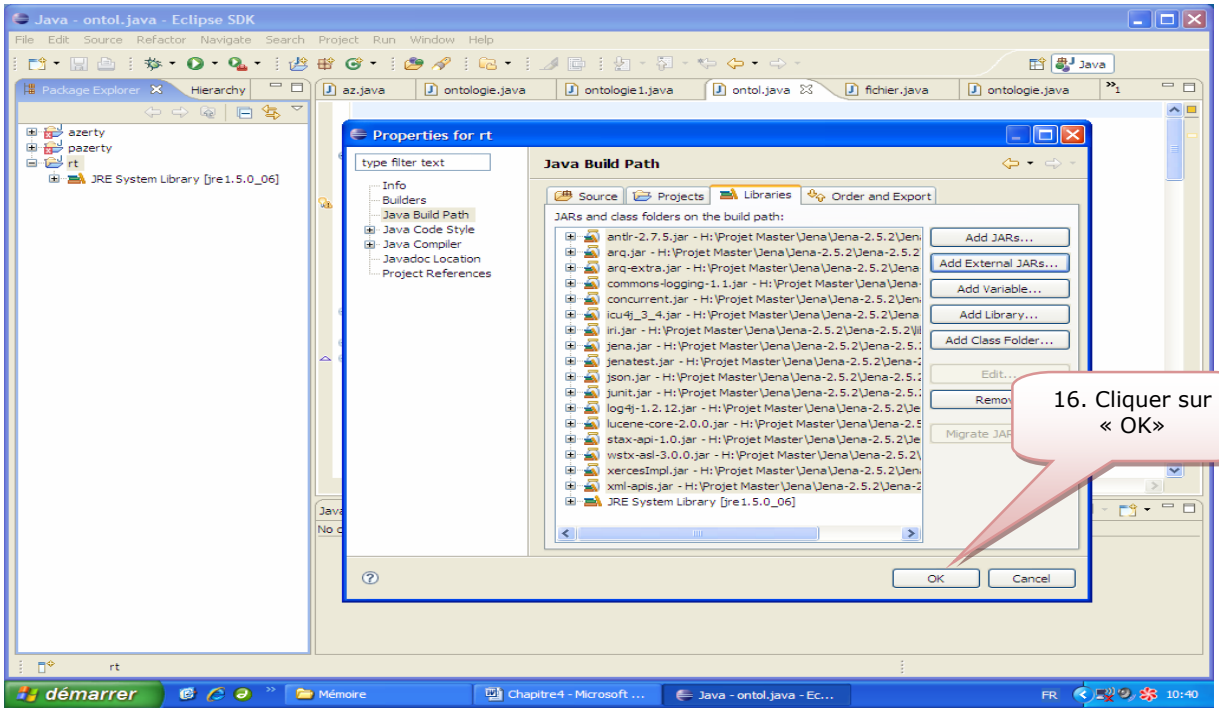


Figure B.7 : Importation de l'API Jena (Etape 4)

Annexe

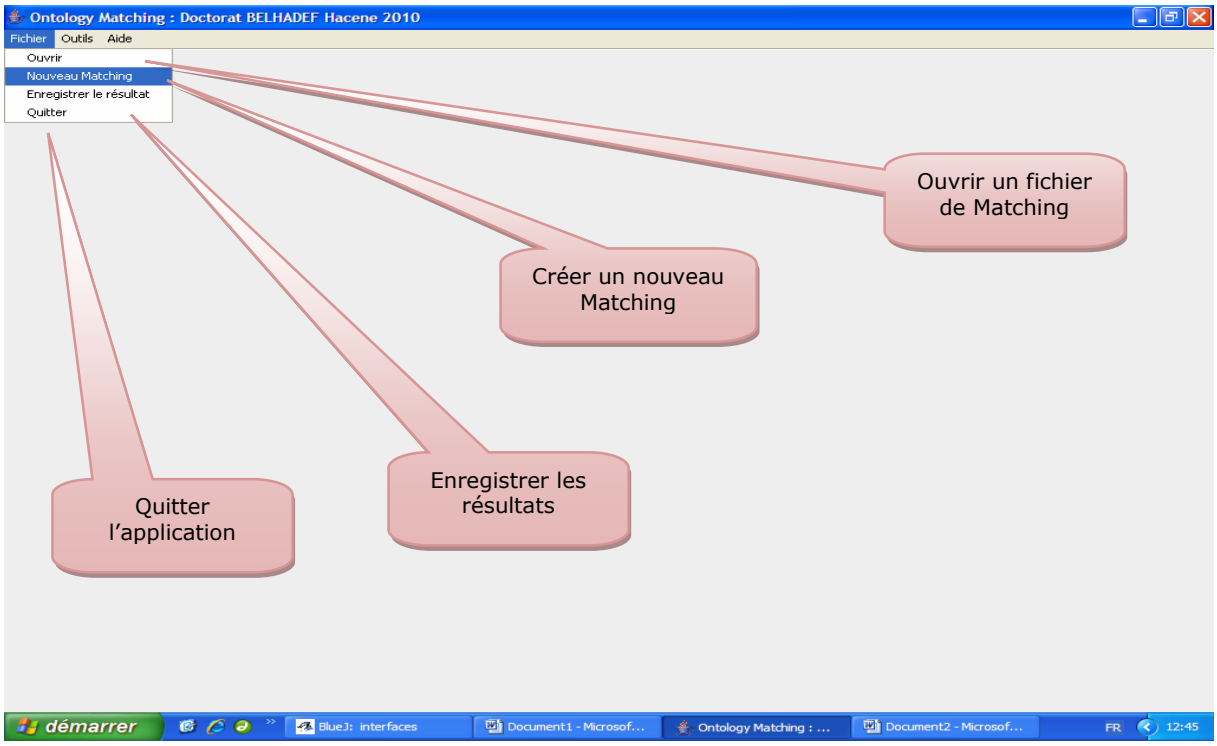


Figure B.8 : Interface de notre application

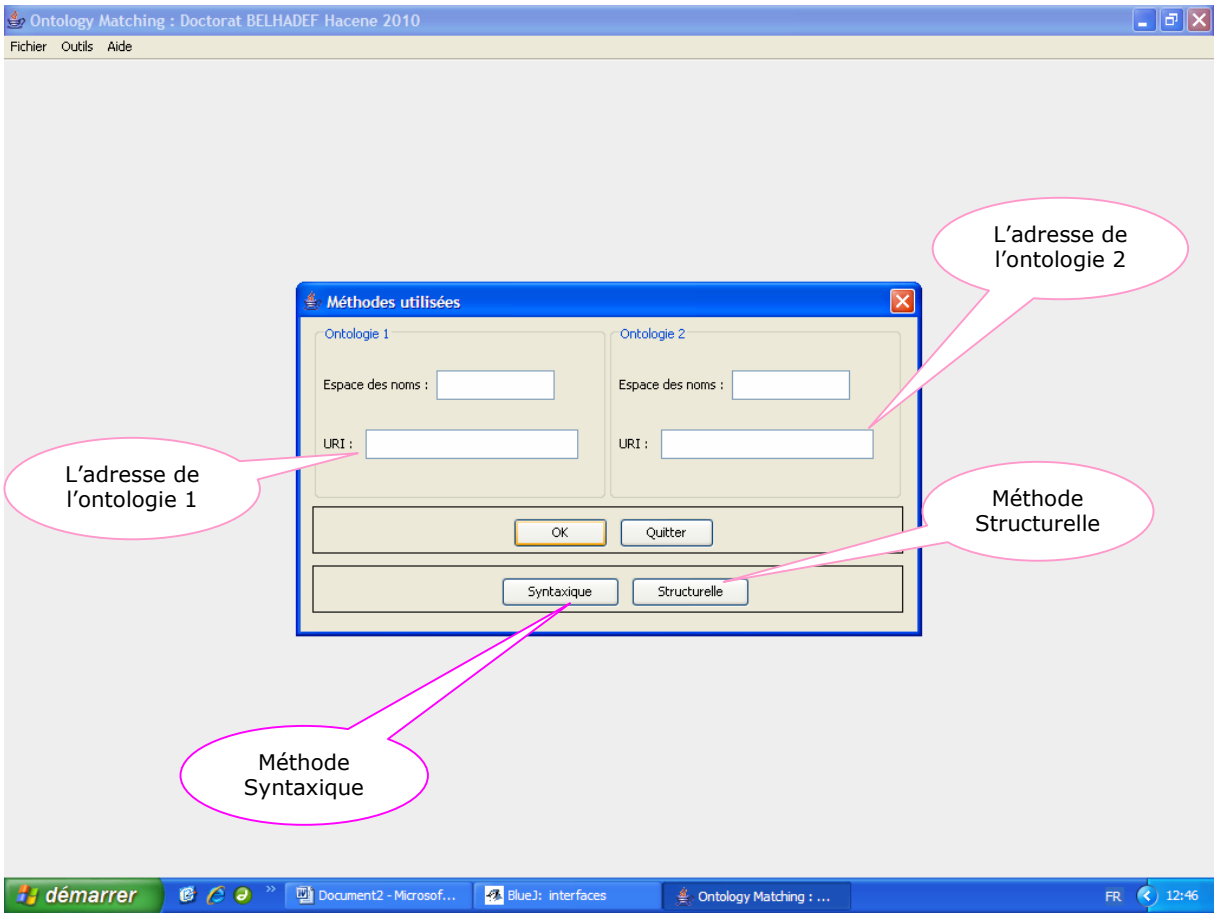


Figure B.9 : Interface d'un nouveau Matching

Annexe

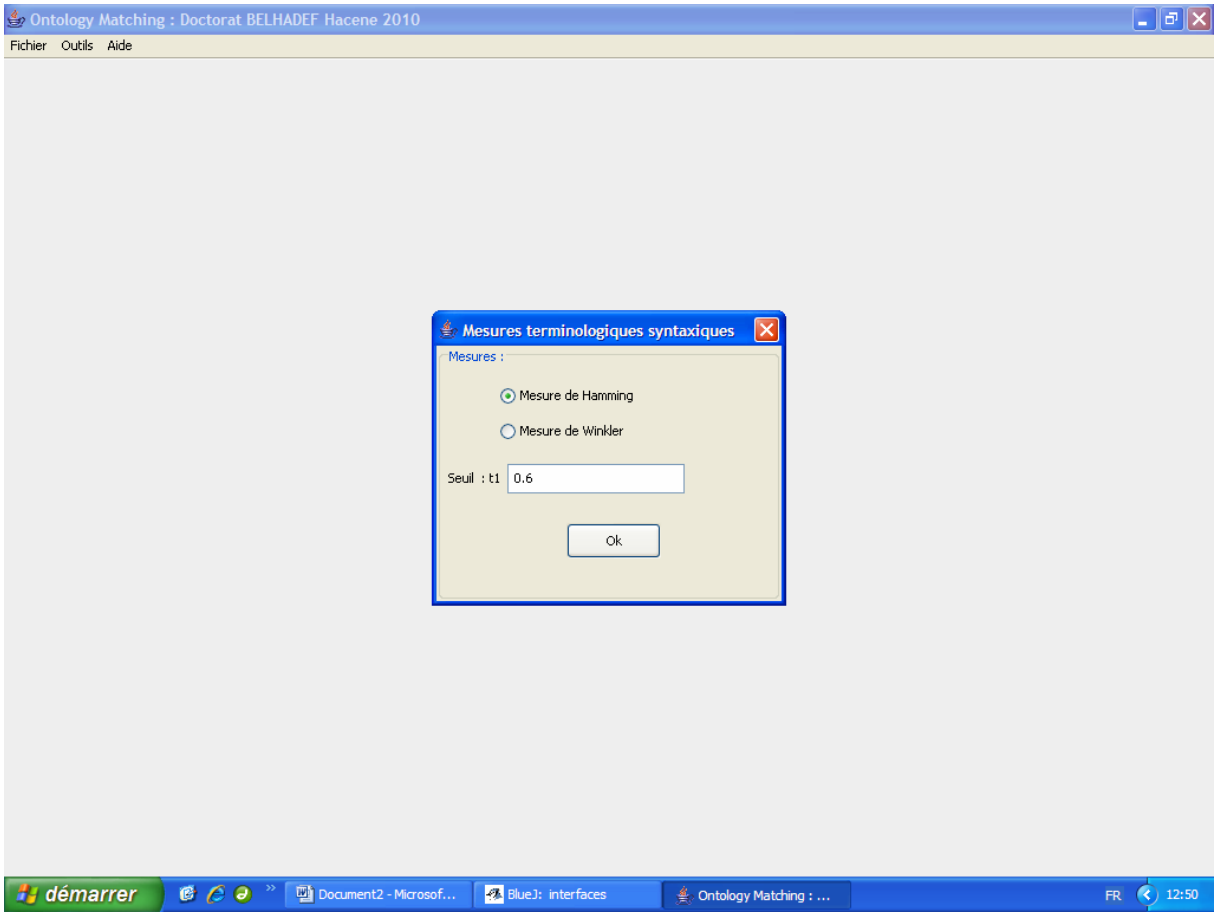


Figure B.10 : Interface des Mesures syntaxiques

Bibliographie et Webographie

Bibliographie

[1] OUDJET Amel, RABIA Asma, « Conception et réalisation d'un Système d'Information Géographique (S.I.G) pour l'aide à la gestion des ressources en eau » Mémoire de fin d'étude INI-Alger , Septembre 2008

[2] Lætitia Perrier Bruslé,Construction d'un SIG et champs d'application : De la mise en forme des données géographiques aux divers champs de la géomatique, Master 1 Géographie U70, université de Nancy

[3] Bakir BOUALEM, Conception et réalisation d'un système d'information géographique pour la promotion du domaine minier Algérien, mémoire de fin d'étude INI, 2006/2007

[4] Frédéric HUBERT, Modèle de Traduction des Besoins d'un Utilisateur pour la Dérivation de Données Géographiques et leur Symbolisation par le Web, Thèse de doctorat, université de CAEN/Basse-Normandie, école doctorale : SIMEM, 07 juillet 2003

[5] Élisabeth HABERT, Qu'est ce qu'un système d'information géographique, Laboratoire de cartographie appliquée - IRD - 2000

[6] Hervé Brunel : Etat de l'art des systèmes d'information géographique, 3 avril 2005

[7] Dangermond Jack, Concepts et spécificités des SIG, 2007

[8] Luis Berardo Borda, Apports des systèmes d'information géographique et l'évaluation de la qualité des eaux côtières. Une contribution à la gestion intégrée de la zone côtière Colombienne, Thèse pour l'obtention du grade de Docteur en Sciences techniques, Ecole Polytechnique Fédérale de Lausanne, 2003

[9] Johann Sorel, Les outils de développement cartographique, 2007

[10] Ahmed Lbath, Marie-Aude Aufaure, Christine Bonhomme. Lvis, Un langage visuel d'interrogation de bases de données spatiales, 2000.

[11] Dominique Hervé Aurélie Toillier, Dynamique d'usage des terres et gestion concertée des ressources naturelles, 2004

[12] Studer, R., Benjamins, V., and Fensel, D., Knowledge engineering: Principles and methods. IEEE Transactions on Data and Knowledge Engineering vol. 25(162), p. 161-197, 1998

[13] R. Gruber, Towards Principles for the Design of Ontologies Used for Knowledge Sharing, In International Workshop on Formal Ontology, Available as technical report KSL-93-04, Padova, Italy, 1993

[14] R. Gruber,: A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition vol. 5(2), pp. 199-220, 1993

[15] Miller, G., Beckwith, R., Fellbaum, C., Gross D., Miller, K.: Introduction to WordNet, an on-line lexical database, International Journal of Lexicography vol. 3 (4), 1990, Revised August 1993,
URL: ftp://ftp.cogsci.princeton.edu/pub/wordnet/5papers.ps

[16] Guarino, N., Masolo, C., Vetere, G.: OntoSeek, Content-Based Access to the Web, IEEE Intelligent Systems vol. 14(3), 70-80, 1999

[17] Tony DUJARDIN, De l'apport des ontologies pour la conception de systèmes multi-agents ouverts, mémoire de Master Recherche Informatique, Laboratoire d'Informatique Fondamentale de Lille (UMR USTL/CNRS 8022), Université des Sciences et Technologies de Lille, 2006

[18] Gómez-Pérez A. Ontological Engineering: A state of the art, Expert Update, 2(3), 33-43. 1999

[19] Quillian, M. R., Semantic memory. In M. Minsky (ed.), Semantic Information Processing, chapter 4. The MIT Press, Cambridge MA, pages 227–270. 1968

[20] J. Charlet_ & P. Laublety, Les ontologies en ingénierie des connaissances, Journée GRACQ, 07 juin 1999

[21] Padadimitriou, C. H., Computational complexity, Addison-Wesley Publishing Company, Massachusetts, É.-U, 1994

[22] F. Baader, D. Calvanese, D. L. McGuiness, D. Nardi, P. F. Patel-Schneider, The Description Logic Handbook: Theory, Implementation, Applications, Cambridge University Press, Cambridge, UK, (ISBN 0-52178-176-0), 2003

[23] Guarino, N., Formal Ontology and Information Systems, IOS Press, 1998.

[24] Hernandez, N., Ontologies de domaine pour la modélisation du contexte en recherche d'information, Thèse de doctorat, Université de Toulouse, 2005.

[25] Mizoguchi R., Kozaki K., Sano T. et Kitamura Y. , Construction and Deployment of a Plant Ontology, The 12th International Conference, EKAW2000, (Lecture Notes in Artificial Intelligence 1937), 113-128, 2000

[26] F. Fürst, L'ingénierie Ontologique (02-07). Nantes: Institut de Recherche en Informatique de Nantes, 2002

[27] N. Guarino, Understanding, building and using ontologies. International J. Human-Computer Studies, 46, 293-310, 1997

[28] Gómez-Pérez, Asunción; Mariano Fernández-Lopez; & Oscar Corcho, Theoretical Foundations of Ontologies, Chapter 1 of Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer-Verlag, pp. 1–45, 2ème edition 2004

[29] Audrey Baneyx, Construire une ontologie de la pneumologie aspects théoriques, modèles et expérimentations, Thèse de doctorat de l'université paris 6 présentée et soutenue publiquement le 06 février 2007

[30] Gilles Kassel, Veille technologique Ingénierie Ontologique, Concepts, méthodes et outils

[31] A. GOMEZ-PEREZ, M. FERNANDEZ LOPEZ, O. CORCHO, Ontological Engineering, Springer, London, 2004

[32] Abdel Kader KEITA, Conception coopérative d'ontologies pré-consensuelles : Application au domaine de l'urbanisme, Thèse doctorat, INSA-Lyon, 06 Juin 2007

[33] Natalya F. Noy et Deborah L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology. Technical Report KSL-01-05Stanford: Knowledge Systems Laboratory, mars 2001

[34] N. Ouafek, H. Belhadef, M. K. Kholladi, A new methodology for the development of computer ontologies, IACe-T'2008 International Arab Conference of e-Technology, Arab Open University, pp 151-156., Amman-Jordan, 2008

[35] DIENG R., CORBY O., GANDON F., GIBOIN A., GOLEBIOWSKA J., MATTA N. & RIBIÈRE M., Méthodes et outils pour la gestion des connaissances : une approche pluridisciplinaire du knowledge management, Dunod, 2 édition 2001

[36] F. GANDON, Ontologies informatiques. Interstices, Journal en ligne de l'INRIA, 2006 Disponible à http://interstices.info/display.jsp?id=c_17672.

[37] Baneyx Audrey, Charlet Jean. Évaluation, évolution et maintenance d'une ontologie en médecine : état des lieux et expérimentations. Revue I3 – Information, Interaction, Intelligence, numéro spécial « Corpus et ontologies », pages 147-173, 2007

[38] HAKIMPOUR F., TIMPF S., Using Ontologies for and Resolution of Semantic Heterogeneity in GIS, 4 th AGILE Conference on Geographical Information Science, 2001

[39] Nadine Cullot, Christine Parent, Stefano Spaccapietra et Christelle Vangenot, Des SIG aux ontologies géographiques, Revue internationale de géomatique. Volume 0-n°0/2003.

[40] T.B. LEE, The semantic Web lifts off, ERCIM News No.51, Octobre 2002

[41] Catherine Roussey, Myoung-Ah Kang, Les Ontologies et leurs Applications en Agriculture, Atelier « Systèmes d'Information et de Décision pour l'Environnement » Congrès Inforsid - 27 mai 2008

[42] Xavier Lacot. Introduction à owl, un langage xml d'ontologies web, Aout 2005

[43] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau et John Cowan. Extensible Markup Language (XML) 1.1. http://www.w3.org/TR/2004/RECxml11-20040204/ (en ligne au 16 juin 2005). Traduction française : Le langage de balisage extensible (XML) 1.1, http://www.yoyodesign.org/doc/w3c/xml11/.

[44] Peter Patel-Schneider Tim Berners-Lee Dan Brickley Dan Connolly Mike Dean Stefan Decker Dieter Fensel Richard Fikes Pat Hayes Jeff Heflin Jim Hendler Ora Lassila Deb McGuinness Lynn Andrea Stein Ian Horrocks, Frank van Harmelen, Specification DAML+OIL, 2001
URL: http://www.daml.org/2001/03/daml+oil-index.html

[45] Mike Dean, Guus Schreiber, Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider et Lynn Andrea Stein. OWL

Web Ontology Language - Reference. http://www.w3.org/TR/2004/REC-owl-ref-20040210/ (en ligne au 16 juin 2005). Traduction française : La référence du langage d'ontologie Web OWL, http://www.yoyodesign.org/doc/w3c/owl-ref-20040210/.

[46] Fournier-Viger, Philippe, Un modèle de représentation des connaissances à trois niveaux de sémantique pour les systèmes tutoriels intelligents, Mémoire de maîtrise (M.Sc.), Université de Sherbrooke, Sherbrooke, Canada, 2005

[47] Jean-François Baget et Yannic Tognetti, Complexité des Logiques de Description, séminaire LIRMMM, Feb. 1998

[48] A. Napoli, Une introduction aux logiques de descriptions, Rapport de Recherche RR-3314, INRIA, 1997

[49] Russell, S. J. et Norvig, P., Articial Intelligence : A Modern Approach, 2ième édition. Prentice Hall, 2002

[50] Étienne André, Construction et utilisation des logiques dans les systèmes d'information, Étude bibliographique, 1er février 2007

[51] Quillian, M. R., Semantic memory. In M. Minsky (ed.), Semantic Information Processing, chapter 4. The MIT Press, Cambridge MA, pages 227–270, 1968

[52] M.L. Minsky. A Framework for Representing Knowledge, Report A.I MEMO 306, Massachusetts Institute of Technology, A.I. Lab., Cambridge, Massachusetts, juin 1974.

[53] Serge Haddad, Chapitre II:LOGIQUES DE DESCRIPTION, Cours de logique pour l'IA, Université Paris-Dauphine : DEA 127, version du 6 novembre 2002

[54] Anne Parrain, Cours, Logiques de Description, l'Université d'Artois Année 2008-09

[55] Francis LAPIQUE, Langages de requêtes pour base de connaissances ,19 décembre 2006

[56] Clark & Parsia. Pellet. http ://pellet.owldl.com/. Présentation et téléchargement du logiciel Pellet, en anglais

[57] Philippe Fournier-Viger. Un modèle de représentation des connaissances à trois niveaux de sémantique pour les systèmes tutoriels intelligents. Master's thesis, Université de Sherbrooke, Sherbrooke, Canada., 2005

[58] Arnaud Zinflou, Système interactif d'aide à la décision basé sur des algorithmes génétiques pour l'optimisation multi-objectifs, Mémoire présenté à l'université du Québec à Chicoutimi comme exigence partielle de la maîtrise en informatique, 29 juin 2004

[59] THOMPSON W.A., WEETMAN G.F, Decision support systems for silviculture planning in Canada. For. Chron. 71, 291-298. 1995

[60] Wiederhold G.: Mediators in the Architecture of Future Information Systems, IEEE Computer 25(3): 38-49, 1992

[61] R. Mizoguchi, A Step Towards Ontological Engineering, Paper presented at the 12th National Conference on AI of JSAI, June 1998

[62] B. Bachimont, Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances. In Z. M. Charlet J., Kassel G., Bourgault D., (Ed.), Ingénierie des connaissances. Évolution Récentes et nouveaux défis Paris: Eyrolles, 305-323, 2000

[63] DIEMER Arnaud, ECONOMIE D'ENTREPRISE, Chapitre 4 : L'entreprise, un centre de décisions, IUFM d'Auvergne CAPET - PLP

[64] H.A. Simon, Administration et processus de décision, Economica, (Chap. IV, p.62)

[65] A. Gachet, P. Haettenschwiler, A Jini-based software framework for developing distributed cooperative decision support systems, Soft. Pract. Exper. 33 221–258 , 2003

[66] Jean-Paul LECLERCQ, Systèmes d'aide à la décision, Cours de Master – Option S.I. Faculté d'Informatique université Notre Dame de la Paix Année académique 2007-2008

[67] Pascale Zaraté, Des Systèmes Interactifs d'Aide à la Décision Aux Systèmes Coopératifs d'Aide à la Décision : Contributions conceptuelles et fonctionnelles, Habilitation à Diriger des Recherches Spécialité Informatique, INPT le 7 Décembre 2005

[68] M. Uschold et M. King, Towards a Methodology for Building Ontologies. Paper presented at the Workshop on Basic Ontological Issues in Knowledge Sharing, 1995

[69] RAMI DIB RIFAIEH, Utilisation des ontologies contextuelles pour le partage sémantique entre les systèmes d'information dans l'entreprise , Thèse doctorat, INSA-Lyon, Soutenue le 20 Décembre 2004

[70] HACENE BELHADEF, MOHAMMED KHIREDDINE KHOLLADI, Urban ontology-based geographical information system ,JATIT, Journal of Theoretical and Applied Information Technology, pages 139-154 , Vol 9. No. 2, 2009

[71] Belhadef hacene, Kholladi M. khireddine Kholladi, Conception of a new ontology for the interoperability of the geographical information system, SETIT 2007, 4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, Hammamet 25-29 March 2007-Tunisia 2007

[72] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, Design Patterns Elements of Reusable Object-Oriented Software, Addison Wesley Professional Computing Series, MA, p.395, , USA, 1994

[73] Yue Ma, Laurent Audibert, Adeline Nazarenko, Ontologies étendues pour l'annotation sémantique, pp 205-216, Actes d'IC 2009

[74] David Sheeren, Méthodologie d'évaluation de la cohérence inter-représentations pour l'intégration de bases de données spatiales, Thèse doctorat de l'université paris 6, 20 mai 2005

[75] Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G, Neumann H and Hübner S., Ontology-based integration of information - A survey of existing approaches, In Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01), Workshop: Ontologies and Information Sharing, 2001

[76] Stoimenov L. and Đorđević-Kajan S. , Framework for Semantic GIS Interoperability, FACTA Universitatis : Series Mathematics and Informatics, 17, pp. 107-125, 2002

[77] Benoit Lavoie, Synthèse de lectures, Notion d'ontologie et construction d'ontologie à partir de corpus de textes, Université du Québec à Montréal, 8 février 2007

[78] BRISSON Laurent Projet EXECO, Mesures d'intérêt subjectif et représentation des connaissances, Rapport de recherche ISRN I3S/RR–2004-35–FR, Octobre2004

[79] NYULAS C. O'CONNOR, M. TU, S, DataMaster – a Plug-in for Importing Schemas and Data from Relational Databases into Protégé, 2007
URL: http://protege.stanford.edu/conference/2007/presentations/10.01_Nyulas.pdf

[80] Sonia Krivine, Jérôme Nobécourt, Lina Soualmia, Farid Cerbah, Catherine Duclos, Construction automatique d'ontologie à partir de bases de données relationnelles: application au médicament dans le domaine de la pharmacovigilance, 20ème journée francophones de l'ingénierie des connaissances IC2009

[81] F, CERBAH, Learning Highly Structured Semantic Repositories from Relational Databases - RDBtoOnto Tool, in Proceedings of the 5th European Semantic Web Conference (ESWC 2008), Tenerife, Spain, June, 2008

[82] F. CERBAH, Mining the Content of Relational Databases to Learn Ontologies with deeper Taxonomies, Proceeding of IEEE/WIC/ACM International Joint Conference on Web Intelligence (WI'08) and Intelligent Agent Technology (IAT'08)pp 9-12., Sydney, Australia, 2008

[83] H.Belhadef N.Ouafek and M.K.Kholladi, A set of mapping rules E/R and relational schema database towards an ontology, ICADIWT 2008 The First IEEE International Conference on the Applications of Digital Information and Web Technologies, pp 289 – 295, VSB- Technical University of Ostrava, Czech Republic, August 4-6, 2008

[84] Z. Xu, X. Cao, Y. Dong, and W. Su, Formal Approach and Automated Tool for Translating ER Schemata into OWL Ontologies , Advances in Knowledge Discovery and Data Mining,Springer-Verlag Berlin Heidelberg, pp. 464–475, 2004

[85] Sana Sellami, Méthodologie de Matching à Large Echelle pour des schémas XML, Thèse doctorat en informatique, INSA-Lyon, 30 Novembre 2009

[86] J. Euzenat, P. Shvaiko: Ontology Matching. Springer, Approx. 340 p., 67 illus., Hardcover, ISBN 978-3-540-49611-3 **,** 2007

[87] M. Ehrig, Y. Sure, Ontology Mapping - An Integrated Approach », Proceedings of the 1st European Semantic Web Symposium, vol. 3053, Springer Verlag, Hersounious, p. 76-91, 2004

[88] Sami Zghal, Sadok Ben Yahia, Engelbert Mephu Nguifo, Yahya Slimani, SODA : Une approche structurelle pour l'alignement d'ontologies OWL-DL, 1ères Journées Francophones sur les ontologies, 18-20 Octobre 2007

[89] Jérôme Euzenat, Quelques pistes pour une distance entre ontologies, 8èmes Journées Francophones Extraction et Gestion des Connaissances, Sophia Antipolis, 29 janvier 2008

[90] P. Shvaiko, J. Euzenat: A Survey of Schema-based Matching Approaches. Journal on Data Semantics (JoDS), IV, LNCS 3730, pp. 146-171, 2005

[91] J. Euzenat, T. L. Bach, J. Barrasa, P. Bouquet, J. D. Bo, R. Dieng-Kuntz, M. Ehrig, M. Hauswirth, M. Jarrar,R. Lara, D. Maynard, A. Napoli, G. Stamou, H. Stuckenschmidt, P. Shvaiko, S. Tessaris, S. V. Acker et I. Zaihrayeu, State of the art on ontology alignment, Deliverable D2.2.3, Knowledge Web NoE, 2004.

[92] Jérôme David, AROMA:une méthode pour la découverte d'alignements orientés entre ontologies à partir de règles d'association,Thèse doctorat, Ecole Polytechnique de l'Université de Nantes, 8 novembre 2007

[93] P. Jaccard, Etude comparative de la distribution orale dans une portion des Alpes et du Jura, Bulletin de la Société Vaudoise des Sciences Naturelles 37, p. 547-579.

[94] W. E. Winkler, The state of record linkage and current research problems, Tech. report, Statistical Research Division, U.S. Bureau of the Census, 1999.

[95] Abdeltif ELBYED, ROMIE : une approche d'alignement d'ontologies à base d'instances, Thèse doctorat de l'institut national des télécommunications de l'université Evry-Val d'Essonne, , 16 Octobre 2009

Webographie

[web 1] http://www.esrifrance.fr : Site officiel des produits ESRI France

[web 2] http://seig.ensg.ign.fr/: Serveur Educatif sur l'Information Géographique

[web 3] http://sig.net.free.fr/ : Introduction aux S.I.G : http://sig.net.free.fr/

[web 4] http://geoconcept.com : Site officiel de Géoconcept

[web 5] http://www.star.be/ : Groupe STAR-APIC Editeur de logiciels dans le domaine des SIG et des technologies géospatiales.

[web 6] http://www.e-Carte.com : Revue de cartographie électronique

[web 7] http://wordnet.princeton.edu/ : Wordnet

[web 8] http://www.aiai.ed.ac.uk/project/enterprise : Enterprise Ontology :

[web 9] http://www.eil.utoronto.ca/tove/ontoTOC.html : TOVE

[web 10] http://estime.spim.jussieu.fr/Menelas/Ontologie/ : MENELAS

[web 11] http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/ : Recommendation W3C XML Schema, 2004

[web 12] http://www.w3.org/TR/rdf-schema/ : Recommendation W3C RDF Schema, 2004

[web 13] http://www.w3.org/TR/owl-ref/ : Recommendation W3C OWL, 2004

[web 14] http://fr.wikipedia.org/ le site Wikipédia en français

[web 15] http://www.mindswap.org/2003/pellet/

[web 16] http://www.racer-systems.com/

[web 17] http://www.cs.man.ac.uk/~horrocks/FaCT/

[web 18] http://dig.sourceforge.net/

[web 19] http://info4.b0nda.com/files/se/decision.pdf : Cours Systèmes d'aide à la décision

[web 20] http:// wapedia.mobi/fr/ : Encyclopédie pour téléphone portable

[web 21] http://sceco.u-strasbg.fr/doc/doc/cours_en_ligne/Netzer/
Entreprise_et_gestion_4.doc : Chapitre 4 : Décision dans les organisations

[web 22] http://geronim.free.fr/ecoent/cours/decision.htm : La prise de décision dans l'entreprise

[web 23] http://www.journaldunet.com : le journal du Net Actualité économique et high-tech

[web 24] http://www2.epfl.ch/webdav/site/coaching/users/156408/public/Anciens_tests/
SIG/91_chapitre%209_1erePartie_Ln.doc : Chapitre 9, Spatial Decision Support Systems