



HAL
open science

Approche ontologique pour la modélisation et le raisonnement sur les trajectoires : prise en compte des aspects thématiques, temporels et spatiaux

Wafa Mefteh

► **To cite this version:**

Wafa Mefteh. Approche ontologique pour la modélisation et le raisonnement sur les trajectoires : prise en compte des aspects thématiques, temporels et spatiaux. Ordinateur et société [cs.CY]. Université de La Rochelle, 2013. Français. NNT : 2013LAROS405 . tel-01124314

HAL Id: tel-01124314

<https://theses.hal.science/tel-01124314v1>

Submitted on 6 Mar 2015

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

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

THÈSE

pour l'obtention du

Doctorat de l'université de La Rochelle
spécialité : Informatique et Applications

20 Septembre 2013

par

Wafa Mefteh

Approche ontologique pour la modélisation et le raisonnement sur les trajectoires. Prise en compte des aspects thématiques, temporels et spatiaux

- Rapporteurs :* GENSEL Jérôme, Professeur
Université Pierre Mendès (Grenoble 2)
BELLATRECHE Ladjel, Professeur
École Nationale Supérieure de Mécanique et d'Aéronautique (ENSMA)
- Examineurs :* CLARAMUNT Christophe, Professeur
Institut de Recherche de l'École navale (IRENav)
- Invités :* LAFAYE Jean-Yves, Professeur
Université de La Rochelle
- Encadreurs* BOUJU Alain, Habilité à Diriger des Recherches
Université de La Rochelle
MALKI Jamal, Maître de conférences
Université de La Rochelle

Remerciements

Je tiens à adresser mes plus chaleureux remerciements à Alain BOUJU, HDR à l'université de La Rochelle, à la fois directeur du département Informatique et mon directeur de thèse, pour m'avoir accueilli au sein du laboratoire L3i et pour m'avoir apporté ses précieuses lumières et ses conseils tout au long de son encadrement de cette thèse. Je le remercie d'avoir eu la gentillesse de me transmettre un peu de son savoir-faire et de son savoir-être. J'ai pu apprécier sa dimension professionnelle, mais aussi sa dimension humaine. J'en profite pour lui exprimer ici ma plus profonde gratitude.

Je remercie très sincèrement Jamal MALKI, Maître de Conférences à l'université de La Rochelle, mon co-encadrant, pour toute sa coopération et toutes nos discussions très enrichissantes qui m'ont guidées tout au long de ce parcours. Je le remercie pour son enthousiasme et la confiance qu'il m'a accordée et qui m'ont chaque fois remotivés. Merci de m'avoir aidé à prendre du recul sur mon travail. Merci pour son soutien continu et pour son aide.

Je tiens également à remercier Ladjel BELLATRECHE et Jérôme GENSEL, pour m'avoir fait l'honneur d'être rapporteurs de cette thèse. Je leur suis très reconnaissante d'avoir accepté cette lourde tâche. Je remercie également, Christophe CLARAMUNT d'avoir accepté d'examiner mon travail ainsi que Jean-Yves LAFAYE qui a accepté de participer à mon jury de thèse.

Je souhaite remercier très chaleureusement tous les membres du laboratoire L3i pour tous les bons moments passés en leur compagnie.

Bien évidemment, je remercie mes chers parents, mes chères sœurs, ma famille et ma belle famille, qui ont toujours cru en moi et qui m'ont toujours soutenue. Malgré la distance qui nous a séparé pendant ces dernières années, vous avez su être toujours près de moi pour m'encourager dans les moments difficiles et pour fêter mes réussites.

Enfin, je souhaite bien évidemment remercier mon extraordinaire mari Anouar qui m'épaula maintenant depuis 6 ans et sans qui rien n'aurait été possible. Je remercie mes deux anges Yassine et Youssef, qui ont vécu avec moi tous les moments passés de cette thèse.

Mes amours, Anouar, Yassine et Youssef, cette thèse vous est dédiée.

Résumé

L'évolution des systèmes de capture des données concernant les objets mobiles a donné naissance à de nouvelles générations d'applications dans différents domaines. Les données capturées, communément appelées « trajectoires », sont au cœur des applications qui analysent et supervisent le trafic routier, maritime et aérien ou également celles qui optimisent le transport public. Elles sont aussi exploitées dans les domaines du jeu vidéo, le cinéma, le sport et le domaine de biologie animale pour l'étude des comportements, par les systèmes de capture des mouvements. Aujourd'hui, les données produites par ces capteurs sont des données brutes à caractère spatio-temporel qui cachent des informations sémantiquement riches et importantes pour un expert. L'objectif de cette thèse est d'associer automatiquement aux données spatio-temporelles des descriptions ou des concepts liés au comportement des objets mobiles, interprétables par les humains, mais surtout par les machines.

Partant de ce constat, nous proposons un processus partant de l'expérience de gestion de données d'objets mobiles du monde réel notamment les navires et les avions, vers un modèle ontologique générique pour la trajectoire. Nous présentons quelques requêtes qui intéressent les experts du domaine et qui montrent l'impossibilité d'exploiter les trajectoires dans leurs états bruts. En effet, l'analyse de ces requêtes fait ressortir trois types de composantes sémantiques : thématique, spatiale et temporelle. Ces composantes doivent être rattachées aux données des trajectoires ce qui conduit à introduire un processus d'annotation qui transforme les trajectoires brutes en trajectoires sémantiques. Pour exploiter les trajectoires sémantiques, on construit une ontologie de haut niveau pour le domaine de la trajectoire qui modélise les données brutes et leurs annotations. Vu le besoin d'un raisonnement complet avec des concepts et des opérateurs spatiaux et temporels, nous proposons une solution de réutilisation des ontologies de temps et de l'espace.

Dans cette thèse, nous présentons aussi notre travail issu d'une collaboration avec une équipe de recherche qui s'intéresse à l'analyse et la compréhension des comportements des mammifères marins dans leur milieu naturel. Nous détaillons le processus utilisé dans les deux premiers domaines, qui part des données brutes représentant les déplacements des phoques jusqu'au modèle ontologique de trajectoire des phoques. Nous accordons une attention particulière à l'apport de l'ontologie de haut niveau définissant un cadre contextuelle pour l'ontologie du domaine d'application.

Enfin, cette thèse présente la difficulté de mise en œuvre sur des données de taille réelles (des centaines de milliers d'individus) lors du raisonnement à travers les mécanismes d'inférence utilisant des règles métiers.

Abstract

The evolution of systems capture data on moving objects has given birth to new generations of applications in various fields. Captured data, commonly called "trajectories" are at the heart of applications that analyze and monitor traffic, maritime and air or also those that optimize public transport. They are also used in the video game, film, sports and field biology to study animal behavior, by motion capture systems. Today, the data produced by these sensors are raw spatio-temporal character hiding semantically rich and meaningful information to an expert data. So, the objective of this thesis is to automatically associate the spatio-temporal data descriptions or concepts related to the behavior of moving objects, interpreted by humans, but also by machines.

Based on this observation, we propose a process based on the experience of real-world moving objects including vessel and plane to an ontological model for the generic path. we present some applications of interest to experts in the field and show the inability to use the paths in their raw state. Indeed, the analysis of these queries identified three types of semantic components : thematic, spatial and temporal. These components must be attached to data paths leading to enter an annotation that transforms raw semantic paths process trajectories. To exploit the semantic trajectories, we construct a high-level ontology for the domain of the path which models the raw data and their annotations. Sight of the need to complete reasoning with concepts and spatial and temporal operators, we propose the solution for reuse of ontologies time space.

In this thesis, we also present our work comes from a collaboration with a research team that focuses on the analysis and understanding of the behavior of marine mammals in their natural environment. We describe the process used in the first two areas, which share raw data representing the movement of seals to ontological trajectory model seals. We pay particular attention to the contribution of the upper ontology defined in a contextual framework for ontology application.

Finally, this thesis presents the difficulty of implementation on real data size (hundreds of thousands) when reasoning through inference mechanisms using business rules.

Table des matières

Introduction	9
1 Contexte	10
2 Description du problème	11
3 Contributions	12
4 Plan de la thèse	13
5 Cadre du travail	15

I Modèles de données et modèles ontologiques pour les trajectoires	17
---	-----------

Chapitre 1	
État de l’art et positionnement scientifique	19
1.1 Introduction	20
1.2 Modèles de données spatiales	20
1.2.1 Introduction	20
1.2.2 Généralités sur les modèle de données spatiales de l’OGC	20
1.2.3 Modèle de données spatiales : GML	21
1.2.4 Modèle de données spatiales : KML	22
1.2.5 Modèle de données spatiales pour le SQL : OGC SFS	24
1.3 Modèles de données temporelles	26
1.3.1 Introduction	26

1.3.2	Modèle temporel dans la norme ISO 19108	26
1.3.3	Algèbre des intervalles temporels d'Allen	27
1.3.4	Modèle temporel dans les bases de données	28
1.4	Modèles de données spatio-temporelles	30
1.4.1	Introduction	30
1.4.2	Modélisation spatio-temporelle avec MADS	30
1.4.3	Modélisation spatio-temporelle avec Perceptory	33
1.4.4	Modélisation spatio-temporelle avec AROM-ST	34
1.5	Modèles de données spatio-temporelles	36
1.5.1	Introduction	36
1.5.2	Modélisation spatio-temporelle avec MADS	36
1.5.3	Modélisation spatio-temporelle avec Perceptory	39
1.5.4	Modélisation spatio-temporelle avec AROM-ST	40
1.6	Synthèse	42

Chapitre 2

Les modèles de données des trajectoires

45

2.1	Introduction	45
2.2	Les trajectoires en sciences physiques et mathématiques	46
2.3	Les trajectoires en bases de données	47
2.3.1	Modèle MOST (Moving Objet Spatio-Temporel)	47
2.3.2	Modèle basé sur les types abstraits spatio-temporels	48
2.3.3	Modèle conceptuel de la trajectoire	49
2.4	Synthèse	51

Chapitre 3

Approche ontologique pour les données spatiales, temporelles et spatio-temporelles

53

3.1	Introduction	54
-----	------------------------	----

3.2	Les ontologies traditionnelles	55
3.2.1	Les différents types d'ontologies	55
3.2.1.0.1	typologie basée la structure de la connaissance	55
3.2.1.0.2	typologie basée le contenu de la connaissance	56
3.2.2	La réutilisabilité des ontologies	56
3.2.3	Les langages de représentation et d'interrogation	57
3.2.3.0.2.1	Les logiques de description	58
3.2.3.0.3	SPARQL : langage d'interrogation	59
3.2.4	Les environnements de construction	60
3.2.5	Le peuplement de l'ontologie	61
3.2.5.1	Les outils	62
3.2.5.1.1	DataMaster	63
3.2.5.2	Les applications	63
3.2.6	Les outils de raisonnement	64
3.3	Approche ontologique : cas des données spatiales	65
3.3.1	Ontologies spatiales : extension de la logique de description . .	65
3.3.2	Ontologies spatiales	66
3.3.3	Raisonnement spatial dans les ontologies	67
3.3.3.1	Architecture hybride pour le raisonnement spatial . . .	67
3.3.3.2	Système de raisonnement PelletSpatial	70
3.4	Approche ontologique : cas des données temporelles	70
3.4.1	Ontologies temporelles : extension de la logique de description .	71
3.4.2	Ontologies temporelles	71
3.4.3	Raisonnement temporel dans les ontologies	73
3.5	Approche ontologique : cas des données spatio-temporelles	74
3.5.1	Cas des données géographiques	74
3.5.2	Cas des bases de données	75
3.6	Synthèse	76

Chapitre 4**Approche ontologique pour les données des trajectoires 79**

4.1	Introduction	80
4.2	Ontologie des objets mobiles	80
4.3	Ontologie modulaire de la trajectoire	82
4.4	Ontologie de trajectoire : espace, temps et thème	83
4.5	Synthèse	84

II Contributions 87**Chapitre 1****De l'expérience au modèle ontologique de la trajectoire 93**

1.1	De l'expérience au modèle conceptuel	93
1.1.1	Introduction	93
1.1.2	Expérience 1 : Étude des trajectoires des navires	94
1.1.2.1	Contexte de l'étude	94
1.1.2.2	Capture des trames	95
1.1.2.3	Analyse des trames	96
1.1.2.4	Modèle UML de la trajectoire	97
1.1.3	Expérience 2 : Étude des trajectoires des avions	99
1.1.3.1	Contexte de l'étude	99
1.1.3.2	Capture des trames	99
1.1.3.3	Analyse des trames	100
1.1.3.4	Modèle UML de la trajectoire	101
1.2	Modèle générique de la trajectoire	102
1.2.1	Objectif	102
1.2.2	Domaine de la trajectoire	102
1.2.3	Domaine de l'objet mobile et de la trajectoire	103

1.3	Modèle de la trajectoire sémantique	104
1.3.1	Annotation	104
1.3.2	Trajectoire sémantique : trajectoire annotée	104
1.4	Modèle ontologique de la trajectoire	105
1.5	Synthèse	107

Chapitre 2

Réutilisation des ontologies dans l'ontologie de la trajectoire 109

2.1	Introduction	109
2.2	Réutilisation de l'ontologie temps	110
2.3	Réutilisation de l'ontologie espace	111
2.4	Correspondance entre des ontologies	111
2.5	Synthèse	112

Chapitre 3

Ontologie de la trajectoire : cas des mammifères marins 115

3.1	Expérience 3 : Étude des trajectoires des mammifères marins	115
3.1.1	Capture des trames	116
3.1.2	Analyse et compréhension des trames	117
3.1.3	Modèle UML de la trajectoire	118
3.1.4	Modèle de la trajectoire sémantique	120
3.2	Intégration du domaine dans l'ontologie de la trajectoire	123

Chapitre 4

Approche bases de données pour la gestion des données ontologiques RDF stores

4.1	Introduction	125
4.2	Stockage des données ontologiques	126
4.2.1	Schéma de stockage	126
4.2.2	Stockage des IRI	127

4.2.3	Compression des préfixes	128
4.2.4	Unité de stockage, propriété et contrôle d'accès	128
4.2.5	Index de données ontologiques	129
4.2.6	Stockage RDF distribué et/ou fédéré	129
4.3	Chargement des données ontologique	129
4.3.1	Chargement incrémental et chargement en masse	129
4.3.2	Effet sur l'architecture de stockage	130
4.3.3	Ajout de données en masse	130
4.3.4	Réutilisation des nœuds anonymes	130
4.4	Inférence	131
4.4.1	Systèmes d'implications standards	131
4.4.2	Interfaçage avec des moteurs de raisonnement tiers	131
4.4.3	Règles définies par l'utilisateur	131
4.4.4	Chaînage avant et chaînage arrière	131
4.5	Interrogation RDF et les données relationnelles	132
4.5.1	Langage d'interrogation	132
4.5.2	Performance des requêtes	132
4.5.3	Optimisation des requêtes	133
4.5.4	Interrogation des données relationnelles dans un contexte RDF	133
4.6	Étude comparative de quelques BDBO	133
4.6.1	Oracle Semantic Data Store	133
4.6.2	Virtuoso Universal Server	134
4.6.3	AllegroGraph	134
4.6.4	Semantics Platform	135
4.6.5	OntoDb	136
4.6.6	Récapitulatif	137

Chapitre 5**Mise en œuvre**

5.1	Approche de mise en œuvre des ontologies	139
5.2	Création des parties déclaratives des ontologies	140
5.3	Peuplement de l'ontologie owlSealTrajectory	141
5.4	Vérification de la consistance des données ontologiques	143
5.5	Inférence et règles	143
5.5.1	Règles d'inférence du langage	143
5.5.2	Règles d'inférence métiers	144
5.5.3	Règles temporelles	145
5.5.4	Règles spatiales	146
5.6	Expérimentation	147
5.7	Synthèse	149
III	Conclusions et perspectives	151
	Bibliographie	157

Introduction

Sommaire

1	Contexte	10
2	Description du problème	11
3	Contributions	12
4	Plan de la thèse	13
5	Cadre du travail	15

1 Contexte

Actuellement, de nombreux objets mobiles sont équipés de capteurs permettant d'obtenir rapidement et fréquemment de nombreuses données. En particulier, nous disposons des données de localisation. Celles-ci sont en général obtenues par l'utilisation de capteur de type GPS¹. La diffusion de ces données utilise des moyens de communication sans fils disponibles comme la VHF² ou la téléphonie mobile. Elles peuvent donc être analysées et diffusées (par exemple sur internet) de façon dynamique. Parmi les données capturées, nous nous intéressons aux données de localisation, et donc aux problématiques liées aux mouvements ou trajectoires. Pour cela, nous considérons deux types d'objets mobiles utilisant un système de suivi basé sur le GPS. Ces objets sont les bateaux et les avions.

Dans le domaine maritime, la réglementation oblige l'équipement d'un nombre de plus en plus important de navires avec un système dit AIS³. Ce système permet l'échange régulier de données entre les bateaux et les stations terrestres, notamment pour améliorer la sécurité. D'autre part, ce système fournit fréquemment des données sur la localisation actuelle du bateau sous forme de latitude et longitude.

Dans le domaine aérien, les avions sont équipés d'un système implémentant une spécification d'échanges de données dite Mode-S. En France, cette spécification est entrée en vigueur depuis mars 2007. Elle permet l'envoi et la réception de données numériques entre un avion et une station terrestre équipée d'un transpondeur. Le Mode-S diffuse des données générales telles que l'immatriculation de l'avion, son numéro de vol et des données de localisation sous forme de latitude, longitude et altitude.

Dans le domaine des jeux vidéo et du cinéma, on sait capturer les mouvements du corps et les expressions faciales d'une personne réelle pour les transmettre à un personnage animé. Le jeu sérieux dans le domaine du sport, de la médecine et des récents projets pour l'assistance aux personnes âgées à domicile exploite massivement les données capturées. Dans le domaine de la biologie ou l'étude du vivant d'une manière générale, les chercheurs collectent et analysent les données provenant des animaux dans leur milieu naturel, en solitaire ou en population, pour comprendre leurs comportements.

Il existe une forme générale des données capturées partagée par les différents domaines d'application cités dans cette thèse. En effet, la donnée capturée peut s'écrire sous la forme d'un quadruplet : (`id_obj`, `coord_spa`, `t_mesure`, [`données_aux`]). Le premier paramètre `id_obj` désigne l'identifiant de l'objet capturé. Il est important de connaître l'origine d'une donnée lorsqu'on se trouve dans un environnement de capture multi-sources. Le paramètre `coord_spa` désigne les coordonnées spatiales d'un ou plusieurs points de l'objet mobile. Le système de référence peut être en deux dimensions, sous forme de (longitude, latitude) dans le cas de la supervision du trafic routier, ou encore en trois dimensions, sous forme de (x,y,z) dans le cas de la capture de mouvements d'un humain. Le nombre de points formant une discrétisation de l'objet mobile et qui seront capturés dépend essentiellement du domaine d'application. Cela peut être un seul point ou plusieurs points

1. Global Positioning System est le système américain de positionnement par satellites

2. Very High Frequency

3. *Automatic Identification System*

sans hiérarchie comme dans le cas de la console Wii ou alors avec hiérarchie tel le cas de la capture de mouvements d'un humain. Le paramètre `t_mesure` indique l'instant de réalisation de la capture. Les systèmes de référence du temps et de l'espace doivent être connus. Enfin, le dernier paramètre `données_aux` désigne la présence ou non d'autres données auxiliaires souvent dérivées, comme la vitesse de l'objet mobile à un instant, ou des mesures effectuées sur le milieu de réalisation de la capture. Nous désignons cet ensemble des données par `trajectoires`.

De plus, plusieurs défis scientifiques doivent être relevés afin de permettre le développement de systèmes de raisonnement spatio-temporelles pour les trajectoires. Parmi ces défis, on trouve en premier lieu le développement de techniques de représentation de données spatiales et temporelles qui soient compatibles avec les langages ontologiques comme OWL afin d'offrir une interopérabilité maximale, ainsi que l'exploitation automatique de ces données par des moteurs de recherche sémantique notamment Swoogle⁴. Dans un deuxième temps, si plusieurs approches pour la modélisation des données spatiales et temporelles existent à ce jour, il faut cependant souligner l'absence de moteurs d'inférence gérant les relations spatiales et temporelles capables d'exploiter ces données pour déduire de nouvelles connaissances et pour répondre à des requêtes.

2 Description du problème

Les données capturées des objets mobiles sont souvent obtenues sous la forme d'un flot de données. Elles sont échangées en temps réel entre les objets mobiles et une station fixe qui gère ce flux. Elles ont un format précis et documenté.

Ces données possèdent des caractéristiques générales ou des méta-données combinées à des données spatio-temporelles permettant de reconstituer une ou plusieurs trajectoires. La dimension spatiale réside dans le caractère géolocalisé exprimé dans différents systèmes selon le domaine d'application. La dimension temporelle exprime la continuité dans le temps de ces données. Par conséquent, ces données sont spatio-temporelles multidimensionnelles. Les trajectoires des objets mobiles forment un volume important de données dès qu'on souhaite les étudier sur une longue période. On peut donc parler dans ce cas de masses de données spatio-temporelles multidimensionnelles.

Les domaines d'applications centrés sur les trajectoires des objets mobiles, fournies par les capteurs cités précédemment (AIS pour les bateaux, Mode-S pour les avions), sont nombreux. Toutefois, elles partagent un même intérêt lié à la caractérisation et ainsi la compréhension du mouvement et par conséquent, le comportement de l'objet mobile dans son environnement. Parmi les applications qui nous intéressent tout particulièrement, on cite :

- la visualisation facile, compréhensible et multi-vues des trajectoires : elle permet aux experts du domaine d'interpréter et de donner un sens (souvent sémantique basée sur des annotations) aux mouvements des objets mobiles durant leurs déplacements, notamment pour décrire des comportements ;

4. <http://swoogle.umbc.edu/>

- l'extraction d'informations supplémentaires sur les trajectoires traçant le déplacement des objets mobiles : ces informations sont le résultat d'opérations d'interrogation, telle que la sélection. Elles peuvent s'appuyer sur les annotations préalablement définies ;
- l'aide à la prise de décision : notamment dans le domaine de la sécurité. En effet, dans le domaine aérien et maritime la compréhension des trajectoires des avions et des bateaux aident les stations à prévoir les différentes évolutions de ces trajectoires. Ainsi, des collisions et autres accidents liés à la météo peuvent être évités. De plus, dans le cas des mammifères marins, que nous l'étudions dans cette thèse, l'évolution de leurs trajectoires est capitale pour la sauvegarde de la population lors de la pollution dans l'océan.

La prise de décision peut être considérée comme des scénarios de cas d'utilisation utilisant les données capturées certes, mais qui nécessitent dans la majorité des cas d'autres sources de données ou d'informations. Tout d'abord, les informations que nous avons sur le domaine dites aussi des règles métiers ou thématiques. Elles sont souvent sémantiques et doivent être liées aux données capturées. Ensuite, le caractère spatio-temporel des données capturées se traduit par des relations spatiales et temporelles et des connaissances sur l'espace géographique dans lequel la capture est réalisée. Ces informations sémantiques forment une nouvelle couche de données, que nous appelons **trajectoires sémantiques**. Le processus qui consiste à tisser les liens entre les trajectoires et les trajectoires sémantiques est souvent appelé **annotation**.

3 Contributions

Les cas d'utilisations présentés à titre d'exemple forment en général des systèmes d'informations évolutifs et complexes. Par conséquent, il est indispensable d'utiliser une modélisation permettant de :

- contrôler les données : en définissant une structuration de cette masse de données. En effet, cette structuration doit s'appuyer sur un modèle de données rigoureux et évolutif pour gérer efficacement : l'indexation, l'interrogation, la comparaison et d'une manière générale la manipulation des trajectoires ;
- comprendre les données : en définissant une structuration des connaissances liées à ces données. Ces connaissances doivent tenir compte des méta-données capturées et des annotations définies par les experts des domaines ;
- prévoir les comportements : en définissant des modèles de passage des données aux connaissances sur celles-ci. Un premier objectif peut être la structuration des modèles d'analyse et de traitement sur ces données, qui sont déjà utilisées dans ces deux domaines, puis définir les liaisons possibles entre les structures de hauts niveaux sur les trajectoires et ces modèles de traitement.

Dans cette thèse, nous présentons une approche ontologique pour la modélisation et l'analyse de trajectoire. Cette approche essaie de répondre à l'ensemble des exigences nécessaires au processus qui va de la donnée jusqu'à la prise de décision. Dans un premier temps, on construit un modèle UML représentant les classes d'analyse de la trajectoire.

On enrichit le modèle de trajectoire pour prendre en compte les annotations métiers. Pour formaliser la trajectoire sémantique, on utilise une approche ontologique. Le modèle enrichi est transformé en une ontologie et les règles d'annotation forment ses axiomes. Pour faire le lien entre les concepts temporels et spatiaux de l'ontologie construite, on adopte une approche de réutilisation d'ontologies existantes. On décrit une ontologie sur le temps et une autre sur l'espace en spécifiant les règles de base que doivent définir ces ontologies. De plus, on prend comme cas d'étude les déplacements de mammifères marins pour appliquer une approche d'annotation automatique. Elle consiste à faire une classification des trajectoires (ou de parties de trajectoires) pour ressortir des invariants classés selon leurs valeurs à l'aide d'une table de décision. L'expert du domaine valide alors les règles utilisées dans le processus d'annotation automatique. Enfin, nous présentons une réalisation dans une base de données à base ontologique avec une évaluation d'expérimentations.

4 Plan de la thèse

Ce mémoire de thèse est organisé en trois parties. Les deux premières parties fixent le contexte et l'état de l'art relatifs aux problématiques de notre travail en rappelant brièvement les modèles de données spatiaux, temporelles, spatio-temporelles et des trajectoires. Ensuite, plus particulièrement, les modèles ontologiques ainsi que les approches de raisonnements spatiaux, temporels et spatio-temporels. Les trajectoires sont le thème central de cette thèse. La troisième partie présente notre contribution. Nous détaillons ci-dessous le contenu de chacun de ces chapitres.

Première partie

Chapitre 1 : ce chapitre est consacré à une présentation de quelques modèles de données spatiales, temporelles et spatio-temporelles. Nous avons privilégié les modèles de données normalisés ou bénéficiant d'une spécification pour une exploitation dans un modèle de données de trajectoire ;

Chapitre 2 : ce chapitre traite de manière plus précise les modèles des données de la trajectoire. En particulier, nous présentons les travaux utilisant une approche conceptuelle.

Deuxième partie

Chapitre 3 : ce chapitre présente quelques importants travaux autour de l'approche ontologique pour la représentation et le raisonnement sur des données spatiales, temporelles et spatio-temporelles ;

Chapitre 4 : ce chapitre présente des travaux de recherche utilisant des approches ontologiques pour la modélisation des objets mobiles et les trajectoires d'une manière générale. La dernière section de ce chapitre présente des travaux concernant la prise en compte de la dimension métier, dite aussi thématique, par les nouveaux modèles ontologiques spatiales/temporelles.

Troisième partie

- Chapitre 1** : ce chapitre présente une méthodologie partant de l'expérience des objets mobiles du monde réel vers un modèle ontologique générique pour la trajectoire. Notre méthode part de l'étude de deux types de trajectoires, des bateaux et des avions. À partir de l'analyse des données obtenues par des capteurs hétérogènes, nous proposons un modèle conceptuel générique de la trajectoire, présenté en UML. Ensuite, nous enrichissons le modèle de la trajectoire en proposant un modèle de la trajectoire sémantique. Nous nous basons sur l'annotation qui représente l'objectif du déplacement de l'objet mobile. À partir de ce point, nous construisons l'ontologie de la trajectoire sémantique ;
- Chapitre 2** : vue le besoin à raisonner avec des concepts et des opérateurs spatiaux et temporels, nous proposons la solution de réutilisation d'ontologies standards. Ainsi, nous détaillons la méthode de correspondance pour faire le lien entre les différentes ontologies en utilisant le langage OWL ;
- Chapitre 3** : dans ce chapitre, nous exploitons la structure générale de la trajectoire représentée par un modèle ontologique afin d'appliquer un raisonnement thématique, spatial et temporel. Nous souhaitons que ce raisonnement soit généralisé et applicable à tout les domaines d'applications exploitant des trajectoires appropriées. Pour cela, nous prenons pour domaine d'application les trajectoires des mammifères marins pour voir concrètement le mécanisme de raisonnement sur les concepts thématiques, spatiaux et temporels de l'ontologie de la trajectoire ;
- Chapitre 4** : dans ce chapitre, nous discutons la gestion des instances d'une ontologie par les systèmes de gestion des bases de données. La première partie est consacrée à l'étude de différentes architectures mis en œuvre dans un système de gestion de base de données pour le stockage des données ontologiques, le chargement des données, le calcul des inférences et l'interrogation. Dans un second temps, on présente l'implémentation de ces techniques dans les systèmes de bases de données sémantiques à travers quatre études de cas : AllegroGraph (Franz Inc.), Virtuoso (openlink Software), Semantic Plateform (Intellidimension) et Oracle Semantic Data Store (Oracle) ;
- Chapitre 5** : dans ce chapitre, nous présentons une démarche pour l'implantation des ontologies de la trajectoire, du temps et de l'espace. Ainsi, nous montrons la création des parties déclaratives et des parties impératives liées aux règles prises en compte par ces ontologies. Le chargement des données dans l'ontologie de trajectoire utilise le principe de correspondance entre une ontologie et une base de données relationnelle. Nous utilisons un chargement en masse. Nous vérifions la consistance de ces données ontologiques. Afin d'interroger les données ontologiques, nous détaillons le processus de calcul de l'inférence basé sur la création d'index sur les données. Enfin, nous discutons la partie interrogation des données ontologiques en utilisant une solution basée sur une fonction table pour les données sémantiques.

Quatrième partie

Enfin, nous concluons ce mémoire en revenant sur les différents travaux présentés et la manière dont ils répondent aux problématiques initiales, tout en essayant de porter

un regard critique sur ceux-ci. Nous envisagerons également certains travaux futurs qu'il nous semble important de garder à l'esprit et dont nous menons actuellement.

5 Cadre du travail

Ces travaux de thèse ont été financés par le ministère de l'enseignement supérieur et de la recherche. Ils se sont déroulés au sein du laboratoire L3i (Laboratoire : Informatique, Image, Interaction), de l'université de La Rochelle. Les activités de notre équipe s'intéressent aux aspects liés à la valorisation et intelligence des données. Les approches développées relèvent de l'ingénierie des données. Une place importante est donnée à l'expression de la sémantique et à celles des contraintes statiques ou dynamiques dans un contexte ontologique ou entrepôt de données.

Première partie

Modèles de données et modèles ontologiques pour les trajectoires

Chapitre 1

État de l'art et positionnement scientifique

Sommaire

1.1	Introduction	20
1.2	Modèles de données spatiales	20
1.2.1	Introduction	20
1.2.2	Généralités sur les modèle de données spatiales de l'OGC	20
1.2.3	Modèle de données spatiales : GML	21
1.2.4	Modèle de données spatiales : KML	22
1.2.5	Modèle de données spatiales pour le SQL : OGC SFS	24
1.3	Modèles de données temporelles	26
1.3.1	Introduction	26
1.3.2	Modèle temporel dans la norme ISO 19108	26
1.3.3	Algèbre des intervalles temporels d'Allen	27
1.3.4	Modèle temporel dans les bases de données	28
1.4	Modèles de données spatio-temporelles	30
1.4.1	Introduction	30
1.4.2	Modélisation spatio-temporelle avec MADS	30
1.4.3	Modélisation spatio-temporelle avec Perceptory	33
1.4.4	Modélisation spatio-temporelle avec AROM-ST	34
1.5	Modèles de données spatio-temporelles	36
1.5.1	Introduction	36
1.5.2	Modélisation spatio-temporelle avec MADS	36
1.5.3	Modélisation spatio-temporelle avec Perceptory	39
1.5.4	Modélisation spatio-temporelle avec AROM-ST	40
1.6	Synthèse	42

1.1 Introduction

L'espace et le temps sont des dimensions essentiels pour la maîtrise et la compréhension de notre environnement. L'usage de l'informatique pour manipuler et analyser les données avec leurs composantes spatiales et temporelles est une problématique scientifique ancienne mais toujours d'actualité. Ainsi, la représentation de l'espace et du temps est un sujet de recherche actif dans différentes communautés scientifiques. On peut citer les communautés qui s'intéressent à la géomatique, aux systèmes d'informations géographiques, aux bases de données (spatiales, temporelles et spatio-temporelles), à l'ingénierie des connaissances sur l'espace, le temps, le raisonnement spatio-temporel, etc. Dans ce chapitre, nous présentons des modèles de données spatiaux, temporels et spatio-temporels étudiés au cours de ce travail de thèse. Nous avons privilégié les modèles de données normalisés ou bénéficiant d'une spécification approuvée pour une exploitation dans un modèle de données pour les trajectoires. Ce modèle de données de plus haut niveau doit prendre en compte des données spatio-temporelles multidimensionnelles.

1.2 Modèles de données spatiales

1.2.1 Introduction

La compréhension et l'interprétation de toute donnée spatiale dépendent toujours du modèle de données spatiales dans lequel cette donnée est exprimée. Nous avons donc étudié différents modèles spatiaux afin de trouver un candidat, de préférence un standard, qui permettra de représenter rigoureusement les données de trajectoires et de garantir l'interopérabilité avec d'autres modèles de données.

Dans ce chapitre, nous proposons une description des modèles de données spatiales que nous avons étudiés. Plus spécifiquement, les modèles de données standards proposés par le consortium géo-spatial OGC (*Open Geospatial Consortium*). Notre objectif est l'exploitation d'une partie d'un standard adéquat pour proposer un modèle de données spatiales pour les données de trajectoires.

1.2.2 Généralités sur les modèle de données spatiales de l'OGC

L'OGC (*Open Geospatial Consortium*) est un consortium industriel international auquel participent des représentants de plus de 270 entreprises et organismes administratifs. Ce consortium s'est fixé comme objectif de développer de manière consensuelle des spécifications publiques pour des applications spatiales. La normalisation des schémas conceptuels de description d'un phénomène géographique a pour but d'améliorer les partages et les échanges des données et des méta-données entre différentes applications. Les spécifications de l'OGC [Percivall, 2003] sont destinées à être utilisées dans les systèmes d'information par les concepteurs, les développeurs ou les utilisateurs des données géographiques ou spatiales afin de manipuler et/ou produire des données spatiales. Grâce aux

spécifications de l'OGC, on devrait être capable de mettre à disposition des usagers et des applications des données spatiales complexes, ainsi que des services accessibles se basant sur celles-ci.

Les standards de l'OGC gravitent autour du concept d'élément (en anglais, *Feature*). Un élément représente une abstraction du monde réel. On parle d'élément géographique si ce dernier est associé à une localisation sur le globe. Les données vectorielles regroupent les primitives géométriques et topologiques utilisées, séparément ou dans le même jeu de données, pour définir des objets exprimant les caractéristiques spatiales d'éléments géographiques. Dans le modèle de l'OGC, chaque caractéristique spatiale est décrite par un ou plusieurs attributs dont la valeur correspond soit à un objet géométrique, soit à un objet topologique :

- la composante géométrique comporte des descriptions sur la position, la taille, l'orientation et la forme du phénomène géographique, ainsi que sur la dimension de l'espace dans lequel le phénomène évolue. Les fonctions associées à la description géométrique d'un objet sont dépendantes du système de coordonnées de référence utilisé pour définir la position de l'objet. La géométrie est la seule partie de la donnée qui est modifiée lorsque l'on change de système géodésique ou de système de coordonnées ;
- la composante topologie est la partie de la donnée qui ne varie pas lorsque l'espace est déformé de manière élastique ou continue. Par exemple, la topologie est invariante par transformation de coordonnées. Dans le contexte de données spatiales, la topologie est utilisée, par exemple, pour décrire les relations de connectivité entre les différents éléments d'un graphe à n dimensions.

Dans les parties suivantes, nous nous intéressons à quelques standards de l'OGC pour la représentation des données spatiales. Nous parlerons de modèles de données spatiales car ces spécifications bénéficient de descriptions rigoureuses des types et des objets dans un cadre formel.

1.2.3 Modèle de données spatiales : GML

Le GML (*Geography Markup Language*) est un langage de type XML pour encoder, manipuler et échanger des données spatiales. Ce standard est développé par le consortium OGC pour garantir l'interopérabilité des données, en particulier dans les domaines des systèmes d'information géographique et de la géomatique. Ce langage fournit une variété de types d'objets pour décrire les données spatiales parmi lesquels : les entités, les systèmes de référence des coordonnées, les géométries, les topologies, le temps, les unités de mesure et les objets généralisés [Percivall, 2003]. Le GML consiste en un ensemble de schémas XML qui définissent un format ouvert pour l'échange de données et permettent de construire des modèles spécifiques pour des domaines spécialisés, comme l'urbanisme, l'hydrologie ou la géologie. Sans essayer de faire une description complète, nous présentons des éléments du langage GML.

La figure 1.1 montre la représentation UML du schéma pour les entités de la spécification GML. Les classes `AbstractGeometry` et `AssociationAttributeGroup` font partie du schéma pour la géométrie. Dans ce modèle, une propriété géométrique est modélisée

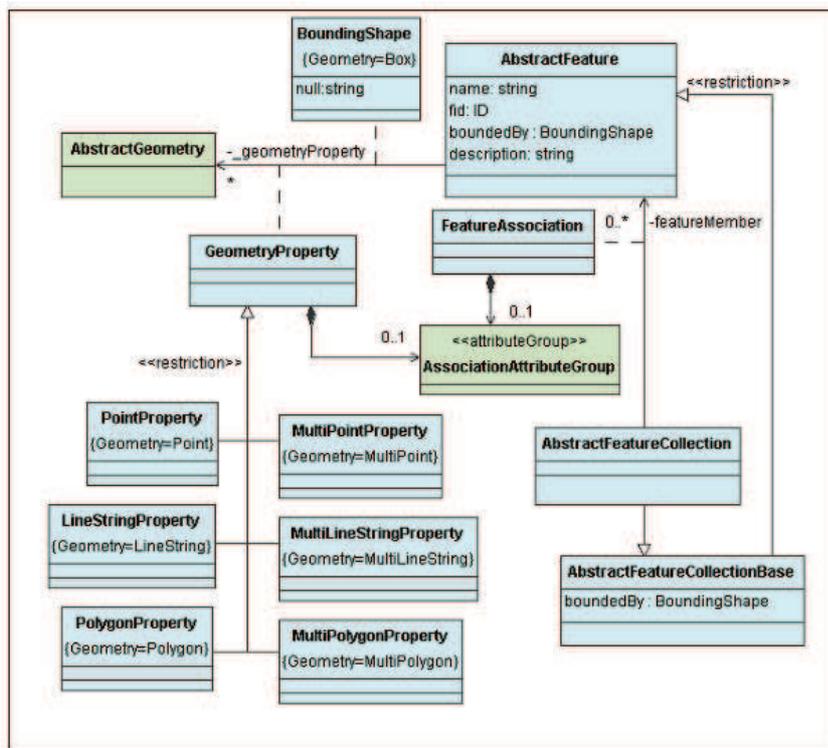


FIGURE 1.1 – Représentation UML du schéma pour les entités dans GML

par une classe association entre une entité et une géométrie. Les propriétés géométriques concrètes, par exemple `PointProperty`, contraignent la géométrie à un type particulier (par exemple `Point`). La figure 1.2 montre tous les types particuliers de géométries dans la représentation UML du schéma pour la géométrie. Toute géométrie est composée à l'aide d'un ou plusieurs éléments `<coord>` de type `Coord` ou `<coordinates>` de type `Coordinates` (extension de base `string`) comme le montre la définition XML-Schema donnée par la figure 1.3.

Le codage des entités avec une géométrie se réalise à l'aide des propriétés géométriques prédéfinies qui peuvent être utilisées afin de mettre en relation les entités avec différents types particuliers de géométries.

1.2.4 Modèle de données spatiales : KML

Le langage KML (*Keyhole Markup Language*) est fondé sur des balises géo-locales. Il est de type XML et destiné à la gestion de l'affichage de données géospatiales dans des systèmes de visualisation d'information géographique tels que : Google Earth, Google Maps, Google Mobile, World Wind, etc. Le KML fait partie maintenant des standards de l'OGC [Bacharach, 2008]. La figure 1.4 montre un extrait du modèle de données du KML sous la forme d'une hiérarchie de classes montrant ses éléments de base. Cette figure montre la racine du modèle `Object`, les éléments abstraits écrits en italique et englobés par des boîtes en pointillé (ex. `Feature`, `Geometry`, ...), et les autres classes

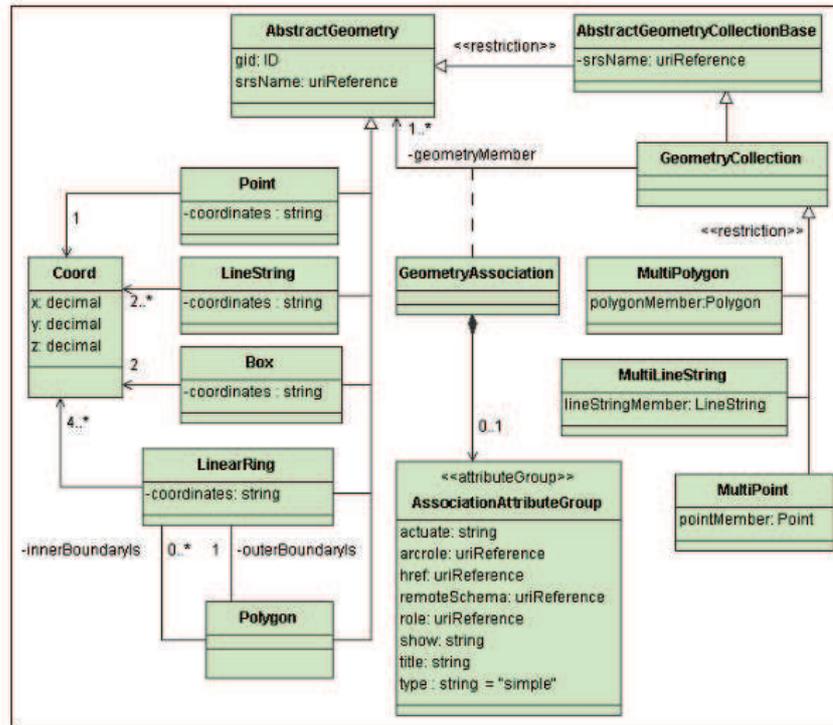


FIGURE 1.2 – Représentation UML du schéma pour la géométrie dans GML

```

<complexType name="CoordType">
  <sequence>
    <element name="X" type="decimal"/>
    <element name="Y" type="decimal" minOccurs="0"/>
    <element name="Z" type="decimal" minOccurs="0"/>
  </sequence>
</complexType>

<complexType name="CoordinatesType">
  <simpleContent>
    <extension base="string">
      <attribute name="decimal" type="string" use="default" value="."/>
      <attribute name="cs" type="string" use="default" value=","/>
      <attribute name="ts" type="string" use="default" value="&#x20;/>
    </extension>
  </simpleContent>
</complexType>
  
```

FIGURE 1.3 – Définition (XML Schema) des deux types d'éléments existants dans GML pour la description des nuplets de coordonnées

instanciables. Les instances sont décrites dans des fichiers XML décrivant généralement des données spatiales. Le code 1.1 donne un exemple de données `LinearRing` dans un fichier de description KML.

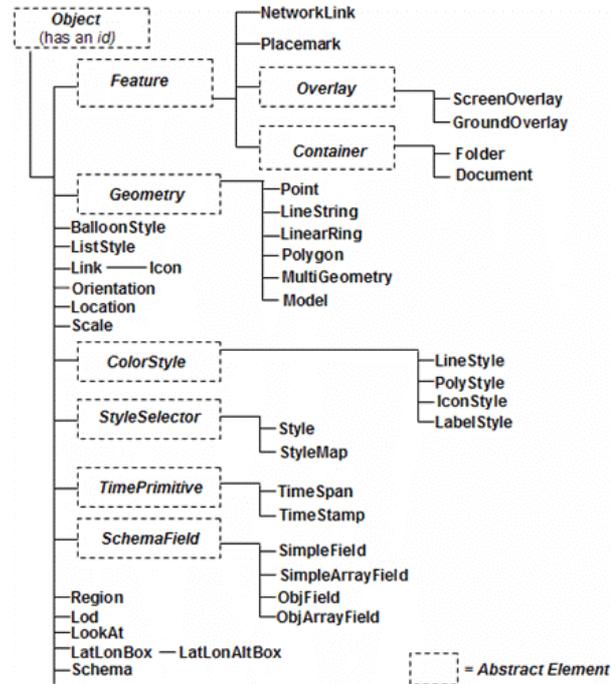


FIGURE 1.4 – L'arborescence des classes du modèle de données KML

```

1 <Polygon id="ID">
2 <!-- specific to Polygon -->
3 <extrude>0</extrude> <!-- boolean -->
4 <tessellate>0</tessellate> <!-- boolean -->
5 <altitudeMode>clampToGround</altitudeMode>
6 <!-- kml:altitudeModeEnum: clampToGround, relativeToGround, or absolute -->
7 <outerBoundaryIs>
8 <LinearRing>
9 <coordinates>...</coordinates> <!-- lon,lat[,alt] -->
10 </LinearRing>
11 </outerBoundaryIs>
12 <innerBoundaryIs>
13 <LinearRing>
14 <coordinates>...</coordinates> <!-- lon,lat[,alt] -->
15 </LinearRing>
16 </innerBoundaryIs>
17 </Polygon>
  
```

Code 1.1 – Extrait d'un fichier KML déclarant des instances du type de données `LinearRing`

1.2.5 Modèle de données spatiales pour le SQL : OGC SFS

La spécification *OpenGIS Simple Features Specification For SQL* [Beddoe et al., 1999] définit un modèle de données SQL standard pour prendre en charge le stockage, la manipulation et la mise à jour des collections d'éléments avec une composante spatiale simple.

Ces derniers constituent les briques de base pour le modèle de données et sont spécifiés par des attributs spatiaux et non spatiaux. Les attributs spatiaux ont des valeurs géométriques.

Le modèle de données de cette spécification est illustré par le diagramme de classes à notations OMT de la figure 1.5. La classe abstraite racine de la hiérarchie **Geometry** a les sous-classes concrètes : **Point**, **Curve**, **Surface**, **GeometryCollection**. Chaque classe est associée à un système de référence spatial **SpatialReferenceSystem**. Ce modèle de données étend les modèles antérieurs de l'OGC par les classes de collections d'éléments : **MultiPoint** pour les objets de dimension 0, **MultiLineString** pour les objets de dimension 1 et **MultiPolygon** pour les objets de dimension 2. Les classes **MultiCurve** et **MultiSurface** sont présentées comme des super-classes d'interface pour manipuler des collections de courbes et de surfaces. Il existe également la norme SQL/MM [Stolze, 2003] qui propose une structure similaire.

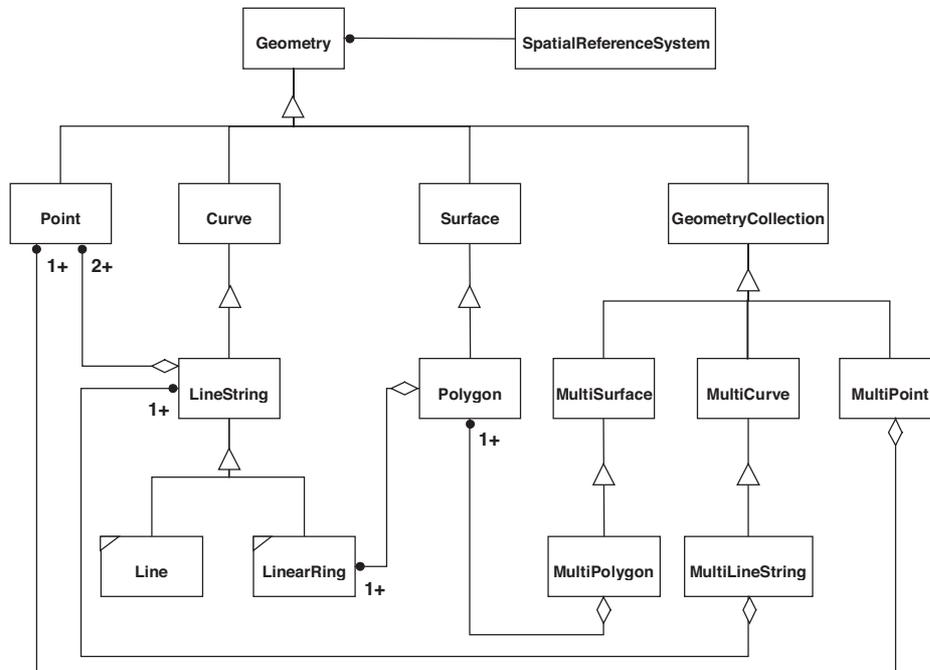


FIGURE 1.5 – Diagramme de classe des simples composantes spatiales De l'OGC

Les opérateurs spatiaux permettent un raisonnement spatial sur les données. L'OGC propose un ensemble d'opérateurs topologiques : **Disjoint**, **Touches**, **Crosses**, **Within**, **Contains**, **Intersects** et **Overlaps**. Ces fonctions sont formellement définies à partir des notions d'intérieur, de frontière et d'extérieur d'objets géométriques. Elles permettent de déterminer la configuration spatiale entre deux objets spatiaux. Dans cette spécification, les relations spatiales entre deux géométries « Geometry-A » et « Geometry-B » sont définies comme suit :

- **Equals**(Geometry-A : Geometry-B) : « Geometry-A » est égale à « Geometry-B ».
- **Disjoint**(Geometry-A : Geometry-B) : « Geometry-A » est disjoint à « Geometry-B ».

- Intersects(Geometry-A : Geometry-B) : « Geometry-A » intersecte avec « Geometry-B ».
- Touches(Geometry-A : Geometry-B) : « Geometry-A » touche « Geometry-B ».
- Crosses(Geometry-A : Geometry-B) : « Geometry-A » intersecte « Geometry-B » et l'intersection résulte en une géométrie comportant une dimension inférieure de un à la dimension maximale des deux géométries source et si l'ensemble d'intersections est intérieur aux deux géométries source.
- Within(Geometry-A : Geometry-B) : « Geometry-B » est dans « Geometry-A ».
- Contains(Geometry-A : Geometry-B) : « Geometry-B » contient « Geometry-A ».
- Overlaps(Geometry-A : Geometry-B) : « Geometry-B » couvre « Geometry-A ».

Conceptuellement, tout élément simple est représenté par un nuplet d'une table dans un système de gestion de bases de données relationnel. Cette table est définie par des colonnes à valeurs géométriques. Dans l'environnement SQL92, une colonne de valeur géométrique est implémentée comme une dépendance de référence dans une table de géométrie. Une valeur de géométrie est stockée en utilisant une ou plusieurs lignes de la table de géométrie.

1.3 Modèles de données temporelles

1.3.1 Introduction

Deux types de données principaux sont utilisés dans les modèles temporels : l'instant et l'intervalle. L'instant est assimilé à un point de la droite réelle. L'intervalle est représenté par un segment de la droite [Bessire *et al.*,]. Cette approche est basée sur une vision discrète du temps, qui permet d'identifier l'instant d'origine d'un événement, ou bien sa période de durabilité.

Dans ce chapitre, nous présentons les modèles de données temporelles que nous avons étudiés. Notre objectif est la recherche d'un standard adéquat pour l'utilisation dans un modèle des données de trajectoires.

1.3.2 Modèle temporel dans la norme ISO 19108

La norme ISO 19108:2002 [ISO-TC-211, 2002] s'intéresse particulièrement aux concepts temporels dans le cadre des systèmes d'information géographique. Toutefois, ce cadre peut être utilisé dans les domaines où la donnée temporelle est présente et doit être modélisée. En effet, la norme reprend et étend des standards existants sur la donnée temporelle comme la norme ISO 8601. La norme ISO 19108:2002 concerne les caractéristiques temporelles de l'information géographique comme elles sont extraites du monde réel, par conséquent, elle met l'accent sur le temps valide plutôt que le temps transactionnel. Elle fournit une base pour la définition des entités temporelles en précisant leurs attributs et leurs opérations ainsi que les associations en utilisant les modèles de domaine UML. La figure 1.6 présente la hiérarchie fondamentale des entités temporelles.

Dans cette hiérarchie (figure 1.6) :

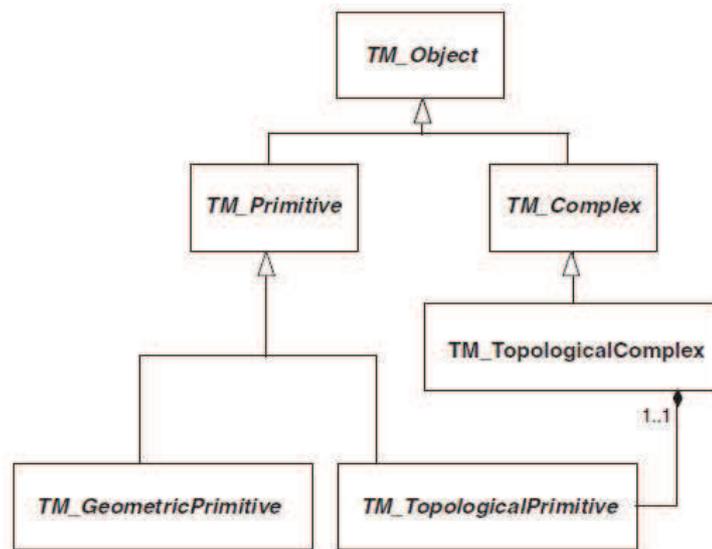


FIGURE 1.6 – Hiérarchie fondamentale des entités temporelles de l'ISO 19108

- `TM_Object` : est la classe temporelle abstraite racine de la hiérarchie ;
- `TM_Primitive` : est la classe abstraite qui représente un élément simple de la géométrie ou de la topologie de temps ;
- `TM_GeometricPrimitive` : est la classe abstraite qui fournit des informations temporelles sur la position ;
- `TM_TopologicalPrimitive` : est la classe abstraite qui fournit des informations sur la connectivité dans le temps ;
- `TM_Complex` : est la classe abstraite représentant est une agrégation de `TM_Primitives` ;
- `TM_TopologicalComplex` : est la seule sous-classe concrète de `TM_Complex` sous la forme d'une agrégation de `TM_TopologicalPrimitives`.

Les deux primitives de la dimension temporelle sont l'instant, la classe `TM_Instant`, et l'intervalle, la classe `TM_Period` comme le montre le diagramme de classes (figure 1.7). Ces primitives sont définies de façon analytique dans le cas où le temps est mesuré sur une échelle d'intervalle, et de façon analogique (i.e. discrète), dans le cas où le temps est mesuré sur une échelle ordinale (quantitative munie d'un ordre). Le modèle de données 1.7 fournit l'interface `TM_Order` munie d'une fonction membre pour déterminer la position entre deux objets de type `TM_Primitive`. Les valeurs retournées sont définies par un type de données énumération `TM_RelativePosition` et s'appuie sur les 13 relations temporelles de l'algèbre d'Allen [Allen, 1983a].

1.3.3 Algèbre des intervalles temporels d'Allen

Afin de représenter des actions et des événements se produisant au cours du temps, Allen et al. [Allen, 1983a; Allen and Ferguson, 1994] proposent un modèle du temps basé sur des intervalles temporels. Les auteurs modélisent un intervalle temporel par une paire de points temporels ordonnés, avec le premier point inférieur au second. Dans [Allen, 1983a], l'auteur introduit l'algèbre des intervalles temporels et définit un calcul pour le

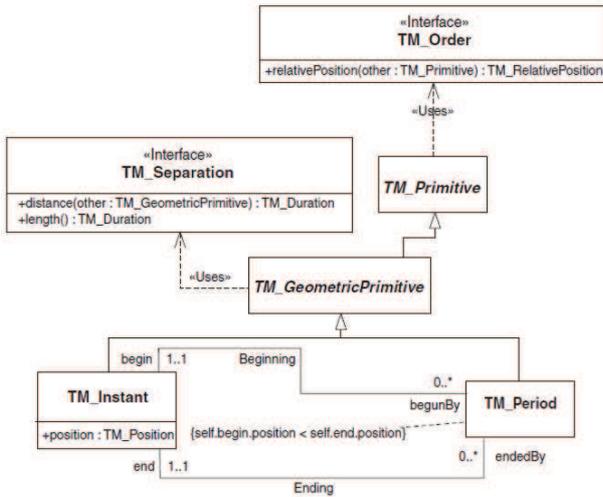


FIGURE 1.7 – Les primitives géométriques temporelles

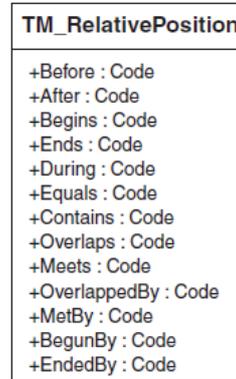


FIGURE 1.8 – TM_RelativePosition

raisonnement temporel basé sur 13 relations de base ou atomiques données par la figure 1.9. Le standard ISO 19108:2002 s'appuie sur cet ensemble de relations et propose une implémentation sous forme d'un ensemble d'opérations.

Relation	Symbole	Inverse	Symbole	Illustration
X before Y	b	Y after X	a	
X meets Y	m	Y met-by X	mi	
X overlaps Y	o	Y overlapped-by X	oi	
X starts Y	s	Y started-by X	si	
X during Y	d	Y contains X	di	
X finishes Y	f	Y finished-by X	fi	
X equals Y	eq	Y equals X	eq	

FIGURE 1.9 – Les relations entre intervalles temporels d'Allen

1.3.4 Modèle temporel dans les bases de données

Les bases de données intègrent l'aspect temporel des données en s'appuyant sur un modèle de données temporelles qui a donné naissance à une version temporelle du langage SQL. Pour le développement d'un SQL temporel, Richard Snodgrass propose en 1992 l'ajout des extensions temporelles au langage SQL92. Cette extension, connue sous

le nom de TSQL2 [Snodgrass and Ahn, 1986], fut alors développée en 1993 par une communauté bases de données. La version définitive des spécifications du langage TSQL2 est publiée en septembre 1994. Des parties de TSQL2 sont incluses dans un sous-standard de SQL3 (ISO/IEC 9075 partie 7, 1999), appelé SQL/Temporal. Dans TSQL2, les auteurs définissent deux dimensions temporelles fondamentales : `VALIDTIME` et `TRANSACTIONTIME`, ainsi qu'un nouveau type de données appelé : `PERIOD`. L'introduction de ces nouvelles définitions nécessite une adaptation du langage SQL. Dans cette partie, nous nous limitons à l'introduction du modèle de données.

Le type de donnée PERIOD

Une période est une durée ancrée qui représente un ensemble de granules de temps contigus dans la durée avec une précision spécifiée. Chaque donnée de type `PERIOD` possède un début et une fin qui peuvent être du type `DATE`, `TIME`, `TIMESTAMP` mais les deux doivent être du même type. Le code 1.2 donne un exemple de déclaration et d'utilisation d'une donnée de type `PERIOD`.

```
1 CREATE TABLE contrat (  
2   contrat_id          VARCHAR(50) NOT NULL,  
3   client_id          VARCHAR(10) NOT NULL,  
4   contrat_duree      PERIOD(DATE) FORMAT 'dd-mm-yyyy' NOT NULL  
5 );  
6  
7 INSERT INTO contrat(contrat_id, client_id, contrat_duree)  
8 VALUES ('ct1001', 'cl1001', PERIOD(DATE '01-01-2010', DATE '01-06-2010'));
```

Code 1.2 – Une table avec un attribut de type `PERIOD`

Le type `PERIOD` bénéficie d'un ensemble de fonctions et de prédicats permettant la manipulation des données temporelles dans la base de données. Les fonctions `BEGIN`, `END`, et `LAST` renvoient respectivement la valeur de début, de fin et la valeur qui représente le dernier granule de la période. La norme propose des prédicats tels que : `CONTAINS`, `EXPAND`, `EXPANDING`, `MEETS`, `NORMALIZE`, `PERIOD`, `PRECEDES`, `SUCCEEDS`.

La dimension VALIDTIME

Le temps valide modélise le temps du monde réel et indique la période de temps durant laquelle un fait, représenté par toutes les informations d'un nuplet, est effectif. Cette information est stockée dans la base de données par le biais du type `PERIOD` dans une colonne de temps valide. Selon la norme, cette dernière est définie en spécifiant `AS VALIDTIME` dans la définition de la colonne. Dans le code 1.2, si la durée d'un contrat est une colonne de type temps valide, alors sa définition doit être :

```
1 contrat_duree PERIOD(DATE) FORMAT 'dd-mm-yyyy' NOT NULL AS VALIDTIME
```

Code 1.3 – Déclaration d'une donnée de temps valide

Notons que dans certaines bases de données temporelles, la notion de temps valide peut s'appliquer à toute la table.

La dimension TRANSACTIONTIME

Le temps transaction est la période de temps durant laquelle un fait est, ou a été effectif dans la base de données. Le temps transactionnel modélise la réalité de la base de données en mémorisant à quel moment des données sont ajoutées ou modifiées dans la base de données. Cette information est stockée dans la base de données par le biais du type `PERIOD` dans une colonne de temps transactionnel. Selon la norme, cette dernière est définie en spécifiant `AS TRANSACTIONTIME` dans la définition de la colonne. La norme précise que le temps transactionnel ne doit pas être modifié manuellement et doit être géré complètement par le système de gestion de la base de données.

1.4 Modèles de données spatio-temporelles

1.4.1 Introduction

Il existe différents modèles conceptuels de données (MCD) spatio-temporels. Nous présenterons trois principaux : MADS, Perceptory et AROM-ST. Ils sont basés respectivement sur les MCD : ER (entité-association), UML et AROM, et ils ont été étendus aux concepts spatiaux et temporels.

1.4.2 Modélisation spatio-temporelle avec MADS

L'approche MADS (Modélisation d'Applications à Données Spatio-temporelles) propose une modélisation conceptuelle des données spatio-temporelles. Elle permet aux concepteurs de s'abstraire des détails liés à la logique d'un SGBD quelconque en produisant des modèles conceptuels aussi proches que possible de la réalité [Parent *et al.*, 1997]. MADS utilise une représentation entité-association des données dans un modèle relationnel étendu. Les modèles MADS reprennent toutes les composantes classiques présentes dans la modélisation orientée objet, en ce qui concerne la partie structurelle des données.

On retrouve donc les concepts de classe, d'objet, d'attribut, d'association, de relation de généralisation/spécialisation, d'agrégation, etc. Les classes sont représentées par des rectangles et les associations par des ellipses. Les cardinalités sont représentées par différents styles de lignes : pointillés, ligne continue, ligne double, etc. La figure 1.15 illustre un schéma MADS ne faisant pas encore intervenir de notion de spatialité ou de temporalité. Ces deux notions peuvent être exprimées au niveau des classes et des associations et aussi au niveau des objets.

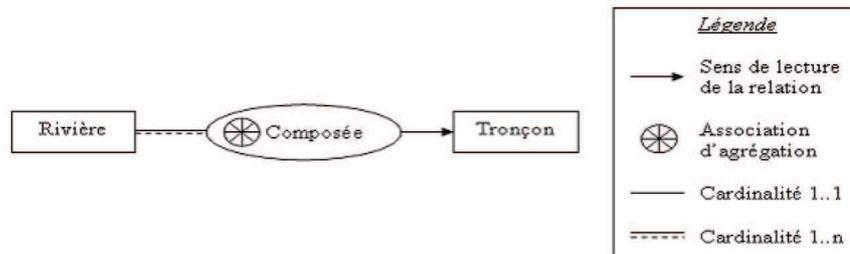


FIGURE 1.10 – Schéma MADS sans référence spatio-temporelle

Le modèle spatial dans MADS

La spatialité dans MADS s'exprime par un ensemble de types abstraits de données spatiaux hiérarchisés (figure 1.16). Ces types sont représentés sur un modèle MADS par des pictogrammes apparaissant au côté des éléments décrits. Les types spatiaux **Point** et

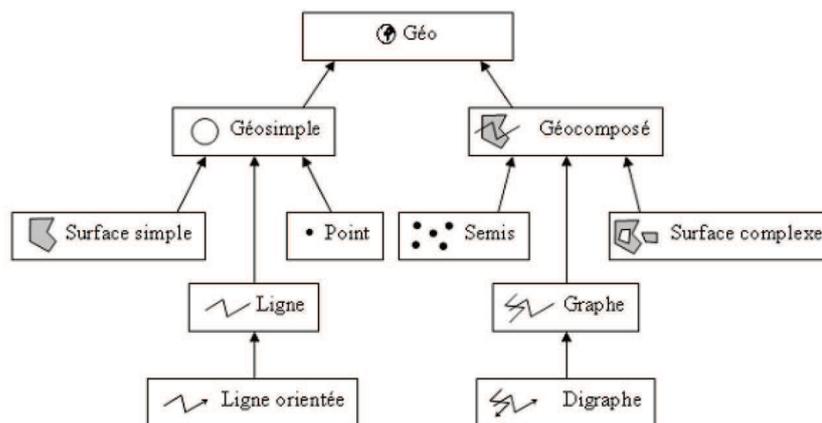


FIGURE 1.11 – Hiérarchie des types abstraits spatiaux dans MADS.

Semis représentent des objets ponctuels. Les types **Ligne**, **Ligne orientée**, **Graphe** et **Digraphe** représentent des objets linéaires (une dimension), et les types **Surface simple** et **Surface complexe** les objets surfaciques (deux dimensions). Les associations peuvent également être décrites à l'aide de pictogrammes spatiaux d'un type particulier : ceux-ci représentent non pas une emprise spatiale mais des relations topologiques (tableau 1.4).

Relation	Pictogramme	Définition
Disjonction		Aucun partage (valable pour toute combinaison d'objets spatiaux).
Adjacence		Partage sans intérieur commun.
Croisement		Partage d'une partie de l'intérieur tel que le nombre de dimensions de l'objet partagé est strictement inférieur à la plus grande dimension des deux objets en relation.
Recouvrement		Partage d'une partie de l'intérieur tel que le nombre de dimensions de l'objet partagé est égal au nombre de dimensions des deux objets en relation (valable pour des objets avec le même nombre de dimensions).
Inclusion		La totalité de l'intérieur d'un objet en relation correspond à une partie de l'intérieur de l'autre.
Egalité		Partage de la totalité de l'intérieur et de la totalité de l'enveloppe (valable pour des objets de même dimension).

TABLE 1.1 – Relations topologiques dans MADS

Dans le cadre du projet MurMur [Parent *et al.*, 2005], un prototype d'atelier de génie logiciel (AGL) visant à traduire un schéma MADS en une base de données Oracle Spatial a été développé. Il comprend une interface graphique qui permet d'assembler aisément les objets qui constituent le modèle, ainsi que des modules de traduction qui génèrent le schéma Oracle Spatial correspondant.

Le modèle temporel dans MADS

Comme le spatial, dans MADS, le temporel est représenté par un pictogramme apparaissant au côté des éléments décrits. Lorsque l'on appose ce pictogramme sur une classe, c'est la durée de vie des objets de cette classe que l'on modélise. On peut également utiliser ce pictogramme sur les attributs (pour symboliser leur plage temporelle de validité) ou, de la même façon, sur les associations. Cette durée de vie ou plage de validité représente le cycle de vie de l'objet : naissance, suspension, activation et mort.

MADS prend en compte les relations temporelles entre intervalles en considérant l'algèbre relationnelle d'Allen [Allen, 1983b]. Ces relations sont aussi représentées par des pictogrammes relationnels. Le haut de la figure 1.17 décrit la composante temporelle d'un modèle MADS. Le milieu de cette figure illustre pour sa part une relation temporelle entre deux objets. Notons également que MADS propose des relations de génération et de transition. Les premières, notées G , permettent de faire le lien entre un objet nouvellement créé et les objets qui ont causé sa naissance. Le bas de la figure 1.17 décrit le cas d'un lac qui se serait formé après un éboulement rocheux sur une rivière. Les transitions, notées

T, permettent de représenter le fait qu'une entité du monde réel a tellement évolué qu'elle nécessite de changer de classification dans le modèle.

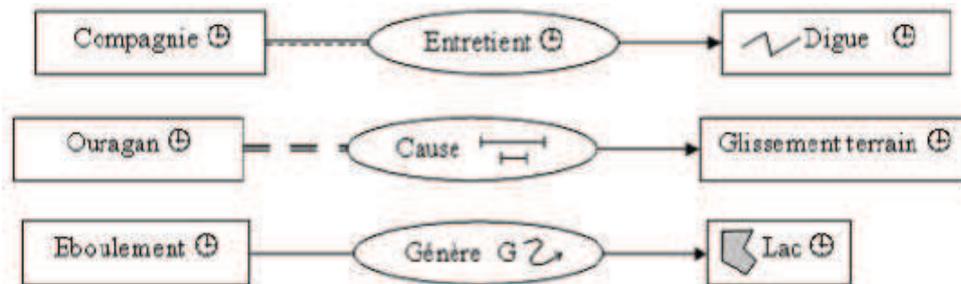


FIGURE 1.12 – Schéma MADS illustrant des relations temporelles

1.4.3 Modélisation spatio-temporelle avec Perceptory

Issu de l'université Laval au Québec, Perceptory propose tout comme MADS, une modélisation conceptuelle pour les données spatio-temporelles [Bédard *et al.*, 2004]. Un module logiciel, sous la forme d'un gabarit (en. Plug-in) pour Microsoft Visio, donne la possibilité de construire graphiquement le modèle et de générer un squelette du code de programmation pour le SGBD Oracle Spatial. Concernant la modélisation, Perceptory utilise les stéréotypes du méta-modèle UML. Ces stéréotypes permettent d'enrichir UML pour des domaines particuliers par la création de nouveaux éléments de modélisation, et éventuellement la définition d'une représentation graphique particulière. Les auteurs de Perceptory ont donc développé leurs propres stéréotypes par le biais de deux PVL (Plug-in for Visual Language) distincts :

- le PVL spatial, pour représenter des données purement spatiales ;
- le PVL spatio-temporel, pour représenter des données spatial et/ou temporel.

Ces PVL ont pris la forme de pictogrammes à ajouter dans les modèles UML, dont le principe ressemble à celui de MADS. Nous présentons ici la version initiale du langage visuel Perceptory, qui utilise des pictogrammes basés sur les lettres de l'alphabet grec.

Le modèle spatial dans Perceptory

Les symboles utilisés pour modéliser la dimension spatiale des entités spatiales sont donnés dans le tableau 1.5. Dans Perceptory, on peut utiliser plusieurs pictogrammes spatiaux si les objets d'une même classe doivent être représentés de plusieurs manières. Par exemple, la classe **Bâtiment** $\beta\epsilon$ peut contenir des bâtiments représentés par des points ainsi que des bâtiments représentés par des polygones. Ces deux représentations seront exclusives. Il est également possible de spécifier qu'une entité peut être représentée par un ensemble complexe de primitives, en fusionnant deux ou plusieurs pictogrammes.

Pictogramme	Description
β	Objet représenté par un point
ω	Objet représenté par une ligne
ϵ	Objet représenté par une surface
ν	Objet représenté par n'importe quelle géométrie
τ	Objet représenté par une géométrie complexe
Ψ	Objet représenté par une géométrie inconnue

TABLE 1.2 – Tableau des pictogrammes spatiaux dans Perceptory

Le modèle temporel dans Perceptory

La modélisation temporelle dans Perceptory suit deux principes fondamentaux : l'existence et l'évolution. L'existence s'apparente à la notion de durée de vie présente dans MADS, à l'exception des notions de **suspension** et d'**activation** qui n'existent pas dans Perceptory. Le Tableau 1.6 présente les pictogrammes temporels employés par Perceptory. Tout comme leurs équivalents spatiaux, les pictogrammes temporels peuvent être assemblés pour modéliser différents aspects de la réalité.

Pictogramme	Description
ζ	Objet dont l'existence est représentée par un instant
ξ	Objet dont l'existence est représentée par un intervalle
φ	Objet dont l'existence est de n'importe quelle type temporel
γ	Objet dont l'existence est représentée par un type temporel complexe
η	Objet dont l'existence est inconnue

TABLE 1.3 – Pictogrammes temporels dans Perceptory

Dans Perceptory, deux types d'évolution cohabitent : l'évolution descriptive, qui définit les variations d'un attribut au cours du temps, et l'évolution spatiale, qui fait varier la spatialité de l'objet. La première se représente en faisant apparaître le pictogramme temporel à côté de l'attribut, et la seconde en accolant le pictogramme temporel au pictogramme spatial qu'il affecte. Les dernières évolutions de l'outil Perceptory montrent une transition vers l'utilisation de pictogrammes plus intuitifs [Bédard *et al.*, 2004]. Le nombre de symboles a été réduit à cinq pour pouvoir représenter des données spatio-temporelles 2D. La grammaire a été enrichie avec des cardinalités pour les différents types d'attributs agrégés, combinée avec la possibilité d'utiliser des géométries composites ou alternatives. Ainsi, le langage est devenu plus expressif, le nombre de combinaisons exprimables à l'aide des pictogrammes est très important.

1.4.4 Modélisation spatio-temporelle avec AROM-ST

AROM-ST est un outil de modélisation spatio-temporelle basé sur le système de représentation AROM [Moisuc *et al.*, 2012]. C'est un outil générique, fondé sur les concepts de

classe et d'association pour représenter les objets et les liens entre les objets. La spécificité de AROM se situe dans l'utilisation d'un langage de modélisation algébrique (LMA). Celui-ci permet d'écrire des règles d'inférence dans un langage très proche du formalisme mathématique, de définir des variables à l'aide d'équations résolues par le système, d'appliquer des contraintes d'intégrité au niveau des classes ou associations, ou d'écrire des requêtes dans les bases de connaissances à l'aide d'un langage de requête (*AROM Query*).

Le modèle spatial dans AROM-ST

Les types spatiaux de AROM-ST (figure 1.18) sont conformes aux spécifications de OpenGIS [Percivall, 2003]. L'ensemble nécessaire et suffisant de types, pour pouvoir représenter les objets spatiaux, est composé des types simples : **Point**, **Polyline** et **Polygon**. Les types **Line** et **LinearRing** sont définis en appliquant des contraintes sur le type **Polyline**. À partir des types simples sont définis les types complexes : **MultiPoint**, **MultiLine** et **MultiArea**.

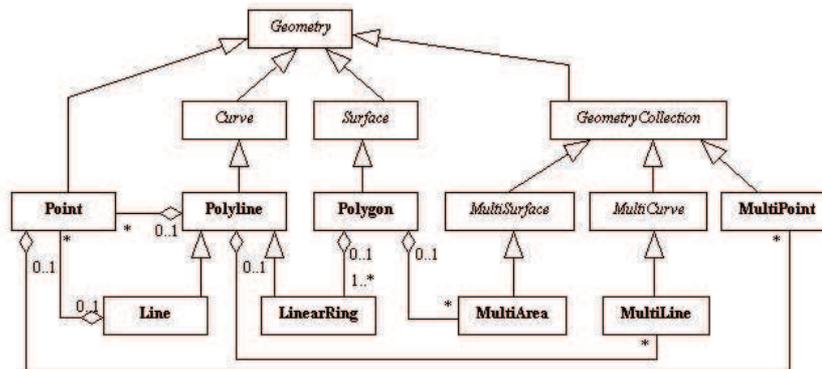


FIGURE 1.13 – Modèle spatial de AROM-ST

Le modèle temporel dans AROM-ST

Les types temporels simples sont le type **Instant** et le type **Interval**. Les types composés, constituant des collections d'instant ou d'intervalles, sont les types **MultiInstant** et **MultiInterval** (figure 1.19).

Les opérations spatio-temporelles dans AROM-ST

Les opérateurs LMA disponibles pour les types spatiaux et temporels sont divisés en trois catégories : les opérateurs topologiques, les opérateurs ensemblistes et les opérateurs de mesure. Les opérateurs topologiques sont des prédicats logiques binaires qui permettent de tester la position relative (spatiale ou temporelle) de deux objets. On considère les opérateurs topologiques définis par l'OpenGIS [Beddoe *et al.*, 1999]. Les opérateurs de

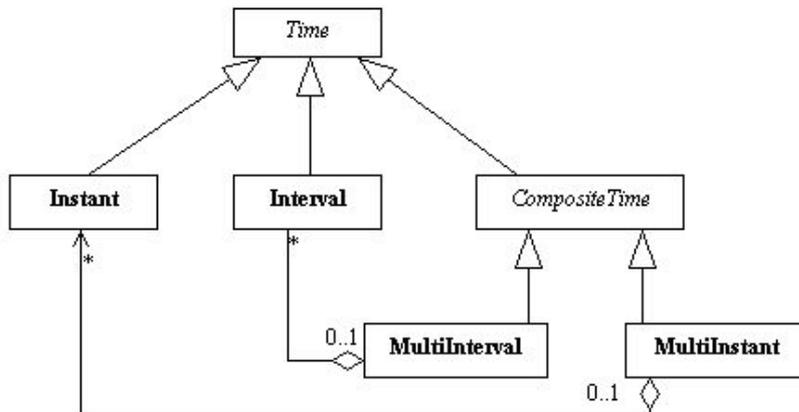


FIGURE 1.14 – Modèle temporel de AROM-ST

topologie temporelle considère les relations de l’algèbre d’Allen [Allen, 1983b]. Les opérateurs ensemblistes permettent de traiter l’espace et le temps, respectivement comme un ensemble de points et d’instant temporels. Pour plus de simplicité, ces opérateurs ont la même forme pour les dimensions spatiales et temporelles : `union`, `intersection`, `symmetricalDifference` et `difference`. Les opérateurs de mesure permettent une description quantitative de l’espace et du temps. Pour la dimension spatiale, les opérateurs définis sont `dimension` et `distance`, leurs correspondants pour la dimension temporelle sont `duration` et `timeLength`.

1.5 Modèles de données spatio-temporelles

1.5.1 Introduction

Il existe différents modèles conceptuels de données (MCD) spatio-temporels. Nous présenterons trois principaux : MADS, Perceptory et AROM-ST. Ils sont basés respectivement sur les MCD : ER (entité-association), UML et AROM, et ils ont été étendus aux concepts spatiaux et temporels.

1.5.2 Modélisation spatio-temporelle avec MADS

L’approche MADS (Modélisation d’Applications à Données Spatio-temporelles) propose une modélisation conceptuelle des données spatio-temporelles. Elle permet aux concepteurs de s’abstraire des détails liés à la logique d’un SGBD quelconque en produisant des modèles conceptuels aussi proches que possible de la réalité [Parent *et al.*, 1997]. MADS utilise une représentation entité-association des données dans un modèle relationnel étendu. Les modèles MADS reprennent toutes les composantes classiques présentes dans la modélisation orientée objet, en ce qui concerne la partie structurelle des données.

On retrouve donc les concepts de classe, d'objet, d'attribut, d'association, de relation de généralisation/spécialisation, d'agrégation, etc. Les classes sont représentées par des rectangles et les associations par des ellipses. Les cardinalités sont représentées par différents styles de lignes : pointillés, ligne continue, ligne double, etc. La figure 1.15 illustre un schéma MADS ne faisant pas encore intervenir de notion de spatialité ou de temporalité. Ces deux notions peuvent être exprimées au niveau des classes et des associations et aussi au niveau des objets.

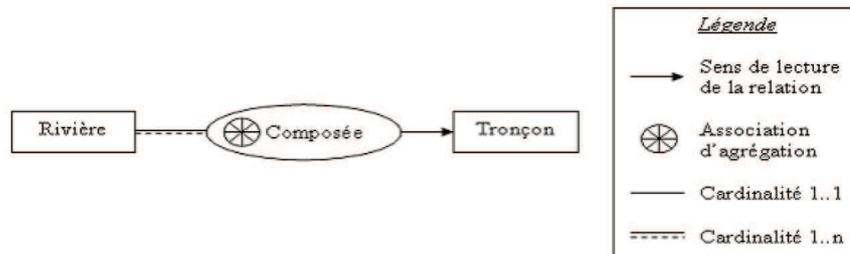


FIGURE 1.15 – Schéma MADS sans référence spatio-temporelle

Le modèle spatial dans MADS

La spatialité dans MADS s'exprime par un ensemble de types abstraits de données spatiaux hiérarchisés (figure 1.16). Ces types sont représentés sur un modèle MADS par des pictogrammes apparaissant au côté des éléments décrits. Les types spatiaux **Point** et

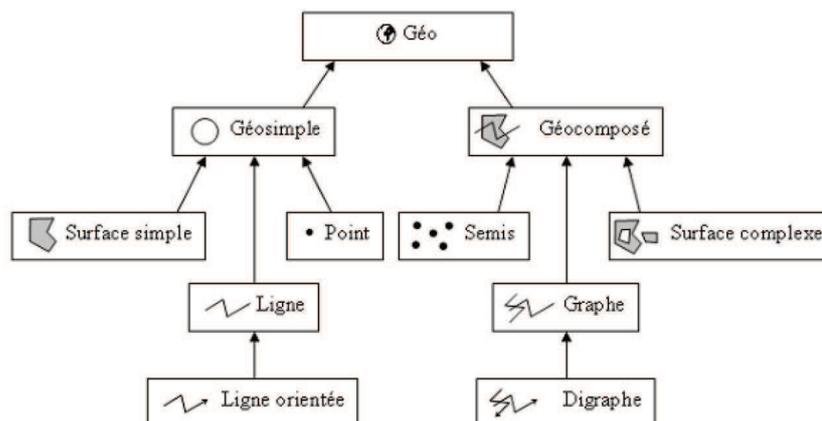


FIGURE 1.16 – Hiérarchie des types abstraits spatiaux dans MADS.

Semis représentent des objets ponctuels. Les types **Ligne**, **Ligne orientée**, **Grappe** et **Digraphe** représentent des objets linéaires (une dimension), et les types **Surface simple** et **Surface complexe** les objets surfaciques (deux dimensions). Les associations peuvent également être décrites à l'aide de pictogrammes spatiaux d'un type particulier : ceux-ci représentent non pas une emprise spatiale mais des relations topologiques (tableau 1.4).

Relation	Pictogramme	Définition
Disjonction		Aucun partage (valable pour toute combinaison d'objets spatiaux).
Adjacence		Partage sans intérieur commun.
Croisement		Partage d'une partie de l'intérieur tel que le nombre de dimensions de l'objet partagé est strictement inférieur à la plus grande dimension des deux objets en relation.
Recouvrement		Partage d'une partie de l'intérieur tel que le nombre de dimensions de l'objet partagé est égal au nombre de dimensions des deux objets en relation (valable pour des objets avec le même nombre de dimensions).
Inclusion		La totalité de l'intérieur d'un objet en relation correspond à une partie de l'intérieur de l'autre.
Egalité		Partage de la totalité de l'intérieur et de la totalité de l'enveloppe (valable pour des objets de même dimension).

TABLE 1.4 – Relations topologiques dans MADS

Dans le cadre du projet MurMur [Parent *et al.*, 2005], un prototype d'atelier de génie logiciel (AGL) visant à traduire un schéma MADS en une base de données Oracle Spatial a été développé. Il comprend une interface graphique qui permet d'assembler aisément les objets qui constituent le modèle, ainsi que des modules de traduction qui génèrent le schéma Oracle Spatial correspondant.

Le modèle temporel dans MADS

Comme le spatial, dans MADS, le temporel est représenté par un pictogramme apparaissant au côté des éléments décrits. Lorsque l'on appose ce pictogramme sur une classe, c'est la durée de vie des objets de cette classe que l'on modélise. On peut également utiliser ce pictogramme sur les attributs (pour symboliser leur plage temporelle de validité) ou, de la même façon, sur les associations. Cette durée de vie ou plage de validité représente le cycle de vie de l'objet : naissance, suspension, activation et mort.

MADS prend en compte les relations temporelles entre intervalles en considérant l'algèbre relationnelle d'Allen [Allen, 1983b]. Ces relations sont aussi représentées par des pictogrammes relationnels. Le haut de la figure 1.17 décrit la composante temporelle d'un modèle MADS. Le milieu de cette figure illustre pour sa part une relation temporelle entre deux objets. Notons également que MADS propose des relations de génération et de transition. Les premières, notées G , permettent de faire le lien entre un objet nouvellement créé et les objets qui ont causé sa naissance. Le bas de la figure 1.17 décrit le cas d'un lac qui se serait formé après un éboulement rocheux sur une rivière. Les transitions, notées

T, permettent de représenter le fait qu'une entité du monde réel a tellement évolué qu'elle nécessite de changer de classification dans le modèle.

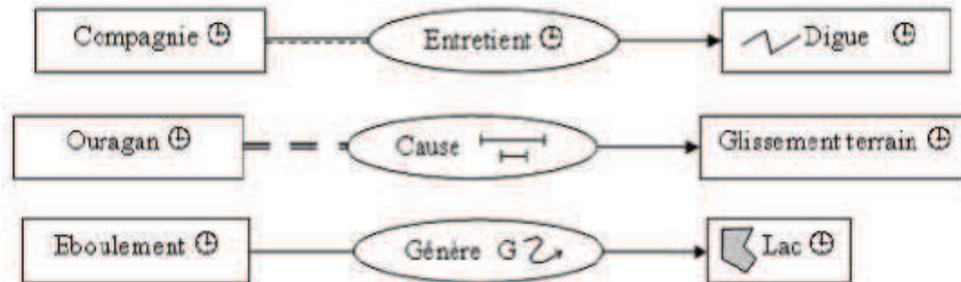


FIGURE 1.17 – Schéma MADS illustrant des relations temporelles

1.5.3 Modélisation spatio-temporelle avec Perceptory

Issu de l'université Laval au Québec, Perceptory propose tout comme MADS, une modélisation conceptuelle pour les données spatio-temporelles [Bédard *et al.*, 2004]. Un module logiciel, sous la forme d'un gabarit (en. Plug-in) pour Microsoft Visio, donne la possibilité de construire graphiquement le modèle et de générer un squelette du code de programmation pour le SGBD Oracle Spatial. Concernant la modélisation, Perceptory utilise les stéréotypes du méta-modèle UML. Ces stéréotypes permettent d'enrichir UML pour des domaines particuliers par la création de nouveaux éléments de modélisation, et éventuellement la définition d'une représentation graphique particulière. Les auteurs de Perceptory ont donc développé leurs propres stéréotypes par le biais de deux PVL (Plug-in for Visual Language) distincts :

- le PVL spatial, pour représenter des données purement spatiales ;
- le PVL spatio-temporel, pour représenter des données spatial et/ou temporel.

Ces PVL ont pris la forme de pictogrammes à ajouter dans les modèles UML, dont le principe ressemble à celui de MADS. Nous présentons ici la version initiale du langage visuel Perceptory, qui utilise des pictogrammes basés sur les lettres de l'alphabet grec.

Le modèle spatial dans Perceptory

Les symboles utilisés pour modéliser la dimension spatiale des entités spatiales sont donnés dans le tableau 1.5. Dans Perceptory, on peut utiliser plusieurs pictogrammes spatiaux si les objets d'une même classe doivent être représentés de plusieurs manières. Par exemple, la classe **Bâtiment** $\beta\epsilon$ peut contenir des bâtiments représentés par des points ainsi que des bâtiments représentés par des polygones. Ces deux représentations seront exclusives. Il est également possible de spécifier qu'une entité peut être représentée par un ensemble complexe de primitives, en fusionnant deux ou plusieurs pictogrammes.

Pictogramme	Description
β	Objet représenté par un point
ω	Objet représenté par une ligne
ϵ	Objet représenté par une surface
ν	Objet représenté par n'importe quelle géométrie
τ	Objet représenté par une géométrie complexe
Ψ	Objet représenté par une géométrie inconnue

TABLE 1.5 – Tableau des pictogrammes spatiaux dans Perceptory

Le modèle temporel dans Perceptory

La modélisation temporelle dans Perceptory suit deux principes fondamentaux : l'existence et l'évolution. L'existence s'apparente à la notion de durée de vie présente dans MADS, à l'exception des notions de **suspension** et d'**activation** qui n'existent pas dans Perceptory. Le Tableau 1.6 présente les pictogrammes temporels employés par Perceptory. Tout comme leurs équivalents spatiaux, les pictogrammes temporels peuvent être assemblés pour modéliser différents aspects de la réalité.

Pictogramme	Description
ζ	Objet dont l'existence est représentée par un instant
ξ	Objet dont l'existence est représentée par un intervalle
φ	Objet dont l'existence est de n'importe quelle type temporel
γ	Objet dont l'existence est représentée par un type temporel complexe
η	Objet dont l'existence est inconnue

TABLE 1.6 – Pictogrammes temporels dans Perceptory

Dans Perceptory, deux types d'évolution cohabitent : l'évolution descriptive, qui définit les variations d'un attribut au cours du temps, et l'évolution spatiale, qui fait varier la spatialité de l'objet. La première se représente en faisant apparaître le pictogramme temporel à côté de l'attribut, et la seconde en accolant le pictogramme temporel au pictogramme spatial qu'il affecte. Les dernières évolutions de l'outil Perceptory montrent une transition vers l'utilisation de pictogrammes plus intuitifs [Bédard *et al.*, 2004]. Le nombre de symboles a été réduit à cinq pour pouvoir représenter des données spatio-temporelles 2D. La grammaire a été enrichie avec des cardinalités pour les différents types d'attributs agrégés, combinée avec la possibilité d'utiliser des géométries composites ou alternatives. Ainsi, le langage est devenu plus expressif, le nombre de combinaisons exprimables à l'aide des pictogrammes est très important.

1.5.4 Modélisation spatio-temporelle avec AROM-ST

AROM-ST est un outil de modélisation spatio-temporelle basé sur le système de représentation AROM [Moisuc *et al.*, 2012]. C'est un outil générique, fondé sur les concepts de

classe et d'association pour représenter les objets et les liens entre les objets. La spécificité de AROM se situe dans l'utilisation d'un langage de modélisation algébrique (LMA). Celui-ci permet d'écrire des règles d'inférence dans un langage très proche du formalisme mathématique, de définir des variables à l'aide d'équations résolues par le système, d'appliquer des contraintes d'intégrité au niveau des classes ou associations, ou d'écrire des requêtes dans les bases de connaissances à l'aide d'un langage de requête (*AROM Query*).

Le modèle spatial dans AROM-ST

Les types spatiaux de AROM-ST (figure 1.18) sont conformes aux spécifications de OpenGIS [Percivall, 2003]. L'ensemble nécessaire et suffisant de types, pour pouvoir représenter les objets spatiaux, est composé des types simples : **Point**, **Polyline** et **Polygon**. Les types **Line** et **LinearRing** sont définis en appliquant des contraintes sur le type **Polyline**. À partir des types simples sont définis les types complexes : **MultiPoint**, **MultiLine** et **MultiArea**.

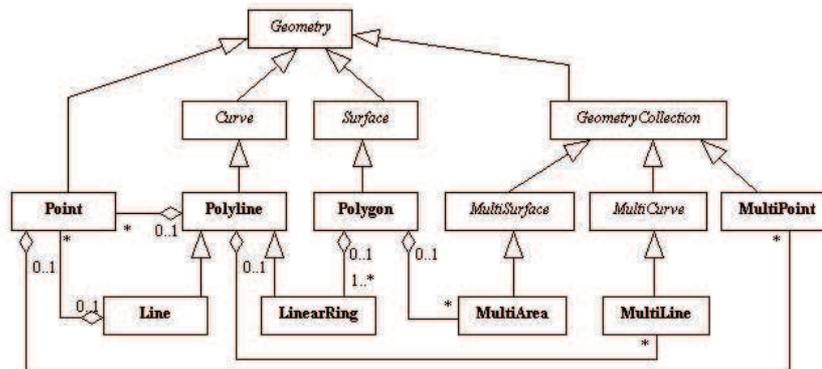


FIGURE 1.18 – Modèle spatial de AROM-ST

Le modèle temporel dans AROM-ST

Les types temporels simples sont le type **Instant** et le type **Interval**. Les types composés, constituant des collections d'instantanés ou d'intervalles, sont les types **MultiInstant** et **MultiInterval** (figure 1.19).

Les opérations spatio-temporelles dans AROM-ST

Les opérateurs LMA disponibles pour les types spatiaux et temporels sont divisés en trois catégories : les opérateurs topologiques, les opérateurs ensemblistes et les opérateurs de mesure. Les opérateurs topologiques sont des prédicats logiques binaires qui permettent de tester la position relative (spatiale ou temporelle) de deux objets. On considère les opérateurs topologiques définis par l'OpenGIS [Beddoe *et al.*, 1999]. Les opérateurs de

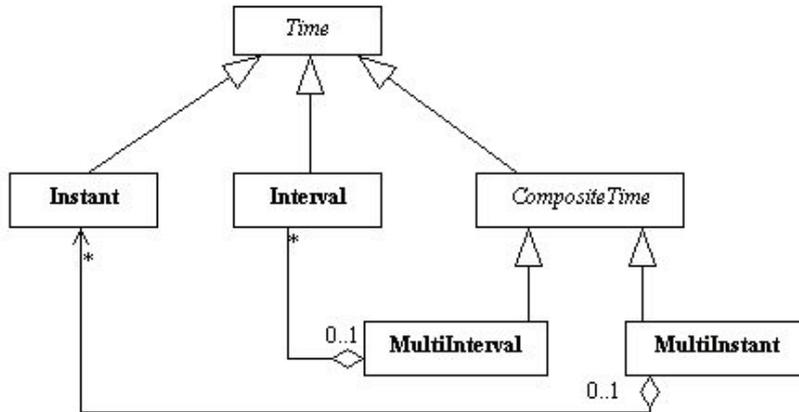


FIGURE 1.19 – Modèle temporel de AROM-ST

topologie temporelle considère les relations de l’algèbre d’Allen [Allen, 1983b]. Les opérateurs ensemblistes permettent de traiter l’espace et le temps, respectivement comme un ensemble de points et d’instant temporels. Pour plus de simplicité, ces opérateurs ont la même forme pour les dimensions spatiales et temporelles : `union`, `intersection`, `symmetricalDifference` et `difference`. Les opérateurs de mesure permettent une description quantitative de l’espace et du temps. Pour la dimension spatiale, les opérateurs définis sont `dimension` et `distance`, leurs correspondants pour la dimension temporelle sont `duration` et `timeLength`.

1.6 Synthèse

Ce chapitre nous a permis de présenter les modèles que nous jugeons les plus importants dans le domaine spatio temporel pour la gestion de l’espace, du temps et de l’espace-temps. Dans la première partie, nous avons étudié les différents types spatiaux et temporels utilisés pour les SIG, ainsi que les opérateurs pour le raisonnement spatial. Plus particulièrement, nous avons présenté le standard OGC SFS, dont le but est de définir un ensemble de types spatiaux de référence qui peuvent être utilisés par les concepteurs des applications spatio temporelles. À côté des modèles de l’espace, nous avons traité les modèles de temps, dédiés à la description des instants et des intervalles. On a cité le modèle standard de la norme ISO 8601, celui du W3C qui propose des types temporels par le schéma XML et le modèle d’Allen qui est le plus utilisé dans le raisonnement temporel. La théorie d’Allen est la principale source d’inspiration pour l’intégration et la réutilisation de l’ontologie de temps OWL-Time dans notre travail. De plus, nous allons utiliser le modèle de référence de l’OGC dans le chapitre 2 des contributions pour la définition d’une ontologie spatiale que nous utilisons dans le raisonnement sur des données spatio temporelles en particulier pour les trajectoires. La deuxième partie de ce chapitre traite des approches et des outils de modélisation spatio temporelles comme « MADS » et « Perceptory » qui utilisent des pictogrammes pour représenter l’espace et le temps pour la

conceptualisation des objets géospatiaux. Nous avons noté que « Perceptory » est un outil gratuit mais il nécessite le logiciel « Visio ». De plus, AROM-ST est un outil de modélisation basé sur le système de représentation de connaissances par objets AROM. Il intègre un langage de modélisation algébrique (LMA).

Chapitre 2

Les modèles de données des trajectoires

Sommaire

2.1	Introduction	45
2.2	Les trajectoires en sciences physiques et mathématiques	46
2.3	Les trajectoires en bases de données	47
2.3.1	Modèle MOST (Moving Objet Spatio-Temporel)	47
2.3.2	Modèle basé sur les types abstraits spatio-temporels	48
2.3.3	Modèle conceptuel de la trajectoire	49
2.4	Synthèse	51

2.1 Introduction

Le développement récent et croissant des moyens de capture des données géoréférencées a donné naissance à des recherches et des applications dont le point central sont les trajectoires. Les données des trajectoires sont souvent considérées par des modèles de données spatio-temporelles, et ne fournissent alors qu'une représentation spatiale et temporelle d'un point de vue général. Dans le chapitre précédant, nous avons présenté d'une manière globale les travaux de recherches pour les modèles spatiaux, temporels et spatio-temporels. Dans ce chapitre, nous voulons nous concentrer sur un nouveau type de donnée spatio temporelle qui est la trajectoire. Pour cela, nous allons introduire des modèles pour la représentation de la trajectoire dans sa globalité. Nous présentons les travaux des communautés de recherches s'intéressant à la modélisation physique et conceptuelle de trajectoire.

2.2 Les trajectoires en sciences physiques et mathématiques

La communauté des sciences physiques et mathématiques considèrent les trajectoires sous forme d'équation de mouvement de l'objet mobile [Geerts, 2004]. L'idée est d'utiliser la deuxième loi de Newton, qui montre que si on connaît le total de la force agissant sur un objet mobile, on connaît son mouvement. La Deuxième loi de Newton relie la force avec l'accélération par le biais de la masse de l'objet. L'équation du mouvement peut être décrite comme un ensemble fini de triplets où chaque triple contient un ensemble de contraintes initiales, un ensemble de contraintes différentielles et un temps de fin dans lequel le triple est valide [Geerts, 2004]. Dans les modèles à base de contraintes, les données sont représentées comme des objets mathématiques [Mokhtar *et al.*, 2002]. Par exemple un modèle où les positions des points mobiles sont représentées par une fonction linéaire ou linéaire par morceau ayant comme paramètre le temps et comme résultat un espace à n dimensions. Une trajectoire est définie comme une fonction continue linéaire par morceau de \mathbb{R} pour le temps dans \mathbb{R}^n et n pour la dimension. Dans l'exemple suivant, la trajectoire est définie dans un espace à trois dimensions :

$$\begin{aligned}
 & \tau = (2, -1, 0)t + (-40, 23, 30) \quad \wedge \quad 0 \leq t < 21 \\
 \vee & \quad \tau = (0, -1, -5)t + (2, 23, 135) \quad \wedge \quad 21 \leq t < 22 \\
 \vee & \quad \tau = (0.5, 0, -1)t + (-9, 1, 47) \quad \wedge \quad 22 \leq t \leq 23
 \end{aligned} \tag{2.1}$$

Dans [Su *et al.*, 2001], les auteurs se basent sur les méthodes de la géométrie différentielle afin de proposer un modèle pour les objets mobiles. Un point mobile en dimension n est une fonction du type $P : \text{time} \rightarrow \text{point}_n$ qui peut être représenté par un vecteur $P(t) = (p_1(t), \dots, p_n(t))$ qui satisfait les conditions suivantes :

- pour chaque $1 < i < n$, p_i est continue dans le temps ;
- il existe m instants $t_1 < t_2 < \dots < t_m$ tel que chaque p_i $1 < i < n$, est infiniment différentiable sur $] -\infty, t_1[$, chaque $]t_{j-1}, t_j[$, $]t_m, \infty[$.

Le vecteur vitesse de l'objet mobile $vel(P)$ correspond à la dérivée de P et le vecteur d'accélération $acc(P)$ est défini par la dérivée de la vitesse. La direction, la vitesse d'un objet et la distance entre deux objets mobiles sont respectivement définies par :

$$dir(P) = \frac{vel(P)}{\|vel(P)\|} \tag{2.2}$$

$$dist(P, Q) = \|P - Q\| \tag{2.3}$$

où $dist(P, Q)$ est la distance Euclidienne entre P et Q .

2.3 Les trajectoires en bases de données

Cette partie présente les travaux de recherche pour la modélisation des trajectoires d'objets mobiles d'un point de vue bases de données. Traditionnellement, ces approches considèrent l'objet trajectoire comme un ensemble de données spatio-temporelles de l'objet mobile. Deux verrous scientifiques sont étudiés : le langage de définition et le langage de manipulation de ces données. On parle alors de bases de données d'objets mobiles (en anglais : *Moving Objects Databases (MOD)*). Les recherches dans la MOD ont défini plusieurs approches pour modéliser la trajectoire. Nous nous intéressons à deux modèles principaux : le modèle de données MOST (*Moving Object Spatio-Temporel*) et le modèle de données fondé sur les types abstraits spatio-temporels.

2.3.1 Modèle MOST (Moving Object Spatio-Temporel)

Les systèmes de gestion de bases de données traditionnels (SGBD) ne sont pas équipés pour traiter les changements continus des données. En effet, ces SGBD considèrent les données comme constantes à moins que des mises à jour surviennent. La prise en compte des données spatio-temporelles implique forcément des mises à jour fréquentes d'où la nécessité de concevoir des systèmes qui peuvent considérer l'évolution continue des données. Dans [Wolfson *et al.*, 1998], les auteurs proposent un modèle de données appelé MOST (*Moving Object Spatio-Temporel*) pour résoudre le problème de changement spatial ou de localisation. Ce modèle représente l'évolution de la localisation en fonction du temps sans faire de mise à jour de façon explicite. Pour cela, on associe à un objet mobile un vecteur de mouvement qui peut être amené à changer mais ce changement est moins fréquent que le changement de la position. Le modèle de données MOST définit la notion d'attribut dynamique comme étant un attribut dont la valeur change continuellement avec le temps sans recourir à des mises à jour explicites de la base de données, par contraste avec un attribut statique. Un attribut dynamique peut changer dans le temps selon une fonction. Par conséquent, ce modèle classe la base de données en deux classes : la classe des objets ayant des attributs statiques, et celle des objets ayant des attributs dynamiques.

Dans [Sistla *et al.*, 1997a], les auteurs représentent les valeurs d'un attribut dynamique à des instants différents par une séquence constituée par des paires (*valeur, instant*) de la forme :

$$(u_1, t_1), \dots, (u_i, t_i), \dots, (u_n, t_n) \quad \text{tel que} \quad t_1 < t_2 < \dots < t_i < \dots < t_n \quad (2.4)$$

L'ensemble $\{t_i, 0 < i \leq n\}$ de l'équation 2.4 désigne les instants de capture des changements implicites des valeurs de l'attribut. Formellement, un attribut dynamique A est représenté par trois sous-attributs : $A.\text{initialvalue}$, $A.\text{updatetime}$ et $A.\text{function}$ ($A.f$). Cette fonction prend comme argument une séquence des paires (*valeur, instant*) et retourne l'ensemble des paires possibles de l'attribut dynamique décrivant ses valeurs dans le futur. À un instant donné, la valeur de $A.\text{updatetime}$ dépend de $A.\text{initialvalue}$ et sa valeur après une mise à jour est définie inductivement [Sistla *et al.*, 1997b].

Un attribut dynamique A est dit *déterministe* si l'ensemble des valeurs retournées par $A.f$ est un singleton. Un changement explicite d'un attribut dynamique peut changer ses sous-attributs sauf $A.updateTime$. Ce dernier a deux interprétations selon les instants validés ou les transactions temporelles. Dans la première interprétation, c'est l'instant où la mise à jour s'est produite dans le monde réel. À titre d'exemple, on peut considérer le cas d'un véhicule qui change son vecteur de mouvement et dont le capteur envoie à la base de données $A.updateTime$. Dans la seconde interprétation, ce sous attribut indique l'instant de validation de la transaction dans le SGBD.

Lorsqu'une requête porte sur un attribut dynamique, le résultat retourné par le SGBD donne la plage des possibles valeurs de l'attribut à l'instant de l'exécution de la requête. Les auteurs définissent alors trois types de requêtes :

- requête instantanée à un instant donné t : définie par rapport à l'ensemble des historiques possibles de la base de données ;
- requête continue à un instant donné t : la séquence des requêtes instantanées pour tout instant $t' \geq t$;
- requête persistante à un instant donné t : la séquence des requêtes instantanées à l'instant t . Les requêtes sont évaluées à chaque instant $t' \geq t$ lorsque la base de données est mise à jour.

Pour résoudre le problème des requêtes persistantes, les auteurs définissent le langage FTL (*Future Temporal Logic*) [Sistla *et al.*, 1997b] basé sur la logique temporelle pour formuler des requêtes prédictives utilisant des opérateurs temporels. Il permet des requêtes persistantes relatives aux états futurs du système pris en charge par un SGBD.

2.3.2 Modèle basé sur les types abstraits spatio-temporels

La nécessité de saisir l'historique complet des trajectoires des objets mobiles favorise le développement de nouvelles approches dans ce domaine. Comme il est clair que les données de localisation changent au cours du temps, la base de données doit décrire non seulement l'état actuel des données spatiales, mais aussi tout l'historique. Pour comprendre l'évolution des données et de les analyser, on doit pouvoir revenir en arrière dans le temps, à tout instant, pour récupérer l'état de la base de données. Dans [Erwig *et al.*, 1999; Forlizzi *et al.*, 2000; Güting *et al.*, 2000; Lema *et al.*, 2003], les auteurs développent cette nouvelle approche où les positions des points et des régions en mouvement sont considérées en trois dimensions : 2D pour l'espace et 1D pour le temps. Des entités de grandes dimensions dont la structure et le comportement sont modélisées en tant que types de données abstraits.

Ces types et leurs opérations sur les valeurs spatiales qui évoluent au cours du temps peuvent être intégrés comme des types de données dans un objet-relationnel, orienté-objet ou tout autre système de gestion de base de données extensibles. Les deux types de base sont : `mpoint` et `mregion`. Le point mobile `mpoint`, respectivement la région mobile `mregion`, est une fonction du temps dans l'espace des points spatiaux, respectivement des régions spatiales. Les auteurs proposent aussi la possibilité d'ajouter de la sémantique à ces fonctions. Dans [Dieker and Güting, 2000], les auteurs proposent une mise en œuvre de cette approche dans le cadre du SGBD SECONDO [Güting, 2007].

Dans [Pelekis *et al.*, 2006], les auteurs exploitent aussi ces travaux et réalisent un framework, appelé *Hermes*, pour la manipulation des objets mobiles dans le SGBD Oracle. Ce cadre fournit des fonctionnalités spatio-temporelles pour ce SGBD et une infrastructure pour la gestion des données historiques des MODs [Pelekis *et al.*, 2005]. Un ensemble de types de données et leurs opérations sont implémentés dans ce SGBD sous le nom *Hermes Moving Data Cartridge (Hermes-MDC)*. La figure 2.1 donne le diagramme de classes du modèle de données de *Hermes-MDC*.

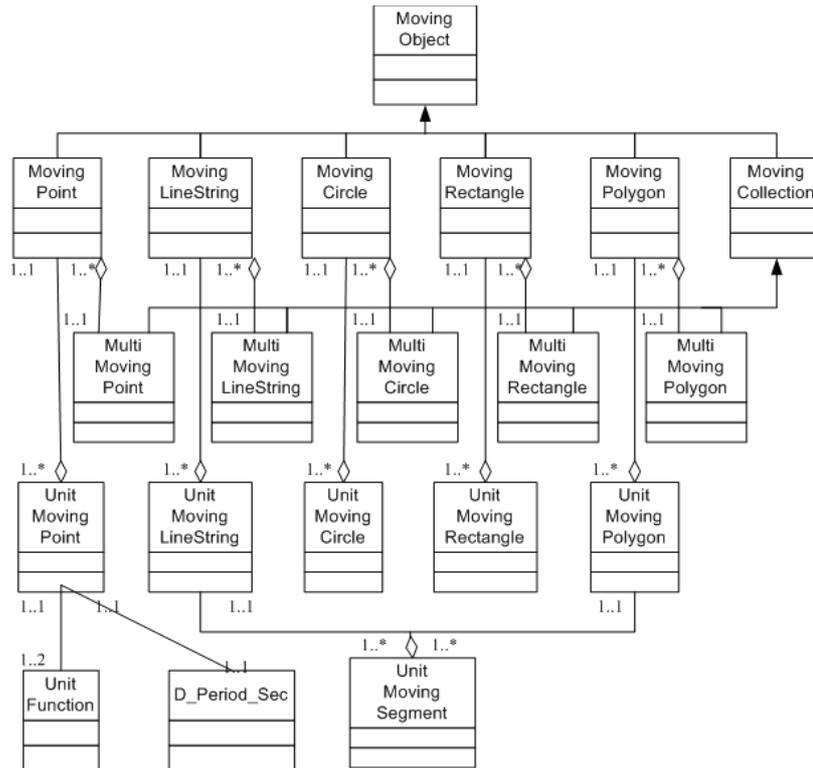


FIGURE 2.1 – Modèle de données de *Hermes-MDC*

2.3.3 Modèle conceptuel de la trajectoire

Dans le cadre du projet européen GeoPKDD⁵, un travail important a été réalisé pour la spécification et l'implémentation des objets mobiles et des entrepôts de données de trajectoires. En effet, dans [Spaccapietra *et al.*, 2008a] les auteurs ont proposé un modèle conceptuel formalisé par deux approches représentant la trajectoire comme une unité sémantique, dépendante du domaine applicatif. Ce modèle fournit aux différentes applications un support direct des trajectoires comme un concept de base. Il a été élaboré à partir d'une analyse profonde des besoins du point de vue d'application (ex : la migration des oiseaux) pour aboutir à une représentation sémantique des mouvements des objets. Dans cette application, on observe la migration des oiseaux et on collecte les données

5. <http://www.geopkdd.eu>

envoyées par les capteurs des Cigognes blanches. L'analyse de la migration des oiseaux est la clé pour améliorer les connaissances sur les comportements des animaux. Durant leur migration, les chercheurs enregistrent les données des trajectoires des animaux durant leur mouvement et leur stationnement et les données sur les conditions environnementales. Les trajectoires sont segmentées en définissant une séquence temporelle de sous intervalles de temps où les positions des objets changent (**move**), et où elles restent fixe (**stop**). Par conséquent, la trajectoire est vue comme une séquence de mouvements **moves** allant d'un arrêt **stop** au suivant arrêt ou comme une séquence d'arrêts séparant les mouvements. Ces composantes d'une trajectoire sont définies comme suit :

- **Stop**
 - une partie de la trajectoire définie par l'utilisateur de l'application pour représenter un arrêt déterminé par un intervalle $[t_{beginstopx}; t_{endstopx}]$;
 - l'arrêt a une durée limité par deux bornes temporelles. Les bornes de deux stops sont disjointes ;
 - le mouvement de l'objet mobile est suspendu. Le domaine spatial de la trajectoire pour l'intervalle $[t_{beginstopx}; t_{endstopx}]$ est un seul point.
- **Move**
 - une partie de la trajectoire limité par deux extrémités qui représentent deux arrêts consécutifs, ou un point de début et le premier arrêt, ou entre le dernier arrêt et le dernier point ;
 - la durée temporelle est définie par un intervalle $[t_{beginmovex}; t_{endmovex}]$;
 - le domaine spatial de la trajectoire pour l'intervalle $[t_{beginmovex}; t_{endmovex}]$ est une ligne spatio-temporelle (pas un point) définie par la fonction de la trajectoire.
- **Begin-End**
 - se sont les deux extrémités de la trajectoire. Chacune est définie par le couple formé du point spatial et d'un instant.

À partir de la définition des besoins intrinsèques et extrinsèques de la trajectoire, les auteurs proposent une modélisation conceptuelle basée sur MADS [Parent *et al.*, 1997] qui supporte les objets et les relations spatiales et temporelles. Cette solution utilise un patron conceptuel pour donner une représentation explicite de la trajectoire et ses composants (**stops**, **moves**, **begin** et **end**). Ce patron conceptuel, donné par la figure 2.2, est un schéma générique qui peut être importé par le concepteur, personnalisé et adapté à différentes applications. Ce modèle a été exploité dans [Alvares *et al.*, 2007a; Alvares *et al.*, 2007b] en intégrant l'information géographique aux composantes de la trajectoire.

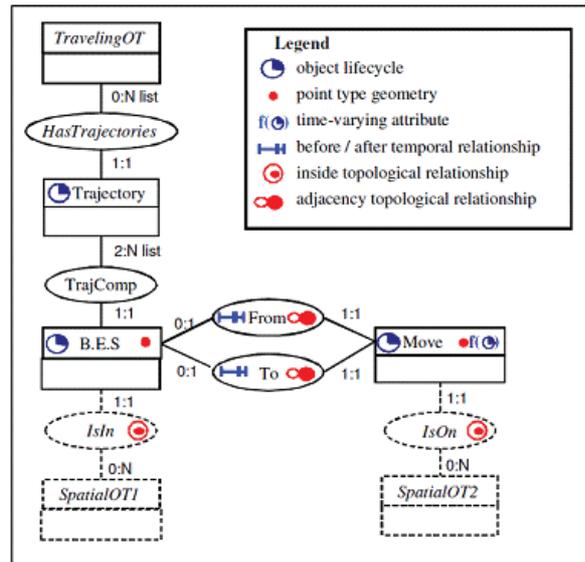


FIGURE 2.2 – Patron basé sur MADS pour les trajectoires

2.4 Synthèse

Ce chapitre porte sur la notion de trajectoire qui est le contexte général de nos travaux. Nous avons passé en revue les travaux que nous jugeons les plus importants dans les différentes communautés de recherche, qui proposent des modèles physiques et conceptuelles pour la trajectoire. La notion de trajectoire est associée au thème général **objets mobile** ou en anglais *Moving Objects*. Des contributions majeures s'intéressent à la définition de nouveaux types de données pour représenter l'objet mobile et ses caractéristiques dans un SGBD. O. Wolfson et al. [Wolfson *et al.*, 1998] identifient clairement ces problèmes et proposent des solutions. Ils introduisent notamment la notion d'attribut dynamique pour représenter le changement avec le temps de la localisation des objets mobiles. Grâce à cette notion, ils définissent ainsi un modèle de données appelé MOST (Moving Objects Spatio-Temporal). Pour formuler des requêtes basées sur MOST, les auteurs définissent un langage appelé FTL (Future Temporal Logic). Les travaux de O. Wolfson et al. représentent la trajectoire d'un objet mobile sous forme de vecteur de mouvements qui change de position en fonction du temps. Par conséquent, la trajectoire d'un objet mobile n'est pas totalement définie. Les types de données et les opérations ne sont pas définies formellement. De notre point de vue, les travaux de R. H. Güting constituent un fondement pour ce thème [Güting *et al.*, 2000]. Ces travaux définissent formellement les types de données nécessaires à la représentation d'un objet mobile sous forme de points ou de régions. Ils définissent en détail les considérations à prendre en compte pour la conception et donnent des exemples pour l'utilisation des opérateurs en SQL. En particulier, dans [Güting *et al.*, 2000] les auteurs proposent une méthode conceptuelle précise en vue d'une implémentation dans un SGBD. Plusieurs travaux fondés sur ceux de O. Wolfson et de R. H. Güting voient le jour, notamment celui de Pelekis et al. [Pelekis *et al.*, 2006] qui

définit une implantation nommée « HERMES » au sein du SGBD Oracle. L'exploitation des trajectoires conjointement avec des données sémantiques fait l'objet de travaux de recherche actifs. Le principe même de trajectoire sémantique n'est défini qu'après dans [Spaccapietra *et al.*, 2008b]. Cet intérêt s'explique par le développement des moyens de capture de données réelles. Ce modèle a été exploité par [Alvares *et al.*, 2007a] en intégrant l'information géographique à travers les *move* et les *stops*. Ce modèle a été proposé à partir de la définition générale de la trajectoire. Dans notre travail, nous allons établir un patron pour la gestion des trajectoires à partir de différentes applications.

Chapitre 3

Approche ontologique pour les données spatiales, temporelles et spatio-temporelles

Sommaire

3.1	Introduction	54
3.2	Les ontologies traditionnelles	55
3.2.1	Les différents types d'ontologies	55
3.2.2	La réutilisabilité des ontologies	56
3.2.3	Les langages de représentation et d'interrogation	57
3.2.4	Les environnements de construction	60
3.2.5	Le peuplement de l'ontologie	61
3.2.6	Les outils de raisonnement	64
3.3	Approche ontologique : cas des données spatiales	65
3.3.1	Ontologies spatiales : extension de la logique de description	65
3.3.2	Ontologies spatiales	66
3.3.3	Raisonnement spatial dans les ontologies	67
3.4	Approche ontologique : cas des données temporelles	70
3.4.1	Ontologies temporelles : extension de la logique de description	71
3.4.2	Ontologies temporelles	71
3.4.3	Raisonnement temporel dans les ontologies	73
3.5	Approche ontologique : cas des données spatio-temporelles	74
3.5.1	Cas des données géographiques	74
3.5.2	Cas des bases de données	75
3.6	Synthèse	76

3.1 Introduction

Les données doivent être récupérées, représentées et réinterprétées. Elles sont transmises à un système ou un programme qui les traite, les modifie et les fait évoluer. Cependant, elles ne sont pas associées à un sens. Pour cela, des connaissances sont nécessaires pour la compréhension du contenu des données en apportant une couche sémantique aidant à leur interprétation. Elles peuvent également aider à la prise de décision de l'utilisateur, en reposant en particulier sur l'exploitation des connaissances déduites par des mécanismes d'inférence appliqués sur les connaissances déclaratives qui sont associées aux données. Enfin, elles peuvent également être utiles dans la compréhension du besoin de l'utilisateur aussi bien par l'utilisateur lui-même que par le système. Elles permettent de donner une vue générale sur les connaissances disponibles dans les données et peuvent aider à spécifier le besoin d'information d'un utilisateur. Dans ce contexte, notre choix s'est porté sur la représentation des connaissances à travers des ontologies. Plus spécifiquement, les ontologies que nous considérons sont des ontologies de domaine, car elles permettent de représenter la connaissance normalisée pour un domaine d'intérêt. Un des intérêts supplémentaires de notre choix réside dans le fait que les ontologies correspondent à des ressources conceptuelles formalisant un degré de connaissance plus élevé qu'une ressource terminologique. Plus spécifiquement, nous générons deux types d'ontologies via une transformation de modèle UML vers une modélisation ontologique : l'une représente le domaine générique de la trajectoire et l'autre correspond au domaine d'application. De plus, au cours de ces dernières années, nous avons constaté un grand intérêt pour l'introduction de l'approche ontologique dans la représentation et l'exploitation des données spatiales, temporelles ou spatio-temporelles. L'engouement pour cette nouvelle approche s'explique par deux points importants :

1. l'utilisation des concepts et des relations formalisés en langage naturel pour définir et caractériser les propriétés spatiales, temporelles ou spatio-temporelles des objets. Ces concepts et ces relations permettent de créer une ontologie du domaine ;
2. le raisonnement spatial ou temporel exploite une sémantique proche du langage naturel compris par la machine. Il peut donc constituer ou faire partie d'un processus d'inférence.

Les ontologies de l'espace et du temps sont en cours de normalisation. À titre d'exemple, les ontologies utilisées par les systèmes d'information géographiques. Au contraire, les ontologies spatio-temporelles sont à leurs débuts, notamment en raison de l'absence d'un modèle approprié, capable de traiter l'espace et le temps, au niveau ontologique, et d'un moteur de raisonnement adapté. Historiquement, les recherches dans le domaine de l'ontologie appliquée à la gestion des données ont été initié par les communautés de l'intelligence artificielle et de représentation des connaissances. Ces communautés s'appuient essentiellement sur la logique pour établir le raisonnement. Pour cette raison, les logiques de description occupent une grande place dans la famille des formalismes pour le développement des modèles ontologiques. Toutefois, les limites des approches basées sur les logiques de description vont contribuer à la naissance d'autres formes de logique plus expressives telles que F-Logic (Logique Frame) et Horn-Logic. Les langages basés sur la logique offrent des possibilités intéressantes pour la construction des ontologies mais leurs

mis en œuvre restent complexes. Pour cette raison, des langages alternatifs vont être développés comme le OWL (Ontology Web Language). L'objectif de la première partie de ce chapitre est de positionner l'utilisation des ontologies par rapport au domaine spatio-temporel. Ainsi, la deuxième partie de ce chapitre présente quelques travaux significatifs concernant l'approche ontologique pour la représentation et le raisonnement spatial, temporel et spatio-temporel.

3.2 Les ontologies traditionnelles

Gruber [Gruber, 1993] introduit la notion d'ontologie comme « an explicit specification of a conceptualization ». Nous adoptons quant à nous la définition suivante : une ontologie est une **spécification formelle consensuelle et référençable d'une conceptualisation d'un domaine**.

- Le terme **conceptualisation** fait référence à un modèle (au sens d'ensemble structuré) de concepts.
- Le terme **formelle** signifie que l'ontologie est représentée sous une forme traitable en machine et que cette description permet de rendre automatique certains traitements.
- Le terme **consensuelle** signifie que l'ensemble des membres d'une communauté se sont mis d'accord sur les concepts définis dans l'ontologie. Ces concepts pourront donc être utilisés pour se comprendre.
- Le terme **référençable** signifie que les différents concepts définis dans l'ontologie peuvent être référencés à partir de tout type d'application indépendamment du modèle de définition de l'ontologie.

3.2.1 Les différents types d'ontologies

3.2.1.0.1 typologie basée la structure de la connaissance Les concepts d'un domaine sont associés à un vocabulaire de termes pour chaque langue naturelle. En fait, certaines ontologies que nous qualifions de terminologiques ou linguistiques visent à définir des mots dans une langue donnée plus que des concepts. Cette constatation permet de classer les ontologies en deux principales catégories [Bellatreche *et al.*, 2004] :

- Les ontologies linguistiques qui définissent non pas seulement des mots, mais également les relations qui les unissent et les règles qui leurs sont associées, telles que la synonymie ou l'antinomie. Les termes peuvent être introduits comme des concepts primitifs ou définis. Le nombre de ces termes définis est généralement très important. Un exemple très connu d'une telle ontologie est WordNet.
- Les ontologies conceptuelles ou formelles visent à définir des concepts indépendamment d'une langue particulière. Le domaine d'étude est appréhendé au travers de concepts représentés par des classes et des propriétés. Des mots d'un langage naturel peuvent être associés mais ils ne définissent pas le sens des concepts. C'est l'ensemble des caractéristiques associées à un concept ainsi que ses liens avec les autres concepts qui en définissent le sens.

3.2.1.0.2 typologie basée le contenu de la connaissance Un autre critère pour la classification des ontologies est le contenu de la connaissance qu'elles représentent, c'est-à-dire le sujet de la conceptualisation [Guarino, 1998].

- Les ontologies génériques définissent des concepts considérés comme génériques à plusieurs domaines. WordNet [Miller, 1998] par exemple est une ontologie dont le but est de représenter la langue naturelle anglaise. WordNet est un système de références lexicales dont la conception a été inspirée par les théories de la mémoire linguistique humaine. WordNet couvre le domaine de la langue générale en intégrant le sens des mots dans différents domaines. Nous partageons le point de vue de Charlet [Charlet, 2002] suivant lequel la limite de ces ontologies générales est leur difficile réutilisation car elles ont pour objectif de recouvrir tous les sens des mots et ne normalisent pas leur sens.
- Les ontologies de domaine sont des conceptualisations spécifiques à un domaine particulier. La connaissance du domaine décrit des situations factuelles du domaine alors que l'ontologie pose des contraintes sur la structure et le contenu de la connaissance du domaine. Comparées aux ontologies génériques, les ontologies de domaine ont pour avantage de permettre une normalisation des concepts dans le cadre du domaine considéré et donc, selon nous, de permettre une meilleure représentation de la connaissance. L'ontologie OWL-Time est un exemple d'ontologie de domaine temporel, celui des données, rassemblant des concepts et leurs relations structurées à partir de la relation **sorte de**. Nous considérons que l'ontologie de domaine est facile à la réutiliser car elles ont pour objectif de normaliser des concepts dans le cadre d'un domaine considéré.
- Les ontologies d'application contiennent toutes les définitions qui sont nécessaires pour modéliser la connaissance propre à l'élaboration d'une tâche ou un contexte particulier. Généralement, les ontologies d'application combinent des éléments d'ontologies de domaine et d'ontologies génériques choisies pour modéliser le contexte visé. Elles sont rarement réutilisables pour une autre application.

L'objectif de nos travaux étant de construire des ontologies liées à un domaine de connaissance donné, les ontologies que nous considérons dans la suite du mémoire sont des ontologies de domaine et des ontologies d'application.

3.2.2 La réutilisabilité des ontologies

La majorité des approches visant à intégrer une ontologie dans leur procédé reposent sur des ontologies existantes [Vallet *et al.*, 2005] [Baziz *et al.*, 2005]. Généralement, l'unique caractéristique prise en compte dans le choix de l'ontologie est le domaine de connaissance représentée dans l'ontologie qui doit couvrir le domaine traité dans les données et les descriptifs de ces données. La construction d'ontologies réutilisables est le but affiché d'un certain nombre de travaux [Lopez *et al.*, 1997] [Uschold *et al.*, 1998] [Pinto and Martins, 2000]. Cependant, de nombreux auteurs considèrent que les ontologies sont non réutilisables. Bachimont affirme en effet que par leur méthode de construction et les travaux épistémologiques qui les supportent, leur réutilisation est impossible [Bachimont, 2004]. De la même façon, Charlet [J. *et al.*, 2000] considère que **les ontologies sont des arte-**

faits construits en fonction d'une tâche précise et ne peuvent être réutilisées, en tant qu'objet formel, pour une autre tâche.

Nous partageons le point de vue de Furst [Fürst, 2004] selon lequel les ontologies sont destinées à être réutilisées. La sémantique qu'elles représentent est liée au cadre applicatif à partir duquel le sens des termes et concepts est défini. Cependant, la représentation ne dépend pas de l'opération faite avec l'ontologie. La sémantique de l'ontologie est liée au contexte mais la représentation n'implique pas que l'ontologie soit utilisée uniquement dans le contexte de sa création.

3.2.3 Les langages de représentation et d'interrogation

Différents langages de spécification d'ontologies sont apparus à partir des années 1990, tels que CycL et KIF [Genesereth *et al.*, 1992], F-Logic [Kifer *et al.*, 1995]. Nous nous concentrons dans cette section sur les langages orientés Web Sémantique. La raison de ce choix est que ces langages sont ou ont été pour la plupart recommandés par le W3C.

- **XML** [Bradley, 1998] est un langage permettant de générer des balises pour la structuration de données et de documents. Il permet la représentation et l'échange de documents semi-structurés. XML-schéma [Fallside, 2001] permet de définir la structure, les contraintes, et la sémantique de documents XML. Ce langage n'est pas vu comme un langage d'ontologies car il a été créé pour vérifier la structure de documents XML. Les primitives qu'il met en place sont plutôt orientées application que concept. En effet, la sémantique définie dans le document est interprétable dans le contexte de l'opération faite sur le document mais ne permet pas d'établir des inférences en dehors de ce contexte. XML et XML-schéma sont considérés comme des langages définissant le format de **message** alors qu'un langage d'ontologies a pour but de **représenter** la connaissance.
- **RDF** [Lassila and Swick, 1999] permet d'encoder, d'échanger et de réutiliser des méta-données structurées. Il a été créé pour gérer les méta-données de documents XML mais peut également être utilisé pour des ontologies. Il permet de définir des ressources avec des propriétés et des états. RDF-Schéma définit les relations entre ces ressources. Le pouvoir sémantique de ces deux langages est limité car les axiomes ne peuvent pas être directement décrits. Le type des relations (symétrique, transitive) ne peut être spécifié.
- **OIL** (Ontology Inference Layer) [Fensel *et al.*, 2000] est à la fois un langage de représentation et d'échange pour les ontologies. Il combine les primitives des langages reposant sur les frames avec une sémantique formelle et des possibilités de raisonnement issues de la logique de description. Pour être utilisé sur le Web, il repose sur les standards RDF(S) et XML. La description de l'ontologie est divisée en trois couches : la couche objet (instances concrètes), la couche de premier méta-niveau (définition de l'ontologie) et la couche de second meta-niveau (définition des caractéristiques de l'ontologie). OIL permet de définir des classes et des relations et un nombre limité d'axiomes. Les relations sont considérées comme des classes et peuvent être organisées hiérarchiquement.

- **DAML+OIL** [Frank van Harmelen and Peter F. Patel-Schneider and Ian Horrocks, 2001] a été proposé par le W3C pour représenter des méta-données et des ontologies. DAML a été transformé en DAML+OIL en intégrant certaines propriétés de OIL [Fensel *et al.*, 2000]. Il repose sur RDF et RDF schéma et fournit en plus des primitives plus riches issues de la logique de description. Les frames définis dans OIL ont été pour la plupart supprimées et remplacées par les assertions faites à l'aide d'un ensemble limité d'axiomes. Le résultat est que le langage est mieux adapté que RDF à l'utilisation et la maintenance d'ontologies mais présente des limites quant à la construction d'ontologie [Bechhofer *et al.*, 2003].
- **OWL Ontologie Web Language** [McGuinness and v. Harmelen, 10 February 2004] est le standard actuellement proposé par le W3C pour représenter les ontologies. Il a été créé pour être utilisé par les applications cherchant à traiter le contenu de l'information et non plus uniquement à présenter l'information. OWL se veut plus représentatif du contenu du Web que XML, RDF et RDF-Schéma en apportant un nouveau vocabulaire avec une sémantique formelle. OWL est une révision de DAML+OIL définie d'après l'expérience acquise lors de la création et l'utilisation de ce langage. OWL ajoute du vocabulaire pour décrire les propriétés et classes, comme par exemple la disjonction de classe, la cardinalité (exactement un), l'égalité, les types de propriétés plus riches, les caractéristiques de propriété (symétrie, transitivité) et les classes énumérées. OWL est décliné en trois sous langages d'expressivité croissante : OWL lite, OWL DL, OWL Full. OWL Lite est fait pour des besoins préliminaires permettant de définir une hiérarchie et des contraintes simples. Il permet de définir facilement des thésaurus ou taxonomies. OWL DL et Full reposent sur OWL Lite auquel sont ajoutés des constructeurs supplémentaires. OWL DL supporte des besoins d'expressivité maximaux tout en garantissant une complétude de calculs et de décidabilité nécessaires aux systèmes de raisonnement. Il repose sur les éléments OWL auxquels il associe un grand nombre de restrictions (par exemple, une classe peut être une sous-classe de nombreuses autres classes, mais pas une instance d'une classe). OWL DL est conçu pour pouvoir supporter la logique de description. Cette logique appartient à un domaine de recherche qui a pour but d'aider au raisonnement sur une base de connaissances.

3.2.3.0.2.1 Les logiques de description reposent sur trois notions de base : les concepts représentant des classes (ensemble d'objets), les rôles (relations liant deux objets) et les individus (objets représentant les classes qu'ilsinstancient). Pour décrire ces éléments, deux structures sont utilisées : la T-BOX et la A-BOX. La T-BOX (boîte terminologique) comprend la description des concepts et des rôles. Cette description est structurée à l'aide du lien hiérarchique sorteDe. Deux concepts particuliers figurent au minimum dans la T-BOX : le concept le plus générique (anything) et le concept le plus spécifique (nothing). La A-BOX (boîte assertionnelle) est constituée des individus, de leur description et des règles qui leur sont attachés. Les inférences reposent sur la reconnaissance d'instances de concepts à partir de leur définition, la détection des concepts plus généraux ou plus spécifiques, et la classification ordonnant les concepts dans la hiérarchie.

- **OWL Full** permet un maximum d'expressivité avec la liberté de syntaxe d'RDF. Il n'impose pas de séparation entre classe, propriété, individu et valeur des données. Il permet donc d'augmenter le sens du vocabulaire pré-défini (en OWL ou RDF). Il lève les contraintes imposées par OWL DL pour rendre certaines valeurs disponibles et utilisables dans des bases de données ou de connaissances, mais il ne supporte pas les raisonnements liés à la logique de description.

3.2.3.0.3 SPARQL : langage d'interrogation Les documents RDF peuvent être interrogés grâce au langage SPARQL (SPARQL Protocol and RDF Query Language) qui constitue la recommandation officielle du W3C [Prud'hommeaux and Seaborne, 2008]. Ce langage permet l'interrogation des triplets des documents RDF et ne réalise aucune inférence. Les requêtes de ce langage contiennent généralement un ensemble de modèles de triplets appelé modèle de graphe élémentaire (basic graph pattern en anglais). Ces modèles de triplets ressemblent à des triplets RDF, à ceci près que le sujet, le prédicat et l'objet peuvent être des variables. Un modèle de graphe élémentaire correspond à un sous-graphe du graphe RDF sur lequel s'applique la requête SPARQL lorsque les termes RDF de ce sous-graphe peuvent être substitués aux variables et le résultat est un graphe RDF équivalent à ce sous-graphe. Les variables de SPARQL sont identifiées par le préfixe ? ou . Le modèle de graphe élémentaire est présent dans une clause WHERE et peut ainsi être vu comme une condition à satisfaire. L'exemple suivant est une requête SPARQL permettant de retrouver tous les noms des personnes présentes dans les données RDF fournies. Le préfixe foaf est utilisé pour identifier la propriété de nom et est spécifié au début de la requête au moyen du mot-clé PREFIX. Le résultat d'une telle requête est une séquence de solutions, comprenant de zéro à n solutions en fonction des données.

```

1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2 SELECT ?name
3 WHERE f ?x foaf:name ?name. g

```

Ce type de requête, avec l'opérateur SELECT, est une requête dite interrogative dans la mesure où elle extrait du graphe RDF un sous-graphe correspondant à un modèle de graphe élémentaire. Il existe également des requêtes constructives, exprimées avec l'opérateur CONSTRUCT, qui créent un nouveau graphe construit sur base d'un template et des triplets résultats de la correspondance avec le modèle de graphe élémentaire. La requête suivante construit un nouveau graphe par extraction des noms des employés et en utilisant la propriété foaf :name au lieu de la propriété org :employeeName. Ainsi, si le document RDF sur lequel la requête est exécutée comprend les triplets (A, org :employeeName, Alice) et (Aorg :employeeId, 12345), le graphe RDF construit par la requête comprendra le nouveau triplet (A, foaf :name, Alice).

```

1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2 PREFIX org: <http://example.com/ns\#>
3 CONSTRUCT f ?x foaf:name ?name g
4 WHERE f ?x org:employeeName ?name. g

```

Le langage SPARQL permet également de réaliser des correspondances en utilisant des littéraux ou des valeurs numériques dans le modèle de graphe élémentaire. Il est également possible de restreindre les résultats à certaines valeurs en utilisant des filtres. La requête suivante réalise un filtre sur le salaire d'un employé et renvoie un graphe RDF comprenant le nom et le salaire des employés.

```
1 PREFIX org: <http://example.com/ns\#>
2 SELECT ?name ?salary
3 WHERE
4 f
5 ?x org:employeeName ?name.
6 ?x org:salary ?salary.
7 FILTER (?salary > 10000)
8 g
```

Enfin, notons qu'il est possible d'appliquer des modificateurs à la séquence de solutions obtenues. Cette séquence étant non ordonnée, le modificateur Order By permet de l'ordonner suivant une expression donnée. L'exemple ci-dessous ordonne les résultats suivant le nom des personnes. Le modificateur Distinct permet de s'assurer que les solutions de la séquence sont uniques tandis que le modificateur Limit est utilisé afin de restreindre le nombre de solutions.

```
1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2 SELECT ?name
3 WHERE f ?x foaf:name ?name. g
4 ORDER BY ?name
```

3.2.4 Les environnements de construction

Différents outils ont été proposés pour aider à la conception manuelle d'ontologies. Ces outils permettent d'éditer une ontologie, d'ajouter des concepts et des relations, etc. Ils intègrent différents langages de formalisation (RDF, OWL). Certains doivent être installés en local alors que d'autres sont distribués sur le Web. Ces outils sont décrits plus spécifiquement.

- **OntoEdit** (Ontology Editor) [Sure *et al.*, 2002] 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.
- **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 [Noy *et al.*, 2000]. Le modèle de connaissances de Protégé2000 est issu du modèle des frames et contient des classes (concepts), des slots (propriétés) et des facets (valeurs des propriétés et contraintes), ainsi que des instances des classes et des propriétés. Protégé2000 autorise la définition de méta-classes, 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.

- L'outil ODE , Ontology Design Environment, développé à l'Université de Polytechnique de Madrid permet de mettre en place la méthodologie METHONTOLOGY décrite dans la section 2. Son successeur pour le Web WebODE [Vega *et al.*, 2003] a pour ambition de couvrir l'ingénierie ontologique à travers les différentes activités liées au cycle de vie d'une ontologie : acquisition de connaissances à partir du Web, édition d'ontologies, test de la consistance d'une ontologie, alignement et fusion d'ontologies, import et export dans des formats variés. Le modèle de représentation de connaissances utilisé associe un modèle de type frame (concepts et attributs) avec des relations entre concepts. Des propriétés conceptuelles (en particulier algébriques) peuvent être associées aux relations. Les axiomes d'une ontologie sont des tautologies du domaine, mais on peut aussi inclure dans l'ontologie des règles susceptibles d'être utilisées pour raisonner dans un moteur d'inférence de type Prolog.
- 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 [Domingue, 1998]. Le modèle de connaissance utilisé est celui du langage OCML (Operational Conceptual Modeling Language), un langage à base de Frames.
- OilEd (Oil Editor) est un éditeur d'ontologies utilisant le formalisme DAML+OIL et les Logiques de Description [Bechhofer *et al.*, 2001]. 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.
- ONTOLINGUA , développé au Knowledge Systems Laboratory de l'Université de Stanford, est un serveur d'édition d'ontologies permettant la construction collaborative d'ontologies [Farquhar *et al.*, 1997]. Une ontologie y est directement exprimée dans un formalisme également nommé ONTOLINGUA.
- Le framework Jena est réalisé en Java et fournit des raisonneurs, permettant ainsi de réaliser de l'inférence à partir de données RDF. Le JRDF a été développé dans ce même langage et ces deux frameworks supportent la sérialisation RDF/XML. CubicWeb permet le développement d'applications dans le langage Python et utilise le langage RQL, similaire au langage SPARQL, afin de réaliser l'interrogation de graphes RDF.

3.2.5 Le peuplement de l'ontologie

Le peuplement de l'ontologie est le processus d'insertion des instances de concepts et des instances de relations dans une ontologie existante. Dans une vue simplifiée, une on-

tologie est considérée comme un ensemble de concepts, des relations entre les concepts et des instances de concepts. La source de ces instances sont variées, notamment des informations extraites d'un corpus de documents ou des données qui sont sauvegardées dans le schéma de la base de données. Dans cette thèse, nous nous intéressons au peuplement à partir des bases de données. La technique de peuplement d'une ontologie s'appuie sur des approches de Correspondance Database Ontology Mapping ou RDB2RDF. Une étude de ces approches actuelles est étudiée dans [Sahoo *et al.*, 2009], [Konstantinou *et al.*, 2008] et [Hert *et al.*, 2011]

3.2.5.1 Les outils

Dans le cadre des projets, la communauté de RDB2RDF a réalisé des outils de Mapping parmi lesquels on cite :

- **Clio** [Haas *et al.*, 2005; Yan *et al.*, 2001] est un outil qui conduit à faire des correspondances d'un ensemble de tables relationnelles et / ou des données XML à un autre. Mais, avec des modifications mineures, il pourrait être appliqué aussi à un schéma d'ontologie . Le processus de correspondance prend comme entrée les correspondances entre les schémas sources et les schémas cibles et génère un schéma constitué d'un ensemble des correspondances logiques. A tout moment lors de la conception, l'utilisateur peut visualiser, ajouter et supprimer des correspondances entre les schémas grâce à un composant d'interface graphique et peut inspecter et modifier les correspondances logiques générés .
- **D2R MAP** a été présentée au début dans [Bizer, 2003]. Il fournit des moyens de mapping de l'ontologie à la base de données d'une façon déclarative. D2R MAP est basée sur le syntaxe XML et constitue une des langues visées à attribuer les concepts de l'ontologie au schéma de la base de données. Il permet le Mapping de structures complexes relationnelles à des ontologies OWL/RDFS, en utilisant des instructions SQL dans les règles de Mapping, sans modifier le schéma de base de données existante. Plus précisément, Le Mapping est représenté par un élément **ClassMap**, contenant des attributs tels que SQL (l'instruction SQL décrivant l'ensemble de données), et les attributs **groupBy** **uriPattern** pour la création d'instances ontologiques. Les résultats sont extraites en RDF, N3, RDF ou les modèles Jéna [Carroll *et al.*, 2004]. D2R MAP peut gérer des structures de tables très normalisés. Cependant, il n'arrive pas à mapper des bases de données faiblement structurées en raison de son expressivité limitée.
- **eD2R** L'étude de cas décrit dans [Barrasa *et al.*,] utilise eD2R pour le Mapping RDB-à-RDF. eD2R est une extension de D2R MAP dans le but de couvrir des situations de cartographie impliquant des bases de données qui sont légèrement structurées ou non sous la première forme normale. Le mapping est basé sur des requêtes SQL qui extraient les enregistrements de la BDR et les fonctions de transformation qui peuvent être appliquées aux valeurs extraites.
- **Relational.OWL** Les travaux de [de Laborda *et al.*, 2005] présentent un format de représentation basée sur OWL pour les données relationnelles et les composants de schéma, appelé Relational.OWL. Il définit une ontologie en OWL Full pour décrire

le schéma et les données d'une BDR.

- **Triplify** [Auer *et al.*, 2009] est une approche légère pour publier **Linked Data** à partir des BDRs. Il est basé sur le **mapping** des requêtes HTTP-URI en requêtes BDR et la traduction des relations résultants en des déclarations RDF. La principale motivation de **Triplify** est que la majorité des informations sur le Web sont déjà stockées sous une forme structurée (en tant que des données dans BDR), mais publié en HTML par des applications Web (par exemple Wiki, Blog). Le **mapping** de **Triplify** est mis en œuvre sous forme de scripts PHP.
- **Virtuoso RDF Views** Le serveur Virtuoso universelle offre une vue RDF [Erling and Mikhailov, 2009] pour présenter les données relationnelles sur le Web sémantique. Il se compose d'un langage de métaschéma déclaratif pour définir le **mapping** des données SQL en un vocabulaire RDF préexistant. Au bas niveau, virtuose les vues RDF de virtuose transforment les résultats d'une requête SQL SELECT en un ensemble de triplets. Le langage de métaschéma ressemble syntaxiquement au SQL DDL.

3.2.5.1.1 DataMaster DataMaster⁶ est un plug-in de Protégé spécialisé dans l'import des structures et des données de bases de données relationnelles [Nyulas and Tu, 2007] qui propose une méthode d'import des tables de la base de données relationnelle comme des concepts OWL. Avec DataMaster, le paradigme relationnel est conservé puisque les données extraites de la base sont simplement inscrites dans l'ontologie générique Relational.OWL [de Laborda *et al.*, 2005].

- **D2RQ** [Bizer, 2004a] s'appuie sur le concept de D2R MAP, conservant ainsi l'élément classMap, mais avec une syntaxe légèrement différente. Avec l'utilisation de D2RQ, une application peut interroger une base de données non-RDF en utilisant RDQL. D2RQ réécrit les requêtes RDQL et les APIs de Jena appellent les requêtes SQL spécifiques de base de données. Le résultat de ces ensembles de requêtes SQL sont transformées en triplets RDF qui sont passés aux couches supérieures du Jena. Étant donné que les **mappings** entre la base de données et RDF et sont créés manuellement, ils doivent être vérifiés après chaque évolution du schéma.

3.2.5.2 Les applications

Les outils mentionnés ci-dessus, ont été utilisés dans différents domaines d'applications, notamment :

- **Sahoo et al.** [Sahoo *et al.*, 2008] décrivent leurs travaux dans le domaine sciences de la vie. Ces travaux intègrent la sémantique de domaine (à partir de plusieurs ontologies) pour créer des correspondances (représentées par des règles XPath en feuille de style XSLT) à partir de RDB en RDF. Un fichier RDF est créé en utilisant une approche par lot **Batch Approach**⁷, et stockées dans Oracle 11g. Le langage SPARQL est utilisé pour interroger l'entrepôt RDF.

6. <http://protegewiki.stanford.edu/index.php/DataMaster>

7. En informatique, un traitement par lots (batch processing en anglais) est un enchaînement automatique de commandes (processus) sur un ordinateur sans intervention d'un opérateur.

- **Byrne** [Byrne, 2008] décrit une application dans le domaine du patrimoine culturel pour convertir les données du Monument National d' Ecosse stockées dans une base relationnelle en RDF. Le système simple d'organisation des connaissances (SKOS) est utilisé pour transformer le thésaurus de la Commission royale d'enquête sur les monuments anciens et historiques de l'Ecosse (RCAHMS) vers le Web sémantique. La transformation de base de données entière (RCAHMS) est réalisée sur les systèmes de Jena et AllegroGraph.
- **Green et al.** [Green *et al.*, 2008] décrivent l'intégration des données spatiales dans RDB pour la modélisation prédictive de la pollution de l'eau en utilisant des ontologies OWL-DL de différents niveaux. Les ontologies au premier niveau (appelées ontologies de données) sont utilisés pour correspondre chaque source de données à des concepts dans les ontologies au niveau suivant (appelées ontologies de domaine). Les ontologies de données sont représentées avec le langage de mapping D2RQ. Au troisième niveau, l'ontologie d'application, fait le lien entre les ontologies et ajoute également des informations spécifiques à l'application. Le moteur D2RQ est modifié pour inclure des opérateurs spatiaux. En plus, il est utilisé pour jouer le rôle d'interface entre les sources de données et les ontologies de données. Le langage de requêtes SPARQL est utilisé pour interroger les graphes RDF virtuels générés à partir des sources de données.

3.2.6 Les outils de raisonnement

Comme nous l'avons vu dans ce chapitre, les langages « RDF Schema » et OWL fournissent du support pour le raisonnement en définissant des hiérarchies de classes ainsi que des relations entre celles-ci et entre des propriétés. Il est par exemple possible de définir une propriété comme étant symétrique par exemple. De telles définitions consistent en des règles d'inférence pouvant être appliquées par des logiciels appelés raisonneurs à la base de connaissance afin d'inférer automatiquement de la connaissance. Ainsi, si le prédicat `kingOf` est défini comme étant l'inverse de `hasKing` et que la base de connaissance contient le triplet (Baudouin, `kingOf`, Belgium), un raisonneur est capable d'enrichir cette base de connaissance avec le triplet (Belgium, `hasKing`, Baudouin). Les principaux services d'inférence proposés par les raisonneurs sont repris ci-dessous.

- **Vérification de la consistance** - Un raisonneur est capable de s'assurer de la consistance d'une base de connaissance, autrement dit de vérifier qu'il n'y a pas de faits contradictoires dans la base de connaissance, tels qu'une assertion dans l'ABox qui soit contraire aux définitions présentes dans la TBox. Ainsi, si la TBox définit une propriété `hasChild` comme devant avoir un individu de la classe `Person` pour sujet et objet, une inconsistance apparaît si l'ABox contient une assertion telle que `hasChild(Alice, rouge)` où Alice est bien une personne mais rouge n'est pas une instance de la classe `Person`.
- **Satisfaisabilité de concept** - Il s'agit de déterminer si une classe peut posséder une instance. Une classe insatisfaisable entraîne une inconsistance si une instance d'une telle classe est définie. Un exemple de classe insatisfaisable est une classe définie comme l'intersection d'une classe `Woman` et d'une classe `Father` étant donné qu'une

personne ne peut être à la fois une femme et un père.

- Classification - La classification consiste à établir les liens hiérarchiques entre toutes les classes d'une ontologie, c'est-à-dire identifier les différentes sous-classes de classes. Ceci permet par exemple de fournir toutes les sous-classes d'une classe donnée.
- Réalisation - La réalisation consiste à établir la classe la plus spécifique d'un individu, ce qui nécessite d'avoir réalisé une classification auparavant. Par exemple, s'il existe une classe Person et une classe Woman qui est une sous-classe de la classe Person, la classe la plus spécifique d'une femme sera la classe Woman et non la classe Person qui est plus générale.

Parmi les raisonneurs existants, citons RacerPro 9, FaCt++ 10 et Pellet [Sirin *et al.*, 2007]. Ce dernier, écrit dans le langage Java, est un raisonneur OWL-DL complet, c'est-à-dire qu'il supporte toute l'expressivité du langage OWL-DL. Il a de plus été étendu afin de supporter certaines propriétés apportées par OWL2, entre autres les assertions de négation de propriétés, les propriétés disjointes et le punning qui est une technique permettant à un individu et une classe de partager la même URI.

3.3 Approche ontologique : cas des données spatiales

3.3.1 Ontologies spatiales : extension de la logique de description

Différentes extensions de la logique de description pour la représentation et le raisonnement spatial ont été proposées. Nous nous intéressons à l'extension dite **ALCRRP(D)** [Haarslev *et al.*, 1998; Haarslev and Möller, 1999] car elle a donné naissance à un tableau de calcul précis et une implémentation dans différents raisonneurs tels que FaCT et RACER.

L'extension **ALCRRP(D)** est basée sur la logique de description **ALC(D)** [Baader and Hanschke, 1991] afin de fournir une base pour le raisonnement spatio-terminologique avec les logiques de description (RP signifie définition de rôles fondées sur des prédicats). Cette approche introduit un raisonnement décidable sur les relations qualitatives entre les régions de l'espace et sur les propriétés des données quantitatives. L'idée principale est de traiter les objets spatiaux et leurs relations à l'aide de prédicats sur le domaine concret des objets et de traiter les connaissances sur les objets abstraits du domaine.

Les relations spatiales de cette théorie sont basées sur un ensemble de relations topologiques par analogie avec Egenhofer [Egenhofer, 1991] ou RCC-8 [Randell *et al.*, 1992]. L'objectif est de développer une logique de description qui fournit des structures de modélisation pouvant être utilisées pour représenter les relations topologiques comme des rôles définis. Dans un modèle de domaine spécifique, les rôles représentant les relations topologiques peuvent être définies en fonction des propriétés entre objets concrets qui, à leur tour, sont associés à des particuliers par le biais des caractéristiques spécifiques. Ainsi, **ALCRRP(D)** prévoit des termes de rôles qui se réfèrent à des prédicats pour un domaine concret. Avec ces constructions, **ALCRRP(D)** étend la puissance expressive de **ALC(D)** (pour une comparaison, voir [Lutz *et al.*, 1997]). Toutefois, afin d'assurer la satisfiabilité de l'algorithme, les auteurs imposent des restrictions sur la forme syntaxique de l'ensemble des

axiomes terminologiques. Bien que la modélisation est plus difficile, les restrictions sur les terminologies assurent la décidabilité de l'approche.

Toutefois, il est nécessaire d'étendre les logiques de descriptions pour satisfaire les besoins d'applications spatiales qui utilisent d'autres types spatiaux comme le point et la ligne et pour ne pas se limiter aux régions. De plus, il est également nécessaire de pouvoir calculer dans la base de connaissances des relations spatiales entre les géométries des objets représentés par les données quantitatives.

3.3.2 Ontologies spatiales

Dans le cadre des ontologies spatiales, nous pensons qu'il est important de faire la différence entre les ontologies géographiques relatives à une thématique liée à un domaine considéré (agriculture, hydrologie, urbanisme, etc.), des ontologies spatiales qui s'intéressent à la spécification des concepts spatiaux. Il existe de nombreux travaux qui proposent des ontologies, essentiellement géographiques, pour la définition des concepts spatiaux quantitatifs représentant les types spatiaux ainsi que les concepts spatiaux qualitatifs modélisant les relations spatiales. Ces ontologies sont majoritairement basées sur les modèles de données fournis par la spécification GML et utilisent le langage de formalisation ontologique OWL. Dans [Abdelmoty *et al.*, 2005], on trouve une étude détaillée portant sur les géo-ontologies dans le cadre du moteur de recherche SPIRIT.

Dans cette partie, nous nous intéressons à la partie déclarative et à la représentation des données spatiales par des ontologies OWL. Nous présentons une ontologie spatiale géographique issus des travaux du groupe de recherche MINDSWAP, et une ontologie OGC dite GeoOWL [Joshua *et al.*, 2007].

Ontologie spatiale Geo-Ontologies

Le groupe de recherche MINDSWAP propose trois ontologies, dites **Geo-Ontologies**, qui expriment un ensemble de base des caractéristiques géographiques comme les villes, les pays et leurs descripteurs spatiaux tels que les points, les polygones ou les multipolygones, et les relations entre ces descripteurs spatiaux. Les composantes de **Geo-Ontologies** sont :

- **geoFeatures** : spécifie certaines caractéristiques géographiques ainsi que les classes pour la description de leurs caractéristiques spatiales (figure 3.1) ;
- **geoRelations** : présente sous forme de classes les relations méréologiques, topologiques, de distance et de direction ;
- **geoCoordinateSystems** : décrit les principaux systèmes de coordonnées.

Ontologie spatiale GeoOWL

L'ontologie géospatiale **GeoOWL** est développée par le groupe de travail *The Geospatial Incubator Group* du W3C [Joshua *et al.*, 2007]. Elle est formalisée en OWL et basée sur le modèle de données et la description des propriétés et des classes géographiques

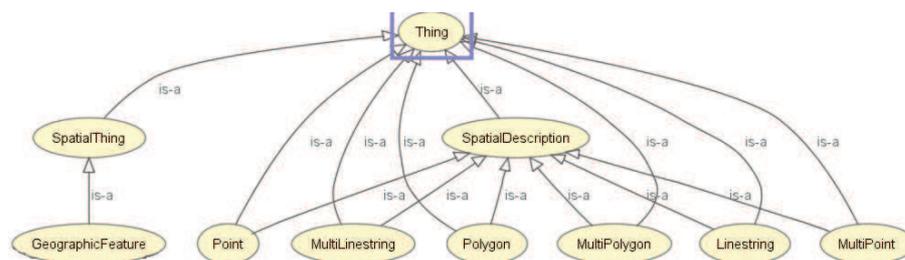


FIGURE 3.1 – L'ontologie geoFeatures

de GeoRSS. L'ontologie GeoOWL constitue une extension compatible de GeoRSS pour une utilisation dans des contextes RDF plus généraux.

L'ontologie GeoOWL définit une relation racine générique `_featureproperty` (figure 3.2) qui a comme domaine toute classe OWL/RDF qui peut être convertie en une entité spatiale. Dans le cas des flux de données Web, `_featureproperty` est associée à la relation `_content`, qui décrit une ressource Web, et donc lui fournit une entité spatiale. La relation `_featureproperty` propose un ensemble de sous relations dont `geo:where` qui associe chaque entité spatiale à un type spatial donné par la relation abstraite `_geometry`. Le modèle GeoRSS possède plusieurs sérialisations, dont deux importantes :

- **GeoRSS-GML** : dans cette sérialisation, la relation `geo:where` utilise un ensemble limité de types spatiaux de la spécification GML tels que : `gml:Point`, `gml:LineString`, `gml:Polygon`, et `gml:Envelope` (figure 3.3) ;
- **GeoRSS-Simple** : dans cette sérialisation, la relation `geo:where` utilise un ensemble de types spatiaux OGC équivalents à ceux de la spécification GML tels que : `geo:Point`, `geo:Line`, `geo:Polygon`, `geo:Circle`, et `geo:Box`.

3.3.3 Raisonnement spatial dans les ontologies

Les travaux cités ne proposent pas d'inférences ontologiques basées sur les données quantitatives. D'un autre côté, les travaux sur le raisonnement spatial qualitatif en OWL peuvent être présentés par deux approches. Une approche proposée dans [Grütter and Bauer-Messmer, 2007a; Grütter and Bauer-Messmer, 2007b; Grütter *et al.*, 2008] qui a pour objectif l'ajout du raisonnement spatial dans le web sémantique. Elle consiste en une modification du système de représentation des connaissances avec des éléments spécifiques au domaine spatial. La seconde approche proposée dans [Stocker and Sirin, 2009] qui réalise une traduction des relations spatiales en axiomes de classes OWL avec une implantation dans le moteur de raisonnement PelletSpatial.

3.3.3.1 Architecture hybride pour le raisonnement spatial

Dans [Grütter and Bauer-Messmer, 2007a; Grütter and Bauer-Messmer, 2007b; Grütter *et al.*, 2008], les auteurs proposent l'architecture d'un système hybride de représentation des connaissances pour le raisonnement spatial dans le web sémantique (figure 3.4). Dans

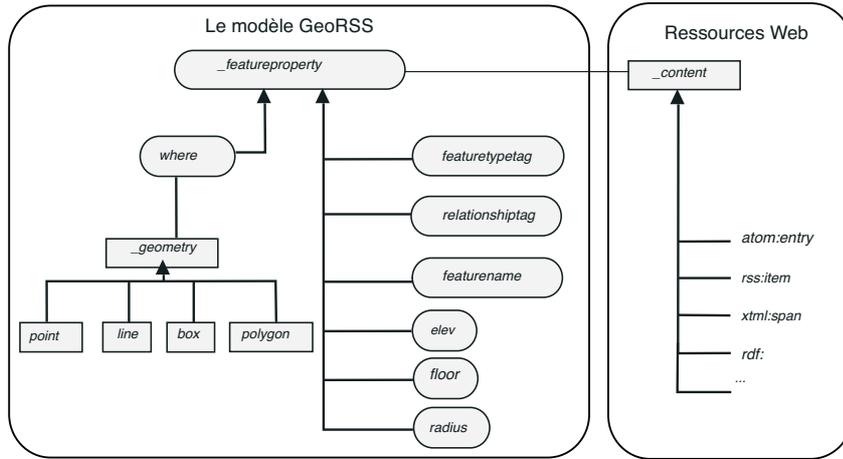


FIGURE 3.2 – Extrait du modèle GeorSS

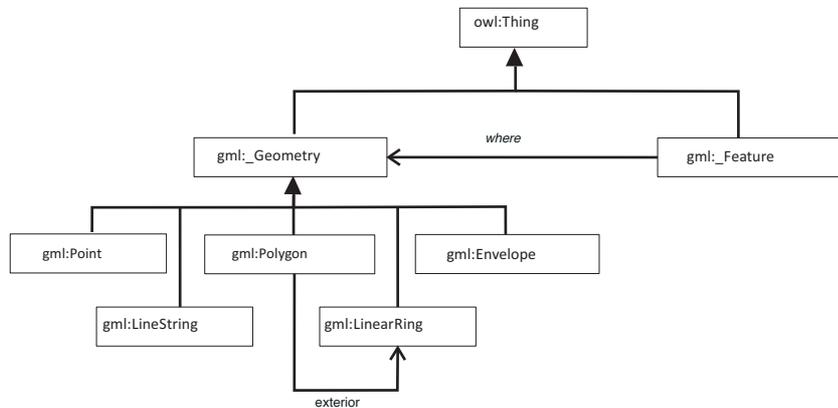


FIGURE 3.3 – Extrait de la sérialisation GeorSS-GML

cette approche, la combinaison entre les relations RCC (tableau 3.1) et OWL se fait au niveau du système de représentation et non pas au niveau du formalisme.

Les relations RCC, pour les ensembles $RCC_{1,...,5}$ (figure 3.5), sont ajoutées dans la TBox sous forme d’une hiérarchie de propriétés. Ces ensembles de relations seront utilisées pour ajouter des assertions dans la ABox permettant de représenter la relation existante entre deux régions. On exploite aussi les tables de composition RCC et on définit la RCCBox pour les différents sous-ensembles de RCC. Le raisonneur utilise les assertions présentes dans la ABox pour déterminer les relations entre les différentes régions et les tables de composition pour s’assurer de la consistance.

Il est possible de déterminer et asserter dans l’ABox toutes les relations RCC, de tous les sous-ensembles mentionnés ci-dessus, entre chaque paire de régions. Toutefois, dans [Grütter and Bauer-Messmer, 2007a], les auteurs déconseillent l’utilisation de cette approche pour deux raisons :

1. l’assertion de toutes les relations mènerait à une base de connaissances de très grande taille, nuisant ainsi aux performances du système ;
2. l’impossibilité de traduire les entrées de la table de composition RCC en axiomes

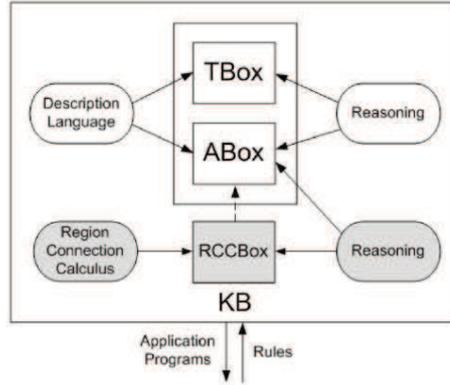


FIGURE 3.4 – Système hybride pour le raisonnement spatial basé sur le RCC

Abbréviation	Définition	Nom de la relation
SR(x, y)	$\equiv_{def} T(x, y)$	(Spatially Related)
C(x, y)	(primitive relation)	(Connects with)
DC(x, y)	$\equiv_{def} \neg C(x, y)$	(DisConnected from)
P(x, y)	$\equiv_{def} \forall z[C(z, x) \rightarrow C(z, y)]$	(Part of)
O(x, y)	$\equiv_{def} \exists z[P(z, x) \wedge P(z, y)]$	(Overlaps)
DR(x, y)	$\equiv_{def} \neg O(x, y)$	(DiscRete from)
EC(x, y)	$\equiv_{def} C(x, y) \wedge \neg O(x, y)$	(Externally Connected to)
EQ(x, y)	$\equiv_{def} P(x, y) \wedge P(y, x)$	(Equal to)
ONE(x, y)	$\equiv_{def} O(x, y) \wedge \neg EQ(x, y)$	(Overlaps Not Equal)
PP(x, y)	$\equiv_{def} P(x, y) \wedge \neg P(y, x)$	(Proper Part of)
PO(x, y)	$\equiv_{def} O(x, y) \wedge \neg P(x, y) \wedge \neg P(y, x)$	(Partially Overlaps)
TPP(x, y)	$\equiv_{def} PP(x, y) \wedge \exists z[EC(z, x) \wedge EC(z, y)]$	(Tangential Proper Part of)
NTTP(x, y)	$\equiv_{def} PP(x, y) \wedge \neg \exists z[EC(z, x) \wedge EC(z, y)]$	(Non-Tangential Proper Part of)

TABLE 3.1 – Relations spatiales RCC. La relation primitive $C(x, y)$ signifiant que x est connecté à y, x et y étant des régions spatiales

OWL-DL de manière décidable, ce qui ne permet pas à un raisonneur OWL de s'assurer de la consistance de la base de connaissances.

Par conséquent, seule une représentation minimale des relations RCC est présente dans l'ABox et les relations supplémentaires ne sont ajoutées que lorsqu'elles sont nécessaires. Dans [Grütter *et al.*, 2008], on présente deux approches pour cette représentation minimale :

1. asserter dans la ABox toutes les relations `connectsWith` pour chaque paire de régions. Les relations RCC supplémentaires, pour chaque sous-ensemble RCC, ne sont alors calculées qu'à l'exécution, si nécessaire, en se basant sur les définitions présentes dans la RCCBox ;
2. asserter dans la ABox, pour chaque paire de régions, la relation RCC valide. Les autres relations RCC, plus générales et appartenant aux autres sous-ensembles RCC, ne sont déterminées qu'à l'exécution, en cas de besoin, en utilisant les axiomes présents dans la TBox. Ce système hybride n'est toutefois pas capable d'asserter une relation entre deux régions sur la base des coordonnées.

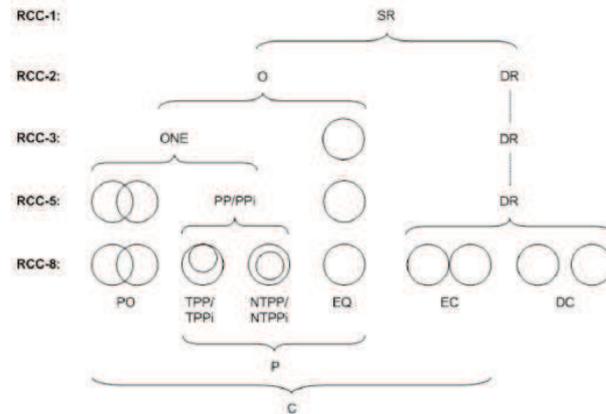


FIGURE 3.5 – Représentation hiérarchique des relations RCC appartenant aux différents sous-ensembles

3.3.3.2 Système de raisonnement PelletSpatial

Le raisonneur *PelletSpatial* étend le moteur de raisonnement *Pellet* en prenant en compte le spatial. Les principaux services de cette extension sont :

1. inférence de nouvelles relations ;
2. vérification de la consistance d'un ensemble de relations spatiales RCC ;
3. réponse à des requêtes SPARQL combinant le RCC et le RDF/OWL.

Le moteur de raisonnement *PelletSpatial* utilise une traduction des relations RCC sous forme d'axiomes de classes OWL-DL en prenant en charge les tables de composition du formalisme spatial. Cette traduction est une implantation directe des travaux de [Katz and Grau, 2005]. Cette technique de traduction engendre des problèmes de performance, ainsi, une approche hybride a été adoptée. Le raisonnement spatial est alors séparé du raisonnement sémantique OWL-DL.

Le système *PelletSpatial* ne peut pas déterminer la relation existante entre deux régions en fonction de leurs coordonnées. Par conséquent, le système nécessite que la ABox contienne déjà de telles relations.

3.4 Approche ontologique : cas des données temporelles

La modélisation des grandes masses de données spatio-temporelles, telles que les trajectoires, nécessite la prise en compte des aspects spatiaux mais aussi les aspects temporels. Dans cette partie, on s'intéresse à une modélisation ontologique des données temporelles. On présente deux approches basées sur des ontologies pour la prise en compte du temporel. La première approche est basée sur la logique de description et la seconde sur le langage OWL.

3.4.1 Ontologies temporelles : extension de la logique de description

Les extensions temporelles des formalismes logiques sont utilisées pour capturer le comportement évolutif des domaines dynamiques. Elles ont été largement prises en compte dans l'intelligence artificielle et l'informatique théorique. En particulier, ces formalismes de logiques temporelles ont été étudiés et appliqués dans des domaines tels que la spécification et la vérification des programmes informatiques [Pnueli, 1986], les systèmes d'information temporels [Chomicki *et al.*, 2003], la planification et le langage naturel [Bentham, 1995].

Depuis que l'incorporation des aspects temporels joue un rôle important dans de nombreux domaines des logiques de description tels que le raisonnement dans les bases de données temporelles [Artale *et al.*, 2003] et le raisonnement sur les actions [Artale and Franconi, 1999], on constate ces dernières années un intérêt croissant pour la logique de description temporelle (TDL). Dans [Baader *et al.*, 2003], on trouve une étude complète sur les extensions temporelles des logiques de description.

Lors de la construction d'une logique de description temporelle, l'une des décisions les plus importantes à faire est de savoir si les instants ou les intervalles de temps devraient être utilisés comme primitives temporelles [Artale and Franconi, 2001]. Comme on le sait de la logique temporelle et d'autres domaines de l'intelligence artificielle, cette décision a un impact important sur l'expressivité et les propriétés calculatoires de la logique qui en résulte [Goranko *et al.*, 2003].

Les TDL basées sur les intervalles ont l'avantage de fournir une expressivité temporelle plus riche que l'expressivité des TDL basées sur les instants. D'autre part, le comportement calculatoire des TDL à base d'intervalles est souvent problématique dans le sens où même les formalismes simples se révèlent être souvent indécidable. Un exemple important est le TDL proposé dans [Schmiedel, 1990], qui est très naturel, mais indécidable car il contient la logique temporelle à base d'intervalles indécidable de Halpern et Shoham [Halpern and Shoham, 1991]. En raison de ces problèmes de calcul, l'un des objectifs principaux de ce domaine de recherche a été d'identifier des TDL décidables et suffisamment expressives pour permettre la représentation des connaissances conceptuelles temporelles dans des domaines pertinents.

Dans [Artale and Lutz, 2004], les auteurs proposent une TDL décidable basée sur les intervalles appelée TL-ALCF. cette TDL est composée de la logique temporelle TL capable d'exprimer les termes du temps quantitatifs et de la logique de description non-temporelle ALCF en étendant le ALC avec des caractéristiques [Hollunder *et al.*, 1990]. La TDL TL-ALCF a été développé pour le raisonnement sur les actions. Une action est représentée par des contraintes temporelles sur un ensemble d'états du monde où chaque état est une collection de propriétés du monde maintenue à un certain temps.

3.4.2 Ontologies temporelles

Les premiers travaux qui ont pointé l'importance de la considération du temps dans les ontologies ont été initié par la communauté du web sémantique [Bry *et al.*, 2005].

Une variété d'approches ont été proposées pour représenter l'information temporelle dans RDF [Manola and Miller, 2004] et OWL [McGuinness and v. Harmelen, 2004], les deux principaux langages du web sémantique. Le modèle temps-valide prédomine dans ces approches. Un des premiers systèmes RDF était BUSTER [Visser, 2004]. C'est un système de recherche documentaire utilisant un cadre de raisonnement temporel pour répondre à des questions orientées temps. Plusieurs extensions du modèle RDF ont été proposées pour faciliter la représentation de l'information temporelle dans les ontologies RDF [Pugliese *et al.*, 2008; Gutierrez *et al.*, 2007a]. Ces extensions comprennent TSPARQL [Tappolet and Bernstein, 2009], qui a été développé comme une extension du langage SPARQL RDF, et TOQL [Baratis *et al.*, 2009] un langage de type SQL pour les requêtes temporelles. Dans [Gutierrez *et al.*, 2007a], les auteurs ont proposé un cadre pour incorporer le raisonnement temporel dans RDF, ce qui a donné naissance à l'approche dite graphes RDF temporels.

Une variété de systèmes temporels basés sur OWL ont été développés. Malheureusement, ajouter une dimension temporelle au langage OWL n'est pas simple. Le formalisme de logique du OWL rend difficile la modélisation de l'information qui change dynamiquement [Krieger, 2008]. En outre, OWL ne fournit pas de constructeurs temporels et parce qu'il ne prend en charge que les prédicats binaires, les relations ne peuvent pas être directement équipées d'un argument temporel. Plutôt que d'étendre le modèle logique de OWL, les recherches se sont focalisées sur la définition des couches temporelles au-dessus de OWL. Par exemple, OWL-Time [Hobbs and Pan, 2004a] propose une ontologie qui fournit des descriptions riches des instants et des intervalles temporels, la durée et les termes du calendrier, etc. Cependant, cette représentation s'intéresse à la description des éléments de données individuels, plutôt que la construction d'un modèle temporel pour décrire systématiquement toutes les informations temporelles dans un système.

Lors de l'élaboration d'un modèle temporel au-dessus de OWL, on distingue deux approches : la réification et le fluent. La réification consiste à introduire un nouvel objet et une relation pour associer une entité avec son étendue temporel ou son temps de validité. L'approche, connue sous le nom fluent [Welty and Fikes, 2006], définit une tranche de temps pour encoder la dimension temporelle occupée par une entité. Les entités sont alors reliées par leurs tranches de temps, plutôt que par des relations de l'ontologie source. Les deux approches peuvent être modélisées au niveau utilisateur sans modifications du OWL lui-même. L'inconvénient majeur de ces approches est que l'application de la sémantique temporelle est à l'extérieur du OWL ce qui se traduit par une limitation sur les processus de raisonnement temporels. Un autre inconvénient est que ces approches introduisent de nouvelles relations et d'objets pour modéliser l'information temporelle associée à une entité. Par conséquent, l'interrogation des informations temporelles peut être alourdie.

Parmi les approches qui ne modifient pas le OWL, on distingue l'approche basée sur le versionnement [Punam and Sudeep, 2007]. Quand une ontologie est modifiée, une nouvelle version est créée pour représenter l'évolution temporelle de l'ontologie. La plupart des implémentations adoptent une variété de stratégies d'optimisation afin de s'assurer que des copies entières de l'ontologie ne sont pas générées pour chaque nouvelle version. Cependant, quelles que soient les optimisations adoptées, le versionnement possède des inconvénients dont les plus importants sont la redondance de l'information et le coût élevé de l'interrogation des événements survenus au cours des intervalles de temps particuliers.

Plusieurs recherches se sont intéressées à définir des extensions du modèle logique du langage OWL pour y intégrer le temps [Kim *et al.*, 2008; Krieger, 2008; Lutz *et al.*, 2008]. Ces approches impliquent la définition de nouveaux opérateurs OWL. Une variété de logiques de description temporelles ont été proposées, sans faire émerger une approche standard.

3.4.3 Raisonnement temporel dans les ontologies

Une fois toutes les données temporelles sont représentées de manière consistante dans une ontologie, on doit alors pouvoir les manipuler en utilisant des méthodes locales ou des langages de requêtes.

Dans [Gutierrez *et al.*, 2005; Gutierrez *et al.*, 2007b], les auteurs introduisent la notion d'implication temporelle. Cette notion considère le graphe RDF comme une base de connaissances à partir de laquelle de nouvelles connaissances, c'est à dire d'autres graphes, peuvent être déduites. L'approche s'appuie sur la temporisation du graphe RDF et introduit des règles temporelles pour les instants et les intervalles. On définit aussi un langage de requêtes adapté au graphe RDF et utilisant les règles définies. Dans [Hurtado and Vaisman, 2006], les auteurs ont présenté un travail basé sur le graphe temporel RDF en définissant les notions de contrainte et d'intervalle temporels. Cette approche avait pour objectif d'inclure explicitement le raisonnement temporel à travers l'utilisation des relations d'Allen entre intervalles sous forme de contraintes. Dans [Perry *et al.*, 2007], les auteurs exploitent la notion de graphe temporel RDF pour représenter des informations temporelles dans une ontologie du domaine militaire. Ce travail a donné lieu à des solutions de stockage des triplets RDF dans une base de données relationnelle ainsi qu'une implantation des inférences RDFS, sous forme d'index, incluant la notion de temps valide pour les déclarations RDF. Les auteurs ont définis des opérateurs temporels, entre autres, qui calculent des paires de sous-graphes dont les intervalles de temps satisfont une relation temporelle donnée. Ces opérateurs temporels ont été implantés dans la base de données sémantique de Oracle 10g pour être utilisés dans un langage de requêtes des données RDF.

Comme le langage OWL lui-même n'a pas d'opérateurs temporels pour manipuler des valeurs de temps, on va voir arriver plusieurs solutions pour ajouter des règles d'inférences générales, tel que le langage SWRL. On va alors utiliser ces langages de règles pour définir des inférences temporelles personnalisées. En effet, SWRL fournit un mécanisme pour créer des bibliothèques de méthodes personnalisées qui peuvent être utilisées dans les règles. Dans [O'Connor and Das, 2011], les auteurs ont utilisé ce mécanisme pour définir une bibliothèque de méthodes qui mettent en œuvre les relations entre intervalles temporels de l'algèbre d'Allen.

Être capable d'écrire des règles temporelles est nécessaire mais insuffisant car il faut aussi être capable d'écrire des requêtes temporelles sur une ontologie. Dans ce cadre, il y a des solutions intéressantes dont le langage de requêtes basé sur SWRL appelé SQWRL (*Semantic Query-Enhanced Web Rule Language*) [O'Connor and Das, 2009]. À l'aide de gabarits prédéfinis, SQWRL définit un ensemble d'opérateurs de type SQL qui peuvent être utilisés pour construire des spécifications de récupération des informations stockées

dans une ontologie OWL [O'Connor and Das, 2011]. D'une façon générale, les opérateurs sont utilisés dans le conséquent d'une règle pour formater les informations appariées avec l'antécédent de la règle. Cet antécédent est effectivement considéré comme un modèle pour la spécification de la requête.

3.5 Approche ontologique : cas des données spatio-temporelles

La représentation conjointe de l'espace et du temps, ainsi que les approches de raisonnement associées, est un défi majeur et récent dicté par des domaines d'application croissants. À titre d'exemple, on peut citer la gestion du trafic routier, prévention des risques naturels, etc. Dans ce chapitre, nous présentons quelques travaux qui réunissent les concepts spatiaux et temporels dans le même cadre ontologique ainsi que les solutions de raisonnement adoptées.

3.5.1 Cas des données géographiques

Les données géographiques possèdent intrinsèquement des composantes temporelles et spatiales exploitées de manière unifiée dans différents domaines d'application. Par conséquent, ces données et leurs caractéristiques sont associées à la sémantique des domaines d'application d'où l'émergence de plusieurs approches basées sur des ontologies pour les données géographiques.

Dans [Bittner *et al.*, 2009], les auteurs ont présenté une ontologie formelle de haut niveau basée sur la logique qui peut être utilisée comme un outil pour spécifier la sémantique de haut niveau des termes utilisés dans les domaines géographiques. Cette approche s'appuie et étend des propositions intéressantes dans ce domaine telles que [Neuhaus *et al.*, 2004; Bittner *et al.*, 2004; Grenon and Smith, 2004a].

En mettant l'accent sur les endurants indépendants de l'ontologie formelle et les relations entre eux, cette approche a permis :

- de mettre l'accent sur les notions géographiquement importantes telles que les partitions, les équivalences météorologiques, etc. ;
- d'examiner plus en détail le comportement temporel et spatial des différentes catégories des endurants indépendants ;
- de se concentrer sur la mise en place de relations météorologiques entre universaux.

Ce travail fournit une contribution majeure dans le cadre des ontologies spatio-temporelles géographiques en élaborant et en appliquant une théorie axiomatique des relations spatiales en fonction du temps d'une manière logique rigoureuse dans le cadre de raisonnement sur l'information géographique. Les auteurs proposent une implémentation de l'ontologie en utilisant une logique de prédicat du premier ordre sous la forme d'un langage objet. Pour les calculs nécessaires au processus de raisonnement, les auteurs utilisent un outil de développement basé sur le langage fonctionnel ML dite *Isabelle*.

Pour être complet, ce travail doit fournir une extension aux entités autres que les endurants, telles que les perdurants. Pour cela, les auteurs doivent donner une description explicite des changements d'états des individus et proposer une ontologie spatio-temporelle intégrant les interrelations entre les entités durables et les processus auxquels ils participent comme celle définie dans [Grenon and Smith, 2004a].

3.5.2 Cas des bases de données

Les bases de données spatio-temporelles sont souvent construites à partir de données provenant de nombreuses sources différentes. Les données à intégrer diffèrent dans leur sémantique et leur représentation. Pour combler ce vide, les approches ontologiques ont été exploitées pour relier ces données. Les ontologies utilisées ont pour objectif d'unifier les sens de ces données et leur utilisation dans un contexte défini. Pour cette raison, ce domaine de recherche ne propose pas une approche globale ou unique car ces ontologies sont souvent construites par des communautés déferentes s'appuyant sur des méthodes et des environnements déferents.

Le projet européen CHOROCHRONOS [Koubarakis *et al.*, 2003] a contribué à de nombreuses idées novatrices dans les domaines de l'ontologie et la modélisation des données ainsi que l'évaluation des requêtes et des prototypes de systèmes de bases de données spatio-temporelles. Ces idées ont déjà trouvé des utilisations dans différents domaines d'application tels que les objets mobiles, les systèmes d'information sur l'environnement, les applications multimédias interactives et les mondes virtuels, etc. Les bases de données spatio-temporelles de telles applications reposent sur des engagements ontologiques. Les décisions concernant le type du système utilisé, comment les identifiants sont gérés, et ainsi de suite, sont tirés d'une vue particulière du monde à laquelle la base de données se rapporte, en d'autres termes d'une ontologie spécifique.

Dans [Frank, 2003], l'auteur affirme que les bases de données spatio-temporelles doivent prendre des engagements forts pour saisir le sens de l'espace et du temps. Cette ontologie est nécessairement plus complexe et la connexion à la zone d'application est plus forte. Le concepteur d'une application de base de données doit concilier les concepts ontologiques de la zone d'application avec l'ontologie construite dans la base de données. Idéalement, une base de données spatio-temporelle intégrant une ontologie implique un engagement minimal sur la façon dont l'espace et le temps sont structurés et doit prendre en compte les améliorations spécifiques à l'application. Pour résoudre ce problème, l'auteur propose l'utilisation d'une ontologie multi-tiers pour capturer la définition des concepts et le système de raisonnement dans des couches séparées.

La description des ontologies pour les bases de données spatio-temporelles est encore plus exigeante. Plusieurs approches proposent d'utiliser des algèbres, qui ne se limitent pas aux relations statiques, mais permettent de décrire des objets et des opérations dans le même contexte pour mieux capturer les aspects temporels et spatiales. D'autres travaux proposent des modèles conceptuels pour des applications spatio-temporelles à base ontologique dans un contexte bases de données.

3.6 Synthèse

Ce chapitre est centré sur les ontologies et leur application dans le domaine spatial, temporel et spatio-temporel, ainsi que les mécanismes de raisonnement développés dans l'espace et le temps. Dans la première partie de ce chapitre, nous avons abordé un des points clés de nos travaux de thèse qui est l'utilisation des ontologies comme moyen de représentation pour la manipulation des connaissances. Nous avons présenté un ensemble d'ontologie afin d'assurer la possibilité de leur réutilisation. Nous avons à l'esprit la réutilisation des ontologies existantes et standards. Ensuite, nous avons passé en revue les langages de représentation standards pour la définition d'annotations sémantiques (RDF(S)) et d'ontologies (OWL)...etc. Le « Resource Description Framework (RDF) » permet de structurer l'information en la représentant sous forme de triplets (sujet, prédicat, objet), les ressources ainsi décrites étant de plus identifiées univoquement par un « Uniform Resource Identifier ». Le RDF Schema est une extension du RDF apportant une structuration en permettant de décrire une hiérarchie de ressources et de propriétés. Ce langage permet par exemple d'inférer le fait que la Belgique est une région, sachant qu'il s'agit d'un pays et qu'un pays est une sous-classe d'une région. Le « Web Ontology Language (OWL) » va plus loin dans cette capacité de raisonnement en ajoutant de nombreux prédicats tels que le fait que deux propriétés soient inverses l'une de l'autre ou qu'une classe corresponde à l'intersection de deux autres classes. Pour cela et suite aux limites de l'expressivité de RDF, nous avons retenu le langage OWL qui répond aux besoins variés des applications. Nous avons également présenté le langage « SPARQL » qui a été développé par le W3C afin de pouvoir interroger les documents RDF. Nous avons également présenté certains environnements pour la modélisation d'ontologies ainsi que plusieurs outils d'inférence. Dans la deuxième partie de ce chapitre, nous avons présenté des approches ontologiques dans le cas de données spatiales, temporelles et spatio-temporelles. Dans cette direction, nous avons présenté l'extension spatiale dans la logique de description et les ontologies liées à l'espace géographique qui définissent un vocabulaire de référence pour la description des données géospatiales. Malgré le caractère spatiale présent dans ces ontologies, elles sont dédiées à des services bien spécifiques comme *geoontology* et à des domaines bien particulier comme SWEET pour les sciences de la terre. En plus, nous avons constaté qu'il y a du vocabulaire inutile pour nos travaux. Concernant la représentation ontologique du temps, nous avons retenu l'ontologie OWL-Time pour nos travaux, recommandation du W3C pour la modélisation des concepts temporels et qui propose également une modélisation pour les treize relations d'Allen. Les raisons du choix de OWLTime ont été :

- elle fournit un vocabulaire pour exprimer des faits sur les relations topologiques entre les instants et les intervalles, avec des informations sur la durée,...
- Dans les grandes ontologies, comme SUMO et upperCyc, les connaissances sur le temps sont dispersées tout au long de l'ontologie. Étant donné que ces grandes ontologies ne sont pas organisées en des sous ontologies, il est difficile d'en extraire toutes les connaissances sur un domaine spécifique.
- De même, il est plus simple d'intégrer deux ontologies implémentées dans le même langage, pour qu'il n'y ait pas besoin de traduction d'une ontologie à l'autre (SUMO est formalisé par une version du langage KIF (Knowledge Interchange Format)).

Nous avons constaté l'absence des axiomes ou les règles liées à ces relations. Pour cela nous allons les implémenter dans la partie mise en œuvre du chapitre 5 des contributions. Ainsi, nous avons présenté les approches ontologiques pour les données spatio-temporelles notamment les données géographiques. En fait, ces données et leurs caractéristiques sont associées directement à la sémantique des domaines d'application qui peuvent être utilisés seulement dans le contexte géographique.

Chapitre 4

Approche ontologique pour les données des trajectoires

Sommaire

4.1	Introduction	80
4.2	Ontologie des objets mobiles	80
4.3	Ontologie modulaire de la trajectoire	82
4.4	Ontologie de trajectoire : espace, temps et thème	83
4.5	Synthèse	84

4.1 Introduction

Les applications basées sur les objets mobiles et leurs trajectoires connaissent un développement important dans différents domaines. Ces applications exploitent conjointement les données spatio-temporelles capturées des objets mobiles, qui forment alors les trajectoires, avec d'autres sources de données liées essentiellement aux sémantiques des domaines considérés. Comme dans les approches ontologiques pour les données spatio-temporelles, on est confronté aux différents problèmes d'exploitation des masses de données spatiales et temporelles en relation avec une sémantique applicative précise. Pour considérer ces problèmes, les recherches actuelles se dirigent vers des modèles ontologiques des trajectoires prenant en compte, d'une façon ou d'une autre, la sémantique du domaine.

Ce chapitre présente des travaux de recherche utilisant des approches ontologiques pour la modélisation des objets mobiles et les trajectoires d'une manière générale. La dernière partie de ce chapitre présente des travaux concernant la prise en compte de la dimension métier, dite aussi thématique, par les nouveaux modèles ontologiques spatiales/temporelles.

4.2 Ontologie des objets mobiles

Dans [Tryfona and Pfoser, 2001], les auteurs s'intéressent à la conception des ontologies pour gérer l'échange et le partage d'informations entre les applications fondées sur des objets mobiles. Ce champ d'applications varie des systèmes de gestion de flotte à la surveillance des utilisateurs de téléphones mobiles pour offrir des services géolocalisés. Les auteurs mettent alors l'accent sur la compréhension de l'objet mobile et ses déplacements. En particulier, ils ont analysé le concept important du mouvement, ses propriétés et ses relations au sein de l'environnement spatio-temporel considéré. Ils décrivent une manière de comprendre, de représenter et interpréter les paramètres communs pour toutes les applications liées aux objets mobiles.

Les auteurs de [Tryfona and Pfoser, 2001] définissent des ontologies mobiles qui permettent de modéliser le monde mobile et ses caractéristiques, où certaines ont tendance à être communes, et pour cela, facile d'être échangées et partagées entre différentes applications mobiles. Cette approche se veut d'être générique pour couvrir différents types d'objets mobiles, tels que les téléphones mobiles, les taxis mobiles, et d'autre part, suffisamment précise pour montrer la nature des données impliquées dans les applications mobiles. Le domaine ontologique mobile, dont un extrait est donné par la figure 4.1, comprend les éléments suivants :

- la trajectoire : le concept central et fondamental de l'ontologie ;
- les objets de la trajectoire : ces objets mobiles peuvent être les véhicules mobiles, les appareils mobiles comme les assistants nomades ou les téléphones, les personnes ou les groupes de personnes en mouvement. De manière analogue, comme ce modèle embarque un cadre de travail 3D, ces objets peuvent être spatio-temporels. La figure 4.2 montre une hiérarchie des objets considérés ;

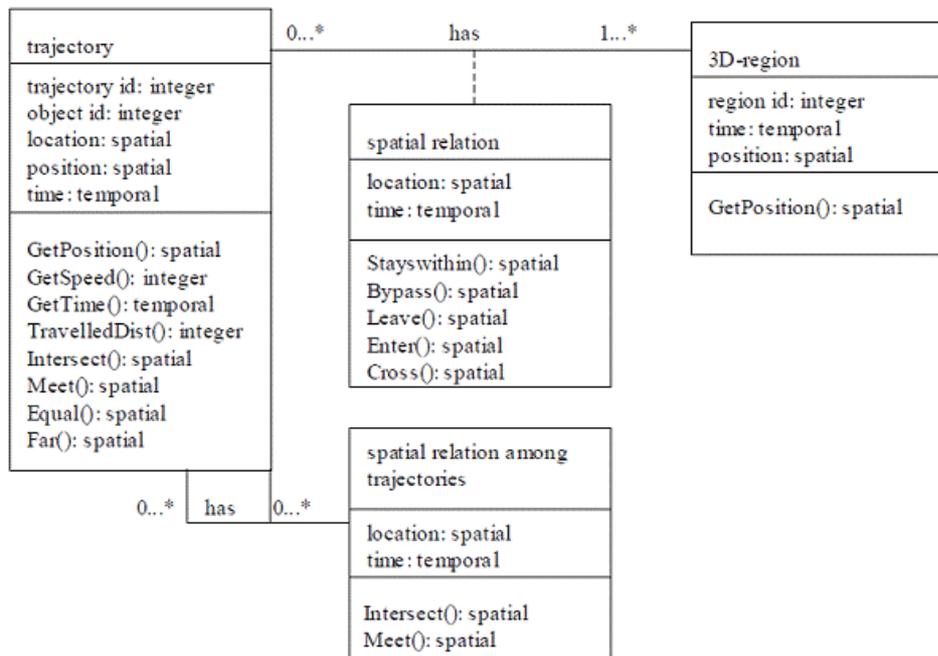


FIGURE 4.1 – Modèle UML de la trajectoire selon [Tryfona and Pfoser, 2001]

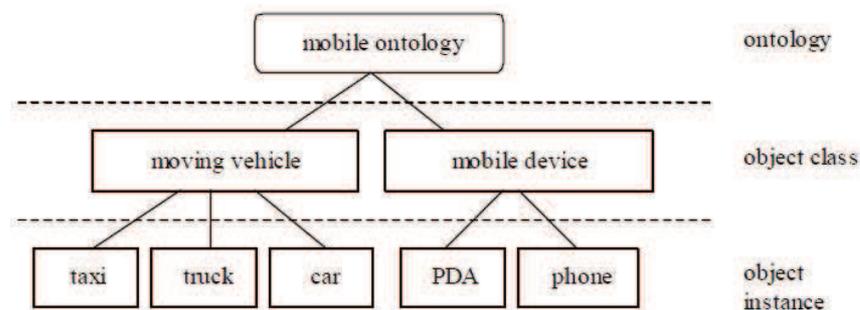


FIGURE 4.2 – Ontologie des mobiles selon [Tryfona and Pfoser, 2001]

- les relations : le modèle gère deux types de relations importantes, à savoir, les relations entre les trajectoires et celle entre les trajectoires et le cadre de travail spatio-temporel considéré. Les auteurs représentent, en outre, les relations fondamentales suivantes :
 - événement spatial : action qui se déroule à un point temporel et spatial précis ;
 - processus : action qui se déroule dans le temps et possède une durée.

Dans [Tryfona and Pfoser, 2005], les auteurs présentent l'utilisation des ontologies des objets mobiles pour les services basés sur la localisation appelés LBS (*Location Based Services*). Ils proposent une architecture ontologique basée sur des composantes ontologiques structurelles indépendantes. Comme le montre la figure 4.3, cette architecture maintient l'ontologie du domaine de la trajectoire indépendante des autres sémantiques.

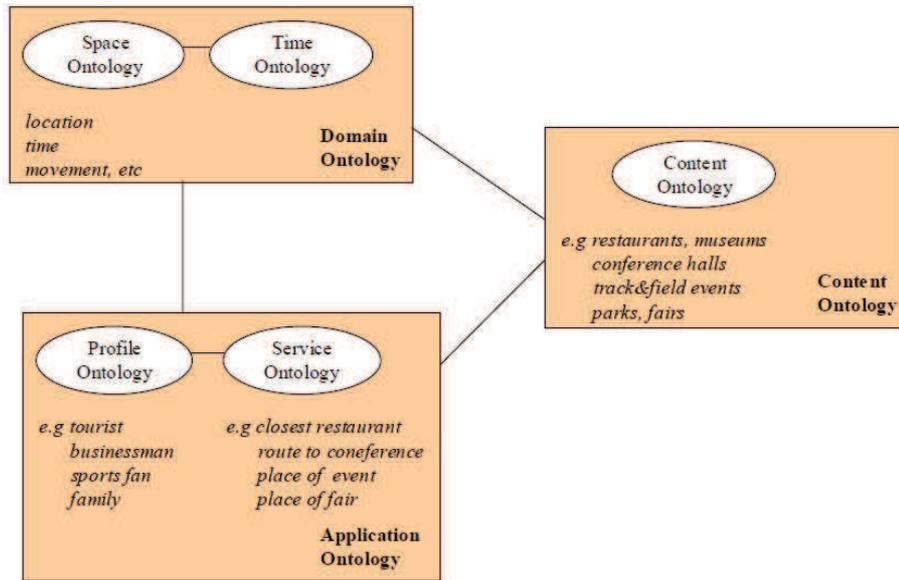


FIGURE 4.3 – Ontologies de services de localisation

4.3 Ontologie modulaire de la trajectoire

En se basant sur les travaux de [Spaccapietra *et al.*, 2008a], dans [Yan *et al.*, 2008; Yan, 2011] les auteurs ont proposé un cadre de travail ontologique formé par des ontologies modulaires. Ils ont montré le besoin des connaissances géométriques, géographiques et du domaine d'application. Pour cela, ils se sont basés sur la modularisation des ontologies, comme dans [Stuckenschmidt *et al.*, 2009], pour fournir trois modules ontologiques : un module de trajectoire géométrique, un module géographique et un module de domaine d'application (figure 4.4).

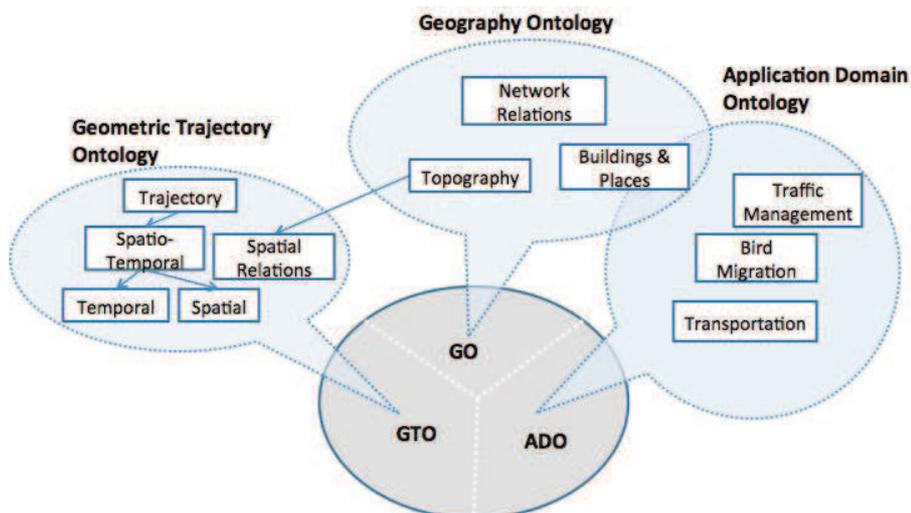


FIGURE 4.4 – Infrastructure ontologique pour la trajectoire

Dans cette infrastructure ontologique pour la trajectoire (figure 4.4) :

- le module géométrique : présente les concepts génériques de la trajectoire. Elle comporte des concepts spatio-temporels utilisés pour spécifier les caractéristiques spatiales, temporelles et spatio-temporelles des données de l'application ;
- le module géographique : présente les concepts qui participent à la description de la couverture géographique. Ce module est lié aux modules de la trajectoire géométrique et de l'application ;
- le module du domaine d'application : rassemble tous les concepts dépendants de l'application. Par exemple des ontologies de gestion du trafic, de migration d'oiseaux et les ontologies de transports.

Les auteurs présentent une ontologie finale qui fournit une description sémantique de la trajectoire liée à une application. La figure 4.5 montre l'ontologie finale de l'application de gestion de trafic.

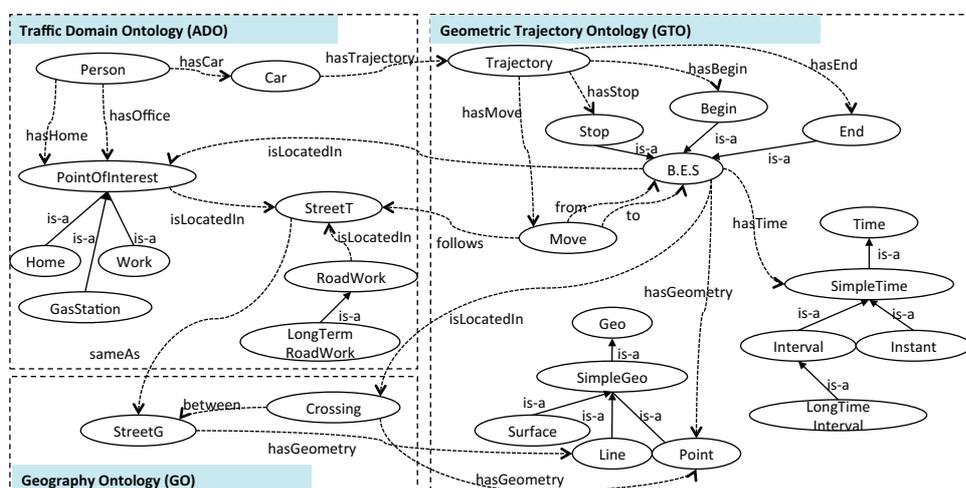


FIGURE 4.5 – Ontologie de la trajectoire sémantique pour l'application de gestion du trafic

4.4 Ontologie de trajectoire : espace, temps et thème

Dans [Perry *et al.*, 2006; Perry *et al.*, 2007; Perry, 2008], les auteurs s'intéressent à la modélisation des données spatio-temporelles capturées basées sur les ontologies en prenant en compte les questions principales de l'utilisateur : le quoi, le où et le quand. Dans cette approche, la dynamique du monde réelle est représentée par trois dimensions : le thème, l'espace et le temps.

Les auteurs présentent une ontologie de haut niveau combinant les dimensions thématique, spatiale et temporelle de la donnée. Comme le montre la figure 4.6, ce niveau supérieur définit une hiérarchie des classes d'entités thématiques et spatiales. Ces classes sont associées par des relations. Les informations temporelles sont intégrées dans l'ontologie en étiquetant les relations entre les instances avec leur temps valide. Pour l'élaboration de ce haut niveau, les auteurs s'appuient sur les travaux de [Grenon and Smith, 2004b].

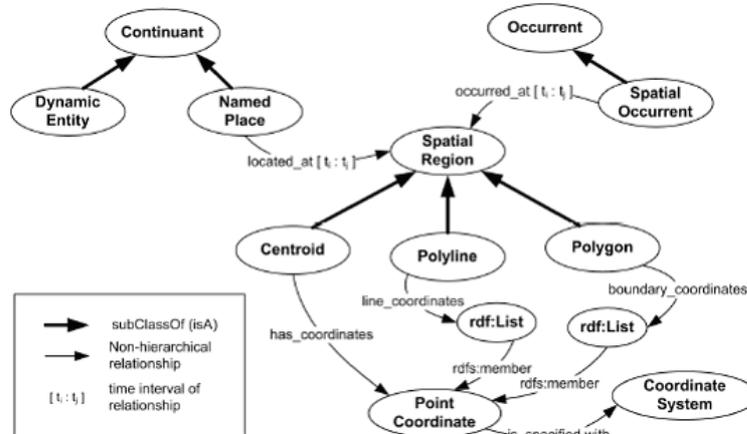


FIGURE 4.6 – Ontologie du thème, espace et temps

Pour représenter les données spatiales à deux dimensions et les données temporelles dans le modèle, les auteurs distinguent les entités qui persistent au fil du temps et maintiennent leur identité à travers le changement nommé « continuants », des événements qui se produisent et disparaissent appelés « occurrents ». La dimension spatiale est présentée par les entités : région spatiale, coordonnées et le système associé. La dimension temporelle est présentée par l'association des intervalles de temps avec les instances de relation dite aussi le temps valide.

Dans ce travail, les auteurs définissent un ensemble d'opérateurs proches des besoins de l'application basés sur l'ontologie de haut niveau. Ces opérateurs supportent la notion de contexte et sont capables de calculer des relations spatio-temporelles implicites. Une implantation SQL des opérateurs et un schéma de stockage des données ontologiques sous forme de triplets RDF sont définis dans le SGBD Oracle 10g.

4.5 Synthèse

Cette partie est centrée sur les travaux de recherche qui s'intéressent à la modélisation ontologique des concepts relatifs à la trajectoire associée à un objet mobile notamment les travaux de Stephano et Tryfona. Ces modèles ontologiques ont été proposés à partir de la définition générale de la trajectoire. Dans notre travail, nous allons établir un patron de la trajectoire à partir de différentes applications et normes. En conclusion, si plusieurs approches pour la modélisation des informations spatiales et temporelles existent à ce jour, il faut souligner le besoin de raisonneurs spatiaux et temporels capables d'exploiter ces données pour déduire des informations implicites et pour répondre à des requêtes. Il existe aussi des travaux qui prennent en considération l'aspect thématique dans la modélisation spatio-temporelle notamment ceux de M. Perry. Dans son travail de thèse, M. Perry [Perry, 2008] s'intéresse à une application analytique dans le domaine militaire. Ces contributions majeures sont la mise en place d'une approche ontologique prenant en compte le thème, l'espace et le temps. Il définit aussi un ensemble d'opérateurs proches

des besoins de l'application notamment par la définition d'une ontologie de haut niveau. Ces opérateurs supportent la notion de contexte et sont capables de calculer des relations spatio-temporelles implicites. Une implantation SQL des opérateurs et un schéma de stockage des données ontologiques sous forme de triplets RDF sont définis dans le SGBD Oracle 10g. L'auteur pointe les problèmes d'inférence RDFS et propose des solutions validées sur des ensembles de données RDF de volumes importants. L'approche de M. Perry nous intéresse tout particulièrement dans sa méthodologie. Cependant, nos données capturées nécessitent une annotation sémantique, ce qui n'est pas traité dans ce travail. De plus nous ferons d'autres choix pour l'intégration ontologique du temps et de l'espace basée sur le principe de réutilisation. Aussi, le mécanisme d'inférence considère les constructeurs RDFS et OWL en plus des règles utilisateur. Cela a des conséquences sur la complexité du raisonnement.

Deuxième partie

Contributions

Actuellement, on dispose de technologies fiables pour capturer les données de localisation d'un objet mobile. On peut citer trois exemples de ces systèmes de capture :

1. sur terre et quand la taille de l'objet est importante : les données de localisation sont obtenues généralement par le système de positionnement et de datation par satellites GNSS⁸ ;
2. sur terre et quand la taille de l'objet est critique : les données de localisation peuvent être obtenues par traitement des images issues de dispositifs de suivi par une caméra embarquée dans un drone [Louvat *et al.*, 2007] ;
3. en pleine mer : on dispose de dispositifs dédiés tels que le Satellite Relay Data Logger SRDL⁹ utilisé pour la localisation des animaux marins par exemple.

Parmi les données capturées, nous nous intéressons à celles qui concernent l'espace et le temps afin de comprendre et analyser les mouvements ou les trajectoires de l'objet mobile. Dans cet objectif, nous considérons deux objets mobiles utilisant un système de suivi basé sur le satellite. Ces objets sont les navires et les avions.

Dans le domaine maritime, la réglementation oblige les navires de commerce à utiliser le système AIS¹⁰. Celui-ci fournit fréquemment les données de localisation du navire sous la forme d'un couple latitude et longitude. D'une manière générale, il permet l'échange régulier de données entre les navires et les stations terrestres, notamment pour améliorer la sécurité.

Dans le domaine aérien, les avions sont équipés d'un système implémentant une spécification d'échanges de données dite Mode-S¹¹. En France, cette spécification est entrée en vigueur depuis mars 2007. Elle permet l'envoi et la réception de données numériques entre un avion et une station terrestre équipée d'un transpondeur. Le Mode-S diffuse des données générales telles que l'immatriculation de l'avion, son numéro de vol et des informations de localisation sous la forme d'un triplet latitude, longitude et altitude.

Les données capturées des objets mobiles sont obtenues sous la forme de flots de données. Elles sont échangées en temps réel ou en différé entre les objets mobiles et une station fixe ou mobile. Elles ont un format précis et documenté. Ces données possèdent des caractéristiques générales ou des méta-données combinées à des données spatio-temporelles permettant de reconstituer une ou plusieurs trajectoires. La dimension spatiale réside dans le caractère géolocalisé exprimé dans différents systèmes selon le domaine d'application. La dimension temporelle exprime la continuité dans le temps de ces données. Par conséquent, ces données sont spatio-temporelles multidimensionnelles. Les trajectoires des objets mobiles forment un volume important de données dès qu'on souhaite les étudier sur une longue période. On peut donc parler dans ce cas de masses de données spatio-temporelles multidimensionnelles.

Les domaines d'applications centrés sur les trajectoires des objets mobiles, fournies par les capteurs cités précédemment (AIS pour les bateaux, Mode-S pour les avions),

8. Global Navigation Satellite System

9. <http://www.smru.st-and.ac.uk/Instrumentation/SRDL>

10. *Automatic Identification System*

11. <http://www.eurocontrol.int/msa/gallery/content/public/documents/EMS-SPE-01-3.11.pdf>

sont nombreux. Toutefois, elles partagent un même intérêt lié à la caractérisation et à la compréhension du mouvement et par conséquent, le comportement de l'objet mobile dans son environnement. Les applications qui nous intéressent tout particulièrement sont :

- la visualisation, compréhensible et multi-vues des trajectoires : elle permet aux experts du domaine d'interpréter et de donner un sens (souvent sémantique basée sur des annotations) aux mouvements des objets mobiles durant leurs déplacements, notamment pour décrire des comportements ;
- l'extraction d'informations supplémentaires sur les trajectoires traçant le déplacement des objets mobiles : ces informations sont le résultat des opérations d'interrogation, telle que la sélection. Elles peuvent s'appuyer sur les annotations préalablement définies ;
- l'aide à la prise de décision : notamment pour assurer la sécurité. En effet, dans le domaine aérien et maritime la compréhension des trajectoires des avions et des navires aident les stations à prévoir les différentes évolutions de ces trajectoires. Ainsi, des collisions et autres accidents liés à la météo peuvent être évités. Dans ce travail de thèse, nous étudions aussi les trajectoires des mammifères marins. Cette étude peut être capitale pour la sauvegarde et la préservation de la population.

Les cas d'utilisations cités à titre d'exemple forment en général des systèmes d'informations évolutifs et complexes. Par conséquent, il est indispensable d'utiliser une modélisation permettant de :

- contrôler les données : en définissant une structuration de cette masse de données. En effet, cette structuration doit s'appuyer sur un modèle de données rigoureux et évolutif pour gérer efficacement : l'indexation, l'interrogation, la comparaison et d'une manière générale la manipulation des trajectoires ;
- comprendre les données : en définissant une structuration des connaissances liées à ces données. Ces connaissances doivent tenir compte des méta-données capturées et des annotations définies par les experts des domaines ;
- prévoir les comportements : en définissant des modèles de passage des données aux connaissances. Un premier objectif peut être la structuration des modèles d'analyse et de traitement des données, puis la définition des liaisons possibles entre les structures de hauts niveaux sur les trajectoires et ces modèles de traitement.

Ainsi, le premier chapitre de la partie contributions aborde la notion de contrôle par la définition d'un modèle de données. Nous présentons nos propositions concernant les différents modèles de données dans les deux exemples cités précédemment. À partir de la notion de trajectoire dans les deux exemples, nous montrons qu'il est possible d'établir une généralisation de la notion de trajectoire d'un objet mobile.

À partir de cette étape, nous organisons les connaissances liées à la trajectoire sémantique dans un modèle ontologique. Cette trajectoire doit être connectée à une ontologie de l'espace et de temps. Pour cela, nous entamons dans le deuxième chapitre la réutilisation des ontologies du temps et l'espace dans l'ontologie du domaine afin d'avoir une trajectoire complète supportant un raisonnement thématique, spatial et temporel. Dans le troisième chapitre, nous souhaitons que ce raisonnement soit généralisé et applicable sur tous les domaines d'application. Pour cela, nous présentons une étude de cas d'application sur les trajectoires des mammifères marins. Nous détaillons le processus qui part des

données brutes représentant les déplacements des phoques jusqu'au modèle ontologique de trajectoire. Dans le quatrième chapitre, nous discutons la prise en charge des données ontologiques par les bases de données et les exigences nécessaires. Ainsi, nous présentons aussi une étude comparative de différentes bases de données à base ontologiques. Dans le dernier chapitre, nous proposons une solution de mise en œuvre ainsi que l'évaluation d'une requête sur la base de données ontologiques.

Chapitre 1

De l'expérience au modèle ontologique de la trajectoire

Sommaire

1.1	De l'expérience au modèle conceptuel	93
1.1.1	Introduction	93
1.1.2	Expérience 1 : Étude des trajectoires des navires	94
1.1.3	Expérience 2 : Étude des trajectoires des avions	99
1.2	Modèle générique de la trajectoire	102
1.2.1	Objectif	102
1.2.2	Domaine de la trajectoire	102
1.2.3	Domaine de l'objet mobile et de la trajectoire	103
1.3	Modèle de la trajectoire sémantique	104
1.3.1	Annotation	104
1.3.2	Trajectoire sémantique : trajectoire annotée	104
1.4	Modèle ontologique de la trajectoire	105
1.5	Synthèse	107

1.1 De l'expérience au modèle conceptuel

1.1.1 Introduction

Dans cette partie, nous abordons la notion de contrôle par la définition d'un modèle de données. Nous présentons nos propositions concernant les modèles de données des trajectoires des navires et des avions. Nous vérifions alors si la notion de modèle de trajectoire a le même sens dans les deux exemples? Quelles sont les entités communes? Est-il possible d'établir une généralisation de la notion de trajectoire d'un objet mobile?

Pour cela, nous présentons la démarche suivie pour aboutir au modèle conceptuel de la trajectoire de chacun des deux objets mobiles (navire, avion). D'une manière générale, cette démarche se compose des étapes suivantes :

1. la capture des trames brutes décrivant la trajectoire de l'objet mobile ;
2. l'analyse et la compréhension des trames ;
3. la transformation des trames capturées en données exploitables ;
4. la présentation du modèle de données. Ce nouveau cadre de travail fait ressortir d'une façon claire la notion de trajectoire, sa relation avec l'objet mobile et l'ensemble des méta-données, pour l'instant intrinsèques, de cette trajectoire.

1.1.2 Expérience 1 : Étude des trajectoires des navires

1.1.2.1 Contexte de l'étude

Dans ce travail de thèse, le domaine maritime est parmi les terrains expérimentaux que nous explorons au sein du laboratoire L3i. Ce travail se fait en partenariat avec le laboratoire IRENAV¹² de l'École Navale depuis plusieurs années. L'objectif est d'étudier les trajectoires et de développer une plate-forme de services de surveillance et de suivi des objets mobiles dans un environnement maritime.

Dans ce domaine, le système AIS est couramment utilisé pour la localisation des navires. Les données reçues et envoyées de l'AIS, aident à la gestion des manœuvres pour éviter les collisions entre les navires et pour assurer la sécurité et la surveillance des zones à fort trafic. L'AIS fonctionne de manière continue et autonome par transmission des trames contenant des données brutes concernant le navire et son mouvement. Le volume des données est important. Par exemple, le système AIS situé dans le port de Brest, peut recevoir jusqu'à 50000 messages AIS par jour. Un chiffre représentatif pour le port de Brest est 13817 positions enregistrées à partir de 26 navires en mouvement, 13915 positions de 20 bateaux se déplaçant pour le port de La Rochelle. Ces positions tracent les trajectoires de ces navires qui sont en mouvement. Cependant, ces mouvements sont effectués sous des contraintes de la navigation et du port. La variation de ces contraintes influent directement sur la précision de représentation des trajectoires des navires. Ainsi, l'analyse de ces trajectoires et ces contraintes contribuent fortement à améliorer les connaissances dans le domaine de la surveillance maritime. La collaboration avec un expert de la capitainerie du port autonome de La Rochelle qui assure l'accueil des navires dans le respect de la réglementation, nous a permis de rassembler des notions et des termes liés à ce domaine et leurs relations avec les données brutes dont nous disposons. L'analyse et la synthèse de ce volume important de données brutes peuvent permettre de répondre à des questions telles que :

- quels sont les navires qui se rapprochent du port pour gérer leurs entrées ?
- quelles sont les zones d'amarrage disponibles lors de l'arrivée des bateaux ?
- quelles sont les zones spatiales critiques à forte densité du trafic ?

12. *Institut de Recherche de l'École navale*

Le centre de surveillance de la capitainerie utilise les cartes numériques couplées aux systèmes de géolocalisation qui permettent de visualiser en temps réel les positions des navires transmises par AIS ainsi que d'autres données comme la vitesse et le cap. Les données de positions ne sont pas suffisantes pour analyser les mouvements et prédire des comportements. Il apparaît cependant que la visualisation de la succession des données de ces navires, sous forme de trajectoires, améliore la compréhension des comportements grâce au couplage de la dimension spatiale avec la dimension temporelle. Pour cela, il est nécessaire de modéliser les déplacements des navires afin de définir des concepts riches telles que la notion de trajectoire et de séquence au sein d'un modèle conceptuel. Plus précisément, la visualisation basée sur un modèle avec des mécanismes de raisonnement sur des trajectoires, permettent l'enrichissement des connaissances ainsi que l'amélioration de la compréhension de situations complexes de la navigation maritime pour l'aide à la gestion et à l'observation de comportements des navires.

1.1.2.2 Capture des trames

Les données de localisation calculées par le GPS sont transmises via des transpondeurs AIS aux utilisateurs ou aux autorités compétentes pour suivre en temps réel l'évolution de l'objet mobile (figure 1.1). Un transpondeur AIS transmet la position déterminée par le GPS et des données caractérisant le moyen de transport via une antenne VHF (figure 1.2). Le transpondeur AIS reçoit les données transmises par d'autres transpondeurs AIS et permet ainsi à un utilisateur de connaître les déplacements et les positions des navires à proximité. Les données reçues peuvent être utilisées par un RADAR¹³ ou couplées à des cartes numériques dans un logiciel répondant à la norme ECDIS¹⁴ de l'OMI (Organisation Maritime Internationale).

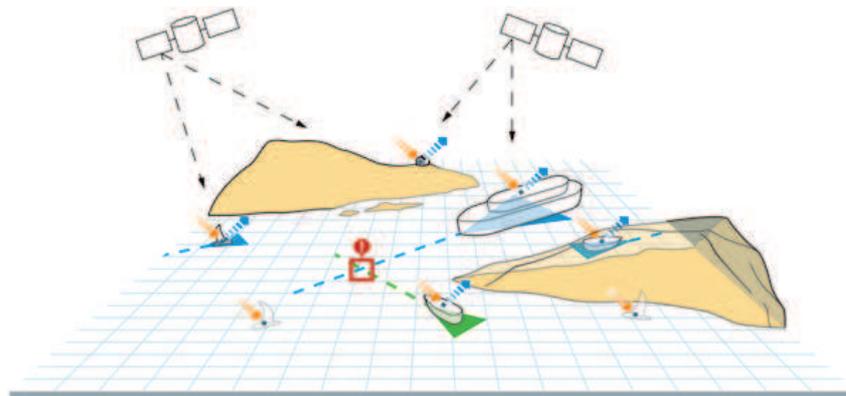


FIGURE 1.1 – Communication entre le AIS et le GPS

Parmi les données envoyées par le système AIS concernant le navire, il y a : le numéro MMSI¹⁵ (identifiant unique du navire), le statut de navigation (ex. : amarré, au

13. *RAdio Detection And Rangin*

14. *Electronic Charts Display Information System*

15. *Maritime Mobile Service Identity*

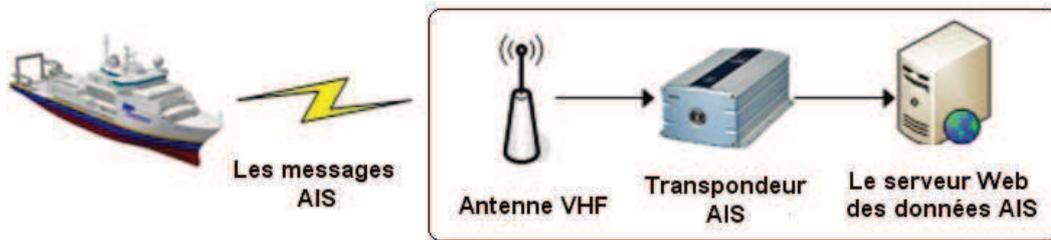


FIGURE 1.2 – Système AIS au L3i pour la capture et l'enregistrement des données

mouillage), la route fond, la vitesse fond, la position (latitude et longitude), le cap vrai, l'heure UTC de la transmission, le nom du navire, le type de bâtiment ou de la cargaison (exemple : marchandises dangereuses), les dimensions du navire, le tirant d'eau, la destination du navire, l'heure prévue d'arrivée à destination. Ces données sont soit renseignées à l'installation du système (MMSI, nom du navire, etc.), soit saisies manuellement (destination du navire, nombre d'hommes d'équipage, etc.) ou calculées automatiquement par le système.

Le laboratoire L3i possède un dispositif permettant de récupérer les trames envoyées par les AIS des navires (figure 1.2). Il comprend deux parties :

1. une partie matérielle : composée d'un récepteur muni d'une antenne. Ce récepteur est relié à un ordinateur et capture en temps réel les trames envoyées par les bateaux ;
2. une partie logicielle : développée au sein du laboratoire L3i en collaboration avec l'IRENav permet de décoder les trames capturées. Ces dernières possèdent un codage normalisé appelé NMEA 0183¹⁶. La figure 1.3 montre un exemple d'une trame de données.

```

IAIVDM,1,1,,B,19NS7Sp02wo?HETKA2K6mUM20<L=,
IAIVDM,1,1,,B,15MqvC00h9G?qinK?VIPhA480@2n,0*1
IAIVDM,1,1,,B,15Mf@6P001G?v68K??4SejL<00SI,0*71
IAIVDM,1,1,,B,15Mn4kPP01G?qNvk>:grkOv<0<11,0*5
IAIVDM,1,1,,B,15O1Pv0022o?GeNKB3f7QV2>00SP,0*
IAIVDM,1,1,,B,15MqvC00h:G?qj0K?Vp@di4B0@5>,0*
IAIVDM,1,1,,B,15NcRf0P3wG?Wq'K>α=RP?vB0<1J,0*

```

FIGURE 1.3 – Exemple de trames AIS

1.1.2.3 Analyse des trames

L'analyse de ces trames fait ressortir les trois catégories de données suivantes :

- **données sur l'objet** : le numéro MMSI (identifiant unique du navire), le type de bâtiment ou de cargaison (ex. : marchandises dangereuses), les dimensions du navire, le nom du navire, etc. ;

16. <http://www.nmea.org/>

- **données sur le voyage** : le statut de navigation (ex. : amarré, au mouillage, faisant route au moteur, à capacité de manœuvre restreinte, échoué, en pêche, handicapé par son tirant d'eau, faisant route à la voile), le tirant d'eau, la destination du navire, le nombre d'hommes de l'équipage, etc. ;
- **données spatio-temporelles** : la localisation (latitude et longitude), la date et l'heure UTC de transmission de la localisation, la vitesse sur le fond (entre 0 et 102 nœuds par pas de 0,1 nœud), la vitesse de changement de cap (de 0 à 720 degrés par minute), le cap vrai (direction en degré), l'ETA¹⁷ (estimation de l'heure d'arrivée à destination).

Suite aux discussions avec l'expert du domaine maritime, nous avons noté d'autres données telles que :

- les différents types de navires : passager, pêche, service comme le remorquer, commerce tels que Cargos et Tankers pétroliers, yachts, etc. ;
- les types de mouvements des bateaux :
 - l'appareillage : le départ, la manœuvre d'un navire qui quitte son mouillage, son poste à quai ;
 - en route : mise en route pour l'arrivée à la destination souhaitée ;
 - l'amarrage : l'action d'immobiliser un bateau en mer au moyen d'une ancre dans une zone d'abri, en utilisant des moyens de mouillage (chaîne, bosses, guindeau ou cabestan, etc.) ;
 - l'accostage : consiste pour un navire ou une embarcation à venir sans erre parallèlement à un quai ou à un autre navire afin de s'y amarrer ;
 - durant son déplacement : le bateau peut décélérer, accélérer, déraper, marcher en arrière, en roulis (un mouvement oscillation latérale du navire, le navire s'inclinant alternativement sur tribord et sur bâbord, s'il s'incline d'un côté seulement on dit qu'il gîte), en tangage (un mouvement d'oscillation d'un bateau d'avant en arrière).

D'autres données peuvent être déduites de la variation de tirant d'eau :

- en charge : à fort tirant d'eau ;
- vide : à faible tirant d'eau ;
- mi-charge : tirant d'eau moyen.

Toutes les données sont capturées à partir des trames reçues et celles fournies par l'expert vont nous permettre d'élaborer le modèle de données pour les trajectoires.

1.1.2.4 Modèle UML de la trajectoire

L'analyse et la compréhension des données nous permettent de ressortir trois principales catégories. En effet, on trouve la catégorie qui définit l'objet mobile, une deuxième catégorie qui regroupe un ensemble de données spatio-temporelles contenant entre autres la trajectoire du bateau et enfin une catégorie pour le contexte. L'étude approfondie des propriétés de chacune de ces trois catégories nous amène au diagramme de classes du domaine dont un extrait est donné par la figure 1.4. Le tableau 1.2 donne une description des principales composantes de ce diagramme de classes.

17. *Estimated time of arrival*

Le modèle 1.4 présente les données statiques de la trajectoire. Les données dynamiques ou les contraintes, telles que les actions du navire, doivent être prises en charge autrement, en particulier à base de règles. Cette partie sera développée dans un chapitre dédié.

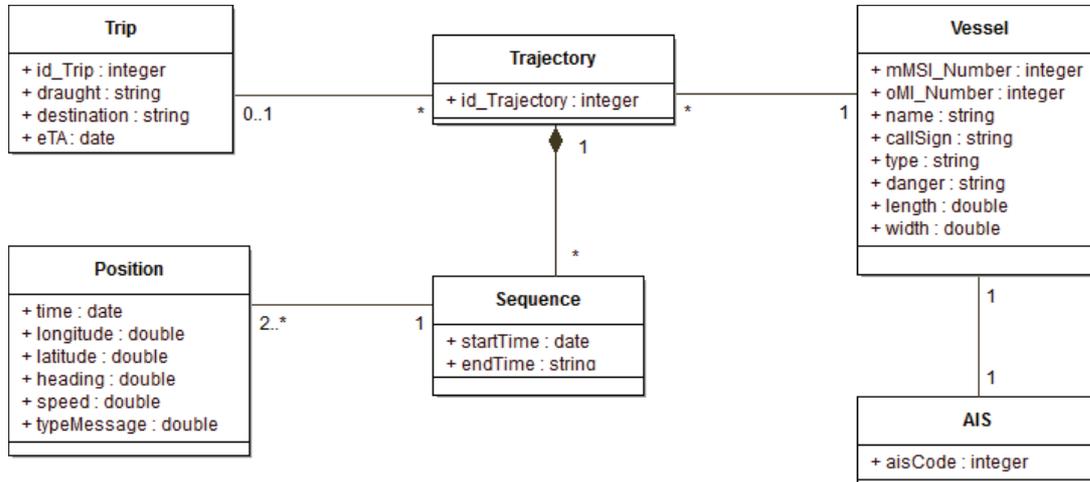


FIGURE 1.4 – Extrait du diagramme de classes du domaine de la trajectoire du navire

Catégorie	Classes	Description
Objet mobile	Vessel, AIS	L'objet mobile équipé d'un capteur AIS identifié par un code unique.
Trajectoire	Trajectory, Sequence, Position	Une trajectoire est une sorte de séquence limitée par un intervalle temporel défini par un début et une fin. Une trajectoire peut être décomposée en un ensemble de séquences. Une séquence est une suite de positions (au minimum 2). Cependant, le contenu spatial est bidimensionnel (latitude et longitude). L'intervalle temporel d'une séquence est déduit de la donnée <code>time</code> de ses positions. La trajectoire ou la séquence généralise le type de mouvement tel que l'appareillage, en route, l'amarrage et l'accostage.
Méta-données	Trip	Le voyage constitue le contexte général du déroulement de la trajectoire. Le système de capture ne mémorise pas le contexte voyage d'une trajectoire.

TABLE 1.2 – Description des principales catégories du diagramme de classes

1.1.3 Expérience 2 : Étude des trajectoires des avions

1.1.3.1 Contexte de l'étude

Dans le domaine aérien, les avions utilisent depuis de nombreuses années des systèmes dits *Mode-A* et *Mode-C* pour la transmission des données aux radars sols. Ces deux systèmes ont montré des faiblesses apparentes en matière d'identification des sources et de la sécurité des données. Par conséquent, un système de transmission des données, dit *Mode-S*, est développé depuis 2003. Il se décline en trois versions : *Mode-S*, *ELS*¹⁸, et la version recommandée depuis mars 2007, dite *EHS*¹⁹.

Le laboratoire L3i a acquis un pack grand public appelé *SBS-1 Real-time virtual radar*, distribué par l'entreprise *Kinetic Avionic*, permettant de capturer les trames *Mode-S-EHS*. Il comprend deux parties :

1. une partie matériel : une antenne installée sur le toit pour obtenir une meilleure réception (allant jusqu'à 250 km), et un boîtier qui décode les trames reçues, puis les envois à l'ordinateur auquel il est relié ;
2. une partie logicielle : une application nommée *Base Station*²⁰ permettant d'obtenir les trames dans un format texte *CSV* et de visualiser les données capturées sur un écran radar virtuel pour suivre en temps réel les trajectoires des avions.

La sécurité aérienne étudie l'ensemble des mesures visant à réduire le risque aérien. Pour cela, l'Organisation de l'Aviation Civile Internationale (*OACI*²¹) mène des recherches continues dans la gestion du trafic et de l'aide à la décision pour éviter tout risque aérien. Par conséquent, l'analyse et la compréhension des trajectoires des avions représentent un des principaux champs de recherche pour la mise en œuvre d'un système automatisé de surveillance, de contrôle, de routage et de guidage des avions.

1.1.3.2 Capture des trames

L'architecture du système d'acquisition des trames utilisé est fermée. Seul le logiciel fournis par *Kinetic Avionic* peut communiquer avec le système *SBS-1*. Le constructeur invoque pour cela des considérations de sécurité aérienne (risque de terrorisme). Néanmoins, le constructeur a laissé une porte d'accès aux données. En effet, en mettant à jour le système *SBS-1* et le logiciel *Base Station*, ce dernier renvoie les informations reçues par les ports USB 30003 sous format texte. La figure 1.5 montre un exemple d'une trame de données capturée.

Dans le mode *S-EHS*, on obtient des données complètes concernant l'objet mobile et sa trajectoire comme le *HexIdent* code d'identification de l'appareil, le *Squawk* code du transpondeur de l'avion, le numéro de vol, la latitude, la longitude, l'altitude, etc.

18. Mode-S Elementary Surveillance

19. Mode-S Enhanced Surveillance (EHS)

20. <http://www.kinetic-avionics.co.uk>

21. en anglais : *International Civil Aviation Organization (ICAO)*

```

MSG,3,14,831,400A9F,51640,2007/10/12,15:07:37.995,2007/10/12,15:12:38.257,,38000,,45.97540,-2.40400,,U,U,U,U
MSG,4,14,194,401079,51620,2007/10/12,15:07:37.394,2007/10/12,15:12:38.257,,448.6,2.9,,0,,0,,0,,0
MSG,5,14,194,401079,51620,2007/10/12,15:07:37.394,2007/10/12,15:12:38.257,,36000,,,,,,,,0,,0,0
MSG,6,14,194,401079,51620,2007/10/12,15:07:37.394,2007/10/12,15:12:38.257,,36000,,,,,,,,7453,0,0,0,0

```

FIGURE 1.5 – Exemple de trames *Mode-S-EHS*

1.1.3.3 Analyse des trames

L'analyse des trames capturées, comme celles données par la figure 1.5, fait ressortir les trois catégories de données suivantes :

- **données sur l'objet** : la donnée la plus intéressante est le *HexIdent* numéro d'immatriculation de l'avion (champ 5), désigné par l'organisation *ICAO*;
- **données sur le vol** : comme le *CallSign* (champ 11) qui correspond au numéro de vol qui doit être stipulé sur le plan de vol de l'avion ;
- **données spatio-temporelles** : la date et l'heure de génération d'une trame (champ 7 et 8), l'altitude (en pieds, champ 12), la vitesse au sol (en nœuds, champ 13), le cap (en degré, champ 14), la latitude (en degré, champ 15), la longitude (en degré, champ 16), la vitesse verticale (en pieds par mn, champ 17).

Les autres champs déterminent le codage des données présentent dans les trames (les champs 1 et 2) où ils sont générés par la tierce partie logicielle qui nous permet l'acquisition de ces trames. Les champs 5 et 11 nécessitent un décodage puis l'interrogation de sources de données externes pour extraire leur sens (pays d'origine de l'avion, sa compagnie, etc.).

Selon l'expert du domaine, ils existent trois principaux types de mouvement lors du trajet d'un avion (figure 1.6) :

- le roulis : l'avion effectue une rotation sur son axe longitudinal ;
- le tangage : l'avion effectue une rotation sur son axe transversal ;
- le lacet : c'est le dernier type de mouvement qu'un avion peut exécuter pour diriger l'avion, en général, pour virer. On le combine avec le roulis. Le lacet est également utilisé pour compenser les effets d'un vent de travers, ou pour maintenir l'avion dans l'axe de la piste au décollage et à l'atterrissage.

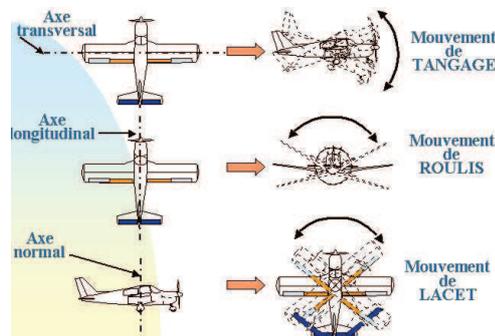


FIGURE 1.6 – Types de mouvements d'un avion

Au cours de son vol, l'avion effectue trois types d'actions notamment « en montée », « en descente » et « en altitude stable ». La variation des types de trajectoire de l'avion

est défini grâce à la variation d'un ensemble de caractéristiques notamment la direction, la vitesse ainsi que les coordonnées spatiales et temporelles.

1.1.3.4 Modèle UML de la trajectoire

L'analyse des trames nous permet de ressortir trois catégories principales de données : un objet mobile, un contexte et un ensemble de données spatio-temporelles qui contient entre autres la trajectoire de l'objet. L'étude approfondie des propriétés de chacune des trois catégories nous amène au diagramme de classes du domaine dont un extrait est donné par la figure 1.7. Le tableau 1.3 donne une description des principales composantes de ce diagramme de classes.

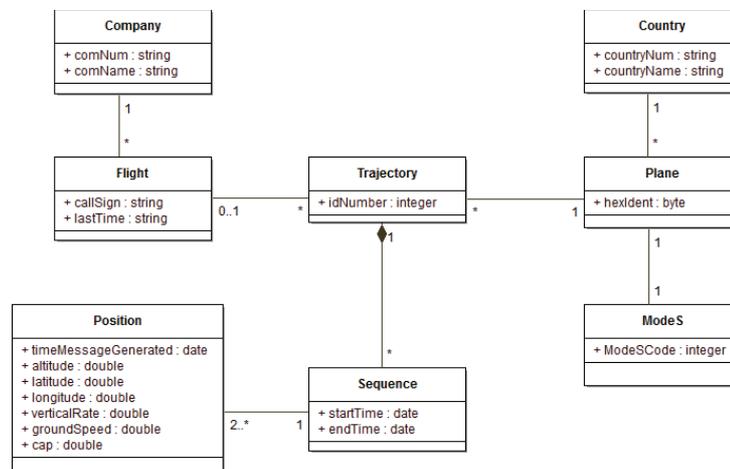


FIGURE 1.7 – Extrait du diagramme de classes du domaine de la trajectoire de l'avion

TABLE 1.3 – Description des principales catégories du diagramme de classes

Catégorie	Classes	Description
Objet mobile	Plane	L'objet mobile étudié identifié par un hexadécimale <code>hexIdent</code> formé par 24 bits.
Trajectoire	Trajectory, Sequence, Position	Ce triplet a la même définition donnée dans le cas du navire. L'intervalle temporel d'une séquence est déduit de la donnée <code>timeMessageGenerated</code> de ses positions. La trajectoire ou la séquence généralise les types de mouvement tels que le tangage, le roulis et le lacet.
Méta-données	Flight, Compagny, Country	Chaque vol est identifié par un indicatif appelé ici <code>callSign</code> . Cette donnée renseigne la compagnie affrétant le vol. Le vol est effectué par un avion. Le système ne mémorise pas forcément le vol d'une trajectoire et inversement.

Comme dans le cas des navires, le modèle 1.7 présente les données statiques de la trajectoire. Les données dynamiques ou les contraintes, telles que les actions de l'avion, doivent être prises en charge par des règles. Cette partie sera développée dans un chapitre dédié.

1.2 Modèle générique de la trajectoire

1.2.1 Objectif

L'objectif de cette étape est de représenter la trajectoire d'une façon simple et générique. Notre démarche consiste à catégoriser les principales entités représentant les parties communes à toutes les trajectoires étudiées et ainsi proposer un modèle ou un patron qui encapsule ces entités communes. Lors de sa réutilisation, ce patron peut être adapté à d'autres applications par la suppression ou l'ajout d'éléments pour maintenir des exigences supplémentaires. Par conséquent, on peut modifier la structure du patron en le paramétrant par des types ou des valeurs afin de permettre l'utilisation du même composant dans des situations différentes.

Cette représentation générique doit être indépendante du domaine d'application. L'objectif est de définir et de modéliser les notions fondamentales de la trajectoire de manière à permettre :

- la prise en compte de l'essentiel est primordiale pour la définition d'une trajectoire ;
- garantir une réutilisation facile et rapide du patron dans des contextes différents.

1.2.2 Domaine de la trajectoire

Le domaine de la trajectoire s'intéresse aux aspects liés aux changements des données. En examinant ces changements dans les applications considérées, on en déduit que le dénominateur commun des domaines des trajectoires est formé par des composantes spatiales, temporelles et spatio-temporelles. Le diagramme de classes 1.8 présente le modèle générique d'une trajectoire en précisant ses composantes, avec :

- **Position** : l'unité spatio-temporelle de base. Elle est caractérisée par ses coordonnées spatiales et son temps de référence. Elle représente un point dans un espace spatio-temporel. La position peut être incluse dans une zone. La zone est un concept défini par des coordonnées spatiales. Elle représente la portion du territoire dans laquelle la position de l'objet mobile est incluse ;
- **Sequence** : l'ensemble des positions sous forme d'intervalles spatio-temporels. Cette entité est structurée en deux parties. Une partie spatialement localisée appelé **Geo-Sequence** qui prendra en charge les spécificités de la trajectoire. Une seconde partie **Specific-Sequence** représente les métadonnées liées à une trajectoire et peut être spatialement localisée. Cette composante fournit des données extrinsèques intéressantes qui peuvent être nécessaires pour l'analyse des trajectoires ;

- **Trajectory** : structuration logique de l'ensemble des séquences, en particulier les composantes **GeoSequence**. Cette entité permet de considérer la trajectoire d'un point de vue globale dans un contexte plus large que celui des séquences.

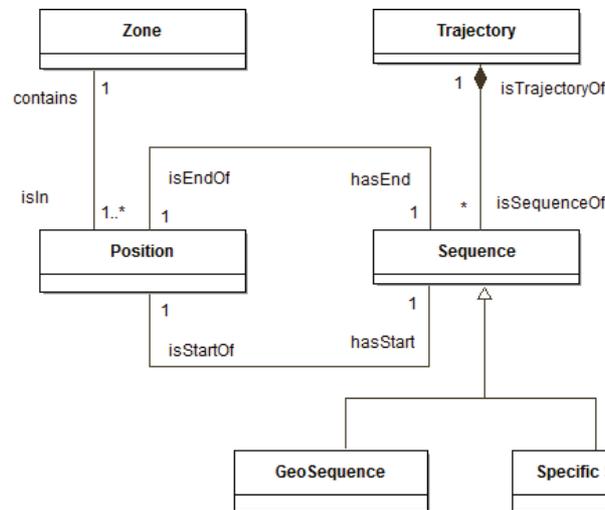


FIGURE 1.8 – Modèle conceptuel générique de la trajectoire

1.2.3 Domaine de l'objet mobile et de la trajectoire

Par définition, les objets en mouvement ou les objets mobiles sont des entités dont les positions ou les attributs évoluent au cours du temps. Cependant, dans de nombreuses applications, la dimension de l'objet n'est pas aussi importante que sa position. Par conséquent, les objets en mouvement sont considérés comme des points mobiles, dont les trajectoires peuvent être visualisées et analysées. Dans tous les cas, les objets mobiles peuvent être classés en deux grands groupes d'objets dynamiques géo-référencés et non-géo-référencés [Dodge *et al.*, 2008]. En d'autres termes, certains sont des objets dynamiques qui se déplacent dans l'espace et peuvent donc être référencés géographiquement comme les humains, les animaux, ou des véhicules, tandis que l'autre groupe comprend des phénomènes dynamiques qui se déplacent dans un espace non-géographique, y compris les mouvements du regard dans les études de mouvements oculaires ou des particules dans une chambre à bulles. Chacun de ces objets dynamiques, à des degrés divers, partagent certaines similarités mais présentent aussi des différences en termes de structure de données, de comportement dynamique et de la nature du mouvement.

Dans le modèle que nous présentons, la classe d'objet mobile **Mobile Object** représente l'objet équipé d'un ou plusieurs capteurs. La classe de déploiement **Deployment** décrit l'utilisation spatio-temporelle du capteur et de l'objet capturé (figure 1.9). L'association entre les deux domaines est réalisée entre les deux classes **Trajectory** et **Mobile Object**.

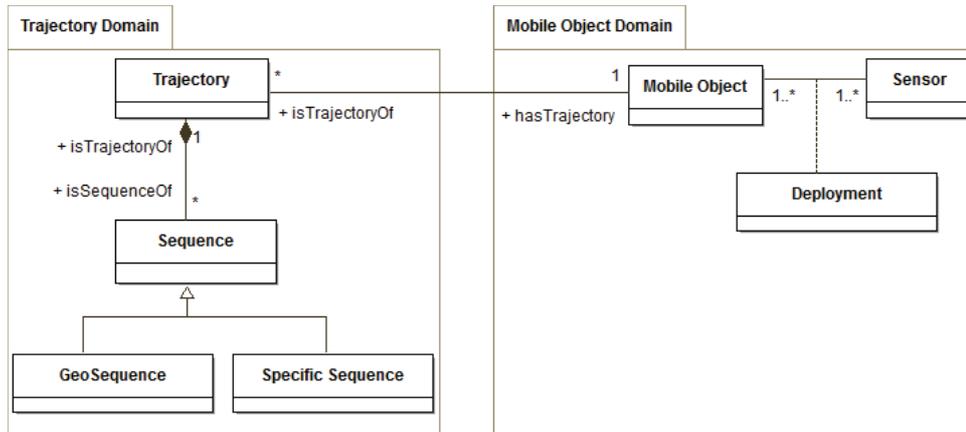


FIGURE 1.9 – Domaine de l’objet mobile et sa relation avec le domaine de la trajectoire

1.3 Modèle de la trajectoire sémantique

1.3.1 Annotation

Dans le langage courant, et d’après le dictionnaire « Le Robert », une annotation est une : *Note critique ou explicative qu’on écrit sur un texte, un livre, etc..* En informatique, il n’existe pas de définition générale ou commune pour définir une annotation. À titre d’exemple, la communauté du Web Sémantique définit l’annotation comme étant un : *Commentaire (libre) rattaché à un document ou une portion de document.* Dans le domaine de la programmation, une annotation est aussi confondue avec le concept de méta-données. Dans le domaine qui nous intéresse, l’ingénierie des données, nous définissons une annotation comme étant : *une description sémantique de plus haut niveau que la donnée décrite. Elle peut être formelle, exprimée dans le même langage ou un langage différent de la donnée associée.*

Dans l’état de l’art, on distingue trois types de processus d’annotation : manuelle, semi-automatique et automatique. La première est effectuée manuellement par un humain chargé d’attribuer les mots clés. Ce type d’annotation augmente la précision mais baisse la productivité. L’annotation automatique est une tâche effectuée par une machine. Ce type d’annotation baisse la précision de la tâche et augmente sa productivité. L’annotation semi-automatique est un compromis entre l’annotation automatique et manuelle.

1.3.2 Trajectoire sémantique : trajectoire annotée

Une trajectoire sémantique est pour nous une trajectoire annotée. Plus précisément, c’est un ensemble de séquences spatio-temporelles annotées. Ces annotations proviennent des connaissances du domaine et on parle alors d’une séquence sémantique. En définitif, une trajectoire sémantique est un ensemble de séquences sémantiques.

Dans le cadre de notre travail, on définit le modèle de trajectoire sémantique par l’association entre le domaine sémantique et le domaine de la trajectoire. Le modèle 1.10

illustre cette association. Le domaine sémantique est constitué par les activités du domaine avec différentes granularités : activité générale et activité de base hiérarchique, selon l'objet spatio-temporel associé.

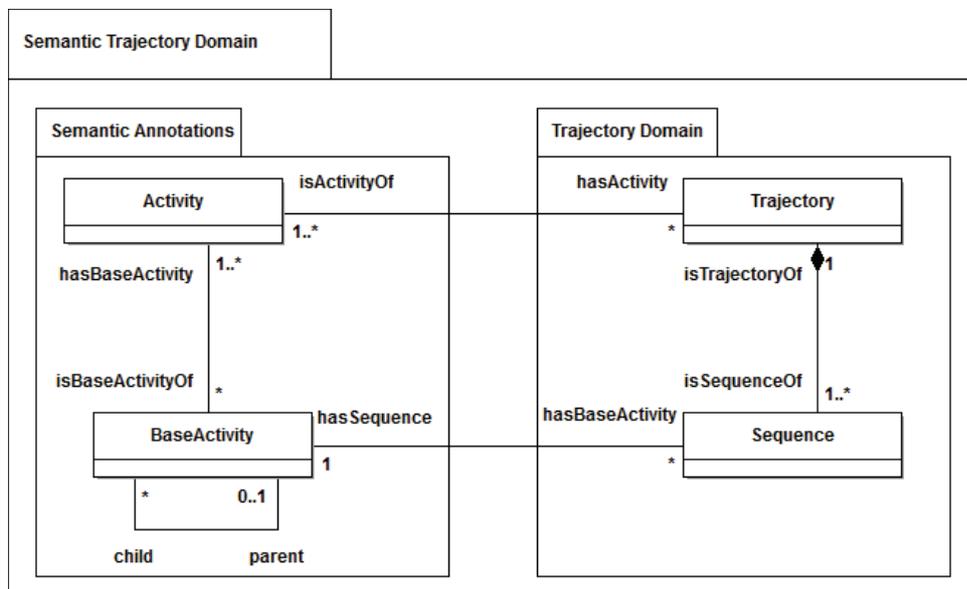


FIGURE 1.10 – Modèle de la trajectoire sémantique

En général, les trajectoires forment une masse de données importantes. Il est donc impensable d'utiliser une annotation manuelle basée sur la visualisation par exemple. On pense alors aux autres types d'annotations. Dans notre cas, on utilise une approche d'annotation automatique supervisée par des experts du domaine afin d'obtenir une annotation de bonne qualité.

1.4 Modèle ontologique de la trajectoire

L'objectif de notre travail est d'intégrer des couches sémantiques métiers aux données. Cependant, l'exploitation des modèles de données, que nous avons établie en UML, est limitée à cause de la complexité dûe à l'ajout des contraintes dans les modèles pour satisfaire totalement les exigences des définitions sémantiques de données. Il existe plusieurs approches qui répondent à l'intégration sémantique comme les méthodes statistiques ou probabilistes pour la description des phénomènes complexes du monde réel et la prise de décision. Dans ce travail, nous avons mis le choix sur l'utilisation du langage ontologique OWL. En effet, ce langage permet une définition stricte de la sémantique qui permettra aux machines d'effectuer des raisonnements automatisés basés les inférences. OWL est un langage standard pour les ontologie proposé dans le cadre du W3C. Il a non seulement la capacité de décrire les concepts dans un domaine mais aussi d'un ensemble plus riche d'opérateurs qui assure la description complète des concepts. De plus, notre choix pour le langage OWL est justifié par la politique scientifique du groupe de travail interne au

laboratoire L3i, qui a choisi le langage OWL comme langage de formalisation des ontologies. Dans notre cas, le diagramme de classes annoté (figure 1.10) constitue le point de départ pour la construction de la partie déclarative de l'ontologie. Nous optons pour une transformation automatique de ce modèle en une ontologie formalisée en OWL. La communauté de l'ingénierie des modèles (IDM) s'est intéressée à cette question et plusieurs approches et outils plus au moins performants ont vus le jour [Faucher *et al.*, 2008]. Nous utilisons le transformateur de modèle appelé *Eclipse uml2owl* [Hillairet, 2007] développé au sein du laboratoire L3i. Ce transformateur, qui repose sur le méta-modèle *eCore* d'Eclipse, prend en entrée un diagramme de classes UML, typiquement un diagramme de classes produit avec l'AGL *Topcased* par exemple, et le transforme en une ontologie OWL-DL. Les transformateurs de modèles possèdent des limites. Les limites du transformateur que nous utilisons viennent d'une part des limites du méta-modèle *eCore* comme par exemple l'absence des classes associations, ou alors des concepts présents en UML qui n'ont pas une équivalence directe en OWL comme les associations de composition. Ces limites doivent être corrigées manuellement sur l'ontologie. La figure 1.11 présente un extrait de l'ontologie finale que nous appelons *owlSemanticTrajectory*.

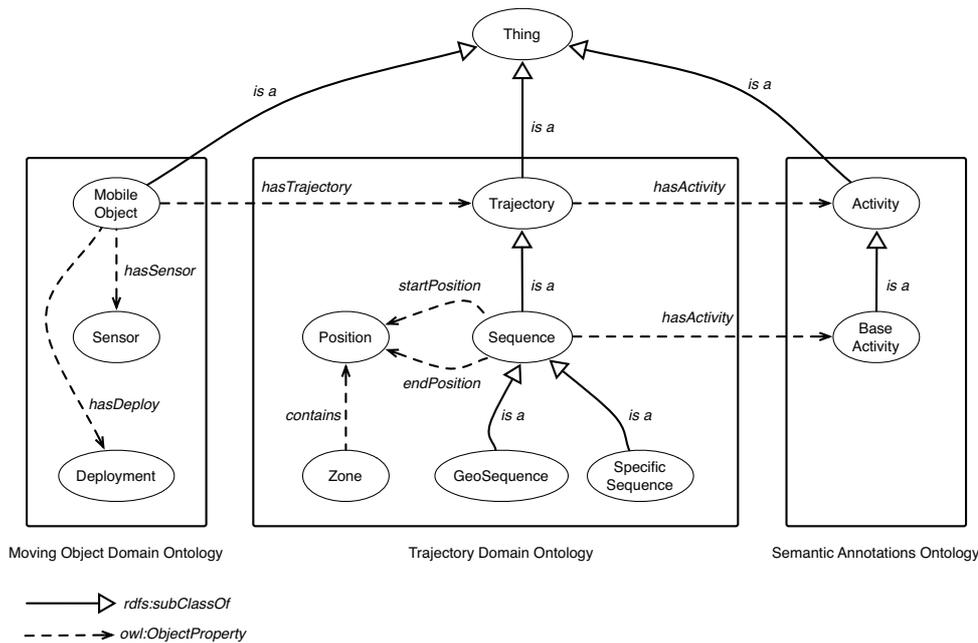


FIGURE 1.11 – Extrait de l'ontologie *owlSemanticTrajectory*

La partie impérative de cette ontologie contient entre autres, les actions ou les règles qui définissent les activités de l'objet mobile. La mise au point de cet ensemble nécessite un langage impératif dépendant du raisonneur utilisé que nous détaillons dans la partie mise en œuvre.

1.5 Synthèse

En résumé, la figure 1.12 montre notre démarche. Elle part de l'étude des trajectoires de navires et d'avions. À partir de l'analyse des données obtenues par des capteurs hétérogènes, nous avons proposé un modèle conceptuel générique de la trajectoire, présenté en UML. Utilisant un processus d'annotation, dans notre cas en s'appuyant sur les objectifs du déplacement de l'objet mobile, nous avons enrichi le modèle de trajectoire par des annotations qui seront associées à des règles métiers pour aboutir à un modèle de la trajectoire sémantique. Ce dernier modèle est la base de l'ontologie de la trajectoire `owlSemanticTrajectory`. La partie déclarative de l'ontologie contient les concepts et les propriétés définissant les données de la trajectoire. La partie impérative contient les actions ou les règles métiers.

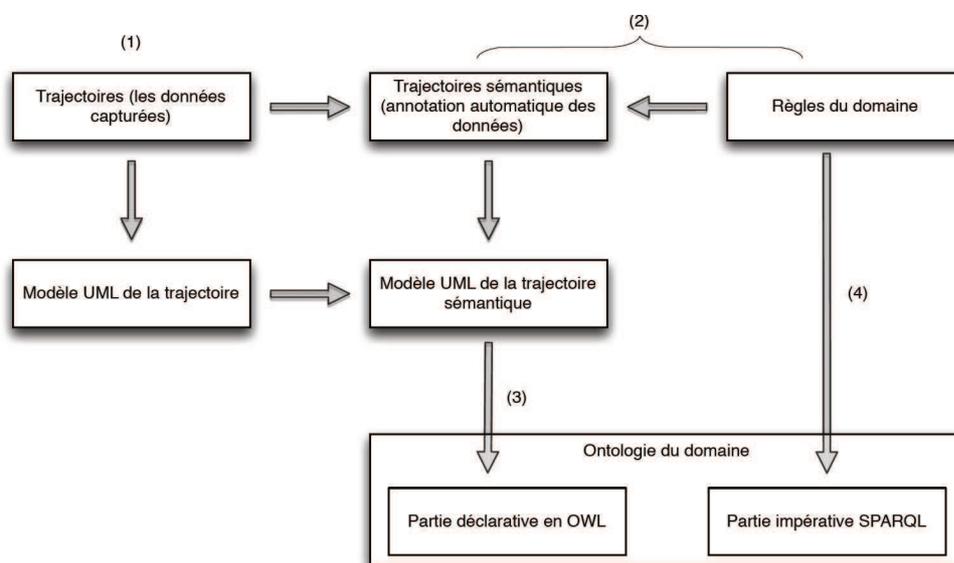


FIGURE 1.12 – Démarche proposée : des données brutes vers une ontologie de trajectoire

Chapitre 2

Réutilisation des ontologies dans l'ontologie de la trajectoire

Sommaire

2.1	Introduction	109
2.2	Réutilisation de l'ontologie temps	110
2.3	Réutilisation de l'ontologie espace	111
2.4	Correspondance entre des ontologies	111
2.5	Synthèse	112

2.1 Introduction

L'ontologie `owlSemanticTrajectory` est formée par trois types de concepts : thématique, temporel et spatial. Pour aboutir à une modélisation complète et donc un raisonnement fiable, nous devons considérer pour chaque type de concept, et en particulier le type spatial et temporel, la possibilité de le connecter à un modèle ontologique existant.

L'apport majeur de la modélisation ontologique des trajectoires est la possibilité de réutiliser des ontologies existantes, et de préférence, celles qui sont testées et prouvées. D'une manière générale, la réutilisation des ontologies peut être partielle ou totale. Dans la réutilisation partielle, on parle d'assemblage, d'extension, de spécialisation ou d'adaptation d'autres ontologies qui seront donc des parties de l'ontologie cible. Dans la réutilisation totale, on parle de fusion de différentes ontologies portant sur le même sujet ou des sujets similaires pour former une seule ontologie. De notre point de vue, la réutilisation des ontologies pose deux problèmes :

1. le problème du choix de l'ontologie à réutiliser ;
2. le problème de la correspondance (ou *mapping*) entre les ontologies.

Plusieurs travaux de recherche ont apporté des solutions à ces questions. Dans [Pinto and Martins, 2000], les auteurs proposent une méthodologie pour la réutilisation des ontologies

et consacrent toute une partie à la question des choix des ontologies à réutiliser. La question de définition de correspondance entre des ontologies a fait l'objet de toute une étude réalisée dans les travaux de [Shvaiko and Euzenat, 2013].

Dans ce travail, nous adoptons l'approche de la réutilisation partielle. En effet, la fusion des ontologies dans l'approche de la réutilisation totale a des conséquences négatives sur le processus d'inférence, comme nous le montrerons dans la partie mise en œuvre de ce travail. Pour résoudre le problème du choix de l'ontologie à réutiliser, nous introduisons une méthodologie basée sur le principe de définition des besoins. Elle repose sur deux principes :

1. à partir de l'ontologie de la trajectoire, on énumère les concepts et les relations qui peuvent être assimilés à des concepts et des relations temporelles et spatiales ;
2. à partir de l'étude des cas d'utilisation, on identifie les besoins en termes de concepts et de relations temporelles et spatiales.

Les ontologies du temps et de l'espace à réutiliser doivent au minimum prendre en compte les exigences temporelles et spatiales identifiées à la fin de ce chapitre.

2.2 Réutilisation de l'ontologie temps

L'application du principe de définition des besoins pour la réutilisation d'une ontologie du temps fait ressortir les concepts temporels : *instant* et *intervalle*. L'identification des relations temporelles conduit à la considération de l'ensemble des relations de l'algèbre temporelle d'Allen [Allen, 1983b].

Comme dans [Peralta *et al.*, 2004], nous avons analysé et testé différentes ontologies candidates à une réutilisation. En particulier *Simple Time Ontology*²² du serveur d'ontologies *Ontolingua* [Gruber, 1992], et les grandes ontologies comme *WordNet* [Miller, 1995], *upperCyc* [Microsystems, 2001], *SUMO* [Niles and Pease, 2001] et *LKIF-Core Ontology* [Hoekstra *et al.*, 2007]. Le point commun à toutes ces ontologies est la théorie du temps d'Allen. Dans les grandes ontologies, les concepts temporels sont éparpillés. Le manque d'organisation de ces ontologies en sous ontologies rend leur réutilisation difficile. La réalisation *Simple Time Ontology* représente une meilleure candidate dans notre cas, car elle est structurée et organisée autour des concepts temporels, mais le manque de documentation pour cette ontologie la rend difficilement réutilisable. L'ontologie *OWL-Time* [Hobbs and Pan, 2004b] développée au sein du consortium W3C, bénéficie d'une spécification précise, se consacre aux concepts et relations temporels comme définis dans la théorie d'Allen, formalisée en OWL, est donc certainement une meilleure candidate. La figure 2.1 montre un extrait de cette ontologie. La partie déclarative de l'ontologie *OWL-Time* peut être consultée à l'URL suivante <http://www.w3.org/2006/time>. La définition de la partie impérative de l'ontologie *OWL-Time* est discutée dans la section mise en œuvre.

22. *Simple Time Ontology* est accessible à travers du site <http://ontolingua.stanford.edu>

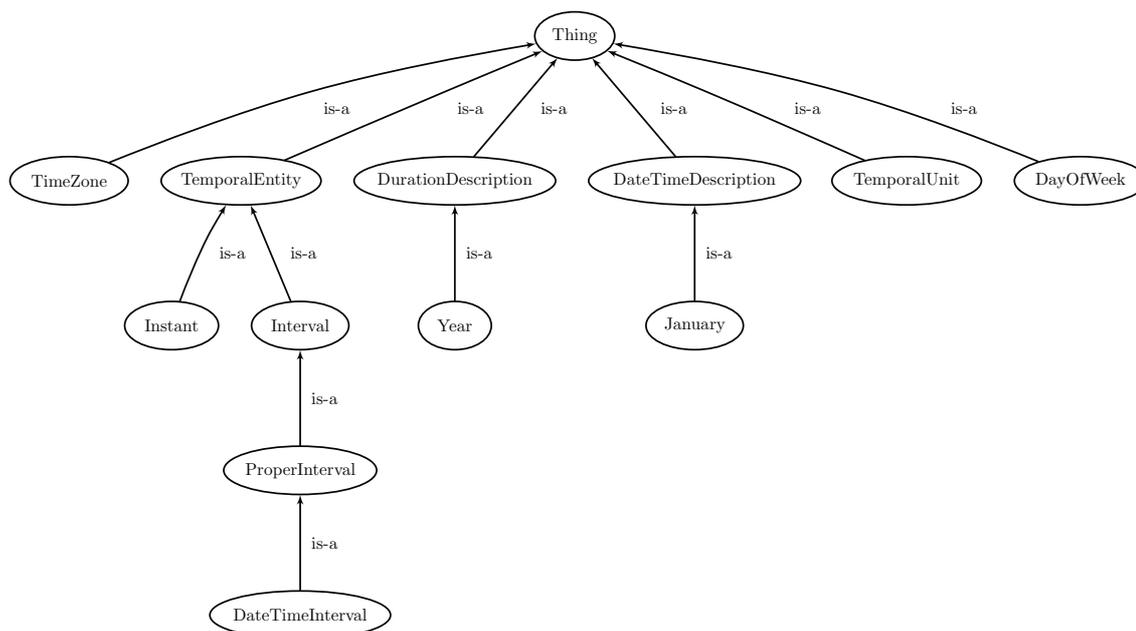


FIGURE 2.1 – Les concepts et hiérarchie de l'ontologie OWL-Time

2.3 Réutilisation de l'ontologie espace

L'application du principe de définition des besoins pour la réutilisation d'une ontologie de l'espace fait ressortir les concepts spatiaux : point, ligne, polygone ainsi que les concepts multi-points, multi-lignes et multi-polygones. L'identification des relations spatiales conduit à la considération de l'ensemble des relations spatiales : Equals, Disjoint, Intersects, Overlaps, Contains, Crosses, Within, Touches.

Cette analyse nous amène à considérer la norme *OGC OpenGIS* [Beddoe *et al.*, 1999] pour les objets spatiaux et les fonctions agissant sur ces objets. Cette norme contient une définition précise des classes et des prédicats spatiaux ainsi que les systèmes de référence. Nous utilisons la technique de transformation de modèle pour définir la partie déclarative de l'ontologie de l'espace à partir du diagramme de classes UML de la spécification OGC. Nous appelons cette ontologie *owlOGCSpatial* (figure 2.2). Nous discutons la partie impérative de cette ontologie dans la section mise en œuvre.

2.4 Correspondance entre des ontologies

Le langage du web sémantique OWL définit le constructeur `owl:imports` qui fournit un mécanisme de type `include`. L'importation des ontologies OWL-Time et owlOGCSpatial fournit l'ensemble de leurs déclarations et de leurs assertions dans l'ontologie owl-SemanticTrajectory. Dans ce cas, on peut utiliser les constructeurs du langage OWL-DL : `rdfs:subClassOf` et `owl:equivalentClass` (resp. `owl:equivalentProperty`). Le constructeur `rdfs:subClassOf` relie une classe spécifique à une classe générale. Le constructeur `owl:equivalentClass` (resp. `owl:equivalentProperty`) sert à indiquer que les deux

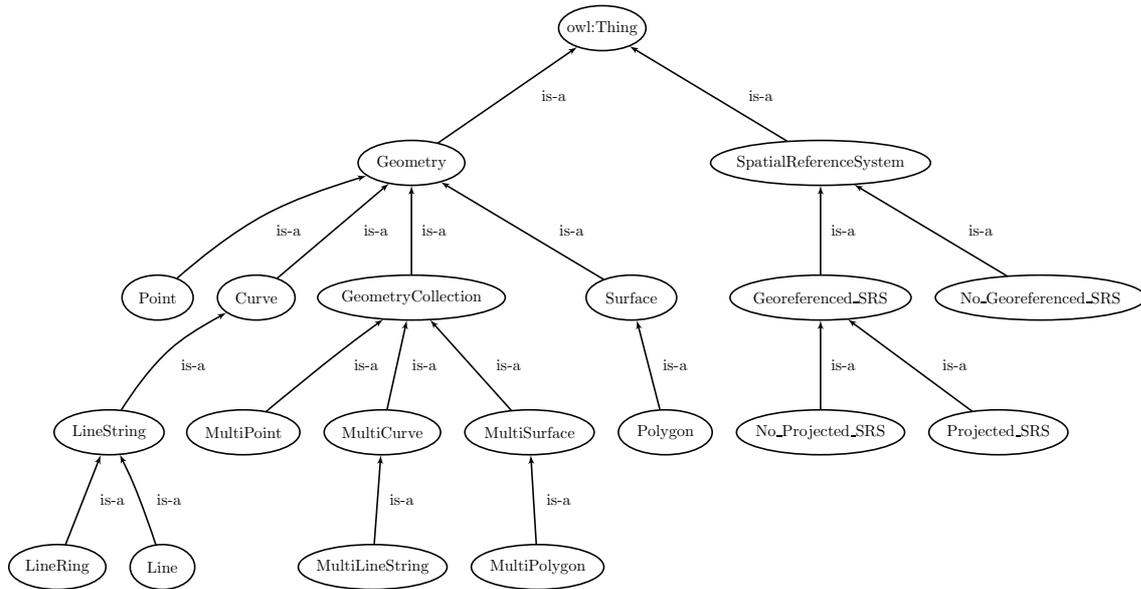


FIGURE 2.2 – L'ontologie spatiale owl:GCSpatial

classes reliées (resp. les deux propriétés objets) ont les mêmes instances. Dans *OWL-DL*, les classes représentent des ensembles d'individus et elles ne sont pas elles-mêmes des individus. Par contre, dans *OWL-Full*, on peut employer le constructeur `owl:sameAs` entre deux classes pour indiquer qu'elles sont identiques en tout point.

Le constructeur `rdfs:subClassOf` ne convient pas dans notre cas, puisqu'il définit une réutilisation totale. Le constructeur `owl:sameAs` peut être intéressant mais il définit une relation très forte entre les concepts des ontologies. Cela peut avoir des conséquences sur le processus d'inférence puisqu'il fait appel à un ensemble de règles non réductible. Par conséquent, on utilise les constructeurs `owl:equivalentClass` et `owl:equivalentProperty`. La figure 2.3 illustre le principe de réutilisation de *OWL-Time* dans *owlSemanticTrajectory* et la figure 2.4 illustre la réutilisation de *owlGCSpatial* (le signe = désigne le constructeur `owl:equivalentProperty`).

2.5 Synthèse

La réutilisation des ontologies existantes permet de faciliter la modélisation et d'améliorer le raisonnement. L'objectif de la réutilisation est l'élaboration d'ontologies de meilleure qualité, tout en réduisant les coûts de développement. En effet, la construction d'une nouvelle ontologie est un travail compliqué qui consomme du temps. Toutefois, le processus de réutilisation doit être bien étudié pour choisir l'opération adéquate entre fusion, intégration, correspondance, etc. Vu le besoin de considérer les concepts et les opérateurs spatiaux et temporels dans notre travail, nous avons proposé la solution de la réutilisation des ontologies standards. Avant de pouvoir importer et réutiliser des parties d'ontologies existantes dans l'ontologie de la trajectoire, nous avons proposé une démarche en trois étapes :

- identifier des ontologies candidates à la réutilisation dans le domaine du temps et de l'espace ;
- localiser dans les ontologies choisies les concepts du temps et de l'espace adéquats ;
- appliquer et tester la correspondance qui lie les différentes ontologies.

Basé sur cette démarche, nous avons détaillé le principe de réutilisation de l'ontologie du temps OWL-Time ainsi que celle de l'espace owlGCSPatial dans l'ontologie de la trajectoire owlSemanticTrajectory.

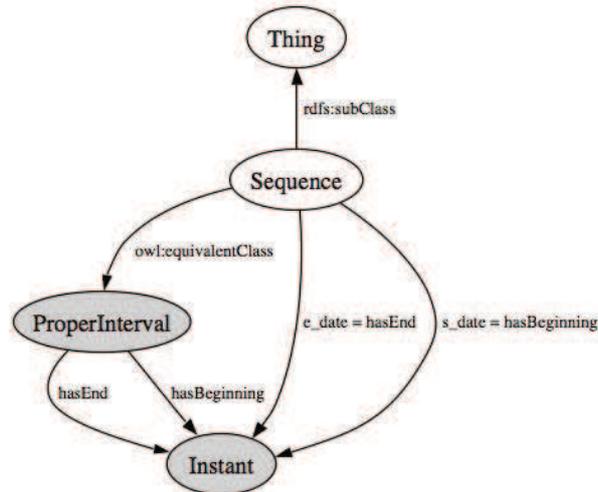


FIGURE 2.3 – Réutilisation de l'ontologie OWL-Time dans owlSemanticTrajectory

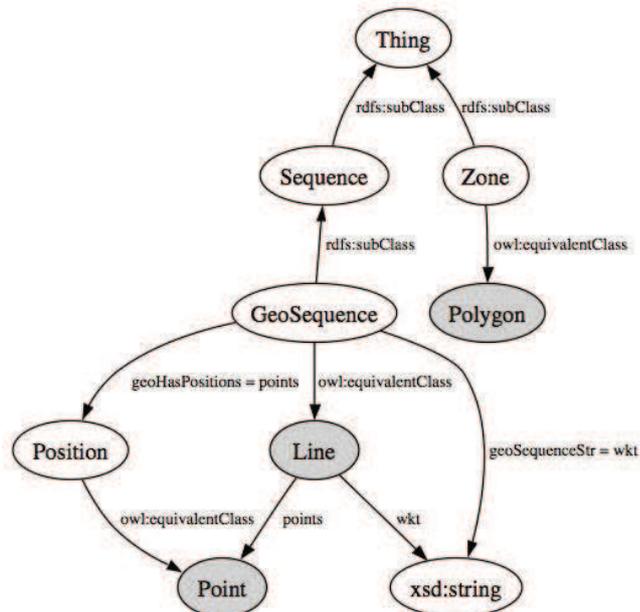


FIGURE 2.4 – Réutilisation de l'ontologie owlGCSPatial dans owlSemanticTrajectory

Chapitre 3

Ontologie de la trajectoire : cas des mammifères marins

Sommaire

3.1	Expérience 3 : Étude des trajectoires des mammifères marins . . .	115
3.1.1	Capture des trames	116
3.1.2	Analyse et compréhension des trames	117
3.1.3	Modèle UML de la trajectoire	118
3.1.4	Modèle de la trajectoire sémantique	120
3.2	Intégration du domaine dans l'ontologie de la trajectoire	123

Ce chapitre s'appuie sur la démarche et sur le modèle de trajectoire sémantique introduits au chapitre précédant pour traiter un troisième cas étude. Alors que les objets mobiles dans les deux premiers cas d'étude exposés sont des navires et des avions, nous nous intéressons ici à la trajectoire de mammifères marins. A la fin de ce chapitre, nous allons montrer comment l'ontologie de niveau supérieur `owlSemanticTrajectory` est spécialisée afin d'obtenir une ontologie de trajectoire des phoques, appelée `owlSealOntology`.

3.1 Expérience 3 : Étude des trajectoires des mammifères marins

Dans cette partie, notre étude porte sur les trajectoires des mammifères marins. Les données de ces trajectoires proviennent de l'équipe de recherche AMARE (réponse des Animaux Marins à la variabilité Environnementale) qui travaille sur la capacité des animaux marins à faire face aux modifications de leur environnement. Cette équipe fait partie de l'unité de recherche LIENSs²³ de l'université de La Rochelle. L'objectif principal de cet équipe est l'étude du comportement des phoques de la manche. Répartis sur plusieurs îles

23. *Littoral Environnement et Sociétés* : <http://lienss.univ-larochelle.fr>

de la côte de la Bretagne, ces pinnothères font des voyages jusqu'aux îles britanniques. Cependant, le mouvement sans contraintes de cette population est encore relativement peu connu. Ainsi, l'analyse des trajectoires des phoques contribue fortement à améliorer les connaissances dans ce domaine en apportant des réponses à des questions simples comme par exemple celles qui s'intéressent à l'activité de chasse :

- quand les phoques chassent-ils (définir les zones temporelles de chasse) ?
- où les phoques chassent-ils (définir les zones spatiales de chasse) ?
- quelles sont les zones spatiales où la densité de la chasse est très élevée ?

D'autres questions concernant cet animal sont aussi importantes, par exemple :

- quels sont les habitats fréquentés par les phoques ?
- où font-ils des escales au cours de leurs voyages ?

L'étude de ces trajectoires suit la démarche utilisée dans les deux cas précédents, à savoir les trajectoires des bateaux et des avions. Nous allons ainsi démontrer, d'une part que notre démarche est applicable à un autre domaine d'application, et d'autre part que les modèles élaborés peuvent servir comme cadre de modélisation pour les nouvelles données. En résumé, dans ce qui suit nous bordons :

- la capture des trames ;
- l'analyse et la compréhension des trames capturées ;
- le modèle UML de la trajectoire ;
- le modèle de la trajectoire sémantique.

3.1.1 Capture des trames

Pour capturer leurs trajectoires, un certain nombre de mammifères marins sont équipés de petites balises appelées :

- GPS phone Tag : capteur basé sur le système GPS ;
- SRDL²⁴ Tag : capteur basé sur le système ARGOS (figure 3.1) et fabriqué par le *Sea Mammal Research Unit*²⁵. Chaque balise est constituée d'un émetteur ARGOS, relié à une antenne émettrice, de plusieurs capteurs (turbine enregistrant la vitesse de nage, capteur d'immersion enregistrant le milieu - air ou eau - dans lequel l'animal se trouve, et enregistreur de pression hydrostatique) reliés à un microprocesseur. Le tout est alimenté par une ou deux batteries au lithium et noyé dans de la résine. La balise obtenue est testée pour résister à des pressions équivalentes à des profondeurs de 500 mètres, et l'ensemble pèse 340 grammes. Les balises émettent toutes les 40 secondes des signaux de fréquence fixe, lorsque l'animal est en surface. Quand la balise est immergée, l'émission est stoppée, ce qui permet de préserver l'énergie des batteries. Seule la première partie de ces signaux est dédiée à l'identification de la balise. Le reste des messages est donc disponible pour les données comportementales et environnementales, transmises par le processeur qui les a préalablement synthétisées selon la programmation élaborée par les utilisateurs.

Ces balises enregistrent jusqu'à 6 mois de données (intrinsèques et extrinsèques) sur les mammifères marins dans leur environnement (localisation, température, vitesse, pro-

24. *Satellite Relay Data Logger*

25. University of St Andrews : <http://www.smru.st-andrews.ac.uk>



FIGURE 3.1 – Balise SRDL

fondeur, etc.). Les balises peuvent transférer les données à travers le réseau mobile GSM-GPRS ou par satellite. En effet, ces données sont stockées dans la mémoire de ce système et elles sont transmises régulièrement. L'équipe AMARE met à notre disposition des bases de données MS-ACCESS contenant les trajectoires capturées de certains mammifères.

3.1.2 Analyse et compréhension des trames

Nous analysons les données capturées pour construire un modèle de trajectoire du phoque (un diagramme de classes UML). Pour faire cette analyse, nous nous appuyons sur la documentation fournie par le fabricant du capteur (dictionnaire des données), les données réellement disponibles et les discussions avec notre expert du domaine appartenant à l'équipe AMARE.

Le système dispose de tous les paramètres d'initialisation portant sur le capteur, l'animal et l'expérimentation en cours. Cette approche permet l'identification facile du triplet (capteur, animal, expérimentation). On distingue les données suivantes :

- données sur le capteur (*Tag info*) : on distingue deux types de données. Des données statiques, comme la référence du capteur, le numéro de téléphone du système embarqué, etc. Des données dynamiques comme le temps dans le système, la profondeur maximale enregistrée à ce jour, etc. ;
- déploiement (*Deployment*) : contient des informations sur l'animal suivi (âge, sexe, poids, vitesses moyennes, etc.) et les informations concernant l'utilisation du capteur sur l'animal comme les dates de début et de fin de l'expérimentation.

Le système embarqué dans la balise surveille continuellement les données capturées et maintient un modèle à trois états. Il prend en compte la position du phoque à la surface de la mer, la profondeur du capteur et les temps d'interactions. Pour représenter cette dynamique, on utilise un diagramme machines à états sur la classe phoque (figure 3.2), avec :

- en surface (*Haulout*) : le début de cette étape est détecté lorsque le capteur est continuellement sec durant un intervalle de temps, généralement 10 min. Elle se termine dès que le capteur redevient humide durant un intervalle de temps, généralement 40 sec. Le système fournit les dates et les positions spatiales de début et de fin de chaque *Haulout* ;

- en croisière (**Cruise**) : cette étape doit durer au minimum 9 min. Durant cette période l’animal reste dans l’eau et se trouve au dessus du seuil de début d’une plongée (paramètre fonction de l’animal et du milieu marin). Le système fournit les dates et les positions spatiales comme dans le cas d’un **Haulout** ;
- en plongée (**Dive**) : lorsque le début d’une plongée est détecté, toutes les données sont enregistrées jusqu’à ce que la fin soit détectée et ainsi on construit une plongée. Le système procède au calcul des points spatiaux intermédiaires (5 en règle générale) qui décrivent le mieux possible la plongée puisque la capture satellitaire du système ne peut pas fonctionner sous l’eau. Le système transmet les groupes consécutifs de plongées avec la date de début et de fin. La durée de chaque plongée est aussi transmise pour calculer la date de début de chacune dans le groupe. Le système transmet aussi des données intrinsèques sur l’animal comme sa vitesse, sa température et la profondeur des plongées.

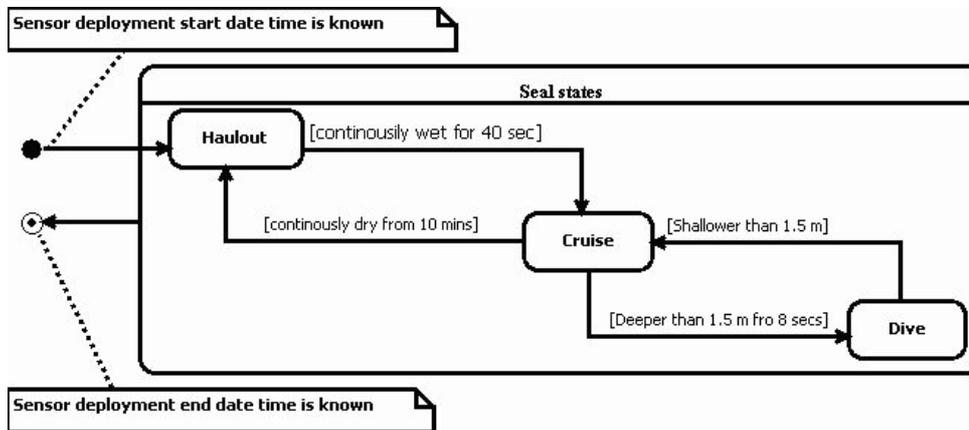


FIGURE 3.2 – États fondamentaux du phoque durant son déplacement

En plus des données localisées dans l’espace et le temps en relation directe avec la trajectoire de l’animal, le système transmet des données calculées et d’autres sur le fond marin visité par l’animal :

- résumé (**Summary**) : contient des données dérivées et envoyées par le système sur une période de temps (4 ou 6 heures). Il s’agit essentiellement des statistiques concernant les trois états précédents ;
- données sur le milieu marin (**CTD Conductivity - Temperature - Depth**) : le système peut capturer des paramètres sur le milieu marin comme la température, la pression, la salinité et la conductivité de l’eau dans le cas de certaines plongées. Ces données sont caractérisées temporellement et spatialement.

3.1.3 Modèle UML de la trajectoire

La synthèse de la phase d’analyse nous conduit au modèle de la trajectoire du phoque donné dans les deux figures 3.3 et 3.4. Bien évidemment, ce modèle se base sur le modèle modulaire et générique 1.9. Il ajoute le domaine d’application qui comporte les concepts et les notions liés aux phoques, nommé le domaine de phoque **Seal Domain**. Il s’agit d’une

spécification issue du domaine de l'objet mobile *Mobile Object Domain*. Pour cela, nous utilisons la relation d'héritage entre la classe *Seal* et la classe *Mobile Object*.

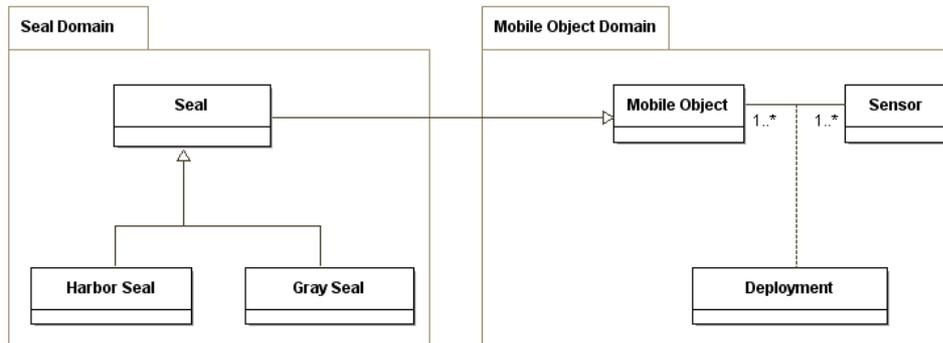


FIGURE 3.3 – Domaine de l'objet mobile Phoque

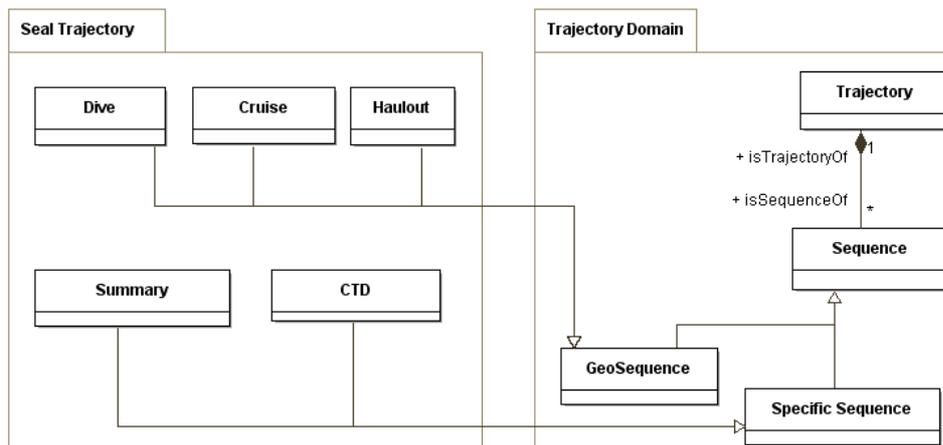


FIGURE 3.4 – Domaine de trajectoire du Phoque

Le modèle 3.3 montre les deux types de phoque : Veau marin *Harbor Seal* et Phoque gris *Gray Seal*, tels que :

- le phoque commun ou veau marin (*Phoca vitulina*) est un mammifère carnivore, de la famille des phocidés (figure 3.5). Sa durée de vie peut atteindre 25 ans pour le mâle et 35 ans pour la femelle, avec des variations importantes selon les zones de vie et les sous-populations ou sous-espèces. Le terme de veau marin n'est réservé qu'à cette espèce de phoque, et ne doit pas être confondu avec le terme veau de mer qui désigne aussi le requin taupe ;
- le phoque gris (*Halichoerus grypus*) est un mammifère carnivore, de la famille des phocidés (figure 3.6). Il est la seule espèce du genre *Halichoerus*. Le phoque gris réside dans l'Atlantique Nord, dans les eaux subarctiques et tempérées froides. On le retrouve en Amérique du Nord, en Europe du Nord et en mer Baltique. En Europe, les grandes colonies sont situées en Angleterre, en Irlande et en Écosse. Les phoques gris fréquentent les zones côtières et viennent sur les côtes sableuses et rocheuses. En

France, il existe des petites colonies sur les côtes bretonnes et plus particulièrement sur l'archipel de Molène et l'archipel des Sept-Îles.

Le phoque commun est, avec le phoque gris, la seule espèce de phocidé qu'il est possible d'observer régulièrement sur certaines plages du nord et du nord-ouest de la France, dont la baie de Somme constitue sa principale zone de reproduction.



FIGURE 3.5 – Phoque commun ou veau marin (*Phoca vitulina*)



FIGURE 3.6 – Phoque gris (*Halichoerus grypus*)

Par ailleurs, le modèle 3.4 spécifie le domaine de trajectoire du phoque *Seal Trajectory* qui comporte les classes et la terminologie métier spécifique aux déplacements des phoques. Il comporte les différentes classes *Dive*, *Haulout* et *Cruise* qui représentent les types des séquences des déplacements des phoques. Elles sont caractérisées spatialement et temporellement. Pour cela, nous les avons défini comme des sous-classes de la classe *GeoSequence*. Nous avons défini les deux classes *Summary* et *CTD* comme sous-classes de la classe *Specific Sequence* qui représente les séquences spécifiques constituées essentiellement par des méta-données.

3.1.4 Modèle de la trajectoire sémantique

Vu la masse importante des données transmises par le capteur, il est impensable de recourir à une annotation sémantique manuelle basée sur la visualisation. Nous adoptons donc une approche d'annotation automatique de toutes les séquences. Dans ce travail, nous nous intéressons aux activités du phoque effectuées pendant ses plongées (les instances

de la classe `Dive`). Les instances des classes `Haulout` et `Cruise` sont en général liées à la notion de voyage.

Selon l'expert du domaine, les instances de la classe `Dive` sont associées à trois activités principales : repos (A_1), déplacement (A_2), chasse (A_3), sommeil (A_4) et déplacement + chasse (A_5). Il existe une corrélation entre la forme géométrique de la plongée et l'activité du phoque. Pour classifier les formes géométriques des plongées, un coefficient appelé TAD ²⁶ est calculé sur un ensemble de données. De cette classification, on distingue trois schémas (figure 3.7) :

- plongée en forme de V : si $0.4 \leq TAD < 0.6$;
- plongée en forme de U+V : si $0.6 \leq TAD < 0.8$;
- plongée en forme de U : si $0.8 \leq TAD < 1$.

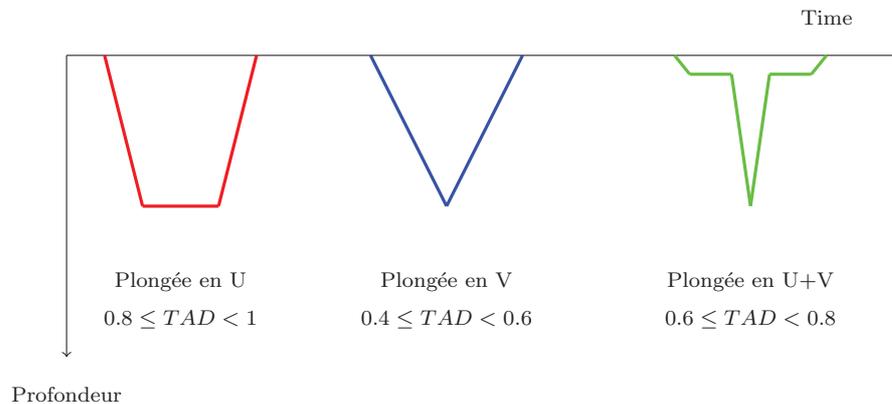


FIGURE 3.7 – Trois formes principales d'une plongée et le TAD correspondant

En plus de la forme géométrique de chaque séquence, il y a deux autres paramètres à prendre en compte pour l'annotation sémantique :

- la vitesse moyenne de plongée (VMP) : les valeurs discrètes prise en compte sont : faibles, moyennes et élevées ;
- la profondeur de plongée par rapport au fond (PP%F) : les valeurs discrètes prises en compte sont : en surface et entre deux eaux PP%F₁, sur le fond PP%F₂, en surface PP%F₃, entre deux eaux et sur le fond PP%F₄.

Par conséquent, ce système contient 36 ($3 \times 3 \times 4$) règles possibles, mais l'expert du domaine ne peut identifier que les 5 actions qui correspondent aux activités du phoque précédemment citées (A_1 à A_5). La table de décision (tableau 3.1) explicite ces actions. Elle sert comme pivot pour lier les instances des classes du modèle de la trajectoire enrichie et les données capturées (figure 3.8).

26. *Time Allocation at Depth*

	VMP			Forme d'une plongée			PP%F			
	faible	moyen	élevé	U	V	U+V	PP%F ₁	PP%F ₂	PP%F ₃	PP%F ₄
A ₁		X		X			X	X	X	X
A ₂			X		X		X			
A ₃			X	X						X
A ₄	X			X				X		
A ₅			X			X				X

TABLE 3.1 – Table de décision associée aux activités de plongées

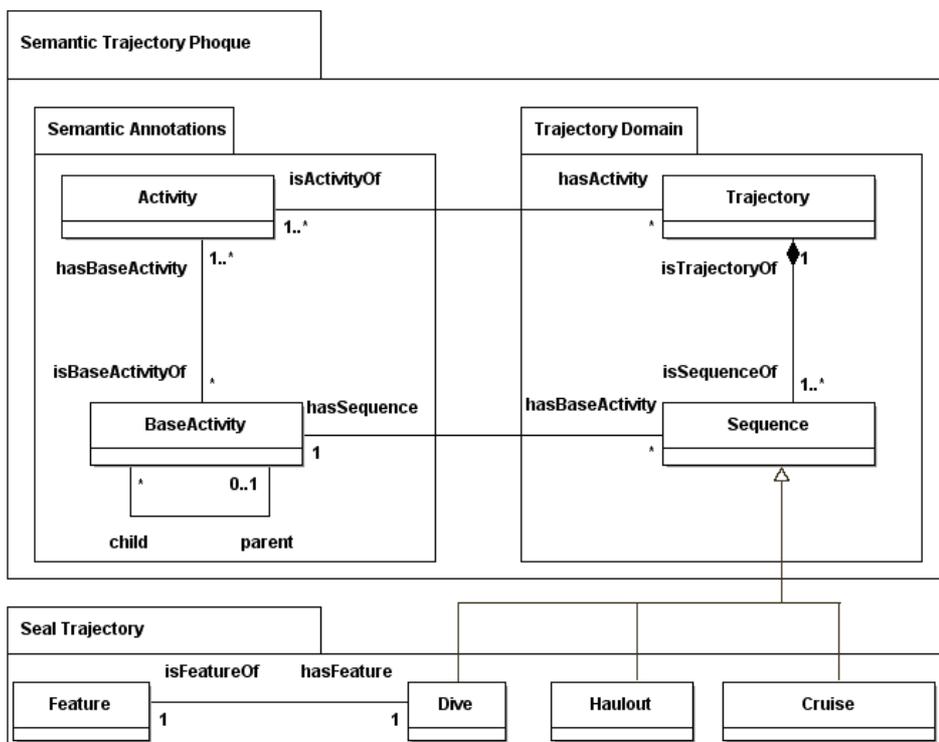


FIGURE 3.8 – Extrait du diagramme de classes de la trajectoire sémantique de phoque

Dans cette partie, nous avons analysé les données capturées pour définir un modèle de la trajectoire du phoque. Ce modèle est ensuite enrichi pour prendre en compte la sémantique du domaine et définir la trajectoire sémantique (figure 3.8). Ce dernier modèle est formé par trois principales composantes :

1. thématique : ce sont les faits réels de la trajectoire (ex. les activités). Dans ce travail, nous nous sommes intéressées aux activités liées aux plongées par le biais de la classe **Feature** qui représente les caractéristiques d'une plongée dont la forme ou le coefficient TAD ;
2. spatiale : la localisation des faits dans l'espace ;
3. temporelle : la localisation des faits dans le temps.

3.2 Intégration du domaine dans l'ontologie de la trajectoire

Dans cette partie nous présentons l'ontologie de la trajectoire du phoque `owlSealTrajectory`. Cette ontologie rassemble tous les concepts qui jouent un rôle important pour définir la notion de la trajectoire du phoque. L'ontologie `owlSealTrajectory` se base sur l'ontologie de haut niveau de la trajectoire définissant une structuration générale des concepts thématiques, spatiaux et temporels (figure 3.9).

La partie inférieur du modèle (figure 3.9) représente l'ontologie du domaine d'application. Elle représente une spécialisation de l'ontologie de haut niveau, à partir de laquelle elle hérite toutes les propriétés, en ajoutant celles liées à son domaine. Parmi les concepts principaux de l'ontologie de haut niveau, on trouve `Mobile Object` qui joue un rôle principal dans la traçabilité de la trajectoire. Dans le modèle 3.9, la trajectoire du phoque est composée par des séquences et plus précisément des géo-séquences comme la plongée `Dive`. Ce dernier concept est associé à des activités comme la chasse `Hunting` ou le déplacement `Displacement`. La formalisation de ces activités est basée sur des restrictions pour ne prendre en compte que des instances du concept `Dive`. Pour l'instant, on ne fait pas encore de distinction entre une activité de base et une activité générale faute de données précises.

La partie impérative de l'ontologie `owlSealTrajectory` contient, entre autres, les actions de la table de décision des activités du phoque (tableau 3.1). Ces actions forment les règles du domaine. La mise au point de ces règles nécessite un langage impératif dépendant du raisonneur utilisé que nous détaillons dans la partie mise en œuvre.

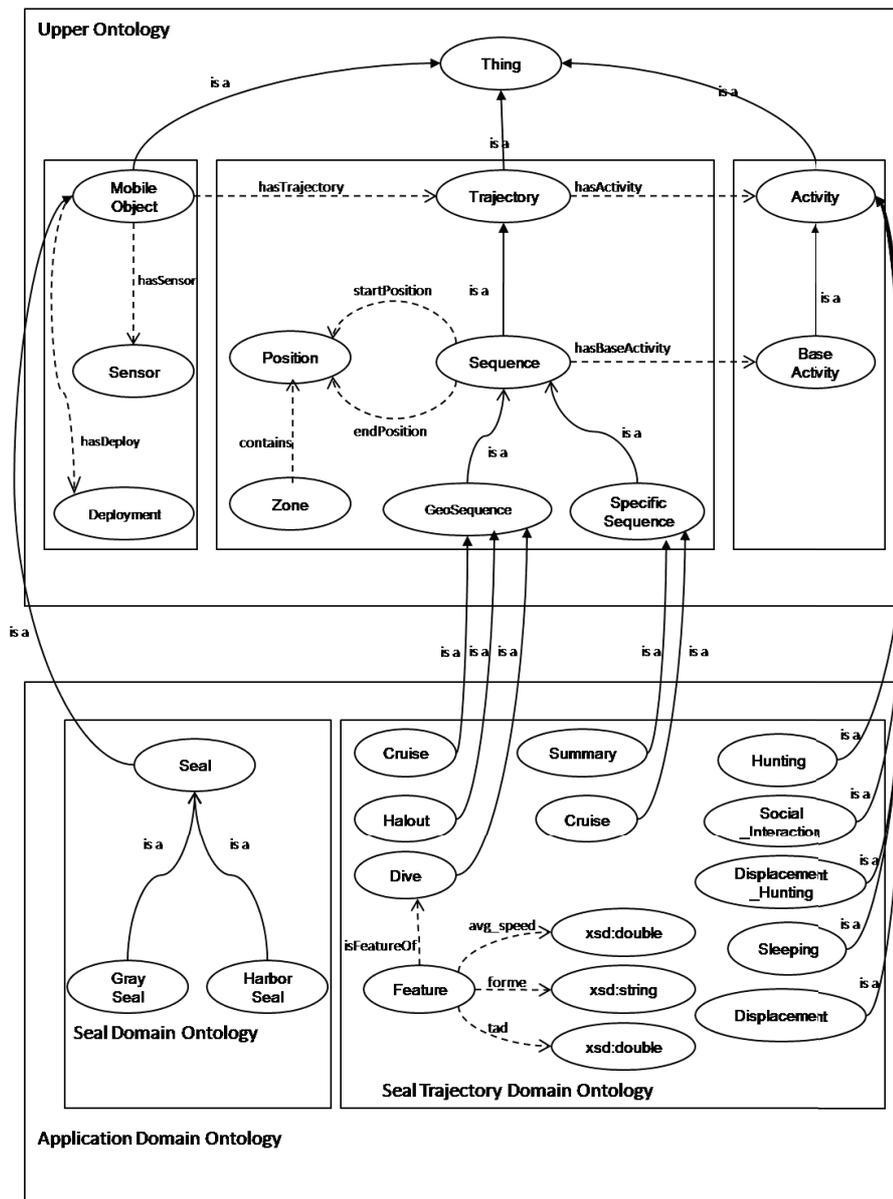


FIGURE 3.9 – Extrait de l'ontologie owlSealTrajectory

Chapitre 4

Approche bases de données pour la gestion des données ontologiques RDF stores

Sommaire

4.1	Introduction	125
4.2	Stockage des données ontologiques	126
4.3	Chargement des données ontologique	129
4.4	Inférence	131
4.5	Interrogation RDF et les données relationnelles	132
4.6	Étude comparative de quelques BDBO	133

4.1 Introduction

Dans ce chapitre, nous discutons la gestion des instances d'une ontologie par les systèmes de gestion de bases de données (SGBD). Des solutions de persistance, d'interrogation et de connexion de ces instances ont été proposées pour stocker ces instances via des SGBD, ce qui fait naître la notion de bases de données à base ontologique (BDBO) [Mbaiossoum *et al.*, 2012], appelée aussi *RDF stores*. Ces solutions ont bénéficié de l'évolution des différentes recommandations et des travaux de recherche dans le domaine de l'ingénierie des données. Leur capacité à accueillir un volume de données toujours croissant s'est amélioré tout comme leur aptitude à interroger plus rapidement les données hébergées ou à s'interfacer avec des données externes. Aujourd'hui, accessibles et matures, ces moteurs de bases de données ontologiques sont des supports viables pour la création d'applications sémantiques pour les environnements d'entreprises, en complément ou en remplacement des supports relationnels classiques.

Comparé au développement des bases de données traditionnelles, ces systèmes ont été développés et proposés dans un espace de temps très court. En 2013, nous avons mené

une étude comparative de quelques solutions BDBO dans le cadre d'un travail de mémoire CNAM de Geoffroy VIBRAC. Comme le souligne les auteurs dans [Mbaiossoum *et al.*, 2012], le développement de nombreuses BDBO résulte principalement de :

1. la diversité des formalismes : chaque BDBO utilise un formalisme particulier pour définir ses ontologies (OWL, PLIB ou FLIGHT) ;
2. la diversité des modèles de stockage : contrairement aux bases de données traditionnelles, où le modèle logique est stocké selon une approche relationnelle, dans une BDBO, une variété de modèles ; stockage (représentation horizontale, spécifique, etc.) sont utilisés pour stocker deux niveaux de modélisation : le niveau ontologie et le niveau des instances ontologiques ;
3. la diversité des architectures cibles utilisées par le système de gestion de bases de données : une BDBO peut utiliser un seul ou plusieurs schémas de base de données pour stocker l'ensemble des données.

Ce chapitre expose les différentes technologies utilisées dans une base de données ontologique pour répondre à ces enjeux. La première partie est consacrée à l'étude des différentes architectures mise en œuvre dans un système de gestion de base de données pour le stockage des données ontologiques ; le chargement des données ; le calcul des inférences ; la recherche des données. Dans un second temps, on présente l'implémentation de ces techniques dans les systèmes de bases de données sémantiques et les illustre à travers trois études de cas : AllegroGraph (Franz Inc.), Virtuoso (openlink Software), Semantic Platform (Intellidimension) et Oracle Semantic Data Store (Oracle).

4.2 Stockage des données ontologiques

Dans la spécification du *Resource Description Framework (RDF)* [Lassila and Swick, 1999], on définit la structure élémentaire d'une donnée ontologique : le triplet. Ce triplet RDF est composé d'un sujet qui représente la ressource à décrire, d'un prédicat qui est le type de propriété applicable à cette ressource et d'un objet qui représente la valeur de la propriété. Ces trois valeurs vont devoir être stockées dans les bases de données et les architectures et les schémas adoptés pour ce stockage auront un rôle déterminant dans la qualité de la plateforme ontologique.

4.2.1 Schéma de stockage

Le stockage des triplets RDF peut être réalisé selon trois organisations différentes [Das and Srinivasan, 2009; Mbaiossoum *et al.*, 2012] : La première approche est le schéma inconscient (schema-oblivious), dite aussi approche verticale. Ce schéma consiste en une seule table constituée de trois colonnes : une pour le sujet, la seconde pour le prédicat et la dernière pour l'objet. Ce format ne change jamais, même si le schéma des données RDF qui doivent être stockées évolue.

Dans la seconde approche, le schéma conscient (schema-aware), dite aussi approche binaire, le schéma de la base de données dépend du schéma des données RDF qui y sont

Subject Resrouce URI	Predicate Property Name	Object Property Value

FIGURE 4.1 – Schéma inconscient d'un triple RDF

stockées. Chaque prédicat est représenté par une table « Property » de deux colonnes représentant le sujet et l'objet du prédicat. Une autre table « Class » stocke chaque classe du schéma RDF. Ce schéma a l'avantage d'être plus compact que le schéma précédent. Les recherches sont ainsi plus rapides car réalisées sur des multiples petites tables plutôt que sur une seule grande table. Par contre, le format des tables peut être amené à changer avec l'ajout de triplets contenant des nouveaux prédicats ou la suppression de triplets contenant un prédicat particulier. De plus, lors de l'exécution de requêtes SPARQL sur tous les prédicats, il faut interroger toutes les tables « Property » ce qui est moins performant.

Property_1	
Subject Resrouce URI	Object Property Value

⋮

Property_n	
Subject Resrouce URI	Object Property Value

Class_1
Subject Resrouce URI

⋮

Class_m
Subject Resrouce URI

FIGURE 4.2 – Schéma conscient d'un triple RDF

La dernière approche, dite hybride ou horizontale, est une variante de la première mais au lieu d'une seule table de trois colonnes, c'est une seule table avec un nombre prédéterminé de colonnes (ou un nombre prédéterminé de tables) basées sur les types de données possible des objets des triplets. La valeur de l'objet est stockée dans la colonne (ou dans la table) appropriée à son type.

Subject Resrouce URI	Predicate Property Value	String_Object Property Value	Decimal_Object Property Value	DateTime_Object Property Value	URI_Object Property Value	Other_Object Property Value

FIGURE 4.3 – Schéma hybride d'un triple RDF

4.2.2 Stockage des IRI

Indépendamment du schéma adopté pour la persistance des données ontologiques, la façon dont sont stockées les IRI (identificateur de ressource internationalisé) [Duerst and Suignard, 2005] est déterminante pour les performances de la solution.

Les données RDF contiennent des occurrences d'IRI long et souvent répétées. Ces IRI peuvent être stockés tels quels. On peut dire que c'est une approche « basée sur la valeur », en effet, on stocke la valeur lexicale directement dans les tables. L'autre solution consiste à établir une correspondance entre ces IRI et un identifiant unique. C'est alors cet identifiant qui est stocké dans les tables contenant les triplets. Cette approche « basée sur un identifiant » requiert une table additionnelle de correspondance entre les valeurs des IRI et les identifiants. Cette solution offre plusieurs avantages : gain de place, index sur les identifiants plus compacts et tests d'égalité plus rapides. Pour ces raisons, cette approche a des résultats meilleurs sur les requêtes et l'inférence. La génération des identifiants peut se faire par l'utilisation de séquences ou par des fonctions de hachage. Cette dernière solution est plus avantageuse car d'une part, la génération d'identifiant est plus simple et d'autre part, il n'y a pas de dépendance sur l'historique des insertions et suppressions des triples.

4.2.3 Compression des préfixes

La plupart des IRI des triplets RDF ont le même préfixe. Il est ainsi possible de factoriser tous ces préfixes communs afin de réduire l'espace de stockage des tables et des index afin d'améliorer la performance des requêtes. Il y a deux approches pour compresser les préfixes.

La première approche consiste en l'association d'un identifiant unique avec chaque préfixe distinct et de stocker cette correspondance dans une table séparée. Pour représenter la valeur lexicale, il suffit d'utiliser la combinaison de l'identifiant et du suffixe lexical. La seconde approche revient à stocker les portions préfixes et suffixes dans deux colonnes différentes d'une table et d'utiliser la compression de table. Cela permet d'éviter la création d'une table séparée et donc de devoir faire des jointures lors des requêtes. Par contre, la compression est limitée dans la mesure où elle ne s'applique que pour les préfixes communs qui sont localisés sur le même bloc des données. Il est d'ailleurs possible de combiner cette technique de compression avec l'approche de stockage des IRI basée sur un identifiant. Préfixes et suffixes sont alors séparés dans deux colonnes de la table de correspondance entre valeur lexicale et identifiant.

4.2.4 Unité de stockage, propriété et contrôle d'accès

Une base de données peut contenir plusieurs modèles de données ontologiques, chacun d'eux peut être considéré comme une unité de stockage, avec un propriétaire et intégrer des contrôles d'accès. Ainsi, quand un utilisateur crée un modèle, le créateur est considéré comme le propriétaire. Le propriétaire peut ensuite allouer des privilèges d'accès à d'autres utilisateurs.

4.2.5 Index de données ontologiques

La duplication de triplets RDF n'est pas autorisée dans le même modèle de données ontologiques. Cette contrainte est maintenue par une clé d'index unique. Cet index est également utilisé pour des raisons de performance.

Une approche du stockage du modèle ontologique consiste à la création d'index prédéfinis et de donner la possibilité d'ajouter des index manuellement voir même de supprimer certains index (à condition de laisser la contrainte d'unicité des triplets). Dans [Harth and Decker, 2005], les auteurs montrent qu'il est possible d'indexer les données RDF de six manières possibles : une pour chaque ordre possible des trois éléments d'une donnée RDF (sujet, prédicat, objet). Les auteurs associent à cette donnée une quatrième composante qui fournit le contexte du triplet. Dans [Wood *et al.*, 2005], on propose une technique similaire d'indexation des données ontologiques où le contexte d'un triplet RDF est remplacé par un méta-item donnant le modèle de provenance de ce triplet. Enfin, dans [Weiss *et al.*, 2008], on trouve une étude analytique concernant les performances de différentes techniques d'indexation.

Il est possible d'économiser l'espace de stockage de ces index avec l'utilisation de la compression des identifiants et la compression de préfixes. Au bénéfice de performance, gagné par l'utilisation des index, il convient de mettre en perspective la perte de performance dû au recalcul des index pendant les opérations de chargement ou de modification de données.

4.2.6 Stockage RDF distribué et/ou fédéré

Gérer des données ontologiques, implique d'être capable de considérer des grandes masses de données. De plus, ces données peuvent être potentiellement réparties, distribuées ou même répliquées. Dans ce domaine, plusieurs travaux se basent sur les techniques traditionnelles de distribution et de réplcation de base de données pour gérer des entrepôts de données RDF distribués ou fédérés. En fait, le modèle des triplets RDF permet de faire des requêtes sur des entrepôts RDF multiples. Cependant, la transformation interne dans le cadre du stockage des IRI basés sur un identifiant complexifie l'exploitation de ce genre de solution car la même valeur lexicale peut se voir attribuer différents identifiants dans deux entrepôts différents. Pour cette raison, les recherches dans ce domaine ont pris d'autres directions tout en se concentrant sur deux points d'entrées fondamentales : les modèles de données distribués et les requêtes distribuées.

4.3 Chargement des données ontologique

4.3.1 Chargement incrémental et chargement en masse

Habituellement, au moins deux types de chargement sont supportés :

- chargement par instruction INSERT, pour une petite quantité de données ;

- chargement en masse par des API hautement optimisées pour assurer un chargement optimal d'une grande quantité de données.

Le format d'entrée pour le chargement en masse est habituellement basé sur un fichier formaté selon l'un des standards du RDF : N-Triple, RDF/XML, N3 ou Turtle. Parfois, les API de chargement s'appuient également sur une table intermédiaire à trois colonnes pour stocker les valeurs lexicales du sujet, du prédicat et de l'objet des triplets.

Le chargement des données inclut également l'analyse des données afin de s'assurer que les valeurs des éléments du triplet sont valides selon les préconisations liées aux données ontologiques et appropriées au contexte. Cette tâche élimine également les doublons pour maintenir l'aspect sémantique du modèle de données. Parfois, le chargement inclut également les calculs d'inférence.

4.3.2 Effet sur l'architecture de stockage

Le système de chargement utilisé pour charger en masse les données s'avère généralement étroitement lié à l'architecture de stockage de la plateforme ontologique. Par exemple, le chargement est simple si on utilise un stockage basé sur la valeur lexicale mais les performances sont amoindries par la nécessité d'écrire un grand nombre de long champ lexical et de construire les index volumineux correspondants. Le chargement des données dans le cadre d'un stockage basé sur des identifiants est plus compliqué car il faut charger une table de correspondance entre valeur et identifiants en même temps que la table des triples. L'utilisation d'identifiants basés sur un hachage des valeurs lexicales permet d'améliorer les performances du chargement car la génération des identifiants est plus simple.

4.3.3 Ajout de données en masse

L'ajout de données, c'est-à-dire le chargement dans un modèle ontologique non vide, requiert la vérification du nouveau lot de triplets afin qu'il n'y ait pas de doublon avec les données déjà présentes. Cela peut aussi supposer une maintenance incrémentale des index. De ce fait, l'ajout de données en masse dans une base qui contient déjà des triplets est moins performant que le chargement en masse dans une base vide.

4.3.4 Réutilisation des nœuds anonymes

L'ajout de nouveaux triplets pose le problème de la réutilisation des noms de nœuds anonymes dans les lots de triplet à charger. En effet, comment être certain qu'on se réfère à la même ressource, si un nœuds anonyme avec son étiquette est déjà présent dans le graphe RDF et que cette même étiquette est utilisée dans le nouveau jeu de données RDF à ajouter au graphe. La réutilisation des étiquettes de nœuds anonymes a l'avantage de ne pas faire de différence, que les données soient chargées dans un seul ou dans plusieurs lots.

4.4 Inférence

L'inférence est un processus de raisonnement qui, s'appuyant sur les connaissances acquises, permet grâce à des règles, d'obtenir de nouvelles données. Propriété majeure des BDBO, c'est à travers la richesse, la performance et la scalabilité de l'inférence, qu'on peut évaluer la qualité d'une plateforme de gestion des données ontologiques.

4.4.1 Systèmes d'implications standards

Il existe plusieurs systèmes d'implications standards : la sémantique RDF, RDFS (RDF Schéma) et OWL (Web Ontology Language). OWL définit trois sous langages : OWL-Lite, OWL-DL (OWL Description Logics) et OWL-Full. La prise en charge de RDF et RDFS dans les BDBO est simplifiée par la disponibilité d'axiomes et de règles, qui font défaut à OWL pour être facilement supporté. L'implication avec le langage OWL-Full nécessite des calculs très complexes ce qui décourage les éditeurs à l'implémenter. Par contre, les BDBO actuelles prennent généralement en charge des sous-ensembles de OWL-Lite et de OWL-DL.

4.4.2 Interfaçage avec des moteurs de raisonnement tiers

La possibilité de s'interfacer avec des moteurs de raisonnement externes permet d'avoir un support plus complet de l'inférence pour un sous-ensemble de données. Par exemple, un support complet d'OWL-DL peut être offert par une plateforme si elle sait s'interfacer avec un moteur d'inférence comme Pellet de Clark & Parsia qui offre un support complet de ce sous langage.

4.4.3 Règles définies par l'utilisateur

Lorsqu'une application a besoin de système d'implication que les langages de règles ne peuvent pas gérer, il convient de donner la possibilité aux utilisateurs de définir leurs propres règles. On parle alors de règles métiers liées au domaine d'application. Il peut aussi s'agir de règles universelles liées à la gestion des données temporelles ou spatiales.

4.4.4 Chaînage avant et chaînage arrière

Le chaînage avant pré-calculé et stocke les triplets inférés tandis que le chaînage arrière détermine les triplets inférés pendant l'interrogation et ne les stocke pas pour un usage ultérieur. Le pré-calcul et le stockage des triplets inférés utilisés dans le chaînage avant utilisent plus d'espace, mais rendent les requêtes plus rapides en évitant de traiter les règles d'inférences pendant la requête. Cependant, dans certaines circonstances l'espace occupé peut être très volumineux. De plus, le maintien des triplets pré-calculés et stockés avec l'utilisation du chaînage avant pose un problème important : tout changement dans

les triplets, dans les axiomes ou dans les règles peuvent impliquer la modification des triplets induits. La qualité d'une BDBO, du point de vue d'un utilisateur, dépend de la rapidité et de l'efficacité avec laquelle l'ensemble des triplets inférés sont vérifiés et mis à jour si nécessaire. La majorité des plateformes actuelles intègrent le chaînage avant.

4.5 Interrogation RDF et les données relationnelles

La capacité d'interroger rapidement et facilement les graphes RDF dans une plateforme BDBO s'avère être l'aspect le plus important du point de vue de l'utilisateur.

4.5.1 Langage d'interrogation

SQL est un puissant langage d'interrogation de base de données éprouvé et optimisé depuis des années. On pourrait ainsi imaginer de l'utiliser pour interroger les données RDF. Cependant, SQL n'est pas adapté pour réaliser des interrogations de graphes.

Ainsi, le W3C recommande, pour l'interrogation des données RDF, l'utilisation du langage SPARQL qui a été conçu pour cette tâche. Il est ainsi plus simple de spécifier une requête en SPARQL qu'en SQL. Cependant, SPARQL souffre d'être un langage autonome qui manque de fonctionnalités essentielles pour le traitement des résultats RDF. Par exemple, dans sa version 1.0, il n'intègre ni fonction d'agrégation, ni sous-requête. Ces fonctions vont apparaître dans la version 1.1.

Avant la parution de la version 1.1, une solution hybride a été d'encapsuler des requêtes SPARQL dans du SQL. Les résultats de SPARQL sont ainsi traités comme une table qui peut ensuite être interrogée en SQL. Cela facilite l'interrogation des données en graphe du fait de l'utilisation de SPARQL et en même temps, les résultats peuvent être traités, combinés, agrégés par les constructions riches de SQL. Une autre approche a été d'étendre le langage SPARQL 1.0 avec les concepts de SQL qui lui font défaut en intégrant, par exemple, des fonctions d'agrégation ou les sous requêtes.

4.5.2 Performance des requêtes

La solution hybride exposée ci-dessus peut être optimisée si la partie de la requête écrite en langage SPARQL peut être réécrite comme une sous requête SQL et ainsi être intégrée au reste de la requête SQL. On peut également améliorer la performance de l'interrogation en incluant des extensions au langage SPARQL comme il existe des extensions dans les implémentations de SQL. Appliquées aux sous-requêtes SPARQL, elles permettraient d'améliorer les plans d'exécution des requêtes. De même dans une architecture basée sur des identifiants, des extensions SPARQL permettent de récupérer uniquement les identifiants à la place des valeurs lexicales évitant ainsi l'utilisation coûteuse des jointures supplémentaires.

4.5.3 Optimisation des requêtes

L'exécution des requêtes peut être optimisée avec des outils tels que les optimiseurs « basés sur les coûts » (CBO) généralement disponibles dans un système de gestion de base de données traditionnel.

4.5.4 Interrogation des données relationnelles dans un contexte RDF

Il y a deux façons pour interroger les données relationnelles et RDF conjointement. La méthode la plus simple consiste à utiliser, dans la requête hybride, une ou plusieurs tables relationnelles et de les joindre dans des sous-requêtes SPARQL. Une autre façon est de traiter les valeurs utilisées dans une colonne de table relationnelle, éventuellement convertie sous forme d'IRI, comme des valeurs d'un graphe RDF. Un nouvel opérateur est défini pour être utilisé dans la clause WHERE de la requête SQL afin de vérifier si la valeur de cette colonne dans les lignes de résultats satisfait une requête SPARQL spécifiée en tant qu'argument de l'opérateur.

Certaines BDBO proposent de visualiser les données relationnelles comme des modèles RDF virtuels afin de permettre l'accès aux informations stockées dans ces bases. Cela permet ainsi de les interroger à l'aide du langage de requête SPARQL et de les croiser avec les données RDF natives. Pour cela, ces plateformes intègrent un langage de correspondance ou mapping permettant de spécifier les liens entre les tables et les classes ainsi qu'entre les colonnes et les propriétés. Elles doivent également intégrer un composant chargé de traduire les requêtes SPARQL sur le modèle RDF en requêtes SQL sur les tables SQL sous-jacentes.

4.6 Étude comparative de quelques BDBO

4.6.1 Oracle Semantic Data Store

Oracle a incorporé des supports pour les langages RDF et OWL dans son système pour la gestion des données sémantique. Ce support a été implanté pour la première fois dans le SGBD Spatial Oracle 10g et est désormais en option dans les bases de données Oracle 11g et 12c. Oracle a défini deux sous-classes d'OWL DL qu'il exploite dans les processus d'inférence : OWLPrime et OWLSIF (figure 4.4).

Oracle utilise une architecture basée sur deux parties : une partie pour la méta-base et une autre pour les données. Dans la méta-base, l'éditeur définit un modèle de stockage des triplets RDF. La table de triplet utilisée a également été améliorée par une décomposition (normalisation). Ainsi, pour éviter de manipuler les longues chaînes de caractères (les URIs), les identificateurs sont générés pour chacune d'elles. Les valeurs lexicales des sujets, prédicats et objets sont mappées en identificateur (ID) entiers générés par le système.

Oracle offre également la possibilité de faire des déductions de connaissances en se basant sur les règles d'inférence définies pour le modèle OWL.

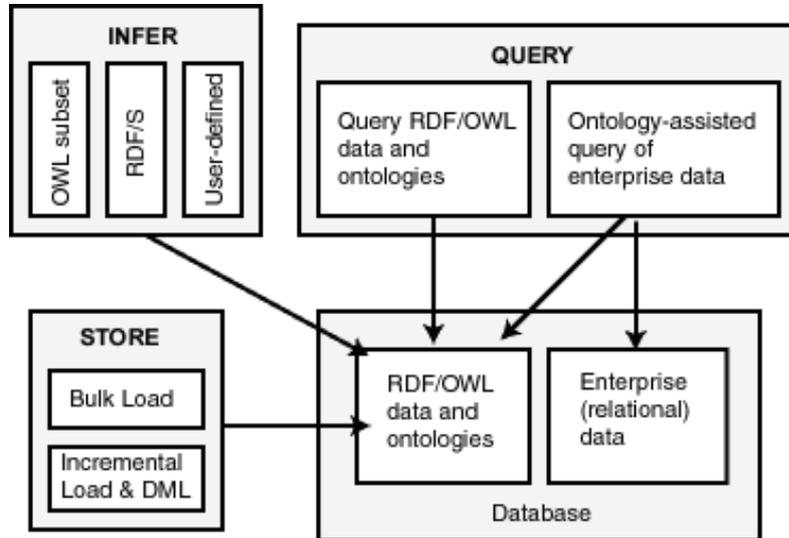


FIGURE 4.4 – Architecture de Oracle Semantic Data Store

4.6.2 Virtuoso Universal Server

Virtuoso Universal Server (appelé par la suite Virtuoso)²⁷, édité par OpenLink Software, n'est pas spécifiquement orienté vers le stockage des triplets RDF. Virtuoso est à la fois un middleware et un moteur de base de données qui regroupe des fonctionnalités telles que : système de gestion de base de données relationnelles ou relationnelles-objets ; gestion de données RDF, XML, texte libre ; serveur d'applications web ; serveur de fichier et serveur de web services (figure 4.5).

Le stockage des données sémantiques dans Virtuoso repose sur un modèle relationnel. Les triplets RDF sont en réalité des quadruplets (mais toujours appelés triplets) avec les quatre attributs : graphe G, sujet S, prédicat P, objet O. Chaque triplet est représenté sous forme d'une ligne de table RDF_QUAD qui contient quatre colonnes (G, S, P, O) pour les quatre attributs. L'unicité des triplets est garantie par une clé primaire sur l'ensemble des attributs.

4.6.3 AllegroGraph

AllegroGraph²⁸ est une solution propriétaire développée par Franz Inc. C'est une plateforme sémantique destinée à la construction d'applications web sémantiques. AllegroGraph utilise pour cela une base de données pour stocker les triplets RDF et propose un

27. <http://docs.openlinksw.com/virtuoso/>

28. <http://www.franz.com/agraph>

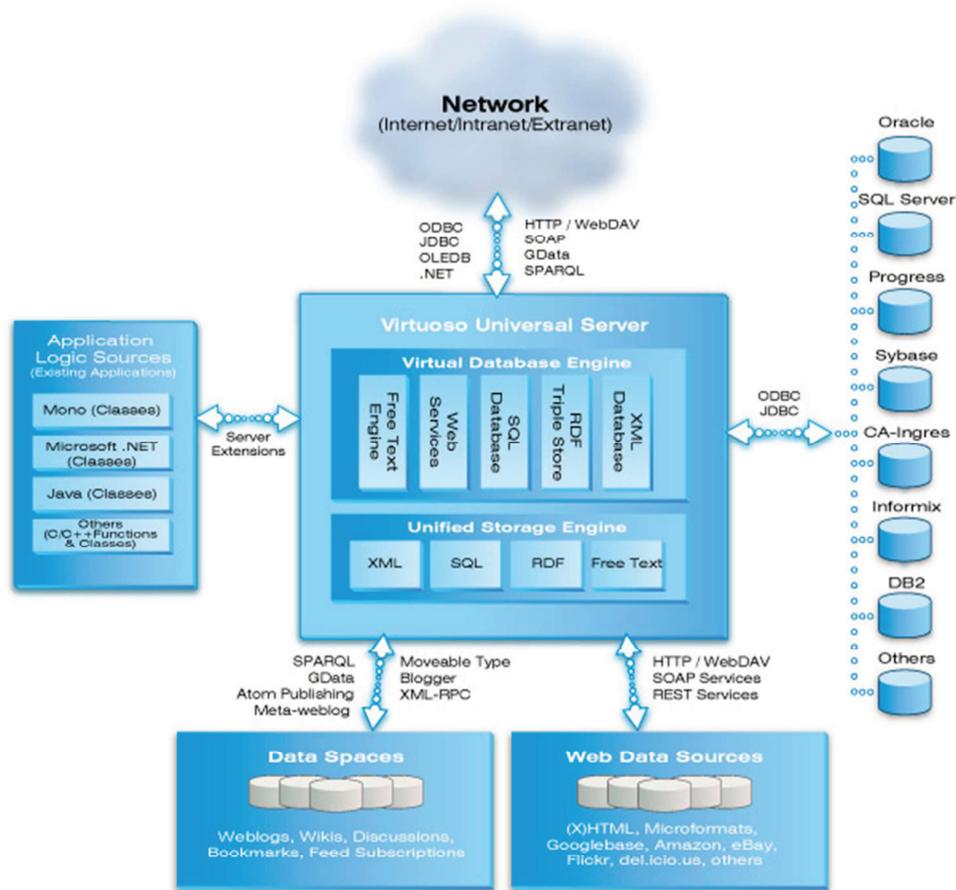


FIGURE 4.5 – Architecture de Virtuoso Universal Server

ensemble d'API dédiées aux requêtes comme SPARQL ou Prolog, et intègre un moteur de raisonnement (figure 4.6). La base de données d'AllegroGraph n'est pas relationnelle mais orientée graphe. En cela, son étude est intéressante car elle apporte une autre vision du stockage des données que celle des autres solutions étudiées conçues autour de bases relationnelles.

4.6.4 Semantics Platform

Semantics Platform²⁹ est un logiciel propriétaire édité par la société Intellidimension. C'est un ensemble de logiciels qui permettent le déploiement d'applications sémantiques reposant sur les technologies Microsoft « .NET » et sur le serveur de base de données Microsoft SQL Server.

Comme le montre la figure 4.7, « Semantics.Framework » est une bibliothèque regroupant des classes permettant d'intégrer les applications sémantiques dans des logiciels et services écrit avec le langage « .NET ». Le composant « Semantics.Server » ajoute une

29. <http://www.intellidimension.com/products/semantics-platform>

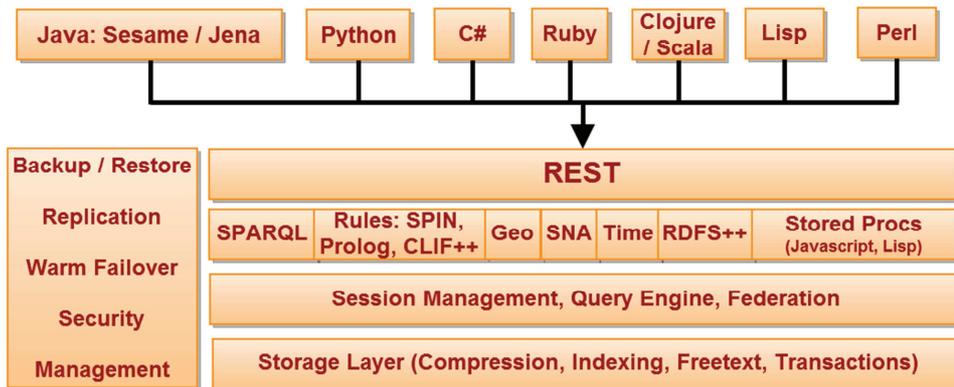


FIGURE 4.6 – Architecture d'AllegroGraph 4.10

couche à MS SQL Server afin de stocker les triplets RDF et de traduire les interrogations SPARQL. Le composant « Semantics.Datacenter » ajoute des fonctionnalités de clustering aux applications sémantiques.

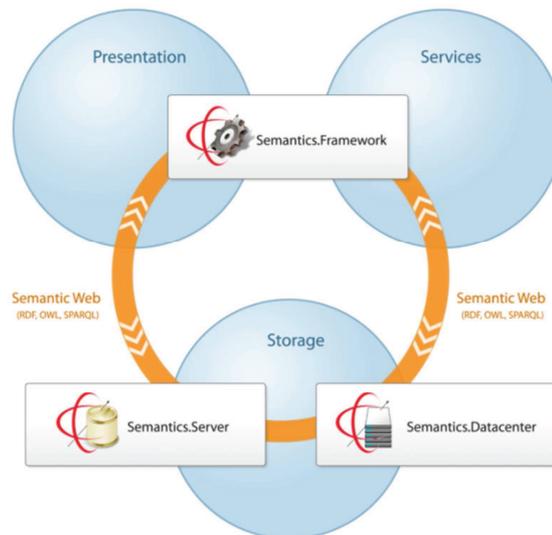


FIGURE 4.7 – Architecture de Semantics Plateform 2.0

4.6.5 OntoDb

OntoDb [Jean *et al.*, 2007] est une base de données à base ontologique conçue par le Laboratoire d'informatique et d'automatique des Systèmes (LIAS). Dans OntoDB, ontologie et données sont toutes deux stockées dans la base de données et font l'objet de mêmes traitements (insertion, mise à jour, requêtes, etc.). La cohabitation se fait de telle sorte qu'à chacune des données présentes dans la base, on associe le concept de l'ontologie qui la définit. Le modèle d'architecture d'OntoDB [Jean *et al.*, 2007] représenté par la figure 4.8, est constitué des deux parties traditionnelles des bases de données (données

et méta-base), d'une partie ontologie, et d'une partie méta-schéma contenant le méta-modèle du modèle d'ontologie utilisé, permettant de rendre générique tout traitement sur les ontologies. L'architecture OntoDB est constituée d'un Système de Gestion de base de données PostgreSQL et du modèle d'ontologie PLIB. EXPRESS est le langage utilisé pour l'interrogation de la base de données.

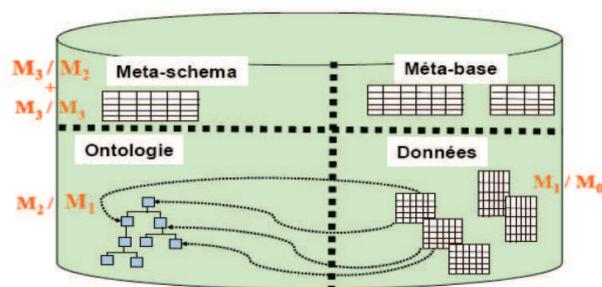


FIGURE 4.8 – Architecture d'OntoDB

4.6.6 Récapitulatif

Une base de données à base ontologique idéale doit concilier un modèle de données économe en espace disque, grâce à des systèmes de compression et d'encodage, mais doit rester rapide à interroger grâce notamment à des index pertinents. Elle doit également permettre le chargement rapide des données grâce à des interfaces variées, riches et garantissant la validité des données chargées. De plus, le système doit être capable de comprendre les langages standards d'ontologie pour réaliser des calculs d'inférence performants et complets. Sa capacité à interroger et retourner les données rapidement est également décisive, comme sa faculté à s'interfacer avec des bases tierces, y compris relationnelles, pour croiser les données. Les systèmes présentés n'intègrent pas l'intégralité de ces caractéristiques.

Le tableau 4.9 dresse un comparatif non exhaustif des principales caractéristiques des BDBO présentées. Dans notre travail, nous avons besoin d'écrire les règles spatiales, temporelles et métiers, appelées aussi règles utilisateurs qui serviront à l'inférence. Nous devons également interroger les données RDF et le résultat de l'inférence via les deux langages SQL et SPARQL. Pour cela, nous avons choisi de travailler avec la base de données ontologiques de Oracle Semantic Data Store pour plusieurs raisons fondamentales :

- modèle de données simple reposant sur la notion de modèle défini dans un réseau et qui peut contenir plusieurs graphes de données RDF séparés ;
- support complet de RDFS et un sous ensemble de OWL très performant ;
- chargement de données en masse simple à mettre en œuvre via des API PL-SQL ;
- inférence basées sur des règles natives et des règles utilisateurs ;
- possibilité d'utiliser des raisonneurs extérieurs comme Pellet ;
- une interrogation des données via SQL, SPARQL et des modules extérieurs.

	Virtuoso	AllegroGraph	Semantics Platform	Oracle Semantic Data Store
STOCKAGE				
Schéma de stockage	Schéma inconscient sur bdd relationnelle	Pseudo schéma inconscient sur bdd graphe	Schéma inconscient sur bdd relationnelle	Schéma inconscient sur bdd relationnelle
Stockage des IRI	Basé sur un identifiant	Basé sur un identifiant	Basé sur un identifiant	Basé sur un identifiant
Compression des préfixes	Dans une table séparée	Non	Non	OUI (api SESAME)
Unité de stockage	Graphe	Entrepôt	Modèle	Graphe
Index (sur la table des triplet)	Création et suppression possible	Création et suppression possible	Création et suppression possible	Création et suppression possible
Vue RDF des données relationnelles	Linked Data Views et sponger	RDB Mapper	Non	Oui (native)
Stockage distribué et/ou fédéré	Oui	Oui	Oui	Oui
Sécurité des données	Oui	Oui	Oui	Oui
Gestion des versions et évolutions des ontologies	Non	Non	Non	Workspace Manager Support for RDF Data
CHARGEMENT				
Chargement incrémental/en masse	Oui	Oui	Oui	Oui
Format d'entrée	N-quad N-tripl Turtle RDF/XML RDFa	N-Quad N-Triple Turtle RDF/XML TriX	N-Quad N-Triple Turtle RDF/XML	N-Quad N-Triple Turtle RDF/XML
Effet sur l'architecture de stockage	Insertions parallélisables	Insertions facilitées par encodage des chaînes	-	Insertions parallélisables
L'ajout de données en masse	Doublons interdits par clé primaire Reindexation manuelle	Suppression des doublons post-insertion Reindexation automatique	Doublons interdits par clé primaire Reindexation manuelle	Doublons interdits par clé primaire Reindexation automatique
Réutilisation des nœuds anonymes	Réutilisation dans le graphe Non réutilisation entre graphe sauf si égalité explicitement spécifiée	Réutilisation totale grâce à l'encodage directe des chaînes	Réutilisation dans le graphe Non réutilisation entre graphe	Réutilisation dans le graphe Non réutilisation entre graphe
INFERENCE				
Systèmes d'implications standards	Support limité de RDFS et d'OWL	RDFS et support limité d'OWL	Support limité de RDFS et d'OWL	RDFS, OWL-Prime, OWLSIF
moteurs d'inférence tiers	Oui (via drivers)	Oui (RacerPro)	Non	Oui
Règles utilisateur	Oui	Oui	Oui	Oui
Chainage avant et chainage arrière	Chainage avant Chainage arrière	Chainage avant Chainage arrière (Racerpro)	Chainage arrière	Chainage avant Chainage arrière
INTERROGATION				
Le langage d'interrogation	SPARQL 1.1 SQL + SPARQL	SPARQL 1.1 Prolog	SPARQL 1.0 + extensions	SEM_MATCH (SQL query) SPARQL 1.0 + extensions
Performance des requêtes	Traduction en SQL par un optimiseur	Pas de traduction SQL nécessaire	Non	Oui
Optimisation des requêtes	CBO	Optimiseur	CBO + cache SQL Server	Optimiseur
Interrogation des données relationnelles dans un contexte RDF	SQL dans SPARQL	Conversion des données relationnelles nécessaire	Non	Oui

FIGURE 4.9 – Tableau comparatif des BDBO étudiées

Chapitre 5

Mise en œuvre

Sommaire

5.1	Approche de mise en œuvre des ontologies	139
5.2	Création des parties déclaratives des ontologies	140
5.3	Peuplement de l'ontologie owlSealTrajectory	141
5.4	Vérification de la consistance des données ontologiques	143
5.5	Inférence et règles	143
5.6	Expérimentation	147
5.7	Synthèse	149

5.1 Approche de mise en œuvre des ontologies

Pour l'implantation des ontologies de la trajectoire, du temps et de l'espace, nous utilisons le support *Oracle Semantic Data Store* basé sur le système de gestion de bases de données Oracle. Cette technologie a évolué depuis la version Oracle 10g, Oracle 11g et prend le nom de *Oracle Spatial and Graph - RDF Semantic Graph*³⁰ dans la version Oracle 12c.

La base de données à base ontologique de Oracle offre un support pour la persistance, l'inférence et l'interrogation des ontologies par une implantation des standards RDF, RDFS et une grande partie du langage OWL. Ce SGBD définit un noyau dans sa méta-base pour supporter les technologies liées aux données ontologiques. Il stocke l'ontologie avec les données sous forme de triplets RDF dans le système sous un schéma appelé *MDSYS*. Chaque triple {sujet, prédicat, objet} est traité comme un objet base de données. La description détaillée de cette technologie dépasse le cadre de ce travail.

Les étapes de la création des ontologies (trajectoire, temps, espace) sont :

1. création des parties déclaratives des trois ontologies ;
2. création des instances et peuplement de l'ontologie de la trajectoire ;

30. <http://www.oracle.com/technetwork/database/options/semantic-tech>

3. vérification de la consistance des instances ontologiques ;
4. définition des règles d'inférence du domaine, du temps et de l'espace.

5.2 Création des parties déclaratives des ontologies

Dans ce paragraphe, nous discutons la création de la partie déclarative de l'ontologie de trajectoire `owlSealTrajectory`. Le même processus est utilisé pour les ontologies du temps et de l'espace. Nous avons choisi d'utiliser une programmation côté serveur en utilisant les API PL-SQL fournies par l'éditeur Oracle.

1. Création du réseau sémantique pour le support des ontologies. Un seul réseau est possible par instance base de données. Pour cela, on utilise `CREATE_SEM_NETWORK`, une fonction du paquetage `SEM_APIS`, ou le paramètre `rdf_tblspace` indique l'espace de table qui va contenir physiquement les déclarations et les instances (code 5.1).

```
1 EXECUTE SEM_APIS.CREATE_SEM_NETWORK('rdf_tblspace')
```

Code 5.1 – Ordre PL-SQL pour la création d'un réseaux

2. Création de la table pour des données RDF. Cette table est considérée comme toute table relationnelle mais inclues les triplets RDF. L'identifiant de chaque triple est optionnel, peut être donné par l'utilisateur ou géré par le système. Le type `SDO_RDF_TRIPLE_S` définit la structure du stockage physique des données, autrement dit les triplets RDF, dans la base de données (code 5.2).

```
1 CREATE TABLE owlSealTrajectory_rdf_data (id NUMBER, triple
SDO_RDF_TRIPLE_S)
```

Code 5.2 – Ordre SQL pour la création de la table des données

3. Création du modèle de données. Ce modèle donne un nom global aux données RDF en précisant la table relationnelle contenant ces données et le nom de la colonne qui contient précisément les triplets RDF (code 5.3).

```
1 EXECUTE SEM_APIS.create_rdf_model('owlSealTrajectory', '
owlSealTrajectory_rdf_data', 'triple')
```

Code 5.3 – Ordre PL-SQL pour la création du modèle

4. Création de la partie déclarative de l'ontologie de trajectoire `owlSealTrajectory`. Pour montrer un aperçu de cette déclaration, nous utilisons une création incrémentale basée sur les ordres SQL `INSERT`. Le code 5.4 donne une partie de la création de cette ontologie.

```
1 INSERT INTO owlSealTrajectory_rdf_data VALUES (1,
2 SDO_RDF_TRIPLE_S('owlSealTrajectory',
3 'http://13i.univ-larochelle.fr/owlSealTrajectory',
4 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type',
5 'http://www.w3.org/2002/07/owl#Ontology'));
```

```
6
7 INSERT INTO owlSealTrajectory_rdf_data VALUES (2,
8 SDO_RDF_TRIPLE_S('owlSealTrajectory',
9 'http://l3i.univ-larochelle.fr/owlSealTrajectory',
10 'http://www.w3.org/2002/07/owl#imports'
11 'http://www.w3.org/2006/time');
12
13 INSERT INTO owlSealTrajectory_rdf_data VALUES (3,
14 SDO_RDF_TRIPLE_S('owlSealTrajectory',
15 'http://l3i.univ-larochelle.fr/owlSealTrajectory',
16 'http://www.w3.org/2002/07/owl#imports'
17 'http://l3i.univ-larochelle.fr/owlOGCSpatial');
18 ...
19 INSERT INTO owlSealTrajectory_rdf_data VALUES (33,
20 SDO_RDF_TRIPLE_S('owlSealTrajectory',
21 'http://l3i.univ-larochelle.fr/owlSealTrajectory#Sequence',
22 'http://www.w3.org/2002/07/owl#equivalentClass'
23 'http://www.w3.org/2006/time#ProperInterval');
24 ...
25 INSERT INTO owlSealTrajectory_data VALUES (40,
26 SDO_RDF_TRIPLE_S('owlSealTrajectory',
27 'http://l3i.univ-larochelle.fr/owlSealTrajectory#GeoSequence',
28 'http://www.w3.org/2002/07/owl#equivalentClass'
29 'http://l3i.univ-larochelle.fr/SpatialOntology#Line');
30 ...
```

Code 5.4 – Ordres SQL pour la création de la partie déclarative de l'ontologie

Le premier ordre SQL déclare l'ontologie `owlSealTrajectory`. Le deuxième ordre importe l'ontologie du temps `OWLTime`, le troisième importe l'ontologie de l'espace `owlOGCSpatial` et les deux derniers relient respectivement les deux concepts `Sequence` et `ProperInterval`, `GeoSequence` et `Line` par la propriété `owl:equivalentClass`, etc.

5.3 Peuplement de l'ontologie owlSealTrajectory

Les techniques de peuplement d'une ontologie à partir de sources de données existantes, typiquement des bases de données, s'appuient sur des transformations appelées aussi correspondance (*Ontology-Database mapping*). Les approches de correspondance ontologie-base de données peuvent être classées en deux catégories :

1. création d'une ontologie à partir d'une base de données ;
2. définition d'une correspondance entre une base de données et une ontologie existante.

Une étude des approches actuelles dans ce domaine est produite par le groupe RDB2RDF au sein du W3C [Sahoo *et al.*, 2009].

Dans notre contexte, nous nous retrouvons dans la seconde catégorie. Autrement, nous souhaitons peupler l'ontologie `owlSealTrajectory` directement et automatiquement par

les données capturées que nous avons migrées vers le SGBD Oracle. Pour cela, nous utilisons la plateforme D2RQ [Christian Bizer, 2007; Bizer, 2004b]. Cette plateforme fournit un langage déclaratif pour définir la correspondance entre une ontologie et une base de données relationnelle. Une ontologie est associée à un schéma de base de données en utilisant `d2rq:ClassMaps` et `d2rq:PropertyBridges`. L'objet central de cette plateforme et aussi l'objet de départ lors de la définition d'une correspondance D2RQ est la `ClassMap`. Elle représente une classe ou un groupe de classes similaires de l'ontologie. Une `ClassMap` précise l'identification des instances de la classe. Elle dispose d'un ensemble de `PropertyBridges` qui précisent la création des propriétés d'une instance. La figure 5.1 illustre un extrait de l'ontologie `owlSealTrajectory` peuplée par les instances.

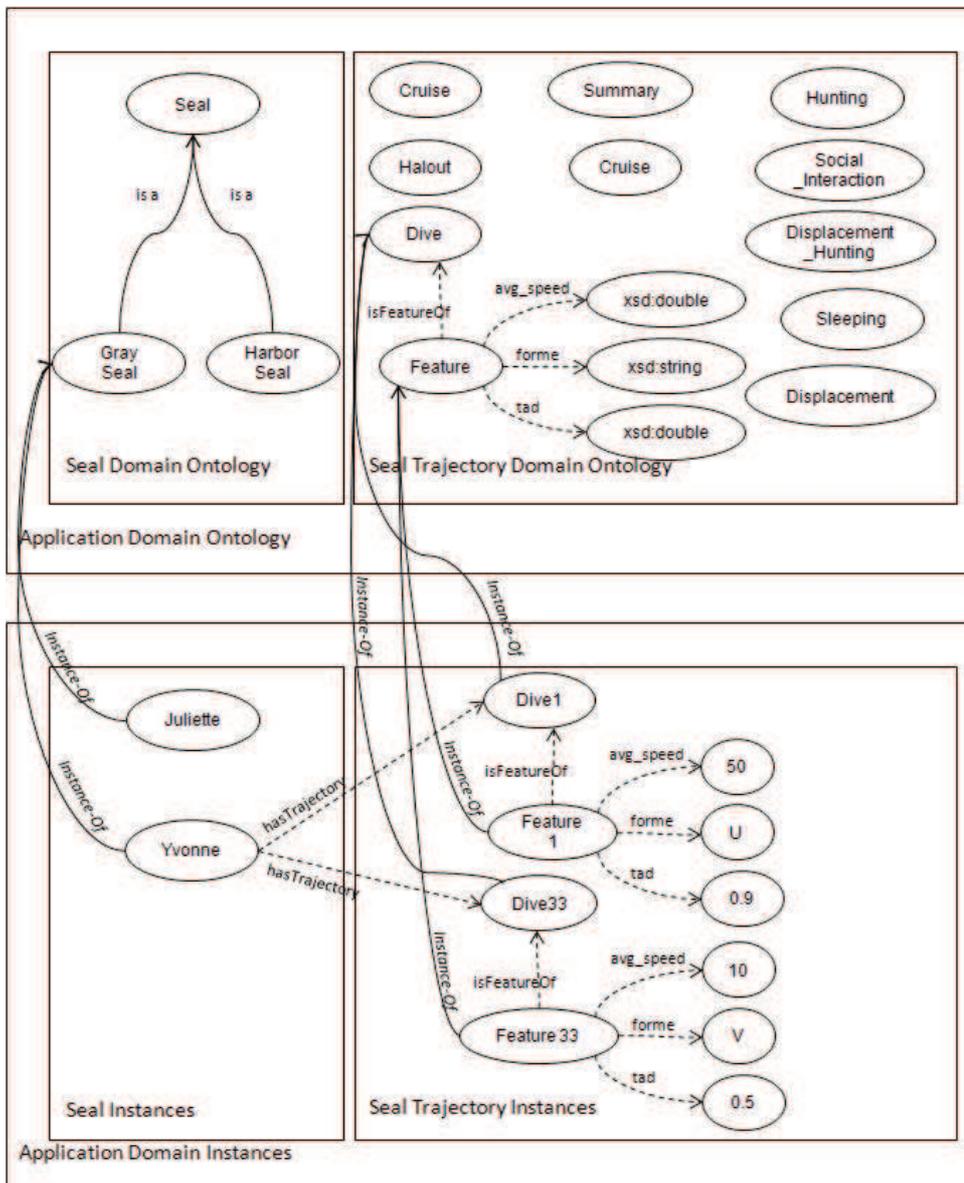


FIGURE 5.1 – Un extrait de l'ontologie `owlSealTrajectory` peuplée

5.4 Vérification de la consistance des données ontologiques

Une ontologie peut contenir des erreurs telles que des classes insatisfiables, des individus appartenant à cette dernière catégorie de classes, ou même des individus affirmés égaux et différents en même temps. L'ordre SQL (Code 5.5) vérifie l'ontologie `owlSealTrajectory` avec ses individus.

```
1 SELECT SEM_APIS.VALIDATE_MODEL(SEM_MODELS('owlSealTrajectory'))  
2 FROM DUAL;
```

Code 5.5 – Validation de l'ontologie `owlSealTrajectory`

5.5 Inférence et règles

Une inférence est un processus de raisonnement qui s'appuie sur des connaissances acquises, et qui s'articule autour de règles fondamentales pour permettre d'obtenir de nouvelles données.

Au sein des bases de données traditionnelles, on connaît l'appariement des données basé sur les valeurs mais aucune nouvelle donnée ne sera produite en dehors des données existantes. On peut éventuellement résoudre quelques déductions liées à des règles de gestion en utilisant des déclencheurs (ie *Triggers*), mais cela reste toujours lié aux valeurs des données et non à la sémantique portée par les données. Ainsi, dans une BDBO, l'inférence est la capacité de faire des déductions logiques fondées sur des règles. Les requêtes utilisent le résultat de l'inférence pour effectuer des correspondances sémantiques basées sur des relations significatives entre les données.

Actuellement, plusieurs moteurs d'inférences gratuits ou commerciaux tels que Racer, Pellet, Fact, Fact++, Surnia, F-OWL et Howlet existent. La plupart de ces moteurs sont conçus pour raisonner sur les logiques de description, mais acceptent en entrée des ontologies 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 ontologiques.

L'inférence implique l'utilisation de règles. On distingue trois types de règles : les règles liées au langage de formalisation des ontologies, les règles définies par l'utilisateur et les règles nécessaires pour le raisonnement spatio-temporel.

5.5.1 Règles d'inférence du langage

Le BDBO *Oracle Semantic Data Store 11g* supporte nativement le vocabulaire *RDFS* et une partie de OWL appelée *OWLPRIME*. Dans sa version actuelle 12c, appelée *Oracle Spatial and Graph RDF Semantic Graph*, elle prend en charge les vocabulaires de *RDFS++*, *OWLSIF* et *OWLPrime*, qui ont une expressivité importante, ainsi que *OWL2RL*. Chaque

vocabulaire soutenu possède une base de règles correspondante, cependant, ces bases de règles n'ont pas besoin d'être peuplées parce que les règles d'inférence sous-jacentes de ces trois vocabulaires sont implémentées en interne. Les vocabulaires supportés sont :

- *RDFS++* : une extension minimale de *RDFS* en incluant les constructeurs `owl:sameAs` et `owl:InverseFunctionalProperty`.
- *OWLSIF* : OWL avec le IF sémantique, le vocabulaire et la sémantique proposés pour *pD** sémantique [ter Horst, 2005].
- *OWLPrime* : les fonctionnalités de OWL suivantes :
 - fondamentales : classe, sous-classe, propriété, sous-propriété, domaine, type ;
 - caractéristiques de propriété : transitive, symétrique, fonctionnel, fonctionnel inverse, inverse ;
 - comparaison de classe : équivalence, disjonction ;
 - comparaison de propriété : équivalence ;
 - comparaison d'instances : même, différent ;
 - expression de classe : complément ;
 - restriction de propriété : `hasValue`, `someValuesFrom`, `allValuesFrom`. Comme avec *pD**, la sémantique prise en charge pour ces restrictions de valeur sont intentionnelles (le IF sémantique).
- *OWL2RL* : décrit dans la section « OWL 2 RL » de la recommandation du W3C concernant le OWL2 Web Ontology Language Profils³¹.
- *OWL2EL* : décrit dans la section « OWL 2 EL » de la recommandation du W3C concernant le OWL2 Web Ontology Language Profils³².

5.5.2 Règles d'inférence métiers

Dans notre domaine d'application, les règles d'inférence métiers sont définies par les règles liées aux annotations sémantiques de la trajectoire, en particulier les règles associées aux activités du phoque. L'ontologie `owlSealTrajectory` contient 5 règles pour les activités du phoque exprimées par la table de décision (tableau 3.1). Cet ensemble de règles est défini dans la base de règle `sealActivities_rb`. À titre d'exemple, nous détaillons la création de la règle liée à l'activité de la chasse (Code 5.6).

```

1 EXECUTE SEM_APIS.CREATE_RULEBASE('sealActivities_rb') ;
2
3 INSERT INTO mdsys.semr_sealActivities_rb VALUES(
4 'hunting_rule',
5 '(?x rdf:type ost:Dive) (?x ost:hasFeature ?fe) (?fe ost:tad ?t) (?fe ost:avg_speed
6 ?a) ',
7 '(t > 0.8) and (t < 1) and (a > 0.5)',
8 '(?x ost:seqHasActivity ?activiteJ) (?activiteJ rdf:type ost:Hunting)',
9 SEM_ALIASES(SEM_ALIAS('ost', 'http://13i.univ-larochelle.fr/owlSealTrajectory#')));

```

Code 5.6 – Création de la règle de chasse

31. http://www.w3.org/TR/owl2-profiles/#OWL_2_RL

32. http://www.w3.org/TR/owl2-profiles/#OWL_2_EL

Dans le code 5.6, la ligne (1) crée la base de règle `sealActivities_rb`. Cela a pour conséquence la création d'une vue nommée `semr_sealActivities_rb` dans le schéma `mdsys`. Cette vue est accessible en lecture et écriture à son créateur (ligne 3), et chaque nouvelle règle insérée dans la base doit avoir un nom unique (ligne 4). La ligne (5) définit la partie conditionnelle sous forme d'un patron de triplet RDF. La ligne (6) introduit un filtre pour limiter le sous-graphe défini par la partie conditionnelle. La conséquence de l'application de la règle est donnée par la ligne (7). Enfin, la dernière ligne permet de qualifier les espaces de noms utilisés dans la définition de la règle. Dans la définition de cette règle, on souligne que la valeur ($a > 0.5$) désigne une vitesse de plongée élevée. Il est certain que ce paramètre doit être étudié. Aussi, la prise en compte du paramètre de *profondeur par rapport au fond* dans la définition des règles n'est pas encore possible à cause de l'absence des sources de données sur la bathymétrie. Les autres règles métiers s'écrivent sur ce même modèle dans la base de règles `sealActivities_rb`.

5.5.3 Règles temporelles

L'ontologie du temps `OWLTime` déclare 13 propriétés-objets qui correspondent aux relations temporelles d'Allen entre intervalles (code 5.7).

```
intervalEquals, intervalBefore, intervalDuring, intervalOverlaps,
intervalStartedBy, intervalOverlappedBy, intervalFinishes, intervalFinishedBy,
intervalContains, intervalMetBy, intervalStarts, intervalMeets, intervalAfter
```

Code 5.7 – Relations temporelles associées aux règles de l'ontologie `OWLTime`

Pour chaque relation, on associe une règle d'inférence dans la base de règle `owlTime_rb` qui décrit sa partie impérative. Pour la mise en œuvre des parties impératives des règles, nous utilisons le calcul sur les intervalles basé sur les opérations définies par la classe `TM_RelativePosition` de la spécification ISO/TC 211 concernant le schéma temporel [ISO-TC-211, 2002]. Le code 5.8 détaille la règle `intervalAfter_rule` associée à la relation temporelle `intervalAfter`. Les autres règles métiers s'écrivent sur ce même modèle dans la base de règles `owlTime_rb`.

```
1 EXECUTE SEM_APIS.CREATE_RULEBASE('owlTime_rb')
2
3 INSERT INTO mdsys.semr_owlTime_rb VALUES(
4 'intervalAfter_rule',
5 '(?x rdf:type owltime:ProperInterval) (?x owltime:hasEnd ?xEnd) (?xEnd :
   inXSDDateTime ?xEndDateTime)
6 (?y rdf:type owltime:ProperInterval) (?y owltime:hasBeginning ?yBegin) (?yBegin :
   inXSDDateTime ?yBeginDateTime)',
7 '(yBeginDateTime > xEndDateTime)',
8 '(?y owltime:intervalAfter ?x)',
9 SEM_ALIASES(SEM_ALIAS('owltime', 'http://www.w3.org/2006/time#')));
```

Code 5.8 – Création de la partie impérative de la règle `intervalAfter_rule`

Dans le code 5.8, la ligne (1) crée la base de règle `owlTime_rb`. Cela a pour conséquence la création d'une vue nommée `semr_sem_owlTime_rb` dans le schéma `mdsys`. Cette vue est accessible en lecture et écriture à son créateur (ligne 3), et chaque nouvelle règle insérée dans la base doit avoir un nom unique (ligne 4). La ligne (5), déclare un intervalle, argument de la relation temporelle, ainsi que sa date de fin. La ligne (6) déclare un intervalle, référent de la relation temporelle, ainsi que sa date début. La ligne (7) opère un filtrage sur les données en vérifiant la ou les conditions nécessaires pour déduire la relation temporelle, comme le montre la ligne (8). Enfin, la dernière ligne fournit la résolution des déclarations en précisant l'espace de noms utilisé.

5.5.4 Règles spatiales

L'ontologie de l'espace `owlOGCSpatial` déclare 8 propriétés objets qui correspondent aux relations topologiques entre des objets spatiaux (code 5.9).

<code>Equals, Disjoint, Intersects, Touches, Crosses, Within, Contains, Overlaps</code>

Code 5.9 – Relations spatiales associées aux règles de l'ontologie `owlOGCSpatial`

Pour chaque relation spatiale, on associe une règle d'inférence dans la base de règles `owlOGCSpatial_rb` qui décrit sa partie impérative. La mise en œuvre des règles fait appel aux fonctions implantées dans la cartouche spatiale du SGBD Oracle. Le code 5.10 décrit la création de la règle `contains_rule` entre deux objets spatiaux.

```

1 EXECUTE SEM_APIS.CREATE_RULEBASE('owlOGCSpatial_rb');
2
3 INSERT INTO mdsys.sem_owlOGCSpatialRelations_rb
4 VALUES(
5 'contains_rule',
6 '(?spatialObjet1 rdf:type os:Geometry) (?spatialObjet1 os:srs ?srsSpatialObjet1) (?
   srsSpatialObjet1 os:srid ?sridSpatialObjet1) (?spatialObjet1 os:type ?
   typeSpatialObjet1) (?spatialObjet1 os:wkt ?strSpatialObjet1)
7 (?spatialObjet2 rdf:type os:Geometry) (?spatialObjet2 os:type ?typeSpatialObjet2) (?
   spatialObjet2 os:wkt ?strSpatialObjet2)',
8 '(evalSpatialRelation(typeSpatialObjet1, strSpatialObjet1, typeSpatialObjet2,
   strSpatialObjet2, sridSpatialObjet1, 'contains') = 1)',
9 '(?spatialObjet1 os:contains ?spatialObjet2)',
10 SEM_ALIASES(SEM_ALIAS('os', 'http://13i.univ-larochelle.fr/owlOGCSpatial#')));

```

Code 5.10 – Création de la partie impérative de la règle `contains_rule`

Dans les lignes (6 et 7), la propriété-donnée (`type`) identifie de façon unique le type de l'objet spatial, la propriété-donnée (`wkt` : well-known text format) représente les coordonnées de l'objet spatial. L'identification et la représentation d'un objet spatial sont décrites par la spécification de l'OGC et implantées par le SGBD. Dans la ligne (8), la procédure PL-SQL `evalSpatialRelation` interroge le SGBD pour évaluer la relation spatiale entre les deux objets.

5.6 Expérimentation

Pour évaluer l'ontologie `owlSealTrajectory`, on considère une requête qui s'intéresse à des faits liés au domaine (les activités du phoque), dans un intervalle de temps et dans une zone spatiale (code 5.11).

```
Trouver toutes les plongées de repos, déplacement et chasse (dans cet ordre)
durant [2003-08-02T00:00:00, 2003-08-09T23:59:00] contenu dans la zone (48, -4,
48, -5, 49, -5, 48, -4)}
```

Code 5.11 – Exemple d'une requête utilisant les trois ontologies

Le patron (repos, déplacement, chasse) correspond à une forte activité de chasse selon l'expert du domaine. On considère une base de données de l'année 2003 d'un seul phoque. Les statistiques de la base de connaissances sont :

- total des triplets est (à peu près) : 200 000 ;
- total des triplets de chaque activité dans le patron (TAP) : 66 000 ;
- nombre de triplets Repos : 17 880, soit 28% du TAP ;
- nombre de triplets Déplacement : 43 710, soit 67% du TAP ;
- nombre de triplets Chasse : 4 410, soit 7% du TAP.

Avant d'interroger la base de données ontologique, on doit d'abord lancer le processus d'inférence (Code 5.12). Pour cela, on utilise la fonction `CREATE_ENTAILMENT` du paquetage `SEM_API` qui crée une inférence, ou index de règles, nommé `owlSealTrajectory_idx`. Une inférence est considéré comme un objet contenant des données précalculées qui peuvent être déduites à partir de l'application d'un ensemble spécifié de bases de règles à un ensemble spécifique de modèles. Une vue est créée, nommée `SEMI_owlSealTrajectory_idx` dans le schéma `MDSYS`, pour accéder aux données inférées.

Dans le code 5.12, la ligne (2) donne un nom à l'index. Les lignes (3,4,5) indiquent les modèles utilisés par le processus d'inférence. Les ligne (6,7,8,9) indiquent les bases de règles qui doivent être prises en compte lors de ce processus. Elles indiquent, la base de règles `OWLPrime` pour assurer la correspondances entre les ontologies, la base de règles utilisateurs `owlSealTrajectory_rb`, la base de règles temporelles `owlTime_rb` et la base de règles spatiales `owlOGCSpatial_rb`. Enfin, la ligne (10) donne des caractéristiques concernant l'index créé.

Pour interroger la base de données ontologiques, on utilise la fonction table `SEM_MATCH`. Cette fonction table est définie par le code 5.13.

L'attribut `query` est une chaîne littérale (ou concaténation de littéraux de chaîne) avec un ou plusieurs patrons de triples, appelés aussi motifs, contenant généralement des variables. Un triple motif est un triple d'atomes suivis d'un point. Chaque atome peut être une variable (par exemple, `x?`), un nom qualifié (par exemple, `rdf:type`) qui est développé sur la base des espaces de noms par défaut et la valeur de l'attribut d'alias, ou une URI complète (par exemple, `<http://l3i.univ-larochelle.fr/owlSealTrajectory/Dive>`). En outre, le troisième atome peut être un littéral numérique (par exemple, `3,14`), un littéral ordinaire (par exemple, « Herman »), un langage balisé littéral ordinaire (par exemple,

```

1 EXECUTE SEM_API.CREATE_ENTAILMENT(
2 'owlSealTrajectory_idx',
3 sem_models('owlSealTrajectory',
4           'owlTime',
5           'owlOGCSpatial'),
6 sem_rulebases('OWLPRIME',
7              'owlSealTrajectory_rb',
8              'owlTime_rb',
9              'owlOGCSpatial_rb'),
10 SEM_APIS.REACH_CLOSURE, null, 'USER_RULES=T');

```

Code 5.12 – Création d'un processus d'inférence

```

1 SEM_MATCH(
2   query      VARCHAR2,
3   models     SEM_MODELS,
4   rulebases  SEM_RULEBASES,
5   aliases    SEM_ALIASES,
6   filter     VARCHAR2,
7   index_status VARCHAR2,
8   options    VARCHAR2,
9   graphs     SEM_GRAPHS,
10  named_graphs SEM_GRAPHS
11 ) RETURN ANYDATASET;

```

Code 5.13 – Définition de la fonction SEM_MATCH

"Herman"@fr), ou un dactylographié littéral (par exemple, "123"^^xsd: int). L'attribut `query` est le point commun entre l'interrogation des données ontologiques de Oracle et le langage SPARQL. Pour cette interrogation, nous avons besoin des modèles, attribut `models`, des bases de règles utilisées, attribut `rulebases`, des alias nécessaires, attribut `aliases`, la condition vérifiée par les données, attribut `filter`. Notons que l'interrogation des données exigent la présence d'un index basé sur un ensemble de bases de règles contenant celles qui figurent dans l'expression de l'interrogation. Cet index peut être valide ou complet ou même incomplet, voire d'autre statut, en utilisant l'attribut `index_status`. L'interrogation peut avoir plusieurs options et peut aussi porter sur des parties distinctes des modèles spécifiés en utilisant les attributs `graph` et `named_graphs`.

La formulation de la requête 5.11 en utilisant la fonction table `SEM_MATCH` sur les données ontologiques définies est donnée par le code 5.14.

```

1 SELECT D1, D2, D3
2 FROM TABLE ( SEM_MATCH(
3 '(?D1 rdf:type ost:Dive) . (?D1 ost:sequenceHasActivity ?activiteD1) . (?activiteD1
4   rdf:type ost:Resting) .
5 (?D2 rdf:type ost:Dive) . (?D2 ost:sequenceHasActivity ?activiteD2) . (?activiteD2
6   rdf:type ost:Displacement) .
7 (?D3 rdf:type ost:Dive) . (?D3 ost:sequenceHasActivity ?activiteD3) . (?activiteD3
8   rdf:type ost:Hunting) .
9 (?D1 ot:intervalBefore ?D2) . (?D2 ot:intervalBefore ?D3) .

```

```

8
9 (?I rdf:type ot:ProperInterval) . (?I ot:hasBeginning ?beginI) . (?beginI ot:
   inXSDDateTime "2003-08-02T00:00:00"^^xsd:datetime) . (?I ot:hasEnd ?endI) . (?
   endI ot:inXSDDateTime"2003-08-09T23:59:00"^^xsd:datetime) .
10
11 (?D1 ot:intervalDuring ?I) . (?D2 ot:intervalDuring ?I) . (?D3 ot:intervalDuring ?I)
   .
12
13 (?Z rdf:type os:Geometry) . (?Z ot:type "5"^^xsd:int) . (?Z os:wkt "48,-4, 48,-5,
   49, -5, 48, -4"^^xsd:string) .
14
15 (?Z os:contains ?D1) . (?Z os:contains ?D2) . (?Z os:contains ?D3)',
16
17 SEM_Models('owlSealTrajectory', 'owlTime', 'owlOGCSpatial'),
18 SEM_Rulebases('OWLPRIME', 'sealActivities_rb', 'owlTime_rb', '
   owlOGCSpatialRelations_rb'),
19 SEM_ALIASES(SEM_ALIAS('ost', 'http://l3i.univ-larochelle.fr/owlSealTrajectory#'),
20 SEM_ALIAS('ot', 'http://www.w3.org/2006/time#'),
21 SEM_ALIAS('os', 'http://l3i.univ-larochelle.fr/owlOGCSpatial#')),
22 null)
23 );

```

Code 5.14 – Formulation SQL de la requête 5.11

Dans le code 5.14, de la ligne (3) à (5) on cherche les activités repos, déplacement et chasse. La ligne (7) définit l'ordre temporel de ces activités. La ligne (9) définit l'intervalle de temps donné par la requête. La ligne (11) exprime la relation entre les intervalles de temps associés aux activités et l'intervalle donné. La ligne (13) définit la zone spatiale donnée par la requête et la ligne (15) exprime sa relation avec les objets spatiaux associés aux activités. Le reste de l'expression indique les modèles, les bases de règles, et les espaces de noms.

Dix plongées de la base de données ontologiques répondent à cette requête. Le temps de l'inférence ajouté au temps de réponse de la requête est de l'ordre d'une heure sur une machine Intel Core(TM) 2 Quad CPU 2.83 GHz, 3.25 Go de RAM. Le temps d'inférence actuel n'est pas encore à un niveau acceptable pour une application exploitable. Cette problématique est en cours d'étude dans les travaux de thèse de Rouaa WANNOUS [Wannous *et al.*, 2013].

5.7 Synthèse

Dans ce chapitre, nous avons présenté une démarche pour l'implantation des ontologies de la trajectoire, du temps et de l'espace. Ainsi, nous avons montré la création des parties déclaratives de ces ontologies. La création des parties impératives liées aux règles prises en compte par ces ontologies a fait l'objet d'une implantation et de tests unitaires et fonctionnelles que nous n'avons pas présentés dans ce manuscrit. Le chargement des données dans l'ontologie de trajectoire utilise le principe de correspondance entre une ontologie et une

base de données relationnelle. Nous avons utilisé un chargement en masse. Nous avons vérifié la consistance de ces données ontologiques. Afin d'interroger les données ontologiques, nous avons détaillé le processus de calcul de l'inférence basé sur la création d'index sur les données. Enfin, nous avons discuté la partie interrogation des données ontologiques. Le diagramme de communication d'applications modélisé dans le cadre de l'architecture TOGAF³³, figure 5.2, montre les principales parties de l'architecture ontologique présentée.

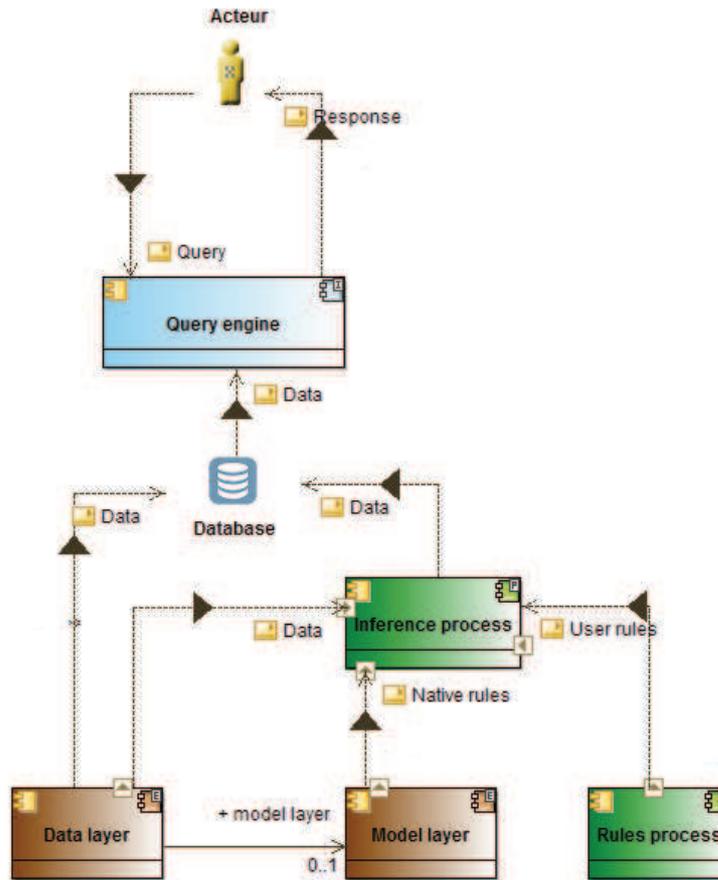


FIGURE 5.2 – Diagramme de communication d'applications pour l'architecture ontologique présentée

33. <http://www.togaf-modeling.org>

Troisième partie
Conclusions et perspectives

RETOURS SUR LES IMPACTS DE LA THÈSE

Réponses aux problématiques initiales

En introduction de ce manuscrit, nous avons résumé la problématique scientifique motivant nos travaux de la manière suivante : gérer de grandes masses de données spatio-temporelles, typiquement des trajectoires, en considérant les règles du domaine. Ainsi, nous avons montré tout au long de cette thèse de quelle manière nous envisagions la considération de ces données, à la fois en termes, d'analyses, de représentations et de modèles. Pour atteindre cet objectif, nous nous sommes basés sur deux domaines d'applications pour établir une démarche d'analyse rigoureuse puis un modèle de données le plus générique possible. Enfin, nous nous sommes intéressés à un troisième domaine d'application pour valider la démarche d'analyse et les modèles. Ce dernier domaine d'application sera en plus notre champ de validation de l'approche globale présentée. Avant de restituer totalement nos travaux et d'y apporter un regard critique, revenons sur les trois axes de recherche majeurs définis dans ce mémoire.

Les modèles de données spatiales, temporelles et spatio-temporelles

Les données des trajectoires possèdent des composantes spatiales et temporelles, voire même des composantes dont les deux aspects sont indissociables, nommées alors spatio-temporelles. Pour prendre en compte ces trois caractéristiques, nous nous sommes intéressés à quelques travaux sur la représentation et la modélisation des données spatiales, des données temporelles et enfin des données spatio-temporelles. Nous n'avons pas cherché l'exhaustivité ou tout savoir sur les approches existantes, mais nous avons pris un fil conducteur lié à l'aspect conceptualisation. Pour chaque composante, nous avons discuté au moins trois types de modélisation : une générale, une liée à une norme ou une spécification et puis une dernière utilisée dans le domaine des bases de données. Dans la suite de nos recherches, nous nous intéressons aux approches définies par cette dernière communauté.

Les modèles de données des trajectoires

Une fois les caractéristiques spatiales, temporelles ou spatio-temporelles d'une donnée est fixée, nous nous sommes intéressés à la considération de ces aspects dans une grande masse de données, dans notre cas, il s'agit des données des trajectoires. Ce sujet est souvent traité de façon très diverses en fonction de la communauté scientifique. Pour cela, nous avons donné un bref aperçu sur la représentation d'une trajectoire en science mathématiques et physique et tout de suite nous nous sommes intéressés aux modèles de données utilisés par la communauté bases de données. Même en se restreignant à cette communauté, on constate que les approches sont multiples et différentes. On souligne aussi un début de réflexion sur une représentation conceptuelle générale de l'objet trajectoire.

Les modèles ontologiques des trajectoires

Le champ de nos recherches dans ce domaine était structuré d'une façon graduelle. Dans un premier temps, nous nous intéressons aux approches ontologiques pour les don-

nées spatiales et temporelles. Dans ce cas, nous avons structuré notre discours autour de l'extension des logiques de description, quelques ontologies types puis nous avons étudié les mécanismes de raisonnement introduits. Dans un second temps, nous avons étudié quelques approches ontologiques pour les données spatio-temporelles. Nous avons montré que cette considération est souvent liée aux domaines d'applications et nous avons abordé le point de vue de la communauté géographique et la communauté base de données. Enfin, nous avons présenté les travaux majeurs utilisant des approches ontologiques pour les données des trajectoires. Nous avons discuté trois types d'approches : une première qui considère la trajectoire du point de vue de l'objet mobile, une deuxième basée sur une approche modulaire, et une dernière approche qui s'intéresse plutôt aux aspects thématiques, spatiales et temporelles de tous les données d'une façon globale.

Vision globale de notre contribution

De l'analyse à un modèle conceptuel des trajectoires

Pour les domaines d'application considérés, nous avons établi une démarche d'analyse des données capturées. Nous avons donné le contexte générale du domaine, la capture et l'analyse des trames puis nous avons présenté un modèle conceptuel de la trajectoire.

Modèle générique de la trajectoire

À partir des différents modèles liés aux domaines d'applications, nous avons défini un modèle générique ou patron conceptuel. Ce modèle représente la trajectoire d'une façon globale indépendamment du domaine d'application. Il montre les principales composantes d'une trajectoire en séparant les exigences sous forme de modules.

Modèle sémantique de la trajectoire

On définit le modèle de trajectoire sémantique par l'association entre le domaine sémantique et le domaine de la trajectoire. Le domaine sémantique est constitué par les activités du domaine avec différentes granularités : activité générale et activité de base hiérarchique, selon l'objet spatio-temporel associé.

Modèle ontologique de la trajectoire

Le modèle ontologique permet l'intégration des couches sémantiques métiers aux données. Pour cela, nous avons utilisé une approche basée sur la transformation des modèles issue du domaine de l'ingénierie des données. Le modèle sémantique de la trajectoire est alors transformé en une ontologie décrivant les concepts et leurs relations. Nous avons fait le choix de l'utilisation du langage ontologique OWL comme cadre de formalisation de l'ontologie de la trajectoire.

Réutilisation des ontologies dans l'ontologie de la trajectoire

L'ontologie de la trajectoire est formée par trois types de concepts : thématique, temporel et spatial. Pour aboutir à une modélisation complète et donc un raisonnement fiable, nous avons considéré pour chaque type de concept, et en particulier le type spatial et temporel, la possibilité de le connecter à un modèle ontologique existant. Nous avons ainsi présenté une ontologie de temps et une ontologie de l'espace. Aussi, nous avons formulé le problème de réutilisation et nous avons présenté une approche basée sur le principe de

réutilisation partielle. Comme notre cadre de formalisation s'appuie sur le langage OWL, alors nous avons présenté clairement le processus de liaison des concepts et des relations entre les différentes ontologies en utilisant les constructeurs de ce langage.

Approche bases de données pour la gestion des données ontologiques

Tout au long de notre travail, nous avons présenté le point de vue des bases de données durant les différentes étapes de notre travail. Dans cette partie, nous avons présenté la problématique de la gestion des instances d'une ontologie par les systèmes de gestion de bases de données (SGBD). Nous avons présenté les principales exigences nécessaires à savoir l'architecture de stockage des données ontologiques, le chargement, l'inférence et l'interrogation. Nous avons conclu cette partie par la présentation de quelques systèmes et un comparatif.

Vision globale sur la mise en œuvre

Mise en œuvre ontologiques

Sur la base de l'étude des système de bases de données ontologiques, nous avons choisi un cadre d'implantation. Sans rentrer dans les détails et difficultés techniques inhérentes à tout système, nous avons montré comment créer la partie déclarative d'une ontologie, comment on peut charger les données dans les instances ontologiques et comment vérifier la consistance de ces données.

Inférence ontologique

Après la définition d'un processus d'inférence, nous avons discuté les différentes règles prises en compte, à savoir, les règles du langage, thématiques, temporelles et spatiales. Pour chaque type de règle, nous avons montré les enjeux associés et l'approche utilisée pour son implantation ainsi que sa relation avec le processus d'inférence.

Interrogation des données ontologiques

Pour interroger les données ontologiques, nous nous sommes basés sur une requête de l'expert du domaine combinant à la fois les aspects thématiques, temporels et spatiaux. Nous avons montré clairement comment on peut la prendre en charge en s'appuyant sur les modèles, les règles et le processus d'inférence ontologiques.

REGARDS CRITIQUES SUR NOS TRAVAUX

Il nous semble également important dans cette conclusion de porter un regard critique sur nos travaux, revenant sur certains aspects qui auraient pu être améliorés ou encore considérés :

- comparer les performances de notre approche à d'autres approches ;
- définir des opérateurs sémantiques de haut niveau, plus complexe combinant à la fois le temps et l'espace. À titre d'exemple, nous souhaitons répondre à la requête qui cherche les phoques qui se croisent. Alors, le concept de croisement des trajectoires des phoques est plus complexe car il s'appuie à la fois sur le croisement dans le temps et le croisement dans l'espace ;

- améliorer les règles thématiques afin de prendre en compte l'objet mobile, ses spécificités et son environnement ;
- résoudre le problème de performance de l'inférence. En effet, le processus d'inférence utilisant toutes les données des trajectoires en tenant compte de toutes les règles nécessite un temps de calcul et un espace de stockage très importants.

PERSPECTIVES

À l'issu de ce travail de thèse, différentes perspectives de recherche venant dans la continuité des travaux présentés dans ce mémoire s'offrent à nous. Nous souhaitons ainsi axer une partie de nos travaux futurs autour des problématiques suivantes qui sont au cœur du travail de thèse WANNOUS Rouaa (2011-2014) :

- étude de l'inférence sur les données ontologiques : il s'agit d'étudier les mécanismes nécessaires pour définir des approches d'inférences moins coûteuses et simples à mettre en œuvre ;
- définition d'une nouvelle approche pour l'analyse des masses de données complexes sémantiques et multidimensionnelles issues de données hétérogènes (capteurs, bases de données, etc.) avec prise en compte de la sémantique du domaine. En effet, au lieu de considérer une approche base de données à base ontologique, nous souhaitons définir un modèle entrepôt de données ontologiques. Pour réaliser cet objectif, quatre verrous scientifiques doivent être levés

1. l'explicitation de la sémantique des sources de données alimentant l'entrepôt de données : la sémantique du domaine peut être capturée par les ontologies. Contrairement aux modèles conceptuels, les ontologies décrivent un domaine dans sa globalité. Pour mieux l'utiliser, les besoins des utilisateurs seront projetés sur cette ontologie afin de définir une ontologie locale spécifique à l'application à concevoir ;
2. la définition de schéma de l'entrepôt de données : cette considération implique alors la définition d'un modèle de données approprié prenant en compte les caractéristiques liées à la nature des données manipulées : spatio-temporelles ;
3. la modélisation et la persistance des besoins utilisateurs prenant en compte des propriétés spatio-temporelles dans l'entrepôt de données ;
4. l'exploitation efficace de l'entrepôt de données sémantique : cette exploitation est réalisée par la définition des algorithmes de fouille de données efficaces prenant en compte les dimensions spatiales et temporelles.

Bibliographie

- [Abdelmoty *et al.*, 2005] Alia I. Abdelmoty, Philip D. Smart, Christopher B. Jones, Gaihua Fu, and David Finch. A critical evaluation of ontology languages for geographic information retrieval on the internet. *J. Vis. Lang. Comput.*, 16(4) :331–358, 2005.
- [Allen and Ferguson, 1994] James F. Allen and George Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4 :531–579, 1994.
- [Allen, 1983a] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11) :832–843, 1983.
- [Allen, 1983b] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11) :832–843, 1983.
- [Alvares *et al.*, 2007a] Luis Otavio Alvares, Vania Bogorny, Bart Kuijpers, Jose Antonio Fernandes de Macedo, Bart Moelans, and Alejandro Vaisman. A model for enriching trajectories with semantic geographical information. In *GIS '07 : Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, pages 1–8, New York, NY, USA, 2007. ACM.
- [Alvares *et al.*, 2007b] Luis Otavio Alvares, Vania Bogorny Jose Antonio Fern, Es De Macedo, Bart Moelans, and Stefano Spaccapietra. Dynamic modeling of trajectory patterns using data mining and reverse engineering. In *In Twenty-Sixth International Conference on Conceptual Modeling - ER2007 - Tutorials, Posters, Panels and Industrial Contributions*, 2007.
- [Artale and Franconi, 1999] Alessandro Artale and Enrico Franconi. Representing a robotic domain using temporal description logics. *Artificial Intelligence in Engineering, Design, Analysis and Manufacturing*, 13(2) :105–117, 1999.
- [Artale and Franconi, 2001] Alessandro Artale and Enrico Franconi. A survey of temporal extensions of description logics. *Annals of Mathematics and Artificial Intelligence*, 30(1-4) :171–210, 2001.
- [Artale and Lutz, 2004] A. Artale and C. Lutz. A correspondence between temporal description logics. *Journal of Applied Non-Classical Logic*, 14(1-2) :209–233, 2004.
- [Artale *et al.*, 2003] Alessandro Artale, Enrico Franconi, and Federica Mandreoli. Description logics for modelling dynamic information. In *Logics for Emerging Applications of Databases. LNCS*, page 368. Springer-Verlag, 2003.
- [Auer *et al.*, 2009] Sren Auer, Sebastian Dietzold, Jens Lehmann, Sebastian Hellmann, and David Aumüller. Triplify – light-weight linked data publication from relational databases, 2009.

- [Baader and Hanschke, 1991] Franz Baader and Philipp Hanschke. A scheme for integrating concrete domains into concept languages. In *Proceedings of the 12th international joint conference on Artificial intelligence - Volume 1, IJCAI'91*, pages 452–457, San Francisco, CA, USA, 1991.
- [Baader *et al.*, 2003] Franz Baader, Ralf Küsters, and Frank Wolter. Extensions to description logics. In Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *Description Logic Handbook*, pages 219–261. Cambridge University Press, 2003.
- [Bacharach, 2008] Sam Bacharach. Ogc approves kml as open standard, 2008. Dernière visite 18/11/2012.
- [Bachimont, 2004] B. Bachimont. *Arts et sciences du numérique : ingénierie des connaissances et critique de la raison computationnelle*. PhD thesis, Université de Technologie de Compiègne, 2004.
- [Baratis *et al.*, 2009] Evdoxios Baratis, Euripides G. M. Petrakis, Sotiris Batsakis, Nikolaos Maris, and Nikos Papadakis. Toql : Temporal ontology querying language. In Nikos Mamoulis, Thomas Seidl, Torben Bach Pedersen, Kristian Torp, and Ira Assent, editors, *Advances in Spatial and Temporal Databases*, volume 5644 of *Lecture Notes in Computer Science*, pages 338–354. Springer, 2009.
- [Barrasa *et al.*,] J. Barrasa, O. Corcho, and A. Gómez-pérez. Fund finder : A case study of database-to-ontology mapping. In *Proceedings of the Semantic Integration Workshop (ISWC 2003)*. Online at <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/>, volume 82. Citeseer.
- [Baziz *et al.*, 2005] Mustapha Baziz, Mohand Boughanem, Nathalie Aussenac-Gilles, and Claude Chrisment. Semantic cores for representing documents in ir. In *Proceedings of the 2005 ACM symposium on Applied computing, SAC '05*, pages 1011–1017, New York, NY, USA, 2005. ACM.
- [Bechhofer *et al.*, 2001] Sean Bechhofer, Ian Horrocks, Carole A. Goble, and Robert Stevens. Oiled : A reason-able ontology editor for the semantic web. In Franz Baader, Gerhard Brewka, and Thomas Eiter, editors, *KI/GAI*, volume 2174 of *Lecture Notes in Computer Science*, pages 396–408. Springer, 2001.
- [Bechhofer *et al.*, 2003] Sean Bechhofer, Raphael Volz, and Phillip W. Lord. Cooking the semantic web with the owl api. In *International Semantic Web Conference*, pages 659–675, 2003.
- [Bédard *et al.*, 2004] Yvan Bédard, Suzie Larrivée, Marie-Josée Proulx, and Martin Nadeau. Modeling geospatial databases with plug-ins for visual languages : A pragmatic approach and the impacts of 16 years of research and experimentations on perceptory. In *ER (Workshops)*, pages 17–30, 2004.
- [Beddoe *et al.*, 1999] D. Beddoe, P. Cotton, R. Uleman, S. Johnson, and J. R. Herring. OpenGIS Simple Features Specification For SQL. Technical report, OGC, May 1999.
- [Bellatreche *et al.*, 2004] Ladjel Bellatreche, Guy Pierra, Dung Nguyen Xuan, and Dehainsala Hondjack. Intégration de sources de données autonomes par articulation a priori d'ontologies. In *INFORSID*, pages 283–298, 2004.

-
- [Benthem, 1995] J. van Benthem. *Temporal Logic*, pages 241–350. Oxford University Press, 1995.
- [Bessire *et al.*,] C. Bessire, J. Euzenat, R. Jeansoulin, G. Ligozat, and S. Schwer. Raisonnement spatial et temporel. In *Actes 6e journées nationales PRC-GDR intelligence artificielle, Grenoble, France*, page 77.
- [Bittner *et al.*, 2004] Thomas Bittner, Maureen Donnelly, and Barry Smith. Endurants and perdurants in directly depicting ontologies. *AI Commun.*, 17(4) :247–258, 2004.
- [Bittner *et al.*, 2009] Thomas Bittner, Maureen Donnelly, and Barry Smith. A spatio-temporal ontology for geographic information integration. *International Journal of Geographical Information Science*, 23(6) :765–798, 2009.
- [Bizer, 2003] Chris Bizer. D2r map - a database to rdf mapping language. In *12th International WorldWide Web Conference (WWW 2003)*, 2003.
- [Bizer, 2004a] Christian Bizer. D2rq - treating non-rdf databases as virtual rdf graphs. In *In Proceedings of the 3rd International Semantic Web Conference (ISWC2004)*, 2004.
- [Bizer, 2004b] Christian Bizer. D2rq - treating non-rdf databases as virtual rdf graphs. In *Proceedings of the 3rd International Semantic Web Conference (ISWC2004)*, 2004.
- [Bradley, 1998] Neil Bradley. *The XML Companion*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1998.
- [Bry *et al.*, 2005] François Bry, Christoph Koch, Tim Furche, Sebastian Schaffert, Liviu Badea, and Sacha Berger. Querying the web reconsidered : Design principles for versatile web query languages. *Int. Journal of Semantic Web and Information Systems*, 1(2) :1–21, 2005.
- [Byrne, 2008] Kate Byrne. Having triplets holding cultural data as RDF. In *Proceedings of the ECDL 2008 Workshop on Information Access to Cultural Heritage, Aarhus, Denmark, September 18, 2008*, September 2008.
- [Carroll *et al.*, 2004] Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena : implementing the semantic web recommendations. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, WWW Alt. '04*, pages 74–83, New York, NY, USA, 2004. ACM.
- [Charlet, 2002] Jean Charlet. *Ingénierie des connaissances (Développements, résultats et perspectives pour la gestion des connaissances médicales)*. Mémoire d’habilitation à diriger des recherches, Université Pierre et Marie Curie, D 2002.
- [Chomicki *et al.*, 2003] Jan Chomicki, Ron van der Meyden, and Gunter Saake, editors. *Logics for Emerging Applications of Databases [outcome of a Dagstuhl seminar]*. Springer, 2003.
- [Christian Bizer, 2007] Richard Cyganiak Christian Bizer. D2rq lessons learned. W3C Workshop on RDF Access to Relational Databases, 2007.
- [Das and Srinivasan, 2009] Souripriya Das and Jagannathan Srinivasan. Database technologies for rdf. In *Reasoning Web*, pages 205–221, 2009.
- [de Laborda *et al.*, 2005] Cristian Perez de Laborda, Stefan Conrad, Cristian Prez, and Laborda Stefan Conrad. Relational.owl - a data and schema representation format based on owl, 2005.

- [Dieker and Gting, 2000] Stefan Dieker and Ralf Hartmut Gting. Plug and play with query algebras : Secondo – a generic dbms development environment. pages 380–392, 2000.
- [Dodge *et al.*, 2008] Somayeh Dodge, Robert Weibel, and Anna-Katharina Lautenschütz. Towards a taxonomy of movement patterns. *Information Visualization*, 7 :240–252, June 2008.
- [Domingue, 1998] J. Domingue. Tadzebao and webonto : Discussing, browsing, editing ontologies on the web. In *11th Knowledge Acquisition for Knowledge-Based Systems Workshop*, 1998.
- [Duerst and Suignard, 2005] M. Duerst and M. Suignard. Internationalized Resource Identifiers (IRIs). RFC 3987, 1 2005.
- [Egenhofer, 1991] Max J. Egenhofer. Reasoning about binary topological relations. In *Proceedings of the Second International Symposium on Advances in Spatial Databases, SSD’91*, pages 143–160, London, UK, UK, 1991. Springer-Verlag.
- [Erling and Mikhailov, 2009] Orri Erling and Ivan Mikhailov. Rdf support in the virtuoso dbms. In *Networked Knowledge - Networked Media*, volume 221, pages 7–24. 2009.
- [Erwig *et al.*, 1999] Martin Erwig, Markus Schneider, Praktische Informatik Iv, and Praktische Informatik Iv. Spatio-temporal data types : an approach to modeling and querying moving objects in databases. *GeoInformatica*, 3 :269–296, 1999.
- [Fallside, 2001] D. C. Fallside. *XML Schema Part 0 : Primer*. 2001. W3C Recommendation.
- [Farquhar *et al.*, 1997] A. Farquhar, R. Fikes, and J. Rice. The ontolingua server : a tool for collaborative ontology construction. *International Journal of Human-Computer Studies*, pages 707–727, 1997.
- [Faucher *et al.*, 2008] C. Faucher, F. Bertrand, and J.Y. Lafaye. Gnration d’ontologie partir d’un modle mtier uml annot. *Numro spcial de la Revue des Nouvelles Technologies de l’Information (RNTI) sur la Modlisation des Connaissances, Cpadus*, 2008.
- [Fensel *et al.*, 2000] Dieter Fensel, Ian Horrocks, Frank van Harmelen, Stefan Decker, Michael Erdmann, and Michel C. A. Klein. Oil in a nutshell. In *EKAW*, volume 1937 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2000.
- [Forlizzi *et al.*, 2000] Luca Forlizzi, Ralf Hartmut Gting, Enrico Nardelli, and Markus Schneider. A data model and data structures for moving objects databases. *SIGMOD Rec.*, 29(2) :319–330, 2000.
- [Frank van Harmelen and Peter F. Patel-Schneider and Ian Horrocks, 2001] Reference description of the daml+oil (march 2001) ontology markup language, March 2001. <http://www.daml.org/2001/03/reference.html>.
- [Frank, 2003] Andrew U. Frank. Ontology for spatio-temporal databases. In *Spatio-Temporal Databases : The CHOROCHRONOS Approach*, pages 9–77, 2003.
- [Fürst, 2004] F. Fürst. *Contribution à l’ingénierie des ontologies : une méthode et un outil d’opérationnalisation*. Thse de doctorat, Universit de Nantes, 2004.

-
- [Geerts, 2004] Floris Geerts. Moving objects and their equations of motion. In *In CDB*, pages 41–52, 2004.
- [Genesereth *et al.*, 1992] R. Genesereth, R.E. Fikes, and Stanford University Computer Science Department. *Knowledge interchange format - version 3.0 : reference manual*. Technical Report Logic. Computer Science Department, Stanford University, 1992.
- [Goranko *et al.*, 2003] Valentin Goranko, Angelo Montanari, and Guido Sciavicco. Propositional interval neighborhood temporal logics. *Journal of Universal Computer Science*, 9(9) :1137–1167, 2003.
- [Green *et al.*, 2008] Jennifer Green, Glen Hart, Catherine Dolbear, Paula C. Engelbrecht, and John Goodwin. Creating a semantic integration system using spatial data. In *International Semantic Web Conference (Posters and Demos)*, volume 401. CEUR-WS.org, 2008.
- [Grenon and Smith, 2004a] Pierre Grenon and Barry Smith. Snap and span : Towards dynamic spatial ontology. *Spatial Cognition and Computation*, 4(1) :69 – 104, 2004.
- [Grenon and Smith, 2004b] Pierre Grenon and Barry Smith. Snap and span : Towards dynamic spatial ontology. *Spatial Cognition & Computation*, 4(1) :69–104, 2004.
- [Gruber, 1992] Thomas R. Gruber. Ontolingua : A mechanism to support portable ontologies. Technical report, 1992.
- [Gruber, 1993] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2) :199–220, June 1993.
- [Grütter and Bauer-Messmer, 2007a] Rolf Grütter and Bettina Bauer-Messmer. Combining owl with rcc for spatioterminological reasoning on environmental data. In *OWLED*, volume 258 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
- [Grütter and Bauer-Messmer, 2007b] Rolf Grütter and Bettina Bauer-Messmer. Towards spatial reasoning in the semantic web : A hybrid knowledge representation system architecture. In *The European Information Society, Lecture Notes in Geoinformation and Cartography*, pages 349–364. Springer Berlin Heidelberg, 2007.
- [Grütter *et al.*, 2008] Rolf Grütter, Thomas Scharrenbach, and Bettina Bauer-Messmer. Improving an rcc-derived geospatial approximation by owl axioms. In *Proceedings of the 7th International Conference on The Semantic Web, ISWC '08*, pages 293–306, Berlin, Heidelberg, 2008. Springer-Verlag.
- [Guarino, 1998] Nicola Guarino. Formal ontology and information systems, 1998.
- [Gutierrez *et al.*, 2005] Claudio Gutierrez, Carlos Hurtado, and Ro Vaisman. Temporal rdf. In *In European Conference on the Semantic Web (ECSW05) (Best paper award)*, pages 93–107, 2005.
- [Gutierrez *et al.*, 2007a] C. Gutierrez, C.A. Hurtado, and A. Vaisman. Introducing time into RDF. *IEEE Transactions on Knowledge and Data Engineering*, 19(2) :207–218, 2007.
- [Gutierrez *et al.*, 2007b] Claudio Gutierrez, Carlos A. Hurtado, and Alejandro A. Vaisman. Introducing time into rdf. *IEEE Trans. Knowl. Data Eng.*, 19(2) :207–218, 2007.

- [Güting *et al.*, 2000] Ralf Hartmut Güting, Michael H. Böhlen, Martin Erwig, Christian S. Jensen, Nikos A. Lorentzos, Markus Schneider, and Michalis Vazirgiannis. A foundation for representing and querying moving objects. *ACM Transactions on Database Systems*, 25(1) :1–42, March 2000.
- [Güting, 2007] Ralf Hartmut Güting. How to build your own moving objects database system. In Christian Becker, Christian S. Jensen, Jianwen Su, and Daniela Nicklas, editors, *8th International Conference on Mobile Data Management (MDM 2007)*, pages 1–2. IEEE, 2007.
- [Haarslev and Möller, 1999] Volker Haarslev and Ralf Möller. Race system description. In *Description Logics*, 1999.
- [Haarslev *et al.*, 1998] Volker Haarslev, Carsten Lutz, and Ralf Möller. Foundations of spatioterminological reasoning with description logics. In *Proceedings of Sixth International Conference on Principles of Knowledge Representation and Reasoning*, pages 112–123. Morgan Kaufmann, 1998.
- [Haas *et al.*, 2005] Laura M. Haas, Mauricio A. Hernández, Howard Ho, Lucian Popa, and Mary Roth. Clio grows up : from research prototype to industrial tool. In *SIGMOD Conference*, pages 805–810, 2005.
- [Halpern and Shoham, 1991] Joseph Y. Halpern and Yoav Shoham. A propositional modal logic of time intervals. *J. ACM*, 38(4) :935–962, 1991.
- [Harth and Decker, 2005] Andreas Harth and Stefan Decker. Optimized Index Structures for Querying RDF from the Web. In *IEEE LA-WEB*, pages 71–80, Buenos Aires, Argentina, 2005.
- [Hert *et al.*, 2011] Matthias Hert, Gerald Reif, and Harald C. Gall. A comparison of rdb-to-rdf mapping languages. In *Proceedings of the 7th International Conference on Semantic Systems, I-Semantics '11*, pages 25–32, New York, NY, USA, 2011.
- [Hewlett-Packard Development Company,] LP (2006) Hewlett-Packard Development Company. Jena - a semantic web framework for java.
- [Hillairet, 2007] Guillaume Hillairet. ATL Use Case - ODM Implementation (Bridging UML and OWL). Technical report, <http://www.eclipse.org/m2m/at1/at1Transformations/>, 2007.
- [Hobbs and Pan, 2004a] Jerry R. Hobbs and Feng Pan. An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing*, 3(1) :66–85, 2004.
- [Hobbs and Pan, 2004b] Jerry R. Hobbs and Feng Pan. An ontology of time for the semantic web. *Acm Transactions On Asian Language Information Processing*, 3(1) :66–85, 2004.
- [Hoekstra *et al.*, 2007] Rinke Hoekstra, Joost Breuker, Marcello Di Bello, and Alexander Boer. The LKIF core ontology of basic legal concepts. In Pompeu Casanovas, Maria Angela Biasiotti, Enrico Francesconi, and Maria Teresa Sagri, editors, *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2007)*, June 2007.

-
- [Hollunder *et al.*, 1990] Bernhard Hollunder, Werner Nutt, and Manfred Schmidt-Schauß. Subsumption algorithms for concept description languages. In *ECAI*, pages 348–353, 1990.
- [Hurtado and Vaisman, 2006] Carlos A. Hurtado and Alejandro A. Vaisman. Reasoning with temporal constraints in rdf. In José Júlio Alferes, James Bailey, Wolfgang May, and Uta Schwertel, editors, *PPSWR*, volume 4187 of *Lecture Notes in Computer Science*, pages 164–178. Springer, 2006.
- [ISO-TC-211, 2002] ISO-TC-211. Geographic information – temporal schema, ISO 19108 :2002, 2002.
- [J. *et al.*, 2000] Charlet J., Zacklad M., Kassel G., and Bourigault D. *Ingénierie des connaissances : recherches et perspectives*, In *Ingénierie des connaissances, évolutions récentes et nouveaux défis*. 2000.
- [Jean *et al.*, 2007] Stéphane Jean, Hondjack Dehainsala, Dung N. Xuan, Guy Pierra, Ladjel Bellatreche, and Yamine Aït-Ameur. OntoDB : It is time to embed your domain ontology in your database. Demo presentation, 2007.
- [Joshua *et al.*, 2007] Lieberman Joshua, Singh Raj, and Goad Chris. W3c geospatial vocabulary. w3c incubator group report 23 october, 2007.
- [Katz and Grau, 2005] Yarden Katz and Bernardo Cuenca Grau. Representing qualitative spatial information in owl-dl. In *Proceedings of the OWL : Experiences and Directions Workshop. Galway*, 2005.
- [Kifer *et al.*, 1995] Michael Kifer, Georg Lausen, and James Wu. Logical foundations of object-oriented and frame-based languages. *J. ACM*, 42(4) :741–843, July 1995.
- [Kim *et al.*, 2008] Sang-Kyun Kim, Mi-Young Song, Chul Kim, Sang-Jun Yea, Hyun Jang, and Kyu-Chul Lee. Temporal ontology language for representing and reasoning interval-based temporal knowledge. pages 31–45. 2008.
- [Konstantinou *et al.*, 2008] N. Konstantinou, D. E. Spanos, and N. Mitrou. Ontology and database mapping : A survey of current implementations and future directions. *Journal of Web Engineering*, 7(1) :1–24, 2008.
- [Koubarakis *et al.*, 2003] Manolis Koubarakis, Timos K. Sellis, Andrew U. Frank, Stéphane Grumbach, Ralf Hartmut Güting, Christian S. Jensen, Nikos A. Lorentzos, Yannis Manolopoulos, Enrico Nardelli, Barbara Pernici, Hans-Jörg Schek, Michel Scholl, Babis Theodoulidis, and Nectaria Tryfona, editors. *Spatio-Temporal Databases : The CHOROCHRONOS Approach*, volume 2520 of *Lecture Notes in Computer Science*. Springer, 2003.
- [Krieger, 2008] Hans-Ulrich Krieger. Where temporal description logics fail : Representing temporally-changing relationships. In Andreas Dengel, Karsten Berns, Thomas M. Breuel, Frank Bomarius, and Thomas Roth-Berghofer, editors, *KI 2008 : Advances in Artificial Intelligence*, volume 5243 of *Lecture Notes in Computer Science*, pages 249–257. Springer, 2008.
- [Lassila and Swick, 1999] Ora Lassila and Ralph R. Swick. Resource description framework (RDF). model and syntax specification. Technical report, W3C, 2 1999.

- [Lema *et al.*, 2003] Jos Antonio Coteló Lema, Luca Forlizzi, Ralf Hartmut Gting, Enrico Nardelli, and Markus Schneider. Algorithms for moving objects databases, 2003.
- [Lopez *et al.*, 1997] Mariano F. Lopez, Asuncion G. Perez, and Natalia Juristo. METHONTOLOGY : from ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium*, pages 33–40, 1997.
- [Louvat *et al.*, 2007] Benoit Louvat, Laurent Bonnaud, Nicolas Marchand, and Gérard Bouvier. Suivi d’objets pour une caméra embarquée dans un drone. In *Actes du 21ème colloque GRETSI sur le Traitement du Signal et des Images, GRETSI 2007*, pages 681–684, Troyes, France, 2007. Département Images et Signal - Département Automatique
Département Images et Signal - Département Automatique.
- [Lutz *et al.*, 1997] C. Lutz, V. Haarslev, and R. Möller. *A Concept Language with Role Forming Predicate Restrictions*. Fachbereich Informatik d. Univ. Hamburg, 1997.
- [Lutz *et al.*, 2008] Carsten Lutz, Frank Wolter, and Michael Zakharyashev. Temporal description logics : A survey. In James Pustejovsky and Peter Revesz, editors, *Proceedings of the 15th International Symposium on Temporal Representation and Reasoning (TIME 2008)*, pages 3–14. IEEE Computer Society, 2008.
- [Manola and Miller, 2004] Frank Manola and Eric Miller. RDF primer. *W3C Recommendation*, 10 :1–107, 2004.
- [Mbaiossoum *et al.*, 2012] Berry Mbaiossoum, Selma Khouri, Ladjel Bellatreche, Stéphane Jean, and Mickaël Baron. Etude comparative des systèmes de bases de données à base ontologiques. In *INFORSID*, pages 379–394, 2012.
- [McGuinness and v. Harmelen, 10 February 2004] D.L. McGuinness and F. v. Harmelen. OWL Web Ontology Language Overview. W3C Recommendation, 10 February 2004. Available at <http://www.w3.org/TR/owl-features/>.
- [McGuinness and v. Harmelen, 2004] D.L. McGuinness and F. v. Harmelen. OWL Web Ontology Language Overview. 2004.
- [Microsystems, 2001] Sun Microsystems. The Upper Cyc Ontology in XTM. Technical report, Sun Microsystems, 2001.
- [Miller, 1995] George A. Miller. Wordnet : A lexical database for english. *Communications of the ACM*, 38 :39–41, 1995.
- [Miller, 1998] G.A. Miller. *Nouns in WordNet*. In : *WordNet. An Electronic Lexical Database.*, pages 23–26. MIT Press, 1998.
- [Moisuc *et al.*, 2012] Bogdan Moisuc, Alina Dia Miron, Marlène Villanova-Oliver, and Jérôme Gensel. Représentation de connaissances spatio-temporelles avec AROM-ST. In Bénédicte Bucher and F. Le Ber, editors, *Développements logiciels en géomatique - innovations et mutualisation*, Information Géographique et Aménagement du Territoire, chapter 4, pages 97–121. Hermès/Lavoisier, 2012.
- [Mokhtar *et al.*, 2002] Hoda Mokhtar, Jianwen Su, and Oscar Ibarra. On moving object queries : (extended abstract). In *PODS '02 : Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 188–198, New York, NY, USA, 2002. ACM.

-
- [Neuhaus *et al.*, 2004] F. Neuhaus, P. Grenon, and B. Smith. A Formal Theory of Substances, Qualities, and Universals. In A. Varzi and L. Vieu, editors, *Formal Ontology in Information Systems (FOIS04)*, pages 49–59. IOS Press, 2004.
- [Niles and Pease, 2001] Ian Niles and Adam Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, pages 2–9, New York, NY, USA, 2001. ACM.
- [Noy *et al.*, 2000] N.F. Noy, R.W. Ferguson, and M.A. Musen. The knowledge model of protg-2000 : Combining interoperability and flexibility. *Lecture Notes in Computer Science*, 1937 :69–82, 2000.
- [Nyulas and Tu, 2007] Csongor Nyulas and Samson Tu. Datamaster a plug-in for importing schemas and data from relational databases into protg. In *In Proceedings of 10 th International Protg Conference*, 2007.
- [O'Connor and Das, 2009] Martin J. O'Connor and Amar K. Das. Sqwrl : A query language for owl. In Rinke Hoekstra and Peter F. Patel-Schneider, editors, *OWLED*, volume 529 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.
- [O'Connor and Das, 2011] Martin J. O'Connor and Amar K. Das. A Method for Representing and Querying Temporal Information in OWL. In Ana Fred, Joaquim Filipe, and Hugo Gamboa, editors, *Biomedical Engineering Systems and Technologies*, volume 127 of *Communications in Computer and Information Science*, chapter 8, pages 97–110. Springer, 2011.
- [Parent *et al.*, 1997] Christine Parent, Stefano Spaccapietra, Esteban Zimanyi, P. Doini, Corinne Plazanet, Christelle Vangenot, N. Rognon, J. Pouliot, and P.-A. Crausaz. MADS : un modèle conceptuel pour des applications spatio-temporelles. *Revue Internationale de Geomatique*, 7(3-4) :317–352, 1997.
- [Parent *et al.*, 2005] Christine Parent, Stefano Spaccapietra, and Esteban Zimányi. The MurMur Project : Modeling and Querying Multi-Representation Spatio-Temporal Databases Information Systems. *Information Systems*, 31(8), 2005.
- [Pelekis *et al.*, 2005] Nikos Pelekis, Nikos Pelekis, Yannis Theodoridis, and Yannis Theodoridis. An oracle data cartridge for moving objects, 2005.
- [Pelekis *et al.*, 2006] Nikos Pelekis, Yannis Theodoridis, Spyros Vosinakis, and Themis Panayiotopoulos. Hermes - a framework for location-based data management. In *In Proceedings of EDBT 2006*, 2006.
- [Peralta *et al.*, 2004] Duarte Nuno Peralta, H. Sofia Pinto, and Nuno J. Mamede. Reusing a time ontology. pages 241–248, September 2004.
- [Percivall, 2003] G. Percivall. OpenGIS reference model, document number OGC 03-040, version 0.1.3. *Open GIS Consortium, Inc.*, 2003.
- [Perry *et al.*, 2006] Matthew Perry, Farshad Hakimpour, and Amit Sheth. Analyzing theme, space and time : An ontology-based approach. In *In : Fourteenth International Symposium on Advances in Geographic Information Systems (ACM-GIS '06); 2006 November 10 - 11*, 2006.
- [Perry *et al.*, 2007] Matthew Perry, Amit P. Sheth, Farshad Hakimpour, and Prateek Jain. Supporting complex thematic, spatial and temporal queries over semantic web

- data. In Frederico T. Fonseca, M. Andrea Rodríguez, and Sergei Levashkin, editors, *GeoS*, volume 4853 of *Lecture Notes in Computer Science*, pages 228–246. Springer, 2007.
- [Perry, 2008] Matthew Perry. *A Framework to Support Spatial, Temporal and Thematic Analytics over Semantic Web Data*. PhD thesis, Wright State University, 2008.
- [Pinto and Martins, 2000] H. Sofia Pinto and J. P. Martins. Reusing ontologies. In *In AAAI 2000 Spring Symposium on Bringing Knowledge to Business Processes*, pages 77–84. AAAI Press, 2000.
- [Pnueli, 1986] A. Pnueli. Application of temporal logic to the specification and verification of reactive systems : A survey of current trends. 224 :510–584, 1986.
- [Prud’hommeaux and Seaborne, 2008] Eric Prud’hommeaux and Andy Seaborne. SPARQL query language for rdf. *W3C Recommendation*, 4 :1–106, 2008.
- [Pugliese *et al.*, 2008] Andrea Pugliese, Octavian Udrea, and V. S. Subrahmanian. Scaling rdf with time. In *WWW ’08 : Proceeding of the 17th international conference on World Wide Web*, pages 605–614. ACM, 2008.
- [Punam and Sudeep, 2007] Bedi Punam and Marwaha Sudeep. Versioning owl ontologies using temporal tags. *World Academy of Science, Engineering, and Technology*, 2007.
- [Randell *et al.*, 1992] D. Randell, Z. Cui, and A. Cohn. A spatial logic based on regions and connections. In *3rd International Conference on Knowledge Representation and Reasoning*, pages 165–176. Morgan Kaufmann, 1992.
- [Sahoo *et al.*, 2008] Satya S. Sahoo, Olivier Bodenreider, Joni L. Rutter, Karen J. Skinner, and Amit P. Sheth. An ontology-driven semantic mashup of gene and biological pathway information : Application to the domain of nicotine dependence. *J. of Biomedical Informatics*, 41(5) :752–765, October 2008.
- [Sahoo *et al.*, 2009] Satya S. Sahoo, Wolfgang Halb, Sebastian Hellmann, Kingsley Idehen, Ted Thibodeau Jr, Sören Auer, Juan Sequeda, and Ahmed Ezzat. A survey of current approaches for mapping of relational databases to rdf. W3C RDB2RDF Incubator Group, 01 2009.
- [Schmiedel, 1990] A. Schmiedel. A temporal terminological logic. In *Proceedings of the 8th National Conference on Artificial Intelligence (AAAI’90)*, pages 640–645, 1990.
- [Shvaiko and Euzenat, 2013] Pavel Shvaiko and Jerome Euzenat. Ontology matching : State of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1) :158–176, 2013.
- [Sirin *et al.*, 2007] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet : A practical OWL-DL reasoner. *Web Semantics*, 5(2) :51–53, 2007.
- [Sistla *et al.*, 1997a] A. Prasad Sistla, Ouri Wolfson, Sam Chamberlain, and Son Dao. Modeling and querying moving objects. In *ICDE*, pages 422–432, 1997.
- [Sistla *et al.*, 1997b] A. Prasad Sistla, Ouri Wolfson, Sam Chamberlain, and Son Dao. Querying the uncertain position of moving objects. In *Temporal Databases, Dagstuhl*, pages 310–337, 1997.

-
- [Snodgrass and Ahn, 1986] Richard T. Snodgrass and Ilsoo Ahn. Temporal databases. *IEEE Computer*, 19(9) :35–42, 1986.
- [Spaccapietra *et al.*, 2008a] Stefano Spaccapietra, Christine Parent, Maria Luisa Damiani, Jose Antonio de Macedo, Fabio Porto, and Christelle Vangenot. A conceptual view on trajectories. *Data & Knowledge Engineering*, 65(1) :126–146, 2008.
- [Spaccapietra *et al.*, 2008b] Stefano Spaccapietra, Christine Parent, Maria Luisa Damiani, José Antônio Fernandes de Macêdo, Fabio Porto, and Christelle Vangenot. A conceptual view on trajectories. *Data Knowl. Eng.*, 65(1) :126–146, 2008.
- [Stocker and Sirin, 2009] Markus Stocker and Evren Sirin. PelletSpatial : A hybrid RCC-8 and RDF/OWL reasoning and query engine. In *Proceedings of the 5th International Workshop on OWL : Experiences and Directions (OWLED)*, 2009.
- [Stolze, 2003] Knut Stolze. Sql/mm spatial - the standard to manage spatial data in a relational database system. In *BTW*, pages 247–264, 2003.
- [Stuckenschmidt *et al.*, 2009] Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, editors. *Modular Ontologies : Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *Lecture Notes in Computer Science*. Springer, 2009.
- [Su *et al.*, 2001] Jianwen Su, Haiyan Xu, and Oscar H. Ibarra. Moving objects : Logical relationships and queries. In *In Proc. 7th Int. Symp. on Spatial and Temporal Databases (SSTD)*, pages 3–19. Springer, 2001.
- [Sure *et al.*, 2002] York Sure, Jrgen Angele, and Steffen Staab. Ontoedit : Guiding ontology development by methodology and inferencing. In Robert Meersman and Zahir Tari, editors, *CoopIS/DOA/ODBASE*, volume 2519 of *Lecture Notes in Computer Science*, pages 1205–1222. Springer, 2002.
- [Tappolet and Bernstein, 2009] J. Tappolet and A. Bernstein. Applied temporal RDF : Efficient temporal querying of RDF data with SPARQL. pages 308–322. Springer-Verlag, 2009.
- [ter Horst, 2005] Herman J. ter Horst. Completeness, decidability and complexity of entailment for RDF schema and a semantic extension involving the OWL vocabulary. *J. Web Sem.*, 3(2-3) :79–115, 2005.
- [Tryfona and Pfoser, 2001] N. Tryfona and D. Pfoser. Designing ontologies for moving objects applications. In *Proc. Intl Workshop on Complex Reasoning on Geographic Data*, 2001.
- [Tryfona and Pfoser, 2005] N. Tryfona and D. Pfoser. Data semantics in location-based services. *J. Data Semantics III*, pages 168–195, 2005.
- [Uschold *et al.*, 1998] M. Uschold, M. Healy, K. Williamson, P. Clark, and S. Woods. Ontology reuse and application, 1998.
- [Vallet *et al.*, 2005] David Vallet, Miriam Fernandez, and Pablo Castells. An Ontology-Based information retrieval model the semantic web : Research and applications. In *The Semantic Web : Research and Applications*, volume 3532 of *Lecture Notes in Computer Science*, pages 103–110. Springer Berlin / Heidelberg, 2005.

- [Vega *et al.*, 2003] Julio Cesar Arprez Vega, scar Corcho, Mariano Fernandez-Lpez, and Asuncin Gmez-Prez. Webode in a nutshell. *AI Magazine*, 24(3) :37–48, 2003.
- [Visser, 2004] Ubbo Visser. *Intelligent Information Integration for the Semantic Web*, volume 3159 of *Lecture Notes in Computer Science*. Springer, 2004.
- [Wannous *et al.*, 2013] Rouaa Wannous, Jamal Malki, Alain Bouju, and Cécile Vincent. Time integration in semantic trajectories using an ontological modelling approach. In Mykola Pechenizkiy and Marek Wojciechowski, editors, *New Trends in Databases and Information Systems*, volume 185 of *Advances in Intelligent Systems and Computing*, pages 187–198. Springer Berlin Heidelberg, 2013.
- [Weiss *et al.*, 2008] Cathrin Weiss, Panagiotis Karras, and Abraham Bernstein. Hexastore : Sextuple Indexing for Semantic Web Data Management. In *VLDB*, Auckland, New Zealand, 2008.
- [Welty and Fikes, 2006] Christopher A. Welty and Richard Fikes. A reusable ontology for fluents in owl. In Brandon Bennett and Christiane Fellbaum, editors, *Formal Ontology in Information Systems : Proceedings of the Fourth International Conference (FOIS 2006)*, volume 150 of *Frontiers in Artificial Intelligence and Applications*, pages 226–236. IOS Press, 2006.
- [Wolfson *et al.*, 1998] Ouri Wolfson, Bo Xu, Sam Chamberlain, and Liqin Jiang. Moving objects databases : Issues and solutions, 1998.
- [Wood *et al.*, 2005] David Wood, Paul Gearon, and Tom Adams. Kowari : A Platform for Semantic Web Storage and Analysis. In *XTech*, 2005.
- [Yan *et al.*, 2001] Ling-Ling Yan, Rene J. Miller, Laura M. Haas, and Ronald Fagin. Data-driven understanding and refinement of schema mappings. In *SIGMOD Conference '01*, pages 485–496, 2001.
- [Yan *et al.*, 2008] Z. Yan, J. Macedo, C. Parent, and S. Spaccapietra. Trajectory Ontologies and Queries. *Transactions in GIS*, 12(s1) :75–91, 2008.
- [Yan, 2011] Zhixian Yan. *Semantic Trajectories : Computing and Understanding Mobility Data*. PhD thesis, Lausanne : EPFL, 2011.