



HAL
open science

Modélisation ontologique des connaissances expertes pour l'analyse de comportements à risque : application à la surveillance maritime

Arnaud Vandecasteele

► To cite this version:

Arnaud Vandecasteele. Modélisation ontologique des connaissances expertes pour l'analyse de comportements à risque : application à la surveillance maritime. Economies et finances. Ecole Nationale Supérieure des Mines de Paris, 2012. Français. NNT : 2012ENMP0077 . pastel-00819259

HAL Id: pastel-00819259

<https://pastel.hal.science/pastel-00819259v1>

Submitted on 30 Apr 2013

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.

Ecole doctorale n° 432 : Sciences et Métiers de l'Ingénieur

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

l'École nationale supérieure des mines de Paris

Spécialité “ Sciences et Génie des Activités à Risques ”

présentée et soutenue publiquement par

Arnaud Vandecasteele

le 30 Octobre 2012

**Modélisation ontologique des connaissances expertes pour
l'analyse de comportements à risque
- Application à la surveillance maritime -**

Directeur de thèse : **Aldo Napoli**

Jury

M. Yvan BEDARD, Professeur, CRG, Université Laval, Québec, Canada
M. Christophe CLARAMUNT, Professeur, IRENAV, École Navale, France
M. Alain BOUJU, Maître de conférence, I3i, Université de la Rochelle, France
M. Bernard ESPINASSE, Professeur, LSIS, Université Paul Cézanne, France
M. Aldo NAPOLI, Chargé de recherche, Mines Paristech, France

Rapporteur
Rapporteur
Examineur
Examineur
Examineur

T
H
È
S
E

« D'incolores idées vertes dorment furieusement »

Noam Chomsky

**« ...En définitive, ces difficultés manifestent l'impossibilité de concevoir
une légende universelle qui soit à la fois indépendante de toutes les cartes
aussi bien que de tous les territoires qui puissent se trouver »**

Bernard Morand

**« Ce n'est qu'en essayant continuellement que l'on finit par réussir.
Autrement dit : plus ça rate, plus on a de chances que ça marche »**

Jacques Rouxel

Remerciements

« *Le paradoxe d'une thèse de doctorat est sans doute d'être un processus d'apprentissage individuel qui ne peut cependant, se concevoir sans une dynamique collective* ». Cette réflexion de Matthieu Noucher résume à elle seule les trois ans passés. En effet, bien que personnelle, une thèse est aussi le fruit d'un travail collaboratif, d'une collaboration aux multiples facettes.

Tout d'abord, celle d'un doctorant à une communauté. Cette communauté qui valide, améliore ou critique la réflexion proposée. Communauté représentée par Yvan Bédard, Christophe Claramunt, Alain Bouju et Bernard Espinasse qui me font l'honneur d'avoir accepté d'être membres de jury et à qui j'adresse mes plus sincères remerciements. Vos commentaires et vos pistes de réflexion seront j'en suis certain une source d'enrichissement supplémentaire.

Mais, cette collaboration est avant tout un duo entre un doctorant et un directeur de thèse. Dans ce rôle, j'ai eu la chance d'avoir à mes côtés Aldo Napoli. J'ai apprécié nos discussions parfois divergentes mais toujours constructives. Tu m'auras enseigné la nécessité de voir toujours plus loin et tu as pleinement participé à l'enrichissement de ma vie de jeune chercheur. Au-delà de cet aspect professionnel, je retiendrai également tes qualités humaines. Merci encore pour toute la confiance et la liberté que tu m'as accordées.

Un doctorant, c'est aussi un laboratoire de recherche. Ces trois années au Centre de Recherche sur les risques et les Crises ont été riches d'enseignement grâce aux différentes personnes que j'ai pu rencontrer. Mes remerciements s'adressent tout particulièrement à Franck Guarnieri, Directeur du laboratoire, qui m'a accepté au sein de son équipe. J'en profite également pour remercier toute l'équipe informatique et administrative qui a grandement participé à ce travail en me déchargeant de préoccupations souvent chronophages. Un grand merci à Sandrine Renaux et Myriam Perrault Lavigne pour leur perpétuelle disponibilité et à Sylvie Michel pour sa capacité à dénicher les ouvrages les plus divers. J'ai également une pensée particulière pour mes différents collègues mais surtout amis. Pour ne pas en oublier, je ne prendrai pas le risque de vouloir les citer; ne m'en voulez pas, vous vous reconnaissez forcément.

Ces quelques 200 pages doivent également beaucoup aux relecteurs qui ont eu la gentillesse d'annoter, corriger et enrichir ce manuscrit. Merci beaucoup à Xavier Chaze, Jonathan Vigneron et surtout à mon père d'y avoir consacré le temps et l'énergie nécessaire.

Enfin, je ne pourrais oublier mes parents, ma famille, ma compagne et mes amis qui de près ou de loin m'ont soutenu et encouragé. Malgré la distance qui nous sépare, je n'ai jamais cessé de penser à vous.

C'est ainsi que s'achève cette thèse, me voilà arrivé au bout de cette aventure mais déjà une autre se dessine. Plus froide, mais tout aussi prometteuse...

SOMMAIRE

Introduction.....	15
<i>Contexte général.....</i>	<i>16</i>
<i>Problématique et contexte scientifique Management et modélisation des connaissances spatiales dans le domaine maritime.....</i>	<i>16</i>
<i>Problématique et contexte applicatif : les systèmes de surveillance maritime de nouvelle génération.....</i>	<i>19</i>
Hypothèse et objectifs de recherche.....	23
<i>Hypothèse</i>	<i>23</i>
<i>objectifs de recherche.....</i>	<i>23</i>
<i>Proposition de recherche et approche adoptée.....</i>	<i>23</i>
Méthodologie & Structure de la thèse	24
<i>Méthodologie.....</i>	<i>24</i>
<i>Structure de la Thèse.....</i>	<i>26</i>
Partie 1 État de l’art.....	29
Chapitre 1. Trajectoires et comportements sémantiques.....	31
1.1. D'un objet mobile à une trajectoire sémantique.....	33
<i>1.1.1. Modélisation conceptuelle des trajectoires.....</i>	<i>34</i>
<i>1.1.2. Enrichissement sémantique des trajectoires spatio-temporelles.....</i>	<i>36</i>
1.2. De la normalité aux anomalies sémantiques.....	40
<i>1.2.1. Détection d'anomalies</i>	<i>40</i>
<i>1.2.2. Création de trajectoires.....</i>	<i>42</i>
<i>1.2.3. Anomalie sémantique</i>	<i>47</i>
1.3. D'une trajectoire anormale à un comportement.....	49
<i>1.3.1. Simple Event Model (SEM).....</i>	<i>49</i>
<i>1.3.2. ISFAR</i>	<i>53</i>
Chapitre 2. Formalisation de la connaissance au sein des ontologies.....	57
2.1. L'ontologie en Intelligence Artificielle.....	59
<i>2.1.1. Origine et définitions.....</i>	<i>59</i>
<i>2.1.2. Les composants d'une ontologie.....</i>	<i>60</i>
<i>2.1.3. Le rôle des ontologies dans les systèmes d'information.....</i>	<i>61</i>
<i>2.1.4. Typologie des ontologies.....</i>	<i>62</i>
2.2. Méthodologies pour la construction d'ontologies.....	64
<i>2.2.1. Principes généraux guidant la construction d'une ontologie.....</i>	<i>64</i>
<i>2.2.2. La méthode On-To-Knowledge</i>	<i>67</i>
<i>2.2.3. La méthode Methontology.....</i>	<i>68</i>

2.3. Formalismes et Langages de représentation d'ontologies.....	70
2.3.1. Formalismes de représentation.....	70
2.3.2. Les langages ontologiques.....	75
2.4. Conception et exploitation d'ontologies.....	84
2.4.1. Éditeur d'ontologies.....	84
2.4.2. Interfaces de programmation pour la manipulation d'ontologies.....	87
2.4.3. Reasonner à partir d'une ontologie	88

Chapitre 3. Représentation et raisonnement spatial dans les ontologies....93

3.1. De la perception de l'espace à sa représentation ontologique.....	95
3.1.1. De la difficulté à représenter l'espace.....	95
3.1.2. Différents niveaux de représentation de l'espace.....	96
3.2. Modélisation des concepts géographiques et spatiaux à l'aide d'une ontologie .	98
3.2.1. Des ontologies géographiques aux ontologies géospatiales.....	98
3.2.2. Domaine d'application des ontologies géospatiales.....	100
3.2.3. Typologie des ontologies géospatiales.....	100
3.2.4. Ontologies géospatiales existantes.....	102
3.2.5. Architecture du Web Sémantique géospatial.....	105
3.3. Raisonnement spatial dans les ontologies.....	109
3.3.1. Le raisonnement spatial qualitatif, le cas de la topologie.....	109
3.3.2. Raisonnement spatial topologique dans les ontologies.....	116

Partie 2 Approche proposée..... 121

Chapitre 4. Une approche ontologique pour l'analyse des comportements de navires..... 123

4.1. Modélisation du contexte sémantique	125
4.1.1. Conceptualisation du modèle.....	125
4.1.2. Ontologie géométrique des trajectoires.....	126
4.1.3. Ontologie géographique	130
4.2. Ontologie du domaine : l'analyse de comportements anormaux de navires....	131
4.2.1. Principales classes de l'ontologie du domaine.....	131
4.2.2. Modélisation des alertes.....	132
4.2.3. Modélisation des scénarios.....	134
4.2.4. Modélisation du contexte maritime	135
4.3. Modélisation des règles d'inférence et enrichissement du SWRL.....	137
4.3.1. Le langage SWRL.....	137
4.3.2. Modélisation des règles classiques.....	137
4.3.3. Modélisation des règles spatiales.....	138

Chapitre 5. Structuration de la plate-forme d'analyse des comportements

anormaux de navires.....	143
5.1. Analyse de comportements dirigé par une ontologie : application à la surveillance maritime	145
5.1.1. <i>Présentation générale du prototype</i>	<i>145</i>
5.1.2. <i>Vue du prototype.....</i>	<i>145</i>
5.2. Module de données.....	147
5.2.1. <i>Présentation des données.....</i>	<i>147</i>
5.2.2. <i>Préparation des données.....</i>	<i>149</i>
5.2.3. <i>Enrichissement des données et création de la trajectoire sémantique.....</i>	<i>152</i>
5.3. Module d'inférence	154
5.3.1. <i>Architecture du module d'inférence.....</i>	<i>154</i>
5.3.2. <i>Processus d'inférence.....</i>	<i>156</i>
5.3.3. <i>Résultats du processus d'inférence</i>	<i>159</i>
5.4. Module de calcul de similarité des comportements anormaux.....	160
5.4.1. <i>Raisonnement à partir de cas.....</i>	<i>160</i>
5.4.2. <i>Architecture du module de calcul de similarité.....</i>	<i>163</i>
5.4.3. <i>intégration de l'ontologie et calcul de similarité.....</i>	<i>165</i>
5.5. Module cartographique.....	168
5.5.1. <i>Architecture du module.....</i>	<i>168</i>
5.5.2. <i>Interface et présentation des comportements.....</i>	<i>168</i>
Chapitre 6. Expérimentations et évaluations.....	171
6.1. Exemples de scénarios type.....	173
6.1.1. <i>Scénario d'un navire dangereux dans une zone réglementée.....</i>	<i>173</i>
6.1.2. <i>Scénario de transbordement de pêche.....</i>	<i>176</i>
6.1.3. <i>Autres scénarios.....</i>	<i>178</i>
6.2. Évaluation du système.....	179
6.2.1. <i>Évaluation des performances du système.....</i>	<i>179</i>
6.2.2. <i>Synthèse des résultats.....</i>	<i>181</i>
6.3. Perspectives d'amélioration.....	182
6.3.1. <i>Amélioration fonctionnelle</i>	<i>182</i>
6.3.2. <i>Amélioration du processus de modélisation des connaissances.....</i>	<i>183</i>
Conclusion.....	189
Conclusion.....	191
<i>Bilan de la thèse.....</i>	<i>191</i>
<i>Perspectives.....</i>	<i>192</i>
Références.....	195

Liste des acronymes

CROSS - Centres Régionaux Opérationnels de Surveillance et de Sauvetage
DPST - Douglas et Peucker Spatio Temporel
GML - Geography Markup Language
OMI - Organisation Maritime Internationale
OWL - Ontology Web Language
OWL-DL - Ontology Web Language Description Logics
RàPC - Raisonnement à partir de Cas
RCC - Region Connection Calculus
RDF - Ressource Description Framework
RDFS - Ressource Description Framework Schema
RIF - Rule Interchange Format
RSQ - Raisonnement Spatial Qualitatif
SEM - Simple Event Model
SGBD - Système de Gestion de Base de Données
SIG - Système d'Information Géographique
SOLAS - Safety Of Life At Sea
SPARQL - SPARQL Protocol and RDF Query Language
SWRL - Semantic Web Rule Language
W3C - World Wide Web Consortium
XML - eXtensible Markup Language

Liste des figures

Figure 1: D'une trajectoire à un comportement sémantique.....	17
Figure 2: D'un modèle à son utilisation opérationnelle.....	18
Figure 3: Un opérateur en charge de la surveillance maritime.....	20
Figure 4: Architecture d'un système de détection d'anomalies et d'interprétation de comportements anormaux.....	22
Figure 5: Les quatre piliers soutenant la réalisation du prototype OntoMap.....	24
Figure 6: Diagramme d'activité de la méthode de recherche.....	25
Figure 7: Plan de la thèse.....	26
Figure 1.1: Couplage AIS et Radar dans le domaine maritime.....	32
Figure 1.2: Dimension spatio-temporelle d'un objet mobile.....	35
Figure 1.3: Processus d'enrichissement sémantique des trajectoires.....	37
Figure 1.4: Représentation ontologique de la notion de trajectoire sémantique.....	37

Figure 1.5: Architecture ontologique pour l'enrichissement sémantique de trajectoires.....	38
Figure 1.6: Cartographie mondiale des navires équipés d'un transpondeur AIS.....	41
Figure 1.7: Illustration de l'utilisation de l'algorithme de Douglas et Peucker spatio-temporel	43
Figure 1.8: Exemple de clusters de trajectoires sur des données maritimes réalisés à l'aide du logiciel M-Atlas.....	44
Figure 1.9: Exemple de couloir spatial à partir de différentes trajectoires.....	45
Figure 1.10: Illustration de la distance de Hausdorff.....	46
Figure 1.11: Taxonomie des anomalies statiques.....	47
Figure 1.12: Taxonomie des anomalies dynamiques.....	48
Figure 1.13: Principales classes du Simple Event Model.....	50
Figure 1.14: Les différentes propriétés du Simple Event Model.....	51
Figure 1.15: Diagramme des flux du système SEM)	52
Figure 1.16: Architecture du prototype ARMAD développé par Roy..	54
Figure 2.1: Les différentes dimensions d'une ontologie.....	58
Figure 2.2: Les cinq étapes du processus On-To-Knowledge.....	68
Figure 2.3: Les différentes phases de Methontology.....	68
Figure 2.4: Architecture du Web Sémantique selon le W3C.....	77
Figure 2.5: Classes et propriétés définies par le RDFS.....	79
Figure 2.6: Implémentation des logiques de description par OWL.....	81
Figure 2.7: Interface du logiciel Protégé.....	85
Figure 2.8: Interface du logiciel TopBraid Composer.....	86
Figure 2.9: Interface du logiciel OntoStudio.....	86
Figure 3.1. Connexions entre le domaine des ontologies et celui des SIG.....	94
Figure 3.2: Les quatre paradigmes de l'information spatiale.....	96
Figure 3.3: Typologie des ontologies selon le niveau de conceptualisation géospatial.....	99
Figure 3.4: Intégration de la dimension spatiale dans l'architecture du Web Sémantique.....	106
Figure 3.5: Représentation schématique du modèle GeoRSS.....	108
Figure 3.6: Représentation du GeoRSS au format OWL.....	108
Figure 3.7: Illustration des 8 relations de base définies par le RCC (RCC8).....	110
Figure 3.8: La décomposition d'une région selon le modèle des 9-Intersections.....	113
Figure 3.9: Matrice du modèle des 9-intersections.....	113
Figure 3.10: Représentation des huit relations topologiques en utilisant la méthode des 9-intersections et leur équivalence dans leur formalisme du RCC8.....	114
Figure 4.1: Les trois principales composantes ontologiques pour l'analyse de comportements anormaux de navires.....	124
Figure 4.2: Modèle ontologique pour la représentation des trajectoires.....	126
Figure 4.3: Représentation graphique de la classe <code>_Geometry</code> de l'ontologie GeoRSS Simple.....	127
Figure 4.4: Représentation graphique de la classe <code>_Geometry</code> de l'ontologie GML.....	128

Figure 4.5: Représentation graphique de la classe TemporalPosition de l'ontologie GML.....	128
Figure 4.6: Les différentes propriétés du pays "France" dans l'ontologie de la FAO.....	130
Figure 4.7: Organisation et principales sous-classes de la classe Alert.....	132
Figure 4.8: Sous-classes de la classe Alert_Spatial.....	132
Figure 4.9: Individus de la classe correspondant à l'analyse de trajectoire.....	133
Figure 4.10: Liste des différentes alertes potentielles liées aux caractéristiques du navire.....	134
Figure 4.11: Sous-classes et instances pour la description des scénarios.....	135
Figure 4.12: Sous-classes de la classe Vessel.....	136
Figure 5.1: Principales étapes nécessaires à l'analyse de comportements de navires.....	144
Figure 5.2: Architecture fonctionnelle du prototype d'analyse de comportements anormaux de navires.....	146
Figure 5.3: Position des navires en méditerranée pour une période de 4 mois.....	148
Figure 5.4: Exemple de graphe de zones pour la région de Gibraltar.....	150
Figure 5.5: Filtre de Douglas et Peucker Spatio-Temporel.....	151
Figure 5.6: Représentation ontologique de la notion de trajectoire sémantique avec ajout de la notion Lost.....	153
Figure 5.7: Étapes et flux du module d'inférence.....	154
Figure 5.8: Principaux composants du raisonneur Pellet.....	156
Figure 5.9: Exemple de fonctions spatiales disponibles et utilisables par le moteur d'inférence.....	157
Figure 5.10: Alerte de type HighSpeed.....	159
Figure 5.11: Alerte spatiale.....	159
Figure 5.12: Alerte maritime.....	159
Figure 5.13: Modèle générique d'un système de raisonnement à partir de cas.....	161
Figure 5.14: Le cycle RàPC re-revisité.	162
Figure 5.15: Étapes et flux du module de calcul de similarité.....	163
Figure 5.16: Architecture de jColibri 2.0.....	164
Figure 5.17: Illustration du calcul de similarité sémantique.....	167
Figure 5.18: Etapes et flux du module cartographique.....	168
Figure 5.19: Interface d'OntoMap.....	169
Figure 6.1: Description du scénario de type attaque de tanker.....	174
Figure 6.2: Liste des propriétés du navire pour le scénario d'attaque.....	175
Figure 6.3: Scénario d'attaque potentielle et indice de criticité.....	175
Figure 6.4: Description du scénario de type transbordement de pêche.....	177
Figure 6.5: Liste des propriétés du navire pour le scénario de transbordement de pêche.....	177
Figure 6.6: Scénario de transbordement potentiel et indice de criticité.....	178
Figure 6.7: Fonctionnement général de D2RQ.....	182
Figure 6.8: Interface du logiciel Snoggle.....	183

Figure 6.9: interface de la plate-forme Cobra.....	184
Figure 6.10: Exemple de représentation de concepts spatiaux à l'aide du logiciel Perceptory.....	185

Liste des tableaux

Tableau 3.1: Représentation des éléments constitutifs d'une trajectoire exprimés en logique de description.....	39
Tableau 1.1: Revue des méthodes d'ingénierie ontologique	66
Tableau 1.2: Exemples de constructeurs et leur correspondance dans les langages.....	74
Tableau 1.3: Analyse comparative des principaux moteurs d'inférence pour les logiques de description.....	89
Tableau 2.1: Définition axiomatique des relations topologiques du RCC.....	111
Tableau 2.2: Tableau de composition pour les relations de l'algèbre RCC-8.....	112
Tableau 2.3: Résumé des relations topologiques pour les modèles 9IM et DE+9IM et pour toutes les dimensions entre objets simples.....	115
Tableau 2.4: Exemple de relation entre deux objets géographiques et leur traduction dans le modèle des 9 Intersections.....	115
Tableau 2.5: Traduction des 8 relations du formalisme RCC8 en OWL DL	117
Tableau 3.1: Représentation des éléments constitutifs d'une trajectoire et ajout du concept Lost..	129

Liste des graphiques

Graphique 2.1: Typologie selon le niveau de conceptualisation.....	63
Graphique 2.2: Exemple simplifié d'un réseau sémantique représentant la phrase "Un pétrolier est un navire transportant du pétrole".....	71
Graphique 2.3: Exemple simplifié d'un schéma portant sur la définition d'un Pétrolier.....	71
Graphique 2.4: Exemple simplifié d'un graphe conceptuel portant sur la définition d'un Pétrolier..	72
Graphique 2.5: Architecture d'une base de connaissances basée sur les logiques de description.....	73
Graphique 2.6: Exemple d'assertion RDF.....	78
Graphique 2.7: Exemple de schéma RDF.....	78
Graphique 2.8: Les familles de langages qui ont conduit au OWL.....	80
Graphique 2.9: Structure d'une règle SWRL.....	82
Graphique 3.1: Architecture hybride d'une base de connaissances incluant des raisonnements spatiaux basés sur le formalisme RCC8.....	118
Graphique 6.1: Répartition du temps de calcul pour la phase de mise à jour.....	179
Graphique 6.2: Répartition du temps de calcul pour la phase d'inférence.....	180
Graphique 6.3: Répartition du temps de calcul pour la phase de RàPC.....	181

INTRODUCTION

CONTEXTE GÉNÉRAL

Le contexte général dans lequel s'inscrit cette thèse puise sa source à la fois dans le domaine du raisonnement spatial et dans celui des ontologies spatiales. Ce choix s'explique notamment par le constat réalisé dans le cadre des projets dans lesquels nous sommes impliqués. Ces projets, qui ont pour objectif de définir l'architecture et les besoins des systèmes de surveillance maritimes de nouvelle génération, ont nécessité l'intégration d'experts du domaine (Jangal et al., 2008, 2009; Morel and Claisse, 2010a, 2010b). Mais, lors des différentes expérimentations réalisées il a été observé une certaine difficulté pour ces experts à manipuler les concepts et prédicats spatiaux ainsi qu'à conceptualiser et partager une vision commune des objets géographiques.

Illustrons cela par un exemple concret. Pour définir un comportement suspect, ces experts utilisent des heuristiques acquises pour la plupart de par leur expérience. Ces heuristiques s'appuient sur des faits comme « *le navire est proche d'une côte* ». Mais, si pour l'expert en charge de la surveillance maritime ce fait est déduit naturellement, sa transcription dans un système informatique est problématique et soulève de nombreuses questions. En effet, de quel type de navires s'agit-il ? Ce fait est-il spécifique à tous les navires ou à une seule catégorie ? Qu'entend-t-on par proche ?

Répondre à ces différentes questions nécessite de disposer d'un formalisme de représentation capable à la fois de traduire la connaissance experte de façon formelle et d'effectuer des raisonnements spatiaux. C'est pourquoi l'idée de ce travail de recherche vise à explorer les potentialités des ontologies spatiales pour l'analyse des comportements anormaux de navires.

PROBLÉMATIQUE ET CONTEXTE SCIENTIFIQUE MANAGEMENT ET MODÉLISATION DES CONNAISSANCES SPATIALES DANS LE DOMAINE MARITIME

Le cœur de ce travail de thèse s'articule autour de la notion de trajectoires sémantiques. Définie par Spaccapietra (Spaccapietra et al., 2008), cette notion consiste à décomposer les données brutes en une série d'éléments (*stop* et *move*). Ce découpage permet alors d'obtenir une modélisation fine des trajectoires qu'il sera ensuite possible d'enrichir sémantiquement.

À partir de ces trajectoires sémantiques, des informations provenant de différents domaines peuvent être ajoutées afin de faciliter l'analyse. Quel que soit le domaine étudié, il est possible de regrouper ces informations sous la forme de trois principales ontologies (Yan, 2011). La première est une ontologie géométrique des trajectoires. Elle offre les composants spatio-temporels nécessaires à la modélisation de la trajectoire d'un objet mobile. La seconde, l'ontologie géographique, contient les différents concepts géographiques portant sur la description du territoire. Enfin, la dernière est spécifique au domaine d'application représenté et permet d'ajouter des éléments informationnels en fonction d'une thématique et de besoins précis.

La problématique de cette thèse repose alors sur l'interprétation d'une trajectoire pour en déduire un comportement. Il s'agit alors à partir de données brutes, d'obtenir dans un premier temps des anomalies puis des comportements sémantiques (Figure 1). Pour y parvenir, le domaine des ontologies et plus particulièrement des ontologies spatiales a été étudié.

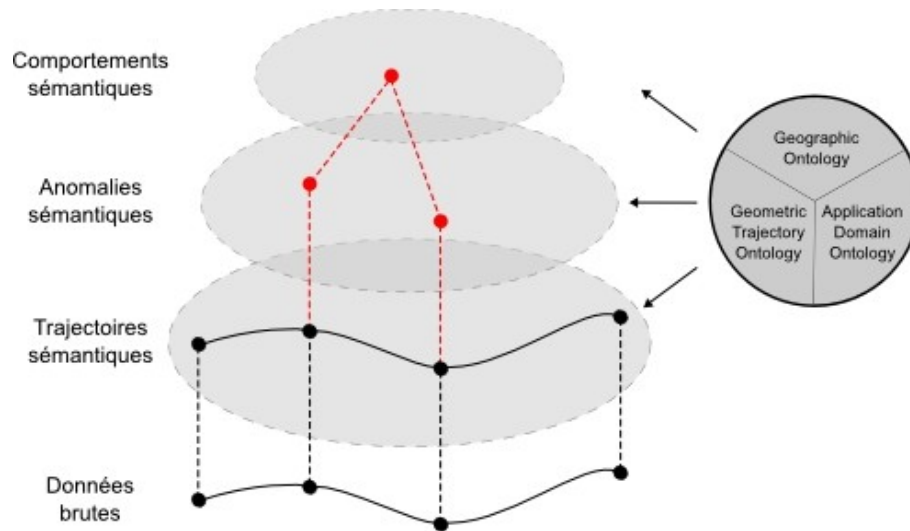


Figure 1: D'une trajectoire à un comportement sémantique

Tout d'abord concept philosophique qui se définit comme l'étude de l'être en tant que être, c'est dans le domaine de l'Intelligence Artificielle que les ontologies sont aujourd'hui utilisées. En effet, face aux limites identifiées lors de la création de systèmes experts, les ontologies apparaissent comme un paradigme clé pour la manipulation de la connaissance. Dans cette vision, l'information n'est plus vue comme une entité unique mais associée par des liens sémantiques à tout un réseau qui permet de définir et de préciser le sens de celle-ci (Varet-Pietri, 2000; Guizzardi, 2007). Il s'agit alors de trouver le modèle le mieux adapté et le plus représentatif de la situation observée. Ainsi, l'objet ontologique est en fait une composition entre un modèle, une conceptualisation, un langage de modélisation et enfin une spécification (Figure 2).

Par leurs caractéristiques, les ontologies permettent d'aboutir à une interopérabilité sémantique aussi bien entre individus, entre systèmes ou entre systèmes et individus (Hepp, 2008). Pour cela, elles offrent les éléments nécessaires à la modélisation des connaissances. Ces éléments sont composés de classes et d'instances pouvant être reliées par des liens, définis ou précisés par des contraintes et enrichis par des attributs. Afin de représenter cette modélisation, différents formalismes et langages ont été proposés. Historiquement, on assiste tout d'abord dans les années 90 à une multiplication des langages traditionnels basés sur le domaine de l'intelligence artificielle. Mais, l'avènement du Web a favorisé la création de nouveaux langages standardisés dont les plus récents sont notamment RDF et OWL (Bechhofer, 2002). Conçus pour être le plus générique et le plus expressif possible, ces langages permettent de répondre à la majorité des besoins imposés par les différents domaines d'application (médecine, recherche d'informations, etc.).

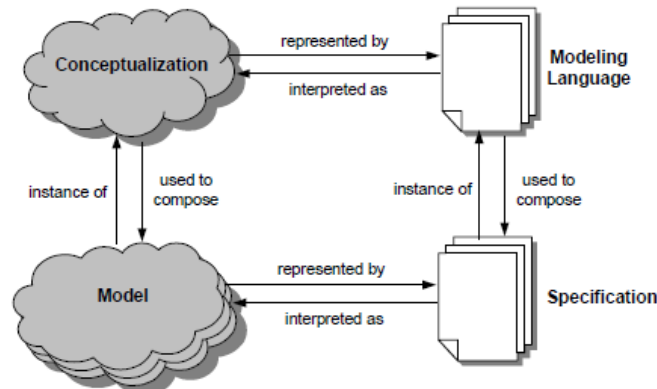


Figure 2: D'un modèle à son utilisation opérationnelle. Source (Guizzardi, 2007)

Dans ce contexte, l'information spatiale occupe une place particulière. En effet, bien que celle-ci ait été considérablement popularisée par l'essor des technologies mobiles (GPS, lecteur RFID, etc.) son intégration dans les ontologies fait toujours l'objet de débat et de travaux de recherches. Cela s'explique notamment par les différents défis tant scientifiques que technologiques qui doivent être auparavant relevés. Parmi ces défis on trouve notamment la standardisation de formalismes de représentation spécifiques à l'information spatiale ainsi que le développement d'agents logiciels permettant d'exploiter celle-ci (Kuhn, 2005; Dia Miron, 2009; Ashish and Sheth, 2011) (Kuhn, 2005; Dia Miron, 2009; Ashish and Sheth, 2011). Dans le cadre de ce travail de recherche, la prise en compte de cette dimension est particulièrement importante. En effet, l'analyse du comportement d'un navire fait nécessairement intervenir des données géographiques. Par exemple, il est nécessaire de pouvoir interroger la base de connaissance à partir de prédicats spatiaux comme :

- Le navire est-il situé dans une zone réglementée ?
- Les deux navires ont-ils une trajectoire parallèle ?
- Quels sont les navires proches d'un pays connu comme étant une plate-forme pour le trafic de drogue ?

Afin de pallier à cette limite, différentes initiatives institutionnelles et travaux scientifiques ont été menés visant à définir les caractéristiques nécessaires à la description des éléments spatiaux dans les ontologies. Dans les années 2000, la communauté des SIG a très largement investi le champ des ontologies. Cela a notamment permis de réaliser une typologie des besoins et une première distinction entre ontologies géographiques et ontologies spatiales (Spaccapietra et al., 2004). Mais pour autant, les différents travaux réalisés se sont principalement appuyés sur les potentialités de ces deux domaines sans pour autant former une véritable discipline commune (Agarwal, 2005). Cela explique notamment la raison de l'absence de types ou de standards spatiaux dans les ontologies. Plus récemment, le Geospatial incubator Group du W3C a publié une série de recommandations qui sont à ce jour les implémentations les plus abouties. Ces recommandations ont établi une catégorisation des différentes ontologies (ontologie des relations spatiales, des noms de lieux, etc.) ainsi qu'un format de représentation le GeORSS Simple (Lieberman et al., 2007a, 2007b). Néanmoins, comme le souligne Dia Miron (Dia Miron, 2009), il n'existe pour le moment aucun consensus concernant un standard de

représentation des caractéristiques et des relations sémantiques spatiales dans les ontologies.

Afin de répondre aux limites identifiées tant au niveau des SIG qu'au niveau des ontologies, Fonseca et Egenhofer (Fonseca and Egenhofer, 1999) ont proposé la création d'un Système d'Information Géographique Dirigé par une Ontologie (ODGIS¹). Dans ce type de système, les ontologies structurent les différents composants et les processus. Ce travail de recherche s'inscrit dans une logique similaire et vise à évaluer les potentialités des ontologies spatiales pour l'analyse des comportements de navires. Pour cela, il est nécessaire de fournir les structures sémantiques appropriées à la description d'une trajectoire, d'une situation et d'un comportement. Ces différents éléments s'appuient sur une composante géographique qu'il faudra également intégrer afin de permettre la formalisation et l'intégration de raisonnements spatiaux.

Puisant son inspiration dans le domaine du raisonnement spatial et celui des ontologies, cette thèse vise à analyser les comportements anormaux d'objets mobiles à partir de différents éléments contextuels. Du fait des activités de recherche dans lesquelles nous sommes impliqués, le domaine maritime a été choisi comme terrain d'expérimentation. Ainsi, ces travaux s'attachent tout d'abord à définir un modèle conceptuel pour l'interprétation des comportements des navires et d'autre part à utiliser ce modèle dans un système d'analyse dirigé par une ontologie.

PROBLÉMATIQUE ET CONTEXTE APPLICATIF : LES SYSTÈMES DE SURVEILLANCE MARITIME DE NOUVELLE GÉNÉRATION

SYSTÈMES DE SURVEILLANCE MARITIME DE NOUVELLE GÉNÉRATION

Chaque jour, plusieurs dizaines de milliers de navires transitent par les océans. Dans ce flot continu et en raison des limites des systèmes de surveillance actuels, certains d'entre eux profitent de cette situation pour exercer des activités illégales (trafic de drogues ou de clandestins, pêche illégale, dégazage, etc.). A titre d'exemple, il est estimé que 70% de la cocaïne est transportée par voie maritime et 82% des réserves halieutiques en Méditerranée sont surexploitées (UNCD, 2011; UNODC, 2011).

Face à ces différentes menaces, les experts du domaine disposent actuellement d'outils spécifiques à la gestion et à l'analyse du trafic maritime comme SPATIONAV pour la France, SIVE pour l'Espagne ou encore MEVAT pour la Finlande. À cela s'ajoutent également des bases de données spécialisées à l'exemple de Equasis ou Traffic 2000 et des réseaux d'informations dédiés comme SafeSeaNet pour le sauvetage en mer.

Si ces outils apportent aux opérationnels une aide certaine, plusieurs limites peuvent également être notées. Celles-ci portent principalement sur les capacités d'analyse, la portée de la zone de surveillance et les interfaces utilisées. À l'heure actuelle les fonctionnalités offertes se résument à la visualisation d'une tenue de situation simple du trafic maritime dans la zone des eaux territoriales (12 à 20 milles). Cette surveillance s'effectue à l'aide d'un réseau de radars déployé le long des côtes dans des sémaphores, mais aussi grâce aux informations transmises par les dispositifs d'identification automatique (AIS²). Concernant les capacités d'analyse, celles-ci se limitent très souvent à la gestion du

1 Ontology Driven Geographic Information System

2 Automatic Identification System, installés sur les bateaux de plus de 300 tonnes, les bateaux de pêche de plus de 15

Introduction

trafic maritime et ne proposent pas de fonctionnalités avancées permettant de comprendre, explorer et visualiser efficacement les informations disponibles (Kristiansen, 2004; Claramunt et al., 2007; Kotis et al., 2009).

Plus concrètement, illustrons ces limites par la description de l'activité de l'expert en charge de la surveillance du trafic maritime. Celui-ci doit obtenir une tenue de situation la plus exhaustive possible à partir d'un écran radar sur lequel sont positionnés des milliers de pistes de navires (Figure 3). Dans cette configuration, le nombre important de données à analyser, les capacités réduites d'exploration offertes ainsi que le délai d'analyse de l'alerte sont autant de facteurs pouvant conduire à une saturation cognitive de l'individu (IALA, 2010; Etienne, 2011).

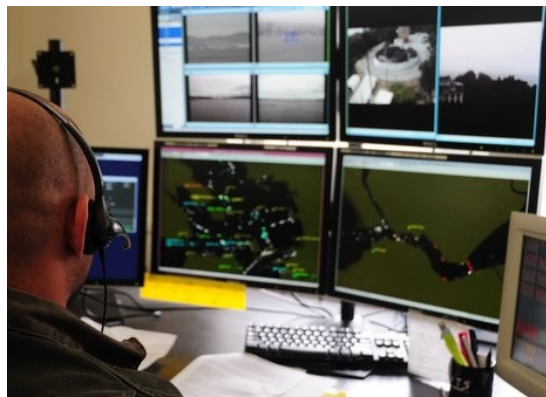


Figure 3: Un opérateur en charge de la surveillance maritime

De ce fait, il est nécessaire de disposer de nouveaux systèmes de surveillances permettant une approche globale de la surveillance maritime. Déjà initié par les États-Unis au travers de la notion de Maritime Domain Awareness (MDA)³, cette approche se définit comme la capacité à disposer de toutes les informations nécessaires au contrôle efficace des mers et des côtes (MDA Report, 2005).

L'un des projets précurseurs dans ce domaine est PANDA⁴ réalisé par le Département de la Défense des États-Unis (DARPA⁵). L'objectif de PANDA est la détection automatique de comportements anormaux de navires. Mais, aucune information à propos des méthodologies utilisées ou des technologies mises en œuvre n'a été divulguée.

En Europe, citons deux autres initiatives. Tout d'abord le projet SECMAR⁶ mené par Thales Underwater System dont le but est la sécurisation des ports. Pour cela, le système proposé étudie les comportements de navires à partir de différentes méthodes statistiques comme les modèles de Markov

mètres et les bateaux transportant des passagers

3 Maritime Domain Awareness (MDA) is the effective understanding of anything associated with the global maritime domain that could impact the security, safety, economy or environment of the United States

4 Predictive Analysis for Naval Deployment Activities

5 Defense Advanced Research Projects Agency

6 SECurity system to protect people, goods and facilities located in critical MARitime area

cachés. La seconde initiative est une suite de projets (SCANMARIS⁷, TAMARIS⁸, SISMARIS⁹ et I2C¹⁰) menée par un consortium composé d'industriels (DCNS, Furuno, etc.), de centres de recherche (Mines Paristech, IRIT, etc.) et de PME-PMI (Intuilab, Sofresud, etc.) (Jangal et al., 2008, 2009; Morel and Claisse, 2010a, 2010b). L'objectif de ces projets est de proposer un système complet allant de l'acquisition des données jusqu'au traitement des menaces en haute mer. Pour cela, de nouveaux moyens de surveillance ont été déployés comme le radar à onde de surface permettant d'étendre la zone de surveillance, des algorithmes de détection de comportements anormaux et des solutions collaboratives.

PROCESSUS D'ANALYSE DE COMPORTEMENTS DANS LE DOMAINE MARITIME

La détection d'anomalies dans le domaine maritime fait intervenir différents outils, techniques et méthodes. C'est pourquoi, Martineau et Roy identifient sept principales étapes allant de l'acquisition des données jusqu'à l'évaluation de la menace et sa diffusion vers les autorités compétentes (Roy, 2008; Martineau and Roy, 2011). Le découpage proposé offre l'avantage d'être suffisamment générique pour s'adapter à d'autres domaines tout en restant en accord avec la littérature du domaine. Dans le cadre de nos travaux, une architecture similaire (Figure 4.) mais divisée en cinq niveaux a été proposée (Vandecasteele and Napoli, 2012a). Les différences se font surtout au sommet de la pyramide ou certaines catégories ont été fusionnées mais la logique reste la même. En résumé, les sept étapes nécessaires à la détection de comportements anormaux sont :

1. **Acquisition de données** : cette étape se situe à la base de la pyramide de détection des anomalies. En effet, la détection d'anomalies dérive directement de l'analyse des données. De ce fait, si celles-ci sont erronées ou insuffisantes les résultats obtenus le seront également. C'est pourquoi, de nouveaux capteurs sont élaborés afin d'assurer une surveillance maritime quel que soit les conditions météorologiques, l'heure (jour/nuit) ou le comportement coopératif ou non coopératif du navire. Pour cela en plus des traditionnelles stations de réception AIS et des radars conventionnels, des capteurs fixes et mobiles sont déployés. Ces derniers visent à améliorer les capacités d'acquisition en augmentant la zone de surveillance ou la détection de petites embarcations. Dès cette étape et en fonction des données obtenues, il est possible de détecter des anomalies. Néanmoins, ce sont ici essentiellement des anomalies individuelles comme l'arrêt d'émission d'un transpondeur AIS ou encore une vitesse excessive (Ristic et al., 2008). Ces anomalies sont particulièrement simples et ne demandent pas d'algorithmes spécifiques.
2. **Fusion des données** : en raison de l'hétérogénéité des capteurs utilisés, une étape de corrélation et d'agrégation spatio-temporelle des données est nécessaire. Celle-ci permettra d'obtenir une position affinée et de réduire les incertitudes tout en réduisant le volume de données à traiter et en améliorant la qualité. Ce processus est lui même composé de différentes étapes qui varient selon les modèles adoptés. L'un des plus connus est notamment le JDL Model¹¹ qui définit six niveaux de raffinement. Dans le domaine maritime ce modèle a notamment été utilisé dans les travaux menés par kokar (Kokar et al., 2009). Là encore, il est possible de détecter certaines

7 Surveillance et Contrôle des Activités des Navires en mer – MARIS

8 Traitement et authentification des Menaces et des RISques en mer

9 Système d'Information et de Surveillance MARitime pour l'Identification des comportements Suspects de navires

10 Integrated System for Interoperable sensors & Information sources for Common abnormal vessel behaviour detection & Collaborative identification of threat

11 Joint Directors of Laboratories

anomalies (Guerriero et al., 2008). Ce sont essentiellement des anomalies simples comme une différence entre la position d'un navire envoyé par son AIS et la position détectée par un radar.

3. **Enrichissement des données** : une fois ces données fusionnées, il est possible de les enrichir en utilisant différentes sources d'information (institutionnelles et/ou privées). Celles-ci permettent d'avoir une vision plus proche de la réalité et participe pleinement au processus décisionnel. En effet, cet enrichissement offre au décideur un niveau d'abstraction plus élevé et permet de comprendre les interactions potentielles entre les différentes entités. Celui-ci peut également s'accompagner d'interfaces spécifiques permettant une visualisation facilitée de l'information.
4. **Détection d'anomalies** : la détection automatique d'anomalies n'a pas pour objectif de supplanter l'opérateur mais plutôt de l'aider dans ces tâches quotidiennes en diminuant la surcharge cognitive que peut entraîner la surveillance et l'analyse d'un grand nombre de données. Ce processus de détection s'appuie sur des modèles composés d'événements récurrents et prédictifs. L'analyse de ces comportements s'appuie sur des méthodes et des algorithmes provenant de différents domaines comme les statistiques ou encore les réseaux de neurones (Hodge and Austin, 2004).
5. **Contextualisation des anomalies** : comme cela a déjà été précisé, le contexte joue un rôle important dans la détection d'anomalies. Il participe à l'amélioration de la détection tout en diminuant les potentiels faux positifs. Par exemple, le changement de trajectoire d'un navire peut s'expliquer par un avis de tempête ou la proximité d'un iceberg. Or, sans la prise en compte de ce contexte, le système aurait automatiquement généré une alerte. Bien évidemment, en raison de différentes limites, il apparaît illusoire aujourd'hui d'imaginer qu'il soit possible d'arriver à une complétude des informations.
6. **Évaluation de la menace** : comme le souligne Roy (Roy, 2008), la notion d'anomalie n'est pas nécessairement synonyme de menace. C'est pourquoi une évaluation est nécessaire. De plus, celle-ci permet une gradation concernant le niveau de dangerosité. A l'heure actuelle, peu de travaux, méthodes et algorithmes existent. Néanmoins, dans le cadre des activités menés par le Centre de Recherche sur les risques et les Crises différentes approches ont été expérimentées. Citons notamment la détection de la menace par une approche Bayésienne (Chaze et al., 2012), par Data Mining (Idiri and Napoli, 2012), ou à l'aide d'ontologies (Vandecasteele and Napoli, 2012b).
7. **Diffusion et affichage** : c'est le sommet de la pyramide. Une anomalie a été détectée et catégorisée comme étant potentiellement une menace. Il s'agit alors de diffuser cette information aux autorités compétentes mais aussi aux autres systèmes et organismes potentiellement reliés. Là encore, d'importants travaux restent encore à réaliser pour aboutir à une présentation claire et effective des alertes (Riveiro et al., 2008; Willems, 2011).

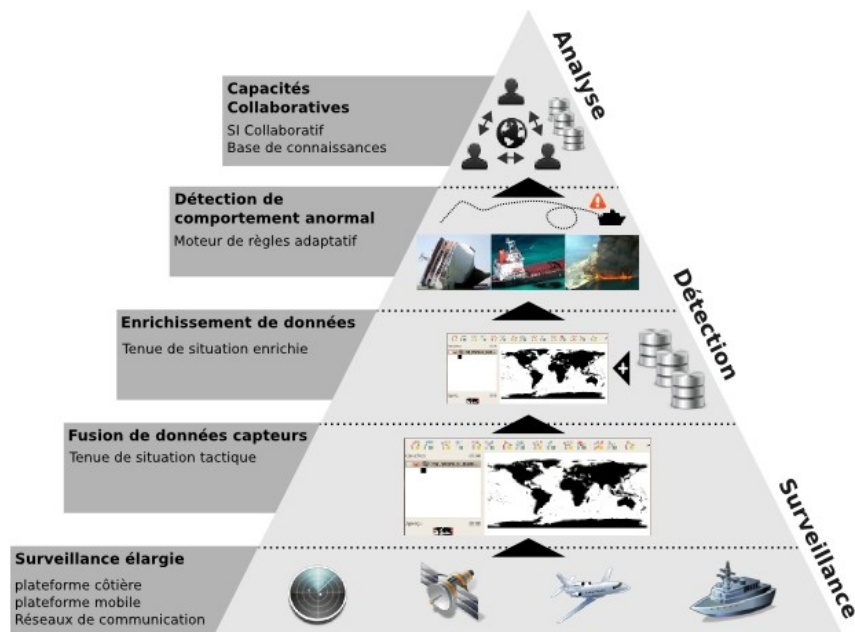


Figure 4: Architecture d'un système de détection d'anomalies et d'interprétation de comportements anormaux.
Source (Vandecasteele and Napoli, 2012a)

HYPOTHÈSE ET OBJECTIFS DE RECHERCHE

HYPOTHÈSE

L'hypothèse principal sur laquelle se fonde ce travail de recherche est qu'une **formalisation enrichie à l'aide de la composante spatiale favorise la description et l'analyse du comportement des navires**. Ce postulat part d'un point central, celui du processus de décision. Dans le domaine maritime, la reconnaissance de comportements anormaux de navires fait partie des décisions que Simon qualifie de non structurées (Simon, 1977). Il est alors nécessaire d'aider l'expert en charge de la surveillance maritime à analyser la situation en lui permettant de manipuler facilement l'information, mais aussi en lui offrant les mécanismes de raisonnement spatial nécessaires à la prise de décision. Cette information peut être enrichie sémantiquement afin d'ajouter à la donnée initiale du contenu informationnel. Ce contenu porte directement sur ses attributs, mais aussi sur les différents liens que cette donnée peut avoir avec les éléments du modèle proposé.

OBJECTIFS DE RECHERCHE

Afin de démontrer cette hypothèse, l'objectif principal de ce travail de recherche est de proposer une nouvelle approche permettant d'améliorer la modélisation des comportements anormaux de navires à l'aide de la composante spatiale. Plus précisément, les objectifs sont :

Introduction

- Étudier la faisabilité d'intégrer, au sein d'un langage de modélisation, la dimension spatiale et de raisonner à partir de celle-ci. Cela permettra ainsi d'offrir un formalisme enrichi permettant une meilleure description des comportements anormaux.
- Concevoir un modèle global pour la surveillance maritime
- Concevoir un système intégrant cette modélisation afin d'analyser automatiquement les comportements anormaux.

Les sujets abordés dans ce manuscrit sont vastes et font toujours l'enjeu d'importants travaux scientifiques. Néanmoins, la méthode ainsi que le prototype présentés démontrent la faisabilité de notre approche et ouvrent la voie à de nouvelles perspectives.

PROPOSITION DE RECHERCHE ET APPROCHE ADOPTÉE

Afin de répondre aux questions décrites ci-dessus, cette thèse se propose d'étudier les potentialités des ontologies spatiales pour l'analyse des comportements de navires. Mais si les ontologies offrent les structures nécessaires à la modélisation des connaissances, elles ne disposent à l'heure actuelle d'aucune formalisation standardisée de la dimension spatiale. Or celle-ci est particulièrement importante pour l'analyse de comportements anormaux. De ce fait, une extension ontologique permettant la création de règles spatiales est proposée.

Plus concrètement cette proposition se traduit sous la forme d'un système nommé OntoMap permettant de décrire et de qualifier le comportement d'un navire. Celui-ci se base sur quatre fonctionnalités majeures (Figure 5) :

1. Tout d'abord l'acquisition de la connaissance. Pour cela, OntoMap intègre différentes ontologies telles que GeoRSS-Simple pour la partie spatiale ou encore Simple Event Model (SEM) pour la description des événements. Ces ontologies sont fusionnées en un seul élément afin d'offrir un cadre conceptuel riche pour la description des comportements de navires.
2. Deuxièmement, le management de la connaissance. Les données sources provenant de l'AIS des navires sont intégrées afin de former une base de connaissances. Ces données sont enrichies sémantiquement afin d'obtenir des trajectoires, des alertes et des comportements sémantiques qui serviront lors des analyses futures.
3. Troisièmement, l'analyse de la connaissance. À partir des informations contenues dans la base de connaissance et des règles d'inférences spatiales qui ont été définies, OntoMap caractérise le comportement de chaque navire. Puis, les comportements suspects sont ensuite classifiés par un module basé sur un raisonnement à partir de cas (RAPC)
4. Et enfin la visualisation de la connaissance. Les résultats obtenus sont présentés à l'utilisateur sous la forme d'une interface cartographique.

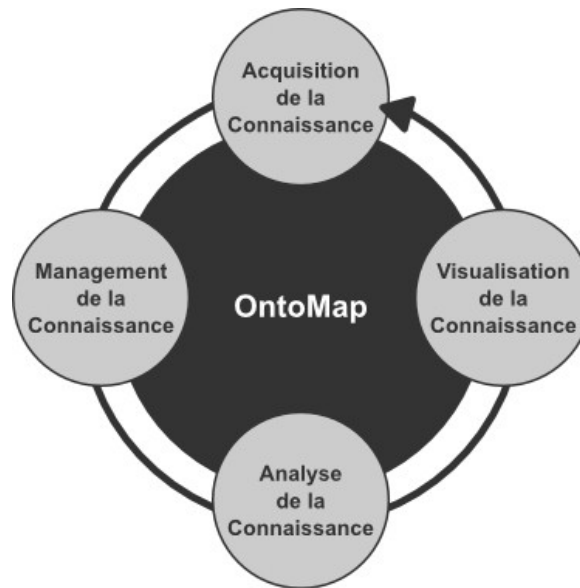


Figure 5: Les quatre piliers soutenant la réalisation du prototype OntoMap

Ainsi, OntoMap est à la fois un environnement de modélisation d'ontologies spatio-temporelles et un système d'inférence capable de raisonner à partir de connaissances spatiales et thématiques pour la détection de comportements anormaux de navires.

MÉTHODOLOGIE & STRUCTURE DE LA THÈSE

MÉTHODOLOGIE

Bien que ce travail de recherche ne soit pas directement rattaché aux projets de surveillance maritime auxquels nous avons participé, celui-ci a été mené en parallèle et puise son inspiration dans les différents constats identifiés. Ces constats ainsi que les apports scientifiques ont permis d'aboutir à la définition du projet de recherche décrit dans ce manuscrit. La méthodologie générale suivie est présentée sur la Figure 6.

Tout d'abord, conjointement aux observations effectuées, une recherche bibliographique a été réalisée afin d'identifier les principaux travaux précédemment publiés. Cette veille bibliographique a notamment porté sur les ontologies, les ontologies spatiales, le raisonnement spatial mais aussi sur différents sujets comme le processus décisionnel, la collaboration ou encore la cognition. Cette étape bibliographique a ainsi permis de définir le contexte de recherche dans lequel s'inscrit cette thèse et de poser un certain nombre d'hypothèses et d'objectifs à atteindre. Cette étape a donné lieu à un premier rapport écrit qui a ensuite été défendu à l'oral devant un comité.

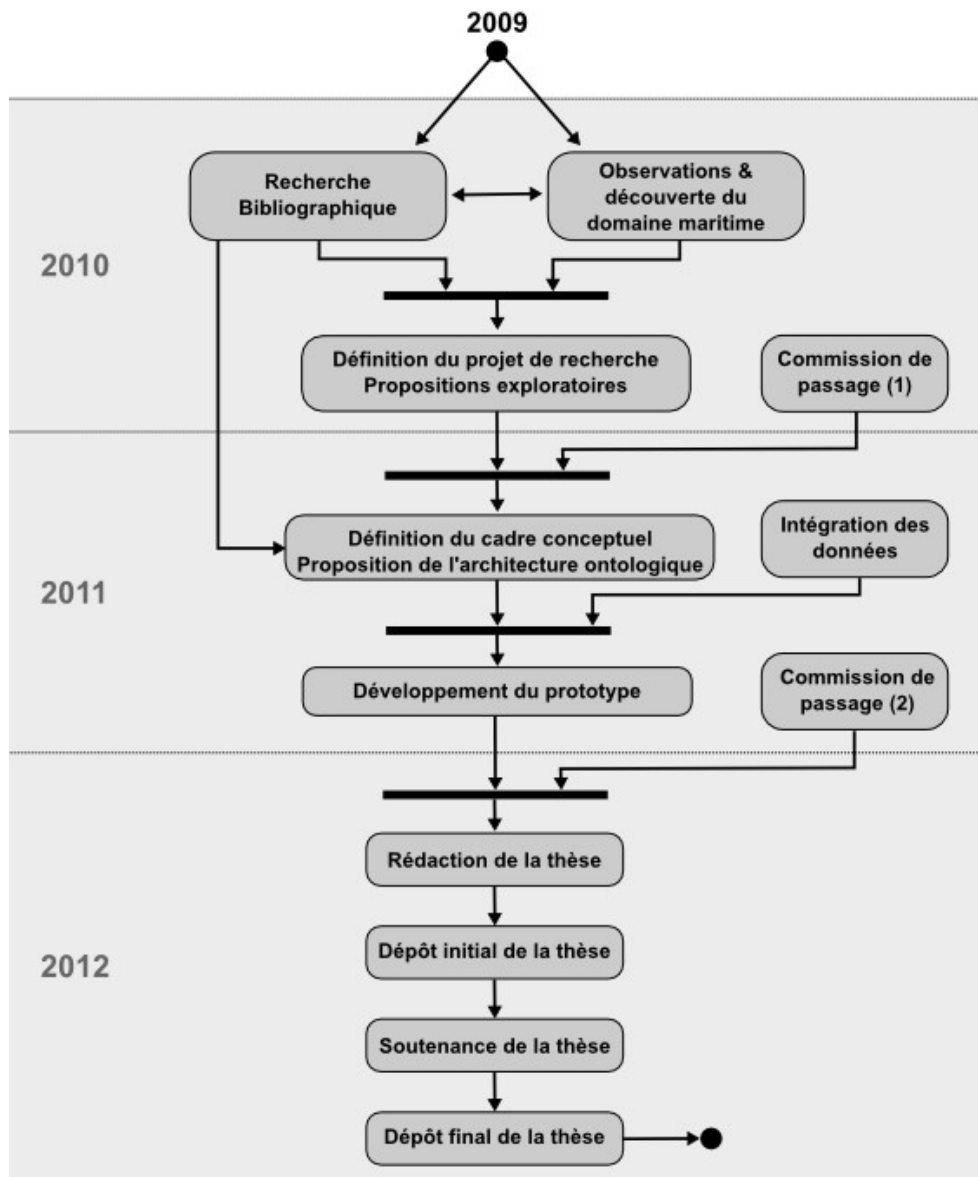


Figure 6: Diagramme d'activité de la méthode de recherche

Suite à la validation de ce rapport, la seconde étape portant sur la création d'un système d'analyse des comportements anormaux de navires a été réalisée. Cette étape peut être divisée en deux principaux cycles. Tout d'abord la création du cadre ontologique à partir de l'état de l'art précédemment effectué. Deuxièmement le développement du système permettant de valider les hypothèses et les concepts exposés lors des étapes précédentes.

Le cadre ontologique proposé s'appuie sur trois ontologies permettant notamment de définir un navire, sa trajectoire, les pays à proximité ou encore les alertes et les scénarios potentiels. Pour cela, une analyse des ontologies existantes a été effectuée afin d'intégrer celles jugées comme pertinentes à notre

problématique. Une ontologie spécifique au domaine maritime a également été réalisée. Celle-ci s'appuie à la fois sur des interviews d'experts et sur la littérature du domaine.

À partir de ce cadre ontologique, un prototype a été développé. Celui-ci offre un cadre informatique riche permettant à partir de trajectoires spatio-temporelles d'analyser les comportements anormaux de navires. Pour cela différentes briques logicielles ont été utilisées. Les principales étant : *OWLAPI* pour la gestion et la manipulation de l'ontologie, *Pellet* pour l'inférence et *jColibri* pour le Raisonnement À Partir de Cas.

Enfin, le prototype ainsi que la méthodologie ont à nouveau été présentés devant une commission de passage. Celle-ci avait pour objectif de valider tant sur le plan scientifique que technique l'approche proposée. Après l'accord de cette commission, l'étape de rédaction de thèse a alors débuté. Celle-ci s'est terminée avec la soutenance et le dépôt final du manuscrit.

STRUCTURE DE LA THÈSE

Dans cette introduction, le cadre général dans lequel s'inscrit cette thèse a été présenté. Cela a permis de mettre en évidence les besoins tant sur le plan scientifique, qu'opérationnel d'une meilleure intégration de la dimension sémantique dans les Systèmes d'Information Géographique. Différentes questions et pistes de réflexion ont été soulevées. Pour y répondre, un parcours composé de deux parties et de six chapitres retrace le cheminement intellectuel qui a conduit cette recherche (Figure 7).

La première partie est consacrée à l'état de l'art et regroupe trois chapitres. Le premier porte sur la modélisation des trajectoires et plus précisément sur la notion de trajectoire sémantique. Dans le second chapitre, le domaine des ontologies et la manière dont celles-ci peuvent être utilisées pour modéliser le monde réel sont présentés. Enfin, le dernier chapitre aborde la question de l'intégration de la dimension spatiale au sein des ontologies.

La seconde partie qui découle directement de l'état de l'art est spécifique aux travaux qui ont été réalisés dans le cadre de cette thèse. Trois chapitres structurent l'approche proposée. Le premier présente l'approche ontologique qui a servi à la modélisation des connaissances. Le second décrit le prototype réalisé ainsi que les différents modules. Puis, le dernier chapitre illustre l'utilisation de ce prototype par différents cas d'utilisation.

Enfin, la conclusion rappelle les apports de cette thèse et présente les différentes perspectives possibles associées à ce travail de recherche.

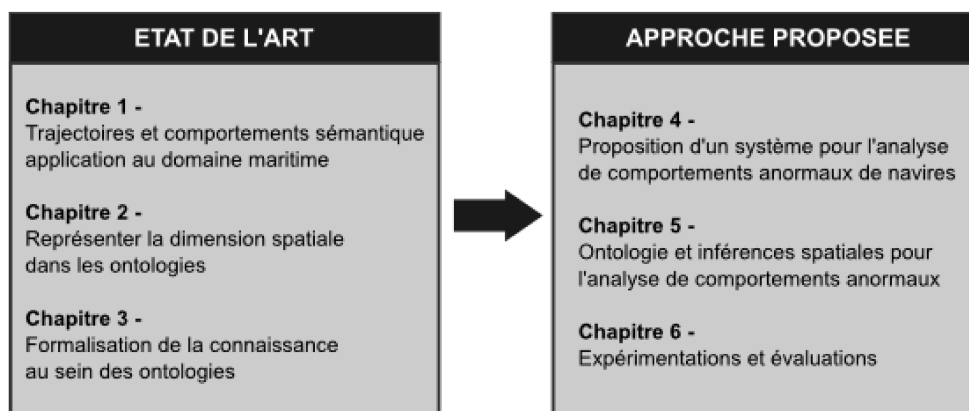


Figure 7: Plan de la thèse

PARTIE 1 ÉTAT DE L'ART

CHAPITRE 1. TRAJECTOIRES ET COMPORTEMENTS SÉMANTIQUES

Chapitre 1. Trajectoires et comportements sémantiques.....	31
1.1. D'un objet mobile à une trajectoire sémantique.....	33
1.1.1. <i>Modélisation conceptuelle des trajectoires.....</i>	<i>34</i>
1.1.2. <i>Enrichissement sémantique des trajectoires spatio-temporelles.....</i>	<i>36</i>
1.2. De la normalité aux anomalies sémantiques.....	40
1.2.1. <i>Détection d'anomalies</i>	<i>40</i>
1.2.2. <i>Création de trajectoires.....</i>	<i>42</i>
1.2.3. <i>Anomalie sémantique</i>	<i>47</i>
1.3. D'une trajectoire anormale à un comportement.....	49
1.3.1. <i>Simple Event Model (SEM).....</i>	<i>49</i>
1.3.2. <i>ISFAR</i>	<i>53</i>

Introduction du chapitre

L'augmentation croissante du nombre de capteurs de positions (GPS, AIS, etc.) fournit de nombreuses informations concernant le déplacement des objets mobiles (Figure 1.1). Ces objets mobiles qui sont par nature spatio-temporels nécessitent de disposer de formes de stockage et d'exploitation spécifiques. Particulièrement actif sur ce sujet, c'est dans le domaine des bases de données que les principaux travaux ont été réalisés. Ces derniers ont permis d'aboutir à la définition de types et de fonctions spécifiques permettant de modéliser et manipuler la notion de trajectoires.

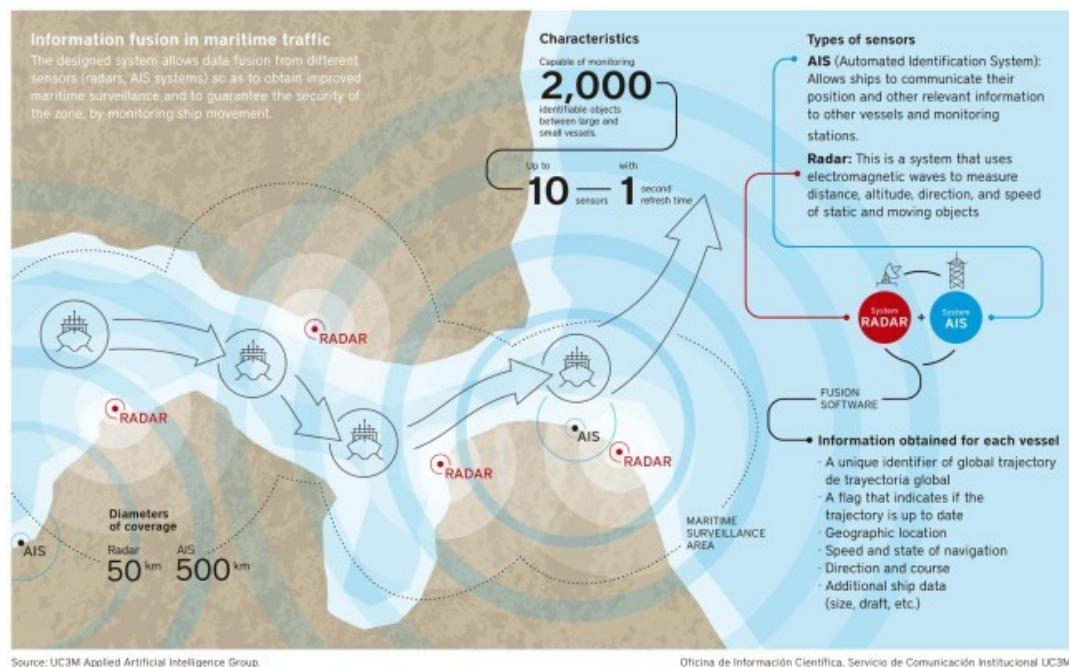


Figure 1.1: Couplage AIS et Radar dans le domaine maritime.
Source (Source: UC3M Applied Artificial Intelligence Group)

À partir de cette modélisation des objets mobiles, il est alors possible d'effectuer des analyses de comportements. Néanmoins, la simple position d'un objet mobile n'est pas suffisante. Il est nécessaire pour cela de disposer d'informations complémentaires qui viendront enrichir la trajectoire initiale. Cette notion, nommée trajectoire sémantique, est à la base de la réflexion initiée dans ce chapitre. Dans un premier temps, le passage d'une position à une trajectoire sémantique est présenté (§1.1). Puis, une étude de la notion d'anomalie appliquée au domaine maritime est réalisée (§1.2). Enfin, un pont entre anomalie et comportement est tracé dans le paragraphe 1.3.

1.1. D'un objet mobile à une trajectoire sémantique

Bien que l'étude des objets mobiles se retrouve dans de nombreux domaines comme la biologie, l'informatique ou la géomatique, les principales contributions ont été réalisées par la communauté des bases de données (Zheng and Zhou, 2011). Celle-ci s'est notamment intéressée à la définition de nouveaux types de données et d'opérateurs de requêtes spécifiques permettant de représenter et d'interroger des objets mobiles au sein d'un Système de Gestion de Bases de Données (GBD). Ces objets mobiles, caractérisés par une forte variabilité temporelle et spatiale nécessitent de nombreux accès en lecture et en écriture dans les GBD. De ce fait, les performances de ces derniers peuvent rapidement se dégrader sans une architecture adéquate (Noyon and Devogele, 2004). Dans les paragraphes suivants, une présentation succincte des travaux effectués dans le domaine des GBD sera réalisée pour ensuite aborder plus concrètement la notion de trajectoire sémantique. Néanmoins, une revue complète des différentes méthodes de modélisation d'objets mobiles est disponible dans la thèse de Noyon (Noyon, 2007).

Historiquement, les premiers travaux sur ce domaine ont été réalisés par Wolfson et son équipe qui ont identifié les principales limitations et les solutions potentielles à l'implémentation des objets mobiles dans les SGBD (Wolfson et al., 1998). Ils proposent pour cela le modèle MOST (Moving Objects Spatio-Temporal) qui décrit des bases de données contenant des attributs dynamiques variant continuellement au cours du temps, ainsi que le langage de requêtes FTL (Future Temporal Logic) basé sur la logique temporelle (Prasad Sistla et al., 1997). Par la suite, les résultats de ces recherches ont été mis en application au sein du projet DOMINO (Databases fOr MovINg Objects tracking) (Wolfson et al., 1999). Bien que les avancées réalisées par Wolfson et son équipe constituent un travail important, elles présentent néanmoins certaines limites. Parmi celles-ci, il a été démontré que le modèle développé est inadapté pour les objets se déplaçant dans un espace non contraint (animaux, navires, etc.). Deuxièmement, comme la notion de trajectoire n'a pas été formalisée, il n'existe pas de types de données ou d'opérations définis (Malki et al., 2009).

Par la suite, différents travaux ont porté sur la formalisation des objets mobiles. L'une des propositions les plus complètes et qui fait aujourd'hui encore référence est celle définie par Güting (Güting et al., 2000, 2006). Dans leur approche, les auteurs couvrent aussi bien la modélisation, que le langage ou encore les requêtes possibles. Pour cela, les types de données (base, spatial, etc.) nécessaires à la représentation d'un objet mobile ont été définis ainsi qu'une algèbre spécifique et une collection d'opérations possibles (*distance*, *direction*, *common_border*, etc.). Différentes implémentations ont ensuite été réalisées comme c'est le cas pour le projet SECONDO (Forlizzi et al., 2000) ou encore pour le projet HERMES (Pelekis et al., 2006).

La formalisation et la conceptualisation des trajectoires dans un système informatique sont une étape nécessaire mais néanmoins insuffisante si l'on souhaite comprendre, exploiter et analyser celles-ci. Afin de faire face à ces limites, différentes approches ont été proposées comme notamment la prise en compte de la dimension temporelle et spatiale des mesures (Sandu Popa et al., 2009) ou une meilleure intégration de la sémantique associée aux trajectoires (Malki et al., 2009). Dans le cadre de cette thèse, c'est cette dernière orientation qui a été explorée au travers notamment de la notion de trajectoire sémantique définie par Spaccapietra (Spaccapietra et al., 2008).

1.1.1. Modélisation conceptuelle des trajectoires

Modéliser le déplacement d'un objet mobile, nécessite d'identifier les événements spatio-temporels permettant l'identification des différentes étapes constituant une trajectoire. Dans la littérature, plusieurs modèles existent. Ces derniers varient tant au niveau des thématiques auxquels ils s'appliquent que des besoins auxquels ils répondent. Par exemple, le modèle proposé par Mouza et Rigaux (Mouza and Rigaux, 2005) permet uniquement l'identification de motifs de trajectoires dont les déplacements seraient déjà connus. Le modèle de Brakatsoulas et son équipe (Brakatsoulas et al., 2004) s'attache quant à lui à définir un modèle sémantique ainsi que les relations potentielles mais uniquement pour le domaine routier. Du fait de ces différentes limites, le choix du modèle de trajectoire s'est porté sur celui proposé par Spaccapietra (Spaccapietra et al., 2008). Celui-ci, plus générique, introduit un ensemble de concepts permettant d'attacher des informations à des événements spécifiques (*stops, moves*) et ainsi d'obtenir une trajectoire sémantiquement enrichie. Les spécificités de celui-ci seront décrits dans les paragraphes ci-dessous.

1.1.1.1 Typologie des notions de trajectoires

Avant d'aborder concrètement la notion de trajectoire, il est nécessaire de spécifier dans quel espace celle-ci s'inscrit. Pour cela (Spaccapietra et al., 2008) distinguent trois principales dimensions : (1) la dimension métaphorique, (2) la dimension géographique et (3) la dimension spatio-temporelle. Chacune d'entre elles renvoie à une notion, un sens, qui peut être différent.

La trajectoire métaphorique n'est pas directement rattachée à la notion de mouvement mais elle fait plutôt référence à l'idée d'évolution. Par exemple, on peut parler du cheminement professionnel d'une personne. Dans ce cas, si la dimension temporelle est bien présente, il n'y a pas nécessairement de dimension spatiale. D'un point de vue informatique, ces évolutions sont gérées comme un changement d'état d'un des attributs de l'entité.

La trajectoire géographique est un cas particulier de la dimension métaphorique. Bien que la dimension spatiale soit plus explicite, elle n'en reste pas moins uniquement textuelle. Ce type de trajectoire est utilisé quotidiennement quand nous souhaitons par exemple indiquer un itinéraire à une personne ou décrire notre dernier voyage. C'est pourquoi les auteurs rapprochent cette trajectoire à la notion de géographie naïve définie par Egenhofer (Egenhofer and Mark, 1995).

La trajectoire spatio-temporelle est celle manipulée dans les SIG. Il s'agit d'une suite de positions exprimées en utilisant des coordonnées géographiques. Si la partie géométrique est aujourd'hui maîtrisée d'un point de vue informatique (du moins pour les géométries en deux dimensions), il en est tout autrement de la partie sémantique. Pour que cette sémantique puisse être appliquée, il est nécessaire au préalable de disposer d'une modélisation de la trajectoire en tant qu'objet. Cela permettra de distinguer les différents éléments qui la composent pour ensuite y rattacher les informations sémantiques appropriées. Cette idée a notamment été explorée par Spaccapietra (Spaccapietra et al., 2008) sous le terme de trajectoire sémantique.

1.1.1.2 Composants d'une trajectoire spatio-temporelle

De par ses caractéristiques, la notion de trajectoire telle que définie par Spaccapietra fait nécessairement référence à une dimension spatio-temporelle. En effet, celle-ci implique le déplacement d'une entité dans une zone géographique et durant un laps de temps déterminé. Celle-ci peut donc être définie sémantiquement de la manière suivante :

trajectory: $[t_{begin}, t_{end}] \rightarrow space$

Cette double vision une à la fois spatiale et temporelle des trajectoires fait écho à l'idée de *geospatial lifeline* définie notamment par Mark et Egenhofer (Mark et al., 1999; Hornsby and Egenhofer, 2002). En effet, on retrouve dans les *geospatial lifeline* une visualisation selon différents niveaux de granularité, les périodes temporelles durant lesquelles une entité occupe un espace donné.

Concernant l'aspect spatial dans lequel évolue une trajectoire, il est courant de caractériser celui-ci en fonction de l'influence qu'il peut entraîner. Par exemple, dans le cas d'une voiture, l'objet mobile est contraint d'évoluer au sein du réseau routier. Dans d'autres cas, comme le domaine maritime, l'espace est dit ouvert. Bien évidemment pour des raisons de rentabilité économique, les navires suivront pour la plupart une trajectoire rectiligne entre différents points d'intérêts. Cette configuration spécifique peut être alors modélisée sous la forme d'un graphe de zones comme cela a été démontré par Etienne (Etienne, 2011) et Devogele (Devogele, 2009). Enfin, du point de vue temporel, il est important de préciser que la trajectoire définie par la fonction $[t_{begin}, t_{end}]$ n'est qu'une période spécifique dans la vie d'un objet mobile. Autrement dit, une trajectoire est un segment temporel réalisé par un objet mobile. De ce fait, elle est marquée par des périodes de mouvements et d'arrêts (Figure 1.2).

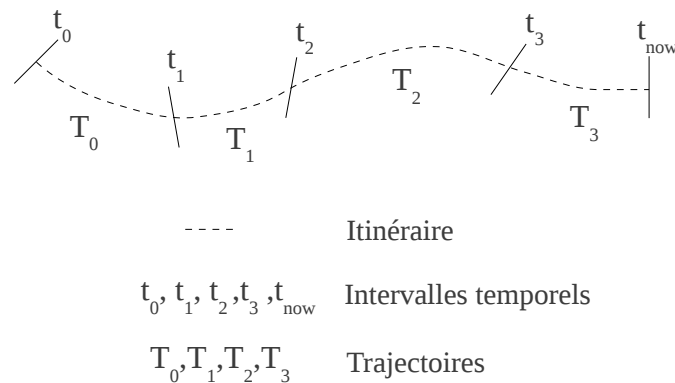


Figure 1.2: Dimension spatio-temporelle d'un objet mobile d'après (Spaccapietra et al, 2008)

Les périodes d'arrêts d'une trajectoire

La notion d'arrêt (*stop*) pour un objet mobile est particulièrement importante. En effet, celle-ci va permettre de délimiter temporellement et spatialement les différentes trajectoires. Mais pour que ces arrêts soient significatifs, il faut les rattacher à un contexte spécifique. Par exemple, dans le domaine maritime du fait du mouvement de la mer, un navire n'est jamais réellement arrêté. Il sera donc nécessaire d'adapter la granularité temporelle de l'arrêt. Ce dernier peut être définie par les caractéristiques ci-dessous :

- l'utilisateur a explicitement défini cette portion de la trajectoire ($[t_{beginstopx}, t_{endstopx}]$) comme étant un arrêt
- l'intervalle temporel de l'arrêt ($[t_{beginstopx}, t_{endstopx}]$) est différent de nul

Chapitre 1. Trajectoires et comportements sémantiques

- L'objet mobile n'est pas en mouvement. Cela signifie donc que la géométrie de l'objet est constituée d'un unique point ainsi que tous les arrêts sont nécessairement disjoints à la fois spatialement et temporellement. Comme cela a été souligné, cette notion doit être évidemment nuancée par le contexte d'application.

Les périodes de mouvement d'une trajectoire

Le mouvement (*move*) est la période temporelle entre deux arrêts durant laquelle un objet se déplace dans l'espace. Celle-ci est donc définie par une suite de positions ordonnées temporellement. Il se définit sémantiquement de la manière suivante :

$\text{move: } [t_{\text{beginmove}}, t_{\text{endmove}}] \rightarrow \text{space}$

Le mouvement d'un objet peut être défini comme faisant partie d'une trajectoire si :

- la partie de la trajectoire concernée est délimitée à ses deux extrémités par deux arrêts de telle sorte que t_{begin} et le premier arrêt ainsi que t_{end} et le second arrêt sont confondus.
- L'intervalle temporel entre ($[t_{\text{beginmove}}, t_{\text{endmove}}]$) est différent de nul
- La géométrie de la trajectoire ($[t_{\text{beginmove}}, t_{\text{endmove}}]$) est une ligne.

Chacune de ces périodes peut être ensuite enrichie sémantiquement en ajoutant par exemple des informations relatives à l'endroit où s'est produit le stop ou encore aux conditions météorologiques.

1.1.2. Enrichissement sémantique des trajectoires spatio-temporelles

Géographique par nature, une trajectoire est constituée d'une succession d'événements (*stops, moves*) spatio-temporels. Bien que cette dimension spatio-temporelle soit nécessaire à l'analyse d'un objet mobile, elle est néanmoins insuffisante. En effet, différentes informations, dites sémantiques, sont nécessaires et doivent être ajoutées. Les paragraphes ci-dessous présentent l'approche choisie ainsi que la notion de trajectoire sémantique.

1.1.2.1 Processus d'enrichissement sémantique et modélisation ontologique

Jusqu'à présent, l'essentiel de la problématique a consisté à proposer une modélisation la plus fidèle possible des trajectoires. Néanmoins, une trajectoire encodée sous la forme d'une suite de coordonnées ne fournit aucune information à propos du contexte dans lequel évolue l'objet. Or ce contexte participe tout autant à la compréhension et à l'interprétation d'une situation. C'est pourquoi en se basant sur la notion de trajectoire sémantique définie par Spaccapietra (Spaccapietra et al., 2008), Baglioni (Baglioni et al., 2008) propose d'augmenter le contenu informationnel d'une trajectoire par un processus d'enrichissement sémantique (Figure 1.3).

1.1. D'un objet mobile à une trajectoire sémantique

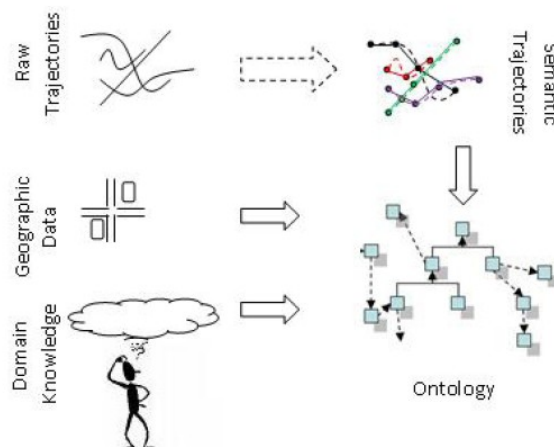


Figure 1.3: Processus d'enrichissement sémantique des trajectoires. Source : (Baglioni et al, 2008)

Ce processus permet à partir de données brutes d'obtenir des trajectoires dont le contenu informationnel a été enrichi par l'ajout d'informations liées à la zone géographique ou encore d'informations ayant trait au domaine d'activité de l'utilisateur. En plus d'apporter des informations complémentaires, ces trajectoires sémantiques doivent être manipulables directement par l'utilisateur en utilisant un langage expressif et humainement compréhensible. De ce fait, il est nécessaire de les modéliser à l'aide d'un formalisme de représentation spécifique. Actuellement, c'est le domaine des ontologies qui offre la plus grande expressivité. C'est pourquoi les auteurs ont choisi d'utiliser le formalisme OWL pour représenter les trajectoires. L'ontologie obtenue (Figure 1.4) est constituée des différents composants d'une trajectoire (cf Paragraphe 1.1.1.2). Dans celle-ci sont définis les deux éléments *Stop* et *Move* permettant de délimiter spatialement et temporellement l'activité d'un objet mobile. Ainsi, dans l'approche des auteurs, la conceptualisation d'une trajectoire est représentée par une séquence d'arrêt connecté à un déplacement. Cette connexion se fait à l'aide de l'une des quatre relations : *fromStop*, *toStop*, *inMove*, *outMove*. De plus, chaque *Stop* s'inscrit dans une dimension temporelle spécifique comme cela est précisé par la relation *StopHasTime*.

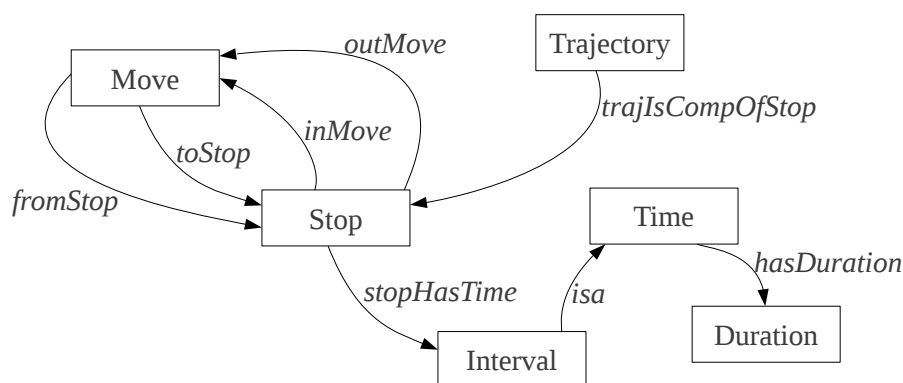


Figure 1.4: Représentation ontologique de la notion de trajectoire sémantique. Source (Baglioni et al, 2008)

Néanmoins, bien que ces trajectoires soient sémantiquement plus riches, ce processus s'accompagne selon les auteurs d'une diminution de la précision géométrique des données. Cela permet notamment une manipulation plus simple des informations contenues pour les langages de représentation. Mais, dans l'hypothèse où un langage ontologique disposerait de types spatiaux

définis, il serait possible d'imaginer de stocker la géométrie exacte et non plus une simplification de celle-ci.

1.1.2.2 Application du processus d'enrichissement sémantique

Basé sur une logique similaire d'enrichissement sémantique, Yan propose dans son travail de thèse une architecture complète pour la création, la gestion et l'analyse des trajectoires (Yan, 2011). Pour cela, il s'appuie sur une infrastructure modulaire composée de trois principales ontologies : une ontologie géométrique des trajectoires, une ontologie géographique et une ontologie du domaine d'application (Figure 1.5).

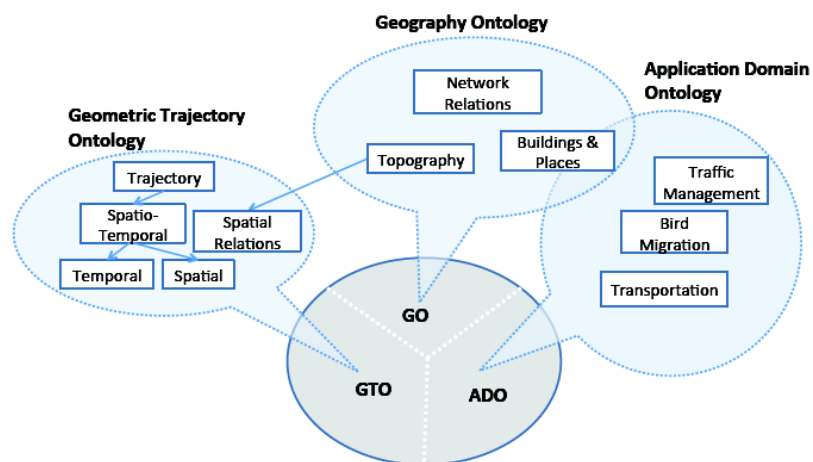


Figure 1.5: Architecture ontologique pour l'enrichissement sémantique de trajectoires. Source (Yan, 2011).

Bien que les notions relatives aux ontologies seront étudiées dans le paragraphe suivant (Chapitre 2. p.57), il est nécessaire de préciser succinctement la signification de ce terme. Conceptuellement, une ontologie peut être vue comme un réceptacle possédant les structures nécessaires à la modélisation des connaissances. En fonction du formalisme utilisé, ces connaissances sont ensuite encodées afin d'être utilisées aussi bien par les individus que par les systèmes. Les trois ontologies nécessaires à la description d'une trajectoire sont détaillées ci-dessous.

L'ontologie géométrique des trajectoires est elle-même constituée de plusieurs sous ontologies spécialisées dans la description des concepts spatiaux ou temporels. Tout l'intérêt de l'approche proposée est de modéliser la trajectoire d'un objet mobile comme une suite de positions ordonnées temporellement. Ainsi, une trajectoire peut être représentée par une suite de triplets $\langle x, y, t \rangle$ contenant la position en x et en y ainsi qu'une référence temporelle t . Cette abstraction des entités géométriques peut être alors modélisée à l'aide des logiques de description (Tableau 3.1) (Yan et al., 2008).

1.1. D'un objet mobile à une trajectoire sémantique

Concept	Définition en logique de description
TimePoint	$\subseteq \exists \text{ hasGeometry.Point} \cap \exists \text{ hasTime.Instant}$
Begin End Stop B.E.S	$\subseteq \text{TimePoint}$ $\subseteq \text{TimePoint}$ $\subseteq \exists \text{ hasGeometry.Point} \cap \exists \text{ hasTime.Interval}$ $\equiv \text{Begin} \cup \text{End} \cup \text{Stop}$
Move	$\subseteq \text{TimeVaryingPoint} \cap \exists \text{ from.D.F.A}$
Trajectory	$\equiv \exists \text{ hasBegin.Begin} \cap \neq 1 \text{ hasBegin} \cap \exists \text{ hasEnd.End} \cap \neq 1$ $\text{hasEnd} \cap \exists \text{ hasMove.Move} \cap \forall \text{ hasStop.Stop}$

Tableau 3.1: Représentation des éléments constitutifs d'une trajectoire exprimés en logique de description. Source (Yan, 2008)

L'ontologie géographique contient les différents concepts géographiques portant sur la description du territoire. Ces concepts peuvent faire référence aussi bien à des éléments naturels (rivière, montagne, etc.) qu'artificiels (routes, bâtiments, etc.). La granularité et la diversité des informations à intégrer seront fonction bien évidemment des besoins de l'application. Ce module occupe une place centrale dans l'architecture car il est en lien avec les deux autres ontologies. En effet, tout concept géographique possède une référence spatiale et peut donc être décrit en utilisant les éléments contenus dans l'ontologie géométrique des trajectoires. De plus, ce module est également lié à l'ontologie d'application du fait que les concepts géographiques peuvent être précisés en fonction des besoins de l'application. Comme son nom l'indique, l'ontologie du domaine d'application est spécifique au domaine sur lequel porte l'application. Les éléments à intégrer dépendront de la thématique et des besoins identifiés.

Ces trois ontologies ont été ensuite combinées afin de fournir l'infrastructure sémantique nécessaire à la description de trajectoires pour une application de gestion du trafic routier. Pour l'exploitation et le stockage des données deux approches ont été explorées. Tout d'abord, une approche purement ontologique s'appuyant sur Protégé et un raisonneur (Racer, Pellet, etc.). Néanmoins, celle-ci entraînait d'importantes limitations notamment sur les performances du système en fonction de l'importance des raisonnements (cf. les notions de Abox et Tbox abordées dans le paragraphe 2.3.1.4 p.72). C'est pourquoi une seconde approche s'appuyant sur une base de données relationnelles possédant une extension spatiale et une extension RDF¹² a été proposée (Yan et al., 2008). Notons que c'est également le choix fait par Malki (Malki et al., 2009; Mefteh et al., 2012) dans ses travaux concernant une application similaire mais appliquée aux trajectoires sémantiques des mammifères marins. Néanmoins, si cette solution offre l'avantage de performances élevées en terme de stockage et d'accès à l'information elle ne permet pas, à notre connaissance, l'utilisation de règles. Or celles-ci sont au cœur de ce travail de thèse. De ce fait, il sera donc nécessaire de s'appuyer sur une architecture ontologique classique.

12 Base de données Oracle avec l'extension Oracle Semantics et Oracle Spatial

1.2. De la normalité aux anomalies sémantiques

La précédente section a permis de mettre en évidence la nécessité de ne pas disposer uniquement de trajectoires brutes, mais de données enrichies en fonction d'un contexte donné. Cette notion, nommée trajectoire sémantique, peut être ensuite utilisée pour détecter des anomalies dans la position ou la trajectoire d'un objet mobile.

1.2.1. Détection d'anomalies

La détection d'anomalies est intrinsèquement liée à la notion de normalité. Il s'agit en effet de trouver un comportement déviant à ce qui a déjà été observé. Une présentation générale du domaine ainsi qu'une typologie des anomalies seront réalisées dans les paragraphes suivants.

1.2.1.1 Présentation générale du domaine de la détection d'anomalies

La détection d'anomalies fait référence à la problématique de trouver un modèle, un comportement non conforme à ce qui est potentiellement attendu. En se basant sur un jeu de données, il s'agit alors de construire des modèles de normalité à partir desquels seront détectées les déviations potentielles. Étudié en statistique dès le 19^e siècle, ce champ de recherche a été appliqué à de très nombreux domaines (bancaire, médical, informatique, etc.) (Chandola et al., 2009). Différentes approches, techniques et méthodes ont ainsi été proposées, certaines très génériques et d'autres spécifiques à un domaine d'application. C'est pourquoi cette notion de normalité et par extension d'anormalité se retrouve sous différentes formes dans la littérature scientifique (Roy, 2008).

Plus récemment, en raison du grand nombre de données disponibles et à l'augmentation croissante du nombre de capteurs (GPS, WIFI, etc.), la détection d'anomalies appliquée aux trajectoires a connu un vif intérêt (Laxhammar, 2011). Néanmoins, la plupart des algorithmes de détection nécessitent des trajectoires complètes avant d'effectuer le processus de classification. Cette procédure, dite *offline*, est une limitation importante pour les domaines nécessitant une réactivité immédiate. De ce fait de nombreux travaux ont été réalisés sur la définition d'algorithmes *online* (ou séquentiel) permettant la détection d'anomalies même sur des trajectoires incomplètes. L'analyse se fait alors en même temps que l'objet mobile évolue (Morris and Trivedi, 2008; Laxhammar, 2011). L'avantage de ces solutions de détection *online* a notamment été abordée dans Piciarelli et Foresti (Piciarelli and Foresti, 2006) ainsi que dans Rhodes et al. (Rhodes et al., 2007).

Dans le domaine maritime, le suivi et la détection de trajectoires anormales ont été facilités par l'obligation pour tous les navires¹³ soumis à la convention SOLAS (Safety Of Life At Sea) de disposer d'un transpondeur AIS¹⁴ (Figure 1.6). Ce transpondeur utilise un signal radio digital afin de diffuser des informations statiques (nom du navire, âge, cargaison, etc.) et dynamiques (cap, vitesse, localisation, etc.) aux autres navires et aux stations côtières équipés d'un récepteur AIS. Mais, ce transpondeur est limité par la puissance du signal radio (30 à 50 milles). C'est pourquoi, l'Organisation Maritime Internationale (IMO), propose d'y ajouter un système LRIT (Long Range Identification and Tracking) afin de permettre une surveillance étendue (1000 milles). Bien que conçus au départ pour éviter la collision entre les navires, ces systèmes ont également été utilisés

13 Les navires de plus de 300 tonnes, les navires effectuant des transports à l'international et les navires affectés au transport de passagers

14 Automatic Identification System

1.2. De la normalité aux anomalies sémantiques

pour détecter des trajectoires anormales comme notamment dans les travaux de Etienne (Etienne et al., 2010) ou encore ceux de Ristic (Ristic et al., 2008).

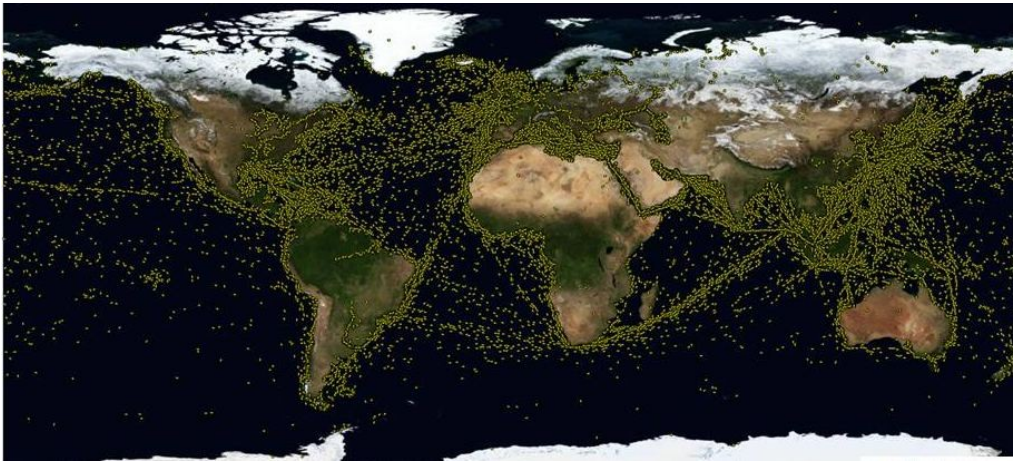


Figure 1.6: Cartographie mondiale des navires équipés d'un transpondeur AIS.
Source (Norwegian Defence Research Establishment (FFI))

Malgré l'augmentation du nombre de capteurs disponibles, il a été prouvé que les méthodes de détection supervisée utilisées habituellement ne sont pas suffisantes. Les raisons principales sont dues notamment au manque de formalisation de la connaissance experte mais aussi à l'absence de données couvrant le spectre des comportements (Patcha and Park, 2007).

1.2.1.2 Typologie des anomalies

Le type d'anomalies à identifier est un facteur important et déterminant concernant le choix des algorithmes à utiliser. Ces anomalies peuvent être regroupées en trois grandes catégories en fonction des données auxquelles elles s'appliquent (Chandola et al., 2009) :

- **anomalies ponctuelles ou individuelles** : celles-ci correspondent à l'anomalie d'un seul individu par rapport à l'ensemble d'un groupe donné. Elles sont parmi les plus simples à détecter et de nombreux travaux et algorithmes sont disponibles.
- **les anomalies collectives** : elles consistent à détecter l'anomalie d'un groupe d'individus par rapport au reste des données. Dans ce cas, si étudiée individuellement une donnée peut apparaître comme étant normale, c'est l'agrégation avec un ensemble plus large qui va permettre de révéler l'anomalie. Quelques travaux dans le domaine géographique existent notamment ceux réalisés par Shekhar (Shekhar et al., 2001) .
- **anomalies contextuelles** : également nommées anomalie conditionnelle (Song et al., 2007), celles-ci sont comme leur nom l'indique, relatives à un contexte donné. De ce fait, l'anormalité est liée à l'objet d'étude mais aussi à l'environnement dans lequel il opère. La notion de contexte est induite par la structure de l'ensemble des données et doit être spécifiée comme faisant partie du problème lui-même. Pour cela chaque donnée est définie au travers de ses attributs contextuels et comportementaux. Les attributs contextuels sont utilisés pour déterminer le contexte dans lequel s'inscrit l'objet. Par exemple, prenons le cas d'un objet géographique, ses données de position (longitude et latitude) sont des attributs contextuels. Inversement, les attributs comportementaux font référence à tout ce qui n'est pas contextuel

comme la valeur d'une mesure (Chandola et al., 2009). Les anomalies contextuelles ont été largement étudiées notamment dans le cas de données géographiques (Kou and Lu, 2006) ou temporelles (Salvador et al., 2004). Néanmoins, si l'apport de la dimension contextuelle permet de réduire les risques de faux positifs et d'améliorer l'analyse cela nécessite de disposer d'une modélisation appropriée ainsi que de données complémentaires.

1.2.2. Création de trajectoires

L'augmentation du nombre de capteurs disponibles a permis une analyse plus fine et continue des trajectoires d'objets mobiles. Ainsi, ce n'est plus seulement la position de l'objet qui est analysée mais son évolution tant sur le plan temporel que spatial. Particulièrement active dans ce domaine, la communauté travaillant à la détection d'anomalies à partir de vidéo a proposé de nombreuses approches (Dee and Velastin, 2008; Morris and Trivedi, 2008). L'objectif de ce travail de thèse n'étant pas d'en réaliser un inventaire exhaustif, seules celles basées sur un regroupement de trajectoires (*cluster*) seront présentées. Ce choix s'explique notamment par le fait que ce sont aujourd'hui les plus couramment utilisées.

1.2.2.1 Processus d'analyse des trajectoires

Le processus d'analyse de trajectoires peut être divisé en trois étapes principales consistant à : (1) traiter les données brutes, (2) réaliser un regroupement des trajectoires (*cluster*), (3) modéliser les trajectoires alors obtenues (Laxhammar, 2011).

Homogénéisation des trajectoires

L'objectif du traitement des données brutes est double. Premièrement, il permet de filtrer les données en éliminant les positions aberrantes et de ne garder que les positions significatives. Différents algorithmes peuvent être utilisés. Par exemple dans ses travaux de thèse, Etienne (Etienne, 2011) se base sur un filtre de Douglas et Peucker spatio-temporel (Bertrand et al., 2007) qui permet de réduire le volume de données de plus de 85% tout en gardant la même cohérence d'analyse (Figure 3.7). D'autres approches peuvent être également signalées comme l'utilisation de l'algorithme de compression Piecewise Linear Segmentation qui est une variante du filtre de Douglas et Peucker mais pour des séries temporelles. Celui-ci a notamment été utilisé par van Hage et son équipe et permet de réduire par 25 le nombre final de données à traiter (van Hage et al., 2009).

Deuxièmement, il vise à fournir une représentation des trajectoires adéquate au processus de regroupement. Par exemple, un des problèmes classiques qui est souvent rencontré lors d'un processus de *clustering* porte sur les longueurs des trajectoires qui peuvent être différentes. Bien évidemment, celles-ci ne sont pas homogènes et ne peuvent donc être efficacement comparées. Pour y remédier deux principales approches peuvent être utilisées. La première consiste à normaliser l'ensemble des trajectoires afin qu'elles aient toutes la même longueur. La seconde vise à redimensionner l'espace d'étude, ce qui a pour effet de réduire la complexité de calcul tout en améliorant le processus de clustering (Morris and Trivedi, 2008).

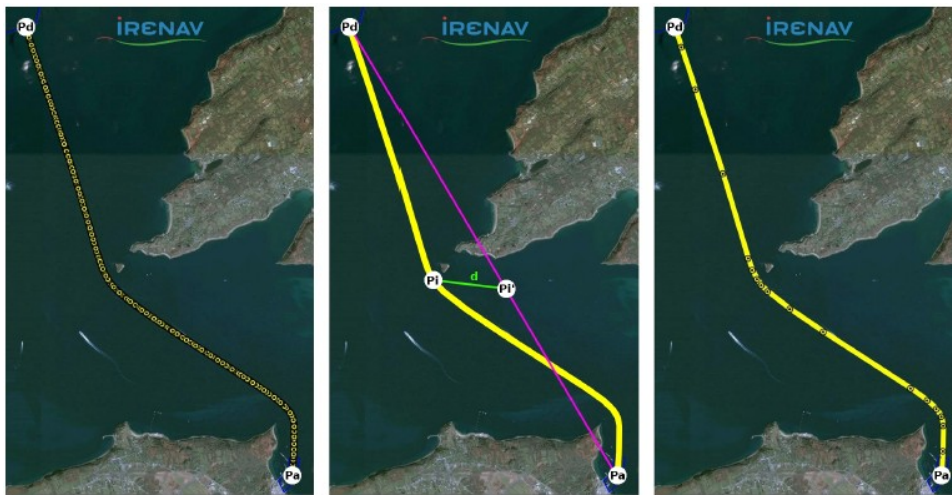


Figure 1.7: Illustration de l'utilisation de l'algorithme de Douglas et Peucker spatio-temporel. Source (Etienne, 2011)

Regroupement des trajectoires

Une fois les données mises en forme, il est possible de réaliser le processus de regroupement de trajectoires (Figure 1.8). Ce processus que l'on retrouve plus généralement dans la littérature sous le terme de *clustering* permet de créer des patrons de normalité qui serviront de structure pour la détection des trajectoires anormales. Différentes techniques peuvent être utilisées pour réaliser ce regroupement. L'objectif n'étant pas de réaliser une description exhaustive de ces dernières, seules les plus importantes seront présentées. Néanmoins, une revue plus complète est notamment disponible dans la thèse de Etienne (Etienne, 2011).

Une première famille d'algorithmes se base sur la technique de "Partitioning Around Medoids" (PAM) (Kaufman and Rousseeuw, 2005). Cette technique s'appuie sur un partitionnement de l'espace dans lequel sont choisis aléatoirement des points de référence nommés *k-medoids*. Les données restantes sont alors appariées de façon récursive à ces points de référence en fonction de leur similarité. À partir de cette technique, différentes extensions ont été proposées comme CLARA (Clustering LARge Applications) qui optimise le calcul des *k-medoids* ou encore CLARANS (Clustering LARge Applications based on RANdomized Search) qui limite le nombre de données testées à chaque itération. Néanmoins, comme le souligne Etienne (Etienne, 2011), l'une des limites de cette approche porte sur le choix du nombre d'objets choisis comme points de référence. En effet, l'efficacité de cette approche dépend de la valeur renseignée par l'utilisateur.

La seconde famille d'algorithmes adopte une approche basée sur une décomposition hiérarchique de l'ensemble des données afin de les représenter sous la forme d'un dendrogramme. La création de celui-ci peut se faire soit par une approche divisive consistant à partir des racines jusqu'aux feuilles ou par une approche agglomérative en partant des feuilles vers les racines. Chacune de ces approches utilise des opérateurs spécifiques permettant de fusionner ou de diviser les ensembles à chaque étape. Contrairement aux algorithmes basés sur la méthode PAM, il n'est pas nécessaire de connaître au préalable le nombre de partitions à obtenir. Néanmoins, une condition d'arrêt du processus de fusion ou de division doit être spécifiée préalablement (Etienne, 2011).

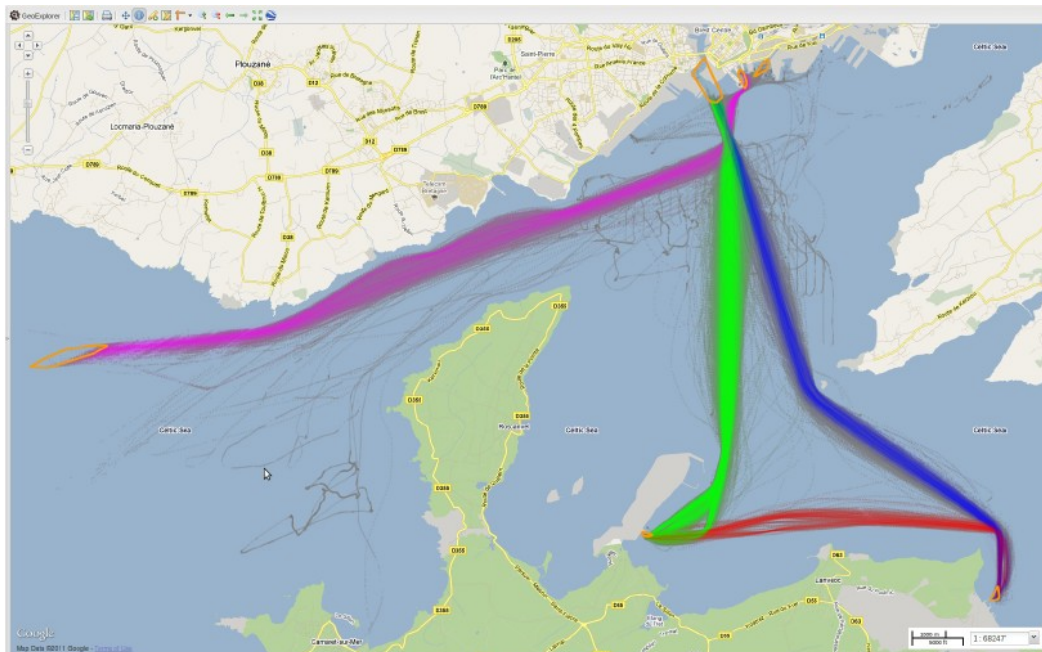


Figure 1.8: Exemple de clusters de trajectoires sur des données maritimes réalisés à l'aide du logiciel M-Atlas. Source : (Etienne, 2011)

Création du patron spatio-temporel

Une fois les trajectoires regroupées, il est nécessaire de modéliser la ou les routes principales. Ces routes, nommées "*path models*" (Morris and Trivedi, 2008) ou aussi "*motion patterns*" (Hu et al., 2006) serviront de patron de référence lors du processus de détection de trajectoires anormales. Deux principales familles de routes peuvent être identifiées. La première considère la trajectoire dans son ensemble, c'est à dire du point de départ à son point d'arrivée. La seconde décompose cette trajectoire en sous-ensemble. En fonction du type choisi, différentes approches et algorithmes existent. Par exemple, Etienne (Etienne, 2011) a proposé l'utilisation du principe des boites à moustaches pour la création de ce patron spatio-temporel (Figure 1.9). Pour cela, quatre étapes sont nécessaires : (1) calcul de la trajectoire médiane, (2) calcul du couloir spatial, (3) calcul du couloir spatio-temporel, (4) calcul statistique des mesures de similarité. Le choix pour l'auteur de se baser sur la trajectoire médiane se justifie par le fait de la diminution des biais par rapport aux trajectoires normales ayant parfois un écart très important.

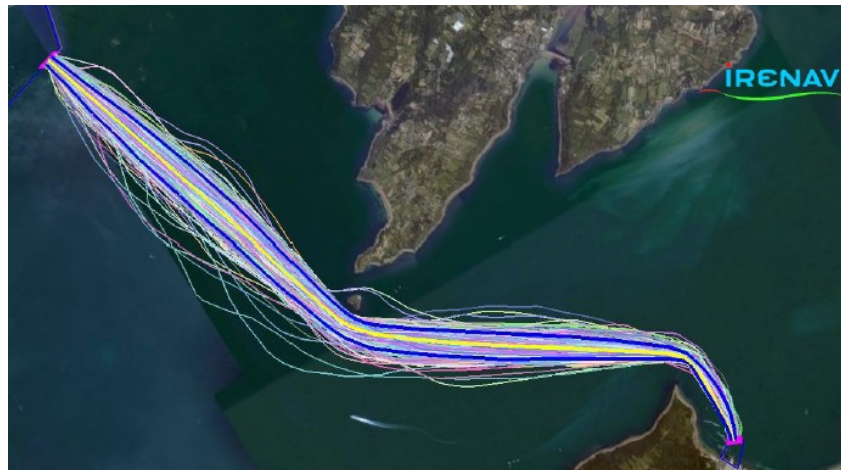


Figure 1.9: Exemple de couloir spatial à partir de différentes trajectoires. Source (Etienne, 2011)

1.2.2.2 Processus d'analyse des trajectoires

Maintenant qu'un patron type de trajectoires a été défini, il est possible de déterminer pour chaque nouvel enregistrement si celui-ci est ou non anormal. L'objectif est d'obtenir au final un indice de similarité qui permettra de catégoriser la différence entre les deux objets comparés. Cette mesure de similarité peut être effectuée par différentes approches comme en analysant les séries temporelles (distance de Minkowski, distance temporelle dynamique, etc.) ou encore en comparant la géométrie des objets. C'est principalement cette seconde approche qui a été adoptée dans ce travail de thèse.

Très utilisée pour la reconnaissance de formes, la comparaison de la géométrie des objets est notamment appliquée dans le domaine du traitement et de l'analyse d'image, de la robotique ou encore de la cartographie. Dans ce cas, il s'agit de calculer la similarité ou au contraire la distance entre deux formes géométriques. Pour cela, différentes métriques au niveau de complexité variable existent. Parmi les plus connus, citons notamment la distance de Hausdorff ou encore la distance de Fréchet.

La distance de Hausdorff (Hausdorff, 1918) recherche pour chaque point de chaque ligne, le point le plus proche de l'autre ligne (Figure 1.10.A). Ainsi, si on note A et B deux éléments de l'espace P, l'expression mathématique de la distance de Hausdorff résultante est la suivante :

$$D_H(L_A, L_B) = \text{Max}(D_1, D_2)$$

sachant que D_1 , et D_2 sont définis de la manière suivante

$$D_1 = \text{Max}_{p_1 \in L_1} (\text{Min}_{p_2 \in L_2} (d(p_1, p_2)))$$

$$D_2 = \text{Max}_{p_2 \in L_2} (\text{Min}_{p_1 \in L_1} (d(p_1, p_2)))$$

Néanmoins, cette distance souffre de ne pas tenir compte de l'ordre des points. Cette limite fait que selon leur configuration deux polygones, même fortement différentes, peuvent avoir un indice de Hausdorff très faible (Figure 1.10.B). Dans ce type de cas, le recours à la distance de Fréchet peut

être une alternative (Alt, 2009).

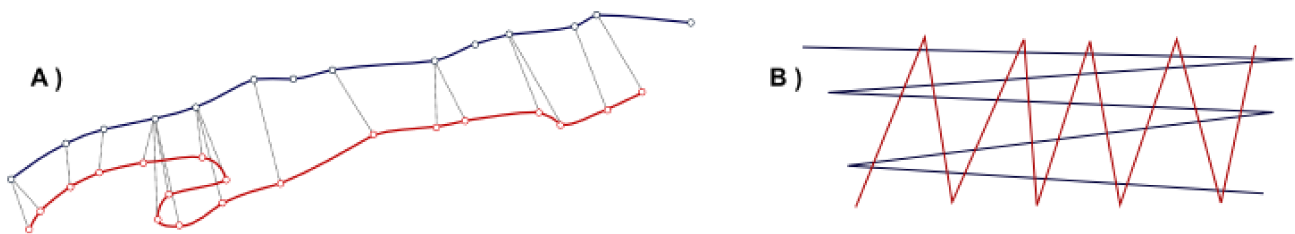


Figure 1.10: Illustration de la distance de Hausdorff.

A) Principe du calcul et d'appariement des points.

B) Exemples de distance de Hausdorff faible malgré une forte différence géométrique

La distance de Fréchet (Fréchet, 1905), permet de calculer la distance maximale entre deux lignes. Son principe est souvent illustré par l'exemple d'un maître promenant son chien en laisse. Chacun suit son propre chemin, s'arrêtant et s'avancant, mais ne pouvant revenir en arrière. La distance de Fréchet est alors représentée par la longueur minimale de laisse nécessaire entre le maître et son chien. Contrairement à la distance de Hausdorff, celle-ci à l'avantage d'être calculée uniquement à partir de couples de points homologues. Néanmoins, et même si des alternatives existent (distance de Fréchet discrète, distance de Fréchet discrète partielle, etc.), elle s'avère complexe à programmer (Devogele, 2009; Etienne, 2011). C'est pourquoi dans le cadre de ce travail de thèse incluant d'importants volumes de données évoluant en temps réel, la distance de Hausdorff a finalement été préférée.

Comme cela a été présenté, différentes méthodes de détection de comportements anormaux dans le domaine maritime existent. Néanmoins, les informations à propos des algorithmes et des méthodes utilisés restent assez rares du fait notamment de leur application directe au niveau militaire (Laxhammar, 2011; Martineau and Roy, 2011). Pour preuve, le premier colloque spécialisé sur ce sujet, nommé Maritime Anomaly Detection (MAD)¹⁵ n'a eu lieu qu'en 2011. Plus d'informations et d'autres méthodes sont notamment disponibles dans les travaux de Johansson et Falkman (Johansson and Falkman, 2007) qui ont proposé l'utilisation des réseaux bayésiens pour la détection de comportements anormaux ou encore ceux de Brax (Brax et al., 2010) avec l'utilisation d'un histogramme multidimensionnel.

1.2.3. Anomalie sémantique

Tout comme la notion de trajectoire sémantique, le concept d'anomalie sémantique consiste à attacher différentes informations (météorologique, géographique, etc.) aux anomalies constatées. La première étape à réaliser est de dresser un catalogue des différentes anomalies potentielles pouvant survenir. Celles-ci doivent être bien évidemment mises en relation avec les éléments contextuels afin d'apporter une meilleure richesse sémantique. Dans le domaine maritime, ce type de travaux est assez rare. À notre connaissance, seul Roy et son équipe ont proposé une description complète d'un système de détection de comportements anormaux basé sur une approche ontologique (Roy, 2008; Roy et al., 2008; Roy and Davenport, 2010). Réalisée dans le cadre d'une étude menée par l'agence R&D du ministère de la défense nationale du Canada (RDDC), Roy propose une taxonomie des situations anormales sous la forme d'une ontologie, une description complète de chaque scénario

¹⁵ <http://mad.uvt.nl/>

1.2. De la normalité aux anomalies sémantiques

potentiel, ainsi qu'un ensemble de règles de détection. Cette taxonomie se divise en deux catégories selon le caractère dynamique ou statique de l'événement constaté.

Les anomalies statiques correspondent par exemple à des incohérences par rapport au numéro IMO ou encore les fréquences radio du navire concerné. Les anomalies dynamiques font référence à la notion de mouvement. Celles-ci ont été subdivisées en deux sous catégories pour différencier les anomalies portant sur la cinématique du navire et les autres. La première (Figure 1.11) nommée "*Dynamic Non-Kinematic Anomaly*" contient six classes principales portant notamment sur les relations avec le port de départ ou d'arrivée, la liste des passagers de l'équipage ou encore sur le type du navire. La seconde (Figure 1.12) nommée "*Dynamic Kinematic Anomaly*" contient cinq classes principales directement relatives à la cinématique du navire comme sa vitesse, ses manœuvres ou encore sa localisation. Dans le cadre de ce travail de thèse, c'est essentiellement cette partie qui a été étudiée.

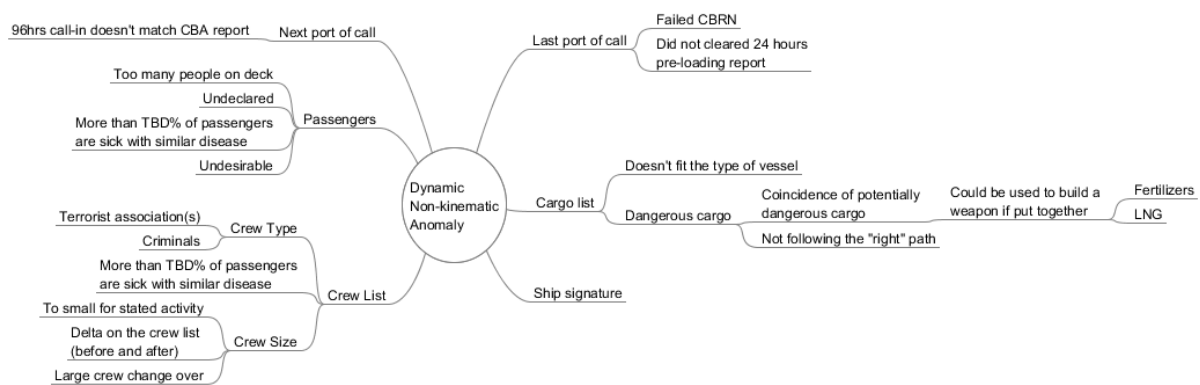


Figure 1.11: Taxonomie des anomalies statiques. Reproduction d'après (Roy, 2008)

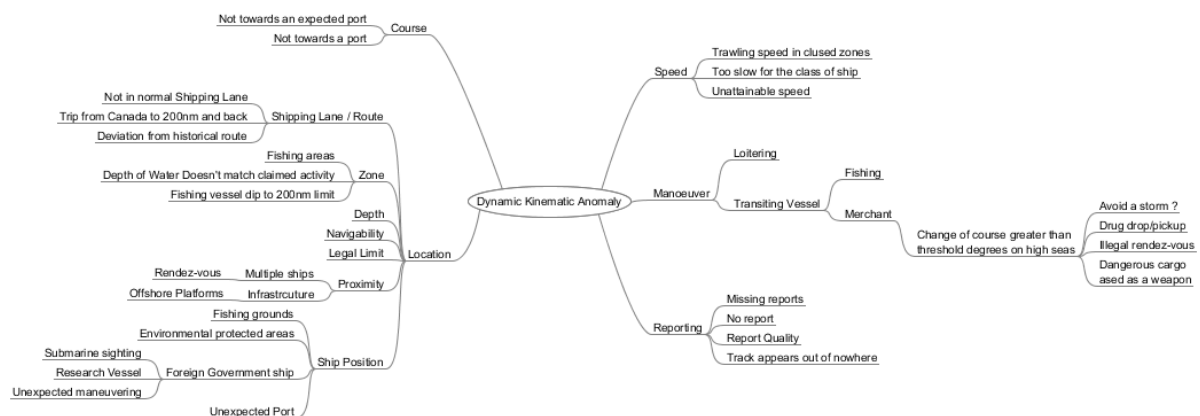


Figure 1.12: Taxonomie des anomalies dynamiques. Reproduction d'après (Roy, 2008)

Au final, cette conceptualisation des anomalies maritimes potentielles a conduit à la création d'une ontologie de haut niveau contenant plus de 350 classes, 100 propriétés, 700 restrictions et 600 instances. Pour cela, les auteurs se sont appuyés notamment sur les connaissances des experts des Forces Canadiennes¹⁶ ainsi que sur différentes ontologies déjà existantes comme JC3IEDM¹⁷ et

¹⁶ <http://www.forces.gc.ca/site/index.asp>

¹⁷ <http://vistology.com/ont/2007/JC3IEDM3.1a/JC3IEDM3.1a.owl>

Chapitre 1. Trajectoires et comportements sémantiques

Ucore¹⁸ (Roy and Davenport, 2010).

Enfin, à partir de cette modélisation, les auteurs ont proposé la création d'un environnement facilitant l'exploitation de différents formalismes de représentation des connaissances (Logiques de description, règles, etc.) nommé ISFAR (Inference of Situational Facts through Automated Reasoning). L'une des applications directe de ISFAR est le prototype ARMAD (Automated Reasoning for Maritime Anomaly Detection) (Roy and Davenport, 2010). Les résultats obtenus ainsi qu'une description plus complète du système seront présentés dans le paragraphe suivant.

Malgré l'intérêt de ces travaux, l'approche présentée est néanmoins peu décrite. En effet, les différentes publications ne présentent que des extraits de l'ontologie et celle-ci n'est pas disponible en libre accès. Il est donc difficile de comprendre comment elle a été construite afin de s'en inspirer. Cela reflète d'ailleurs le constat réalisé par Laxhammar (Laxhammar, 2011), qui souligne le fait que malgré les travaux existants sur la détection de trajectoires anormales dans le domaine maritime peu d'entre eux font l'objet d'une véritable valorisation scientifique. Selon lui, cela s'explique notamment par l'enjeu sur lesquelles portent ces applications ainsi que par leur caractère militaire. Mais, la tenue en 2011 du premier colloque sur la détection d'anomalie maritime¹⁹ (MAD – Maritime Anomaly Detection) pourrait permettre à l'avenir une plus grande diffusion des avancées scientifiques.

18 <https://metadata.ces.mil/ucore/index.html>

19 <http://mad.uvt.nl/>

1.3. D'une trajectoire anormale à un comportement

Bien que la détection d'une anomalie soit une étape importante, celle-ci ne permet pas pour autant d'expliquer totalement la situation maritime (Ekman and Holst, 2008). En effet, une anomalie est un événement ponctuel à la fois temporairement et spatialement. De plus, des événements d'apparence totalement normaux peuvent également participer à la compréhension de cette situation. Par exemple, afin de dissimuler une activité illicite, les auteurs veilleront justement à ne commettre aucune infraction. Il est difficile dans ce cas de ne se baser que sur la détection d'anomalies et il apparaît nécessaire d'apporter un niveau supplémentaire dans l'analyse (van Hage et al., 2011).

Pour illustrer cela, prenons l'exemple d'un navire qui s'arrêterait par une nuit de pleine lune à proximité de la côte d'un pays connu pour être un lieu de passage pour le trafic de drogue. Après quelques heures, ce même navire quitte cette zone pour se rendre à vive allure dans le port d'un autre pays. Comme l'illustre ce scénario, ce n'est pas l'analyse d'une anomalie unique qui permet de comprendre cette situation, mais l'enchaînement des actions entre elles. C'est pourquoi, cette thèse se propose d'enrichir les potentialités d'analyse des systèmes de surveillance maritime par la notion de comportement. Un comportement se définit alors par toutes les interactions potentielles qu'a le navire avec son environnement. Pour cela il est nécessaire de disposer d'un éventail d'information le plus complet possible. Mais la modélisation de ce comportement, tout comme l'intégration des différentes informations est une tâche complexe. De ce fait, il est nécessaire de disposer de formalismes de représentation suffisamment riches permettant de traduire la représentation du monde réel. A l'heure actuelle, c'est dans le domaine des ontologies que les principaux travaux sont réalisés. Bien qu'il n'existe encore aucune ontologie générique portant sur la modélisation de comportements anormaux de navires deux projets peuvent être signalés. Le premier nommé Simple Event Model s'attache à la description des événements. Le second, qui s'intègre au prototype ISFAR, est spécifique au domaine maritime et porte sur la modélisation des comportements. Ces deux projets sont décrits en détails dans les paragraphes suivants.

1.3.1. Simple Event Model (SEM)

Simple Event Model (SEM) fournit l'architecture ontologique nécessaire pour la description d'événements²⁰ (van Hage et al., 2009, 2011). L'objectif de SEM est de faciliter l'intégration des connaissances à partir de données provenant de différentes sources (internet, capteurs, documents, etc.). L'un des avantages de SEM est qu'il a été conçu pour s'adapter facilement à des données qui peuvent être incomplètes ou partielles. Pour cela, les auteurs ont proposé une architecture composée de trois strates principales : *Core Classes*, *Types* et *Constraints* (Figure 1.13).

1.3.1.1 Modélisation ontologique de SEM

Au sommet de cette architecture, on retrouve les classes formant le cœur de l'ontologie SEM. Quatre classes ont été définies et permettent de modéliser tout événement pouvant se produire. Celles-ci sont :

- *sem:Event* (l'événement), permet de spécifier ce qui est en train de se passer. Celle-ci est reliée aux trois classes suivantes par les propriétés *sem:hasActor* pour indiquer le ou les participants en cause, *sem:hasPlace* afin de localiser le lieu et *sem:hasTime* pour préciser le

²⁰ Disponible à l'adresse suivante : <http://semanticweb.cs.vu.nl/2009/11/sem/>

moment.

- *sem:Actor* (le ou les participants), en plus d'indiquer les participants (humains ou non) d'un événements, cette classe possède un niveau d'abstraction supplémentaire permettant de modéliser des cas où l'objet en cause serait une partie de la classe *sem:Actor*. Pour illustrer cela, prenons l'exemple d'un conteneur chargé à bord d'un porte-conteneur. Si celui-ci passe par dessus bord, ce conteneur peut être alors défini en tant que *sem:Object* sous classe de *sem:Actor*.
- *sem:Place* (le lieu), elle indique l'endroit où s'est passé l'événement. Cet endroit peut être un lieu précis défini par ses coordonnées géographiques ou alors être complètement symbolique.
- *sem:Time* (le moment), tout comme *sem:Place*, il est possible de définir aussi bien un instant précis ou alors symbolique.

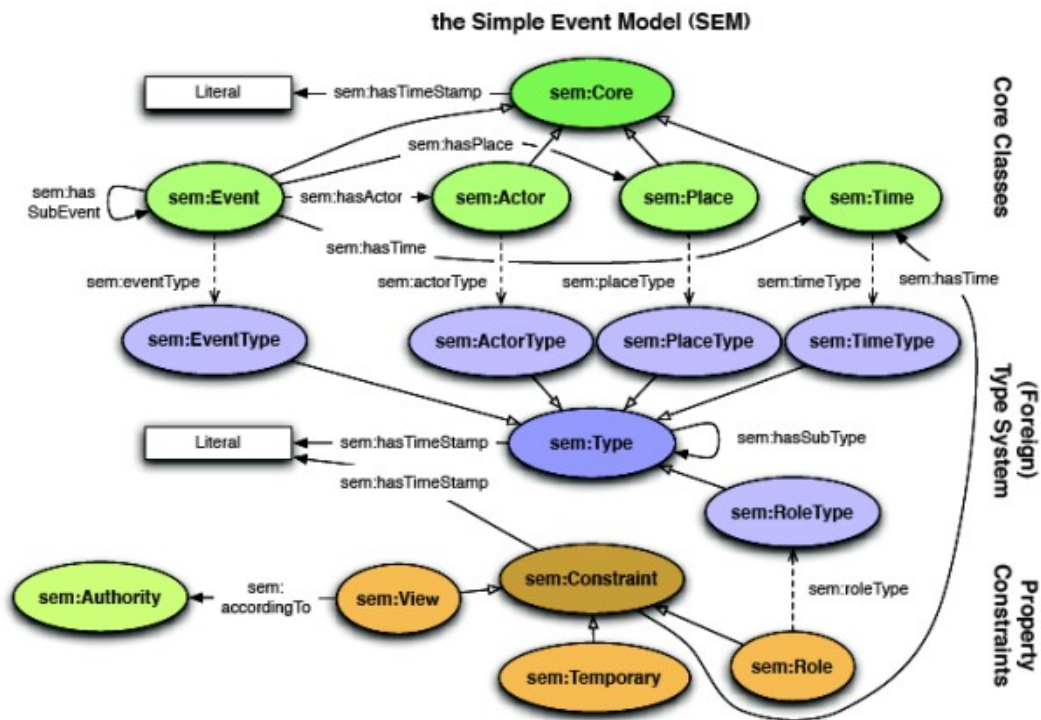


Figure 1.13: Principales classes du Simple Event Model. Source : (van Hage et al., 2011)

Ces classes sont reliées entre elles par différentes propriétés. Trois grand types peuvent être distinguées celles-ci sont *sem:eventProperties*, *sem:type* et d'autres propriétés mineures telles que *sem:accordingTo* (Figure 1.14).

Les propriétés de type *sem:eventProperties* permettent de relier la classe *sem:Event* aux autres classes principales. Celles-ci sont par exemple *sem:hasPlace*, *sem:hasActor*, *sem:hasTime*. La propriété *sem:type* fournit les mécanismes nécessaires afin de préciser le type de chaque classe. Cette modélisation permet de réduire la complexité de raisonnement, car comme chaque *Core Classes* est déjà liée par un lien *sem:type* il n'est pas nécessaire pour le moteur d'inférence de le déduire par subsomption.

1.3. D'une trajectoire anormale à un comportement

La dimension temporelle est spécifiée grâce à l'une des sept propriétés *sem:hasTimeStamp*. Celles-ci peuvent aussi bien représenter un événement ponctuel (*sem:hasTimeStamp*) qu'un intervalle (*sem:hasBeginTimeStamp*, *sem:hasEndTimeStamp*) où même un événement dont la durée ou l'heure n'est connue que de manière approximative (*sem:hasEarliestBeginTimeStamp*, *sem:hasLatestBeginTimeStamp*).

Toutes les propriétés de l'ontologie SEM peuvent être précisées à l'aide de différentes contraintes. Ces contraintes, représentées comme des classes sont : *sem:Role*, *sem:Temporary*, *sem:View*. La première, *sem:Role*, définit le rôle joué par un l'individu d'une classe dans le contexte d'un événement spécifique. La seconde, *sem:Temporary*, spécifie l'intervalle temporel dans lequel s'inscrit la propriété. Et enfin, *sem:View*, permet de préciser le ou les points de vue des différents acteurs impliqués dans un événement.

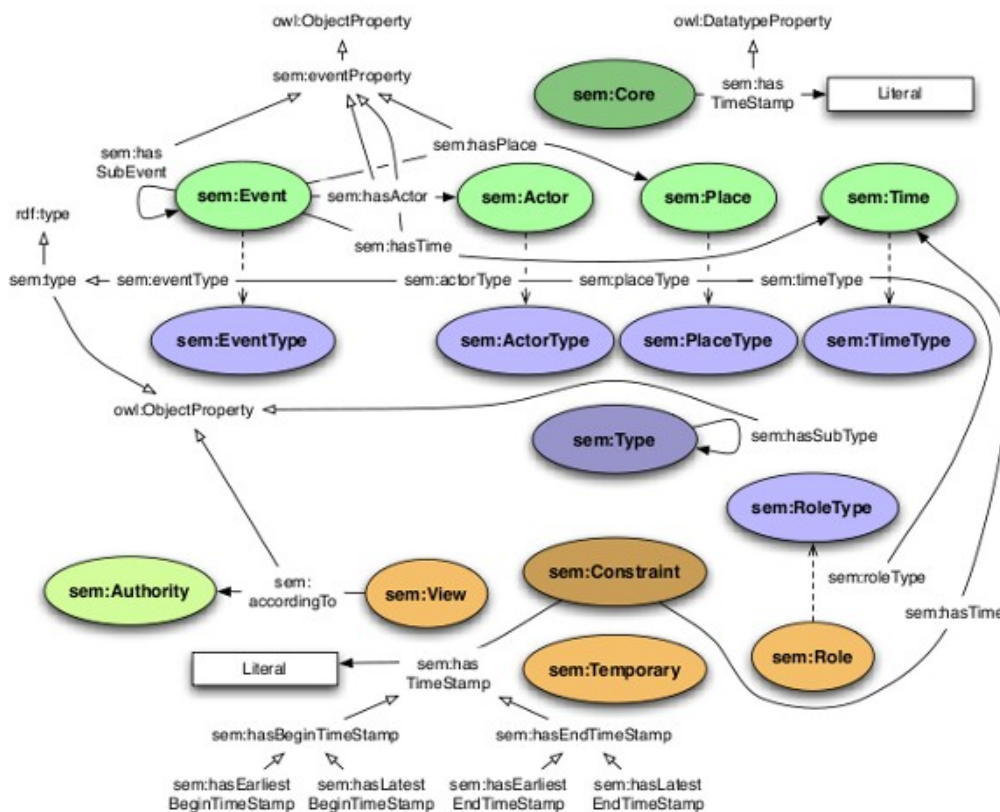


Figure 1.14: Les différentes propriétés du Simple Event Model. Source (van Hage et al., 2011)

1.3.1.2 Caractérisation des comportements

La modélisation présentée a ensuite été intégrée au sein d'une application permettant de caractériser les comportements des navires à partir de leur trajectoire sémantique et des événements identifiés (ralentissement, accélération, etc.). Pour cela cinq étapes principales sont nécessaires (Figure 1.15).

Tout d'abord, il s'agit bien évidemment des données sources. La position des navires a été obtenue à partir de leur signal AIS. Mais pour le moment, il ne s'agit encore que de simples positions. Il est donc nécessaire de dériver des trajectoires à partir de celles-ci. L'algorithme de compression

Chapitre 1. Trajectoires et comportements sémantiques

Piecewise Linear Segmentation a alors été utilisé (Andrienko and Andrienko, 2007). Celui-ci permet de réduire le volume de données à traiter et facilite par la suite les procédures de raisonnement. Au final, seuls les positions significatives (Stop, Move) telles que définies par Spaccapietra (Spaccapietra et al., 2008) sont enregistrées dans une base de données en tant que trajectoires. Le passage à la dimension sémantique est ensuite réalisé grâce à D2RQ Server²¹. Cette application permet de transformer les données de trajectoires stockées dans une base relationnelle en triplet RDF. Chaque enregistrement de la base est alors traduit sous la forme d'un segment sémantique correspondant à une instance de *sem:Event*. La cinématique du navire est spécifiée grâce à *sem:eventType* qui peut alors prendre comme valeur *Stopped* ou *Moving*. Des informations additionnelles peuvent également enrichir cette trajectoire. Celles-ci proviennent de différents sites internet spécialisés tel que marine trafic²². Grâce au numéro MMSI il est alors possible de déterminer le type du navire, sa taille ou encore sa vitesse. Grâce à l'ontologie GeoNames²³, il est également possible d'apporter des informations complémentaires par rapport à la position initiale enregistrée. Cela peut être intéressant afin de savoir notamment si le navire est à proximité d'une côte ou d'un port. Pour cela, une bibliothèque de programmation nommée SWI-Prolog Space Package²⁴ a été développée par les auteurs. À partir de ces trajectoires sémantiques, il est maintenant possible de déduire des comportements. À cet effet, une base de règles a été créée. Mais peu d'informations sont disponibles sur celles-ci. Le seul exemple donné, porte sur la caractérisation du trajet d'un ferry.

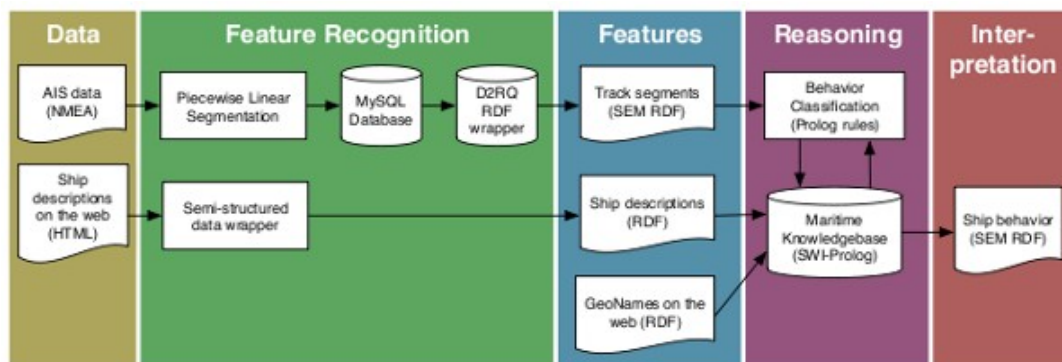


Figure 1.15: Diagramme des flux du système SEM. Source (van Hage et al., 2011)

L'approche Simple Event Model présentée par van Hage (van Hage et al., 2011) est particulièrement intéressante. En effet celle-ci offre un cadre conceptuel souple permettant la description d'événements relatifs aussi bien à un navire qu'à sa cargaison. L'architecture proposée s'appuie à la fois sur les potentialités de stockage des bases de données et sur la richesse sémantique qu'offre les ontologies. Néanmoins, peu d'informations concernant les procédures de raisonnement, la volumétrie des données ainsi que les performances du système sont disponibles.

21 <http://d2rq.org/d2r-server>

22 <http://www.marinetraffic.com/ais/>

23 <http://www.geonames.org/>

24 <http://www.swi-prolog.org/pldoc/package/space.html>

1.3.2. ISFAR

Tout comme la détection d'anomalies dans le domaine maritime, les exemples d'applications portant sur la notion de comportement sont très rares. Là encore, à notre connaissance, l'essentiel des propositions à été réalisé par Roy et son équipe (Roy and Davenport, 2010). Pour cela, les auteurs ont développé un prototype d'analyse de comportements anormaux nommé ISFAR (Figure 1.16). Celui-ci se base sur une ontologie comportant près de 350 classes et 600 instances. Le moteur d'inférence Pellet a été utilisé à la fois lors de la phase de développement pour vérifier la consistance de l'ontologie mais aussi lors de la phase de production pour réaliser les procédures d'inférences.

1.3.2.1 Architecture du système

Le système proposé fonctionne en trois temps (Figure 1.16). Tout d'abord, l'ontologie du domaine maritime est fusionnée avec les données (faits) provenant du *Situation Facts Management Service* ce qui conduit à la création d'une base de connaissances (Temporary Augmented Ontology). À partir de celle-ci des procédures de contrôle sont lancées par le moteur d'inférence qui détermine alors en fonction des règles spécifiées les navires suspects. Ces navires suspects peuvent être ensuite interrogés grâce au service *Ontology Query Service*. L'une des parties essentielles du système proposé est constituée par le moteur d'inférence. C'est pourquoi les auteurs ont effectué une série de tests sur les différentes solutions existantes aussi bien Open Source que commerciales. Sur la quarantaine de moteurs testés, c'est Pellet²⁵ qui a été choisi en raison notamment de sa documentation riche, de son interface de programmation (API) ou encore de ses temps d'exécution.

Pour l'intégration des données et l'utilisation du service, deux modes ont été proposés. Le premier, dit continu, ajoute en temps réel les nouveaux faits à la base de connaissances. Néanmoins, même sur une ontologie de petite ou de moyenne taille la procédure de classification et de création nécessite un temps considérable. C'est pourquoi, le second mode, dit *batch* c'est à dire fonctionnant de manière asynchrone, a été préféré. Dans celui-ci, les nouveaux faits sont ajoutés selon un intervalle de temps régulier.

25 <http://clarkparsia.com/pellet/>

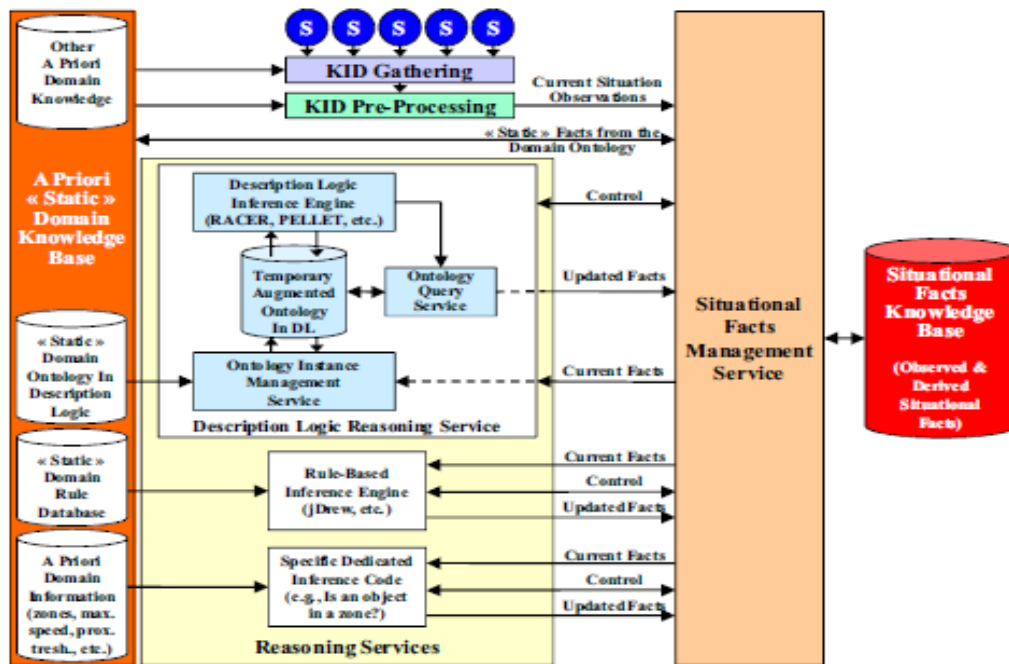


Figure 1.16: Architecture du prototype ARMAD développé par Roy.
 Source : (Roy and Davenport, 2010).

1.3.2.2 Performances et limites du système

Concernant les performances du système proposé, celui-ci a été testé en utilisant plus d'une centaine de faits correspondant aux navires, leur position, leur parcours et les dossiers s'y rattachant. Bien que les résultats soient concluants, les auteurs notent néanmoins des problèmes de performance quand le nombre de navires augmente. Pour les auteurs, ces résultats s'expliquent notamment par la modélisation complexe des classes en cas d'absence de règles, l'impossibilité d'effectuer certains raisonnements ou encore sur l'impossibilité de gérer certains cas intégrant une dimension temporelle variable.

Au-delà de la modélisation, certaines limites sont également à mettre en relation avec le moteur d'inférence Pellet. En effet celui-ci est basé sur des algorithmes de vérification de satisfiabilité à base de tableaux (*tableau-based algo*), ce qui signifie que pour chaque restriction, il est nécessaire d'ajouter au graphe le nœud correspondant. Ainsi, si un grand nombre de restrictions a été défini cela entraîne la création d'un nombre de nœuds exponentiel. Or l'ontologie développée contient plus de 400 restrictions ce qui pourrait expliquer les ralentissements constatés.

Différentes solutions ont été imaginées afin d'améliorer ces performances. Deux d'entre elles consistant au passage de OWL-DL à OWL Full et la séparation en plusieurs sous-ontologies ont été explorées mais abandonnées par la suite en raison d'absence de résultats significatifs. Par contre, la restructuration de l'ontologie au regard des limites observées a permis d'améliorer considérablement les performances du prototype. Cette restructuration orientée "*evidence centered*" a notamment été rendue possible par l'intégration de built-in SWRL. De plus, le passage à la nouvelle version de Pellet a permis de diviser considérablement les temps d'inférences. Par exemple, la classification de deux navires prenait plus de 400 secondes avec la première version de Pellet et plus que 2,5 secondes avec la nouvelle (Roy and Davenport, 2010).

Conclusion du chapitre

Bien qu'essentielle, la position et par extension la trajectoire d'un objet mobile sont insuffisantes pour analyser son comportement. L'intégration d'informations complémentaires décrivant notamment le contexte dans lequel il évolue apparaît comme nécessaire. Ces informations sont alors reliées à l'objet mobile permettant ainsi d'aboutir à une trajectoire sémantique. Cette trajectoire sémantique peut être ensuite analysée afin d'identifier des anomalies ou des comportements suspects.

Le cadre structurant cette trajectoire sémantique est constitué de trois principales ontologies : l'ontologie géométrique, l'ontologie géographique et l'ontologie du domaine. Le cœur de ce travail de thèse a porté sur la modélisation de cette dernière ainsi que sur l'intégration en un ensemble cohérent des différentes ontologies. Gérées indépendamment, elles sont ensuite fusionnées afin d'offrir aux experts de la surveillance maritime l'ensemble des notions nécessaires à la description d'un comportement.

Mais se pose alors la question de la modélisation de ces informations. Comment doivent-elles être intégrées, comprises et analysées ? Comment créer des notions partagées par l'ensemble des acteurs ? Ces questions trouvent leur réponse dans le domaine de l'ingénierie de la connaissance et plus particulièrement dans celui des ontologies.

Le chapitre suivant décrit les différentes facettes des ontologies à la fois comme objet conceptuel et composant informatique. Les méthodologies de construction existantes ainsi que les formalismes et langages disponibles sont également présentés.

CHAPITRE 2. FORMALISATION DE LA CONNAISSANCE AU SEIN DES ONTOLOGIES

Chapitre 2. Formalisation de la connaissance au sein des ontologies.....	57
2.1. L'ontologie en Intelligence Artificielle.....	59
2.1.1. <i>Origine et définitions.....</i>	59
2.1.2. <i>Les composants d'une ontologie.....</i>	60
2.1.3. <i>Le rôle des ontologies dans les systèmes d'information.....</i>	61
2.1.4. <i>Typologie des ontologies.....</i>	62
2.2. Méthodologies pour la construction d'ontologies.....	64
2.2.1. <i>Principes généraux guidant la construction d'une ontologie.....</i>	64
2.2.2. <i>La méthode On-To-Knowledge</i>	67
2.2.3. <i>La méthode Methontology.....</i>	68
2.3. Formalismes et Langages de représentation d'ontologies.....	70
2.3.1. <i>Formalismes de représentation.....</i>	70
2.3.2. <i>Les langages ontologiques.....</i>	75
2.4. Conception et exploitation d'ontologies.....	84
2.4.1. <i>Éditeur d'ontologies.....</i>	84
2.4.2. <i>Interfaces de programmation pour la manipulation d'ontologies.....</i>	87
2.4.3. <i>Raisonnement à partir d'une ontologie</i>	88

Introduction du chapitre

Modéliser la connaissance nécessite de disposer de structures sémantiques et de formalismes de représentation permettant de traduire la complexité de la pensée humaine. À l'heure actuelle, c'est dans le domaine des ontologies et plus particulièrement dans celui du Web Sémantique que les avancées et les travaux effectués sont les plus importants.

Une ontologie peut être vue alors comme un artefact cognitif permettant une conceptualisation et une exploitation partagée de la connaissance. Celle-ci est composée d'un vocabulaire du domaine dans lequel sont précisés le sens des termes et les relations entre les différentes notions. En plus de ses capacités descriptives s'ajoutent également la possibilité d'effectuer des raisonnements à partir des notions définies (Figure 2.1).

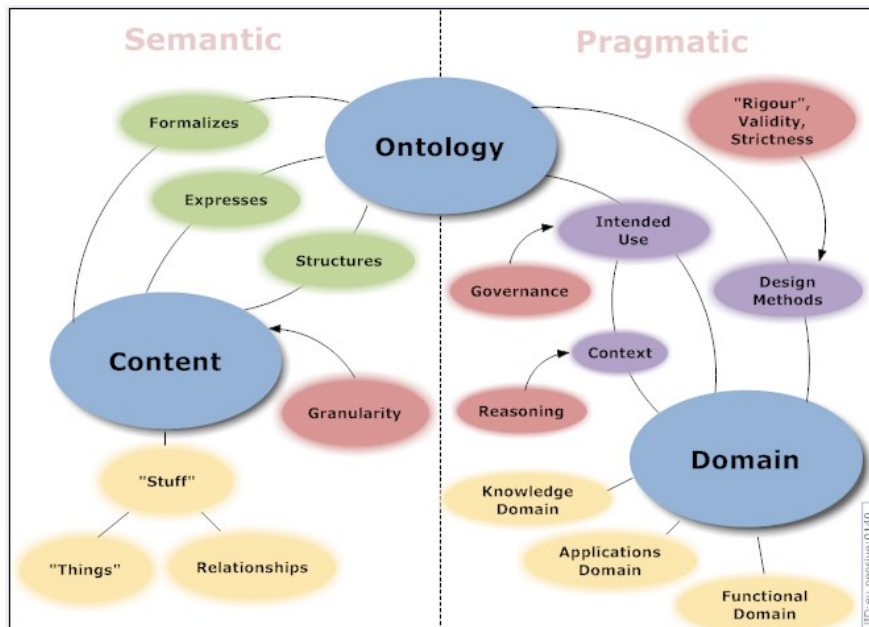


Figure 2.1: Les différentes dimensions d'une ontologie.

Source (Semwebtech : <http://keet.wordpress.com>)

Dans ce chapitre un retour historique sur la notion d'ontologie ainsi qu'une présentation des principaux concepts sont effectués (§2.1). Les méthodologies existantes et plus particulièrement On-To-Knowledge et Methontology sont ensuite analysées (§2.2). Puis, les formalismes de représentation ainsi que les langages ontologiques actuels sont décrits. Enfin, les principaux outils et applications permettant de construire et exploiter une ontologie sont présentés (§2.4).

2.1. L'ontologie en Intelligence Artificielle

Situées au carrefour de plusieurs grands domaines informatiques (ingénierie des connaissances, intelligence artificielle, etc.) mais aussi en psychologie ou en cognition, les ontologies sont aujourd'hui considérées comme l'un des paradigmes clés pour l'interopérabilité sémantique. Ainsi, de l'intelligence artificielle au web sémantique l'information n'est plus considérée uniquement au travers du prisme de la donnée, mais aussi au travers de la représentation des connaissances.

2.1.1. Origine et définitions

2.1.1.1 Contexte d'apparition des ontologies

Bien qu'originellement rattachée au domaine de la philosophie²⁶, la notion d'ontologie est considérée dans ce manuscrit sous l'angle de l'Intelligence Artificielle (IA). Conçues comme une réponse aux problèmes posés par l'intégration des connaissances au sein des systèmes informatiques, les ontologies apparaissent comme une clé essentielle pour la manipulation automatique de l'information au niveau sémantique.

Historiquement, l'origine des ontologies remonte aux années 80 lors de la réalisation des premiers systèmes experts. Les tentatives de modélisation des processus cognitifs, sous la forme de bases de connaissances, s'avèrent être un processus complexe nécessitant un temps considérable (Schreiber et al., 1993). Dès lors, afin de favoriser le partage des connaissances, il apparaît nécessaire d'implémenter de nouveaux formalismes capables de connaître à la fois les termes utilisés, mais aussi la sémantique associée. De ce fait, et notamment sous l'impulsion d'un groupe de chercheurs américains participant au projet Knowledge Sharing Effort²⁷, une nouvelle forme de représentation des connaissances répondant aux contraintes observées a été définie : l'ontologie (Psyché, 2007; Mondary, 2011).

2.1.1.2 De multiples définitions pour une même notion

D'une vision purement philosophique à son application en intelligence artificielle (IA), de multiples définitions et interprétation de la notion d'ontologie ont été proposées. Ainsi, Guarino note que ce terme est rattaché à sept différentes notions (Guarino and Giaretta, 1995). Originellement la première définition, et aussi la plus couramment citée, est celle donnée par Gruber (Gruber, 1995) pour qui une ontologie est :

« une spécification explicite d'une conceptualisation »

Néanmoins, de nombreux auteurs lui reprochent son caractère trop vague et générique. C'est pourquoi c'est celle donnée par Studer (Studer et al., 1998) qui est habituellement préférée. Celle-ci ajoute à la définition de Gruber la notion de partage. Une ontologie se définit alors comme :

« une spécification formelle et explicite d'une conceptualisation partagée »

Cette définition fait ressortir trois notions particulièrement importantes dans le domaine des ontologies :

- **Formel** : signifie que la conceptualisation et la représentation du domaine doivent être

²⁶ Doctrine ou théorie de l'être en tant qu'être (Encyclopaedia Universalis)

²⁷ <http://www-ksl.stanford.edu/knowledge-sharing/papers/kse-overview.html>

standardisées et utilisables par un système informatique

- **Explicite**: spécifie que les concepts utilisés tout comme les contraintes sont définis de façon déclarative
- **Conceptualisation** : souligne le fait qu'une ontologie n'est qu'une abstraction du monde réel et que les termes utilisés ainsi que leurs relations doivent être décrits sans ambiguïté.
- **Partage** : implique que les ontologies favorisent une connaissance consensuelle

D'une manière plus formelle, une ontologie est constituée d'un vocabulaire spécifique permettant une description du domaine sous la forme d'un graphe constitué de concepts et de relations. Celui-ci peut alors être décrit de la manière suivante : $O:\{C, R, HC, rel, A0\}$ où C est un ensemble de concepts, R un ensemble de relation entre ces concepts, HC une hiérarchie entre ces concepts, rel une fonction spécifiant la relation de r et enfin $A0$ un ensemble d'axiomes exprimé dans un langage ontologique (Baglioni et al., 2011).

2.1.2. Les composants d'une ontologie

Comme cela a été mentionné, les ontologies fournissent un vocabulaire du domaine et définissent, de façon plus ou moins formelle, le sens des termes et des relations entre ces derniers. Pour cela, les connaissances intégrées dans une ontologie sont formalisées en s'appuyant sur cinq types de composants : les concepts, les relations, les fonctions, les instances et les axiomes (Gruber, 1993) :

- **Concept** : terme que l'on retrouve également dans la littérature sous le nom de classe, il fait référence à la représentation d'un objet matériel, d'une notion ou d'une idée dans l'ontologie. Il s'agit, d'abstractions pertinentes d'un fragment du monde réel en fonction d'un domaine d'application. Un concept peut être classé selon plusieurs dimensions : le niveau d'abstraction (concret ou abstrait), l'atomicité (élémentaire ou composé) et le niveau de réalité (réel ou fictif) (Gómez-Pérez, 1999). Selon Uschold (Uschold and King, 1995), un concept se compose de trois parties : (1) un ou plusieurs termes, (2) une notion et (3) un ensemble d'objets. Le ou les termes permettent d'identifier le concept. La notion, aussi appelée intention du concept, désigne la sémantique du concept défini au travers de ses propriétés et de ses attributs. Enfin, l'ensemble d'objets forme l'extension du concept, il s'agit de toutes les instances du concept. Lors de la création d'une ontologie, le problème qui se pose est la sélection de ces concepts. Quels sont ceux à intégrer ? Comment définir les hiérarchies ? Comment surmonter la variabilité des représentations ? Pour répondre à ces questions, Bachimont (Bachimont, 2000) propose de recourir à une normalisation sémantique des concepts en s'appuyant sur le paradigme différentiel. Celui-ci permet d'éviter toute ambiguïté de sens en définissant chacun des concepts en fonction des quatre principes différentiels que sont : (1) le principe de communauté avec le père, (2) le principe de différence avec le père, (3) le principe de différence avec les frères et (4) le principe de communauté avec les frères. Enfin, un concept se définit également au travers du lien qu'il peut entretenir avec les autres concepts de l'ontologie. De ce fait, Guarino (Guarino and Welty, 2000) propose de qualifier un concept en se basant sur des propriétés formelles de rigidité, d'identité, d'unité et de dépendance.
- **Relations** : Dans une ontologie, les relations correspondent aux types d'interaction possibles entre les concepts d'un domaine. De ce fait, elles se définissent à la fois par leur signature mais aussi par le contenu sémantique entre les concepts qu'elles unissent (Bachimont, 2000). Par exemple, deux relations peuvent porter sur deux mêmes concepts (signature sémantique égale) sans toutefois avoir le même sens (contenu sémantique différent). Dans une ontologie, les relations permettent ainsi de structurer les connaissances et de définir les

interrelations entre les concepts. Par exemple si l'on souhaite définir une relation de spécialisation entre les concepts d'un domaine, on utilise pour cela une relation de type *sous-classe-de* (*subClassOf*). Ces relations peuvent également être utilisées afin de faire référence à des propriétés spécifiques afin d'exprimer une valeur qui peut être textuelle ou encore algébrique. Ainsi, deux grands types de relations peuvent être distingués (Hernandez, 2005) : les relations taxonomiques et les relations associatives. Les premières, également appelées subsomption ou encore relation de spécificité/généricité permettent d'organiser hiérarchiquement un ensemble de concepts. Les secondes désignent toutes les relations entre les concepts qui ne sont pas des relations de taxonomiques. Ces relations peuvent être enrichies par des notions de symétrie, de transitivité ou encore de non-réflexivité.

- **Fonctions** : ce sont un cas particulier de relations dans lesquelles un élément est défini en fonction des éléments précédents.
- **Instances** : elles sont utilisées pour représenter les individus d'une ontologie. Ces individus correspondent à une instance concrète de la classe à laquelle ils appartiennent. Dans le cas d'une ontologie contenant des instances, celles-ci devient alors une base de connaissances.
- **Axiomes** : sont des assertions toujours vraies à propos des abstractions (concepts et relations) du domaine modélisé. Ils permettent de combiner des concepts, des relations et des fonctions pour définir des règles d'inférence. L'ajout d'axiomes dans une ontologie peut avoir notamment pour objectif de définir la signification des composants, de restreindre la valeur des attributs ou encore de vérifier la validité des informations spécifiées (Hernandez, 2005)

2.1.3. Le rôle des ontologies dans les systèmes d'information

Loin de n'être qu'un objet purement informatique, les ontologies servent de cadre unificateur et de réceptacle de connaissances. Comme le souligne Gandon (Gandon, 2008) celles-ci interviennent à différents niveaux dans un système d'information :

« Une ontologie informatique offre un cadre unificateur et fournit des primitives améliorant la communication entre les personnes, entre les personnes et les systèmes, et entre les systèmes »

Dans cette perspective, l'ontologie est au cœur du système d'information (*Ontology Driven Informatic System*). Plusieurs points particulièrement importants sont à souligner. Tout d'abord, les ontologies offrent un cadre unificateur. Comme cela a déjà été abordé, une ontologie spécifie un langage commun, un corpus de connaissance à propos d'un domaine donné. Cela permet ainsi d'éliminer toute ambiguïté à propos d'un terme donné et favorise donc la communication. Cette communication peut se faire entre les personnes, entre les personnes et le système, mais aussi entre les systèmes. A cette amélioration de la communication, il est également nécessaire d'ajouter les potentialités d'inférence informatique ainsi que celui de la réutilisation des connaissances. De ce fait, l'intérêt des ontologies peut être regroupé en fonction des trois points présentés ci-dessous (Hepp, 2008) :

1. Communication

- entre humains,
- entre systèmes informatiques,
- entre humains et systèmes informatiques.

2. Inférence informatique
 - pour la représentation et la manipulation des informations,
 - pour l'analyse des structures, algorithmes, des entrées/sorties d'un système.
3. Réutilisation de la connaissance
 - pour la structuration et l'organisation d'un domaine.

Plus concrètement, dans un système informatique, les ontologies peuvent être utilisées à différents niveaux. Basé sur son expérience, (Guarino, 1998) en dénombre sept :

- la spécification et à l'analyse des besoins du système,
- la maintenance du système faisant office de documentation ou en permettant la vérification d'incohérences,
- la coopération et au partage en tant que format d'échange,
- la recherche d'informations en servant de base d'index ou de métadonnées,
- l'interopérabilité entre diverses sources de données hétérogènes,
- la compréhension du schéma conceptuel et du vocabulaire du système à travers sa visualisation,
- l'exécution et au traitement de requêtes exprimées en langue naturelle.

Ainsi, l'ontologie n'a pas comme vocation unique d'être seulement un objet informatique mais plutôt un support de connaissance entre toutes les composantes d'un système. De ce fait, elle peut être appliquée à de nombreux domaines. Par exemple, parmi les applications les plus connus citons notamment le projet Cyc²⁸ qui vise à développer une ontologie globale pour l'Intelligence Artificielle ; le projet SUMO²⁹ qui propose une collection d'ontologies couvrant des thèmes aussi divers que la linguistique et le raisonnement ; ou encore le projet LinkingOpenData³⁰ qui a pour objectif de référencer l'ensemble des données RDF disponibles.

2.1.4. Typologie des ontologies

Si les définitions de la notion d'ontologie sont très nombreuses, leurs typologies ne le sont pas moins (Van Heijst et al., 1997). Plusieurs d'entre elles ont en effet été proposées selon par exemple le degré de conceptualisation (Guarino, 1998), le niveau de représentation des connaissances (Bachimont, 2000) ou encore le niveau d'expressivité (Mizoguchi, 2003).

Ce document n'a pas pour ambition de faire une liste exhaustive des typologies existantes, mais plutôt d'en présenter deux qui semblent traduire concrètement les différentes représentations d'une ontologie.

2.1.4.1 Typologie selon l'objet de conceptualisation

La typologie proposée par Guarino est l'une des représentations les plus couramment citées (Guarino, 1998). Celle-ci se décompose en quatre niveaux dépendant du degré de conceptualisation (Graphique 2.1) :

- **Ontologies de haut-niveau** : Ces ontologies que l'on retrouve également sous le terme de top level ontologies (Guarino, 1998), ontologies de sens commun/général (Mizoguchi and Ikeda, 1996) ou encore ontologies génériques (Van Heijst et al., 1997) modélisent les concepts les plus généraux. Elles sont réutilisables d'un domaine à un autre et sont conçues pour réduire les incohérences des termes définis en aval de la hiérarchie. Parmi les

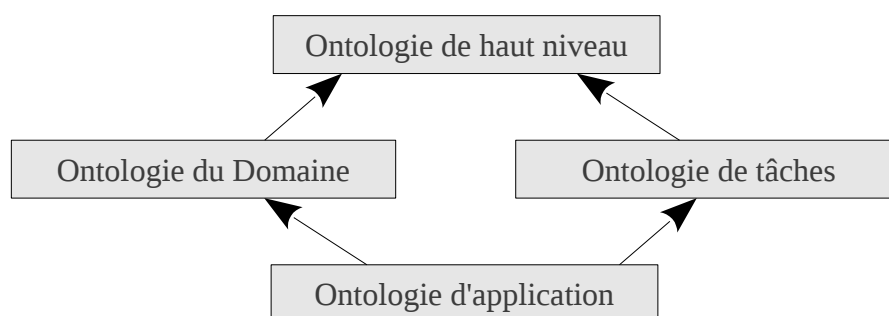
28 <http://cyc.com/>

29 <http://www.ontologyportal.org/>

30 <http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

ontologies de haut niveau les plus connus, citons notamment le projet Basic Formal Ontology³¹ (BFO) ou encore le projet Suggested Upper Merged Ontology³² (SUMO).

- **Ontologies de tâches** : elles décrivent les concepts utilisés pour résoudre un ou des problèmes d'une activité spécifique indépendamment d'un domaine quelconque. Elles fournissent un ensemble de termes et de relations au moyen duquel il est possible de décrire la manière de résoudre un problème. Selon Mizoguchi (Mizoguchi and Bourdeau, 2000), une ontologie de tâches est la structure computationnelle d'un système à base de connaissances.
- **Ontologies domaine** : elles sont spécialisées pour un certain type d'artefact et s'attachent à décrire le vocabulaire relatif à un domaine donné. Elles permettent de spécialiser les termes et les notions des ontologies de haut niveau.
- **Ontologies d'application** : c'est l'ontologie la plus spécifique, elle est utilisée pour modéliser les concepts d'un domaine particulier dans le cadre d'une activité spécifique. Elles sont à la fois une union et une spécialisation des ontologies de tâches et de domaines (Maedche and Staab, 2001).



Graphique 2.1: Typologie selon le niveau de conceptualisation (Guarino 1998)

À ce découpage en quatre niveaux, Gómez-Pérez (Gómez-Pérez, 1999) reprend cette typologie en y ajoutant 3 autres catégories :

- **ontologies de représentation des connaissances** : celles-ci décrivent les concepts généraux utilisés pour formaliser les connaissances.
- **méta-ontologies** : également nommées ontologies génériques ou core ontologies, elles contiennent des concepts moins abstraits que les ontologies de haut niveau mais réutilisables dans plusieurs domaines.
- **ontologie de tâche-domaine** : celles-ci sont relatives à une tâche spécifique d'un certain domaine.

2.1.4.2 Typologie selon le niveau de formalisation

À la proposition de Guarino de classifier les ontologies selon l'objet de conceptualisation, ajoutons également celle de Uschold et Gruninger (Uschold and Gruninger, 1996) qui porte sur le niveau de formalisation. En effet, selon le langage ou les formalismes de représentation utilisés, le degré de formalisation de l'ontologie peut être variable. Cette variabilité est exprimée sous la forme d'une typologie en quatre niveaux :

31 Basic Formal Ontology : <http://www.ifomis.org/bfo>

32 Suggested Upper Merged Ontology : <http://www.ontologyportal.org/>

- **Ontologies informelles** : exprimées en langue naturelle elles sont de ce fait facilement compréhensibles par les utilisateurs. Mais l'absence de formalisation rend leur utilisation dans un système informatique et leur validation problématiques.
- **Ontologies semi-informelles** : la sémantique du langage est plus structurée et limitée
- **Ontologies semi-formelles** : exprimée formellement dans un langage artificiel
- **Ontologies formelles** : exprimée dans un langage artificiel disposant d'une sémantique formelle permettant d'effectuer des vérifications. C'est le cas par exemple des logiques de description qui seront présentées dans le chapitre 2.3.1.4.

Cette typologie met en évidence plusieurs niveaux de représentation des ontologies. Nous verrons dans le chapitre sur les méthodes de développement d'ontologie (Chapitre 2.2) que le processus de la conceptualisation (ontologie informelle) du domaine à l'opérationnalisation (ontologie formelle) de l'ontologie consiste au passage d'un niveau à un autre.

2.2. Méthodologies pour la construction d'ontologies

Les ontologies étant destinées à être utilisées dans un système d'information, leur développement doit s'appuyer sur des principes similaires à ceux utilisés en génie logiciel. Afin qu'une ontologie soit pérenne et réponde aux objectifs attendus, il est donc nécessaire de la construire en respectant certaines règles.

De nombreuses méthodologies de construction d'ontologie existent sans qu'il y ait toutefois de consensus autour des meilleures pratiques à adopter (Fernández-López and Gómez-Pérez, 2002; De Nicola et al., 2009). Néanmoins, quelle que soit la méthodologie retenue, les auteurs s'accordent sur un processus de développement découpé en trois principales phases : (1) la spécification sémantique, (2) la conceptualisation, (3) la formalisation. Cette dernière peut être complétée par une phase d'ontologisation et une phase d'opérationnalisation. Enfin, tous soulignent également que la création d'une ontologie n'est pas linéaire mais que de nombreux allers-retours entre les phases de conceptualisation et de formalisation sont nécessaires. Le cycle de développement est donc par nature itératif et incrémental.

Dans les paragraphes suivants, quelques principes devant guider la construction d'une ontologie seront présentées ainsi que deux des méthodes les plus connues à savoir On-To-Knowledge et Methontology. Ce choix s'explique notamment par le fait que, contrairement aux autres, elles couvrent l'essentiel du cycle de vie des ontologies. Néanmoins, une vue plus synthétique et générale s'inspirant des travaux de Keita (Keita, 2007) et Psyché (Psyché, 2007) est également disponible dans le Tableau 1.1.

2.2.1. Principes généraux guidant la construction d'une ontologie

Avant d'aborder en détail les méthodologies existantes, il convient de rappeler les principes généraux qui doivent guider le modélisateur lors de la construction d'une ontologie. Il ne s'agit pas à proprement parlé d'une méthode de construction mais plutôt de points essentiels qu'il convient d'appliquer afin d'obtenir une ontologie cohérente. A partir des travaux de différents auteurs du domaine, Psyché (Psyché, 2007) propose une liste de onze recommandations à respecter :

1. **Clarté et objectivité** : afin d'éviter toute ambiguïté, l'ontologie doit fournir des définitions claires et objectives des termes indépendamment de tout contexte. Pour cela il est nécessaire d'associer aux entités ontologiques des définitions en langage naturel, une documentation et des exemples (Gruber, 1995).
2. **Complétude et perfection** : dès que cela est possible, un axiome logique doit être exprimé

en listant toutes les conditions nécessaires et suffisantes (Gruber, 1995).

3. **Cohérence** : les axiomes, les concepts et les inférences doivent être logiquement consistants (Gruber, 1995).
4. **Extensibilité monotone maximale** : lors de la modélisation de l'ontologie, la structure proposée doit anticiper les ajouts futurs. Ainsi, l'ajout de nouveaux concepts qu'ils soient généraux ou spécialisés ne doit pas entraîner la révision des concepts déjà existants (Gruber, 1995).
5. **Engagement ontologique minimal** : lors de la modélisation, les affirmations au sujet du monde réel doivent être les plus minimales. La spécialisation sera ensuite réalisée pendant la phase de construction (Gruber, 1995).
6. **Distinction ontologique** : les classes correspondant à différents critères d'identité doivent être disjointes. La notion d'identité étant définie comme les propriétés invariables pour une classe donnée. Par exemple, deux individus aux propriétés différentes peuvent être néanmoins rattachés à la même classe car ils possèdent un même noyau, une même identité.
7. **Modularité minimale** : l'ontologie doit fournir des définitions claires et objectives des termes indépendamment de tout choix d'implémentation. Cela signifie qu'il est nécessaire d'associer aux entités ontologiques des définitions en langage naturel.
8. **Subdivision des taxinomies** : dès que cela est possible, il faut subdiviser les taxinomies afin de gagner en précision sémantique et ainsi augmenter les capacités d'inférence.
9. **Distance sémantique minimale** : la distance sémantique entre des concepts provenant d'un même parent doit être la plus faible possible. Il est donc nécessaire de les grouper et de les représenter sous la forme d'une sous-classe.
10. **Normalisation des termes** : dès que cela est possible, il faut normaliser les termes utilisés. Cela permet de gagner en homogénéité et d'unifier le vocabulaire utilisé.
11. **Différentiation sémantique** : selon Bachimont (Bachimont, 2000) l'héritage des propriétés ainsi que la place d'une unité déterminent la signification du concept exprimé. Une construction organisée et méthodique d'une ontologie doit donc se faire en respectant les quatre points ci-dessous :
 - **principe de communauté avec le père** : toute unité de l'ontologie se détermine par rapport à son parent. Il faut donc préciser en quoi ils sont similaires
 - **principe de différence avec le père** : tout en étant similaire, l'unité fils est néanmoins différente de l'unité père. Les deux sont donc distincts. Il faut donc expliciter cette différence.
 - **principe de différence avec les frères** : tout comme la précédente, il est nécessaire d'explicitier en quoi une unité ontologique est différente des autres unités du même parent.
 - **principe de communauté avec les frères** : la aussi, bien qu'une unité soit différente elle a obligatoirement un lien avec les autres unités du même parent qu'il sera nécessaire d'explicitier.

Méthodes analysées	Principaux Auteurs	Description des Processus				Informations sur la méthode				
		Spécification	Acquisition de connaissances	Conceptualisation	Formalisation	Niveau de détail de la méthode	Formalisme de représentation	Stratégie d'identification des concepts	Stratégie d'ingénierie ontologique	Ontologies ou environnements développés
On-to-Knowledge	(Staab et al., 2001)	Très bonne description	Très bonne description	Description parcellaire	Très bonne description	Moyen	XML	Top-Down	Semi-dépendante de l'application	Text-To-Onto, OntoEditor
SENSUS	(Swartout et al., 1997)	Description parcellaire	Pas de prise en compte	Pas de prise en compte	Pas de prise en compte	Moyen	Réseaux sémantiques	Top-Down	Semi-dépendante de l'application	Strategy Development Assistant
METHON - TOLOGY	(Fernández-López et al., 1997)	Très bonne description	Très bonne description	Très bonne description	Très bonne description	Beaucoup	Liberté du choix	Middle-Out	Indépendante de l'application	(ODE, WebODE)
KACTUS	(Bernaras and Laresgoiti, 1996)	Description parcellaire	Pas de prise en compte	Description parcellaire	Très bonne description	Très peu	Aucun en particulier	Top-down	Dépendante de l'application	Ontology for diagnostic of faults
TOVE	(Grüninger and Fox, 1995)	Très bonne description	Description parcellaire	Très bonne description	Très bonne description	Peu	Logique	Middle-Out	Semi-dépendante de l'application	TOVE, Entreprise Design Workbench
Uschold & King	(Uschold and King, 1995)	Description parcellaire	Description parcellaire	Pas de prise en compte	Pas de prise en compte	Très peu	Aucun en particulier	Middle-Out	Indépendante de l'application	Entreprise Ontology, Entreprise Toolset

Tableau 1.1: Revue des méthodes d'ingénierie ontologique d'après (Keita, 2007) et (Psyché, 2007)

2.2.2. La méthode On-To-Knowledge

Développée dans le cadre d'un projet européen³³, la méthode On-To-Knowledge se décompose en cinq étapes dont seules les quatre dernières portent réellement sur le développement d'ontologies (Staab et al., 2001) (Figure 2.2).

La première étape (Feasibility Study), bien que nécessaire, ne peut être considérée comme faisant partie du cycle de développement de l'ontologie. Il s'agit en fait d'identifier le problème et de réaliser une étude de faisabilité par rapport à celui-ci. Cela permettra alors de sélectionner les outils nécessaires et les personnes ressources.

Les quatre phases suivantes sont quant à elles directement rattachées au cycle de vie de l'ontologie :

- **Phase de lancement** (*kickoff*) : durant laquelle les objectifs sont définis, les conditions requises sont spécifiées et les sources d'information identifiées. Cela permet d'aboutir à une première ontologie qualifiée de semi-formelle.
- **Phase d'amélioration** (*refinement*) : l'ontologie précédemment créée est améliorée en fonction des retours des experts du domaine. Les relations et les axiomes sont également précisés. Cela permet d'aboutir à une ontologie opérationnelle (*target ontology*) qui doit être conforme aux objectifs identifiés durant la phase de *kickoff*.
- **Phase d'évaluation** : cette phase passe par trois niveaux (technologique, utilisateur, ontologique) et permet d'aboutir à une ontologie utilisable en production.
- **Phase de production et d'évolution** (*application & evolution*) : une fois dans le système, l'ontologie va continuer à évoluer. Il est donc nécessaire que soient identifiées la ou les personnes qui auront en charge sa maintenance et son évolution.

Comme cela est illustré par les boucles de rétro-action partant des phases d'évaluation et de production vers la phase d'amélioration, le processus de création de l'ontologie n'est pas linéaire. Ces itérations successives permettent de vérifier à chaque étape si l'ontologie créée est en adéquation avec les besoins précédemment exprimés. Comparativement à certaines autres méthodes plus parcellaires, l'un des points forts de On-To-Knowledge est de couvrir la totalité du cycle de vie de l'ontologie.

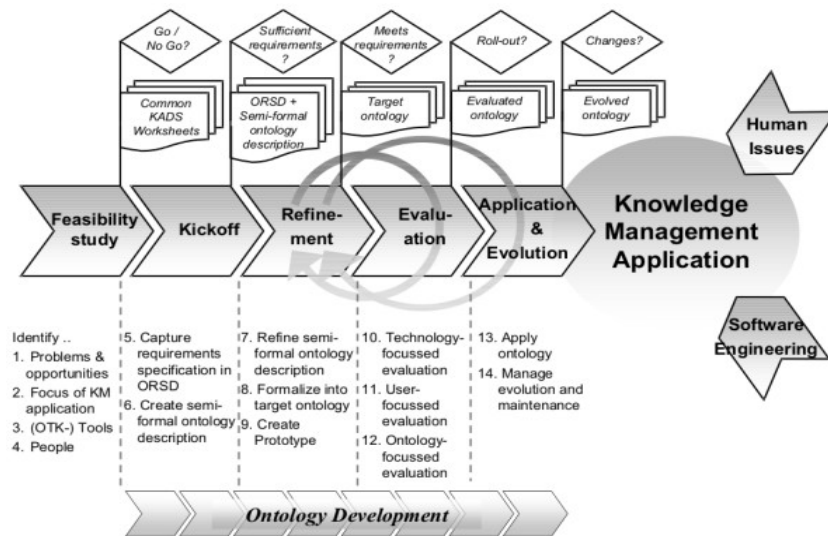


Figure 2.2: Les cinq étapes du processus On-To-Knowledge. Source (Staab et al., 2001)

2.2.3. La méthode Methontology

Développée par le Ontological Engineering Group de l'Université Polytechnique de Madrid, Methontology s'inspire des travaux déjà effectués en ingénierie des connaissances et en génie logiciel (Gómez-Pérez et al., 2004). Basée sur un prototypage évolutif, elle couvre l'ensemble du cycle de vie d'une ontologie (Figure 2.3) (Fernández-López et al., 1997). L'étude comparative menée par Corcho (Corcho, 2003) montre que Methontology est la méthode la plus avancée, ce qui lui vaut d'être recommandé par la FIPA³⁴ en matière de construction d'ontologies.

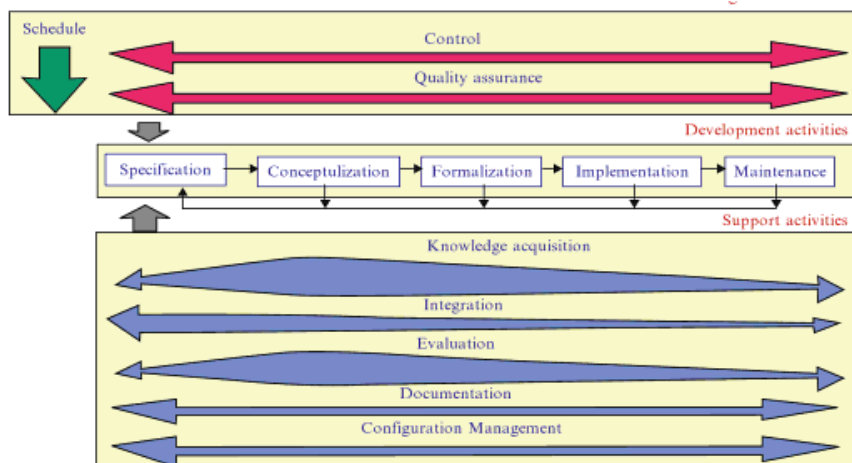


Figure 2.3: Les différentes phases de Methontology. Source (Corcho et al., 2005)

34 <http://www.fipa.org/about/index.html>

2.2. Méthodologies pour la construction d'ontologies

Bien que les noms des activités de développement diffèrent, une logique similaire aux cinq phases de développement définies dans On-To-Knowledge peut être observée :

- **Spécification** : définit l'objectif de l'ontologie, les utilisateurs finaux ainsi que son dimensionnement.
- **Conceptualisation** : il s'agit ici d'organiser la connaissance. Pour cela, onze activités principales ont été définies allant de la définition des termes jusqu'à la description des instances et des règles. C'est l'une des phases les plus importantes car l'essentiel de la connaissance est acquis au début de la construction.
- **Formalisation** : c'est dans cette phase que les connaissances sont traduites sous la forme d'une ontologie. Il s'agit de passer d'un modèle purement conceptuel à une implémentation informatique.
- **Implémentation** : l'ontologie est alors traduite dans un langage ontologique comme le OWL ou le RDF.
- **Maintenance** : la phase de maintenance corrige ou améliore l'ontologie créée

Methontology est l'une des méthodologies les plus complètes disponibles actuellement. Elle décrit en détails les différentes phases, mais aussi l'ordre des tâches à effectuer dans chacune d'entre elles. Cette complétude fait qu'elle a été utilisée dans des projets et domaines aussi divers que l'industrie chimique, la minéralogie ou encore la détection de polluants.

Les méthodes qui ont été évoqués offrent un cadre conceptuel pour la création d'ontologies. Dans ce travail de thèse c'est Methontology qui a été choisie pour guider le processus de création. Néanmoins, la phase de maintenance n'a pour le moment pas encore été appliqué. Il s'agit en effet d'un prototype dont l'utilisation reste limitée.

Grâce aux méthodes décrites précédemment, il est possible d'aboutir à une ontologie sémantiquement correcte et partageable entre différentes personnes. Il est maintenant nécessaire de traduire les différents concepts et relations identifiés dans un formalisme informatique. Le paragraphe suivant synthétise les différents travaux actuels et passés pour se focaliser ensuite sur la notion de Web Sémantique.

2.3. Formalismes et Langages de représentation d'ontologies

Que cela soit pour les individus ou pour les systèmes, représenter la connaissance nécessite de disposer de formalismes et de langages spécifiques. Ces derniers varient en fonction des formes de représentation et des capacités descriptives offertes. Dans ce chapitre, un rappel historique des formalismes de représentation est réalisé ainsi qu'une description des différents langages.

2.3.1. Formalismes de représentation

Toute connaissance se base sur un schéma cognitif propre à chaque individu. Un des enjeux de l'Intelligence Artificielle (IA) consiste alors à modéliser un domaine et à implémenter cette cognition sous une forme manipulable aussi bien par les humains que par les machines. En d'autres termes, il s'agit d'implémenter sous une forme pragmatique et opérationnelle, les structures et les mécanismes de connaissances manipulés par les individus. A cet effet, différents formalismes de représentation ont été inventés. Ces derniers varient en fonction du niveau d'expressivité, de la rigueur et de la sémantique offerts (Bechhofer et al., 2002).

Dans l'histoire de ces formalismes, deux grandes périodes peuvent être distinguées. Tout d'abord avant les années quatre-vingt où l'on distinguait les formalismes basés sur la logique (descriptive, de 1er ordre, etc.) et les représentations non logiques (les réseaux sémantiques, les schémas, etc.). Puis ensuite une seconde phase, qualifiée d'hybride, car mélangeant différents formalismes de représentation comme les schémas avec de la logique. Cela a conduit à l'apparition de nouveaux paradigmes telle que la logique de description qui est majoritairement utilisée pour la description d'ontologie (Gensel, 2006).

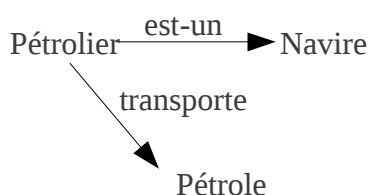
Dans ce chapitre, les quatre principaux formalismes que sont les réseaux sémantiques, les graphes conceptuels, les frames (schéma), et les logiques de description seront présentés. Ce choix s'explique tout d'abord par l'importance qu'ils ont eu sur le domaine de la représentation des connaissances mais aussi par l'héritage conceptuel qu'ils ont laissé et qui est aujourd'hui toujours utilisé. Pour les lecteurs intéressés, une revue plus complète est notamment disponible dans (Brachman and Levesque, 2004; Baader et al., 2007).

2.3.1.1 Les réseaux sémantiques

Issus des travaux de Quillian (Quillian, 1966; Collins and Quillian, 1969), les réseaux sémantiques ont été initialement conçus pour illustrer le fonctionnement de la mémoire dans le cadre de la linguistique. Ils ont par la suite été étendus pour devenir un langage de représentation (Anderson and Bower, 1980).

Dans cette théorie, la connaissance est stockée sous la forme d'un graphe orienté et annoté. Celui-ci est composé de nœuds représentant des concepts ou des individus, reliés entre eux par des arcs correspondant à des liens d'action, de spécialisation ou d'appartenance (Graphique 2.2). Cette forme de schématisation permet de mettre en avant certains mécanismes comme la généralisation/héritage (passage d'une classe à une sous classe) ou encore la notion d'instanciation (création d'un individu à partir d'une notion).

2.3. Formalismes et Langages de représentation d'ontologies

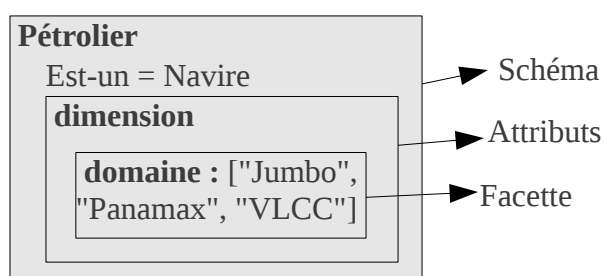


Graphique 2.2: Exemple simplifié d'un réseau sémantique représentant la phrase "Un pétrolier est un navire transportant du pétrole"

Malgré l'apport des réseaux sémantiques dans le domaine de la modélisation de la connaissance, ces derniers souffrent d'importantes limites. En effet, l'accent est mis sur la structure de l'information et non sur sa sémantique pouvant amener à des confusions entre les relations ou les classes (Brachman, 1977). A cela s'ajoute également la question de la granularité de représentation et de la hiérarchisation des informations. Ce constat a alors amené au développement des nouveaux formalismes dont notamment les schémas (frames).

2.3.1.2 Les Schémas (frames)

Issu des travaux de Minsky (Minsky, 1974), le principe de schéma (également nommé frame ou cadre conceptuel) part du postulat que notre représentation du monde se fait sous la forme d'un agencement d'unités d'informations élémentaires nommées schéma. Contrairement aux réseaux sémantiques, pour qui la mémoire sémantique est de type associatif, l'information peut être alors divisée en sous-systèmes ayant potentiellement des liens entre eux. Dans cette théorie, un concept est représenté sous la forme d'un schéma composé d'un individu nommé *stéréotype* possédant des propriétés et des attributs (*slots*) (Graphique 2.3). Les attributs sont eux-mêmes décrits par des facettes qui peuvent être soit déclaratives (elles spécifient des contraintes pour la valeur de l'attribut) ou procédurales (elles peuvent être vues comme des représentations locales de procédure de raisonnement). Les schémas peuvent être organisés sous une forme hiérarchique (classes/sous-classes) grâce à des mécanismes d'héritage.



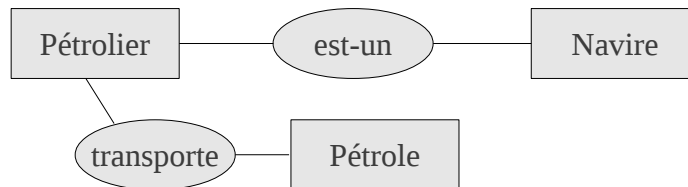
Graphique 2.3: Exemple simplifié d'un schéma portant sur la définition d'un Pétrolier

2.3.1.3 Les graphes conceptuels

S'inspirant des graphes existentiels de Pierce et des réseaux sémantiques, Sowa (Sowa, 1983, 1999) propose un nouveau mode de représentation s'appuyant sur la logique des prédicats de 1er ordre. Si les graphes conceptuels partagent les mêmes avantages que réseaux sémantiques, ils offrent par contre une meilleure standardisation ainsi qu'une plus grande formalisation.

Pour cela, Sowa propose de définir un système formel composé d'un ensemble de graphes, appelé

base canonique, qui contient les graphes sémantiquement corrects. Avec les apports de la théorie des graphes, il devient possible d'effectuer des raisonnements poussés à partir d'opérations de projection, de généralisation ou de spécialisation. Enfin, la compréhension du modèle est également facilitée en raison d'une représentation graphique différente en fonction de la connaissance (les concepts sont ainsi représentés par des rectangles et les prédicats par des ronds) (Graphique 2.4) (Carloni, 2008).



Graphique 2.4: Exemple simplifié d'un graphe conceptuel portant sur la définition d'un Pétrolier

2.3.1.4 Les logiques de description

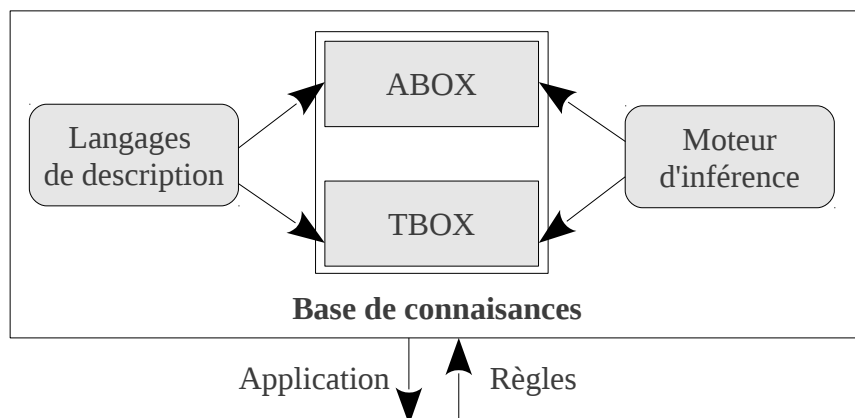
S'inspirant du système KL-ONE, des réseaux sémantiques et des frames, les logiques de description³⁵ se distinguent par leur sémantique formelle basée sur la logique du 1er ordre et la théorie des ensembles (Baader, 2003). Tout d'abord limitées à des logiques de description peu expressives et à la résolution de problèmes d'inférence en temps polynomial, celles-ci vont à partir des années 90 s'enrichir d'algorithmes de raisonnement plus efficaces³⁶ ouvrant la voie à de nouvelles applications telles que le Web Sémantique (Fournier-Viger, 2005).

Les deux niveaux de description

Pour modéliser un domaine de connaissance, le formalisme des logiques de description s'appuie sur deux niveaux de description. Le premier nommé TBOX (T pour terminologique) décrit les connaissances générales (concepts et rôles) d'un domaine (Graphique 2.5). Le second nommé ABOX (A pour assertionnel) décrit un état spécifique du domaine dans lequel sont définis les individus qui sont des instances de concepts. A chaque Abox est associée une Tbox, ce qui signifie qu'un individu correspond nécessairement à une entité générique qui peut être relié à d'autres individus par des relations binaires (Napoli, 1997).

35 Aussi appelées logiques terminologiques

36 Algorithmes de vérification à base de tableaux



Graphique 2.5: Architecture d'une base de connaissances basée sur les logiques de description (Baader & Nutt 2003)

Dans cette logique, l'interprétation des connaissances est faite selon l'hypothèse d'un monde ouvert. Par opposition au monde fermé, que l'on retrouve par exemple dans les bases de données, le monde ouvert suppose qu'il peut exister des instances ou des relations qui n'ont pas encore été définies. De ce fait, l'absence d'une information ne retourne pas nécessairement un résultat négatif. Néanmoins, cela implique des raisonnements logiques plus complexes que dans un monde fermé. En effet, pour vérifier la présence d'une information, il est alors nécessaire d'envisager plusieurs situations alternatives (Baader, 2003).

Expressivité des logiques de description

Le niveau d'expressivité des logiques de description dépend des constructeurs employés. Ainsi, bien que la logique minimale nommée *Attributives Language (AL)* soit peu expressive, elle peut être enrichie par différents constructeurs afin d'exprimer par exemple la négation (*ALC*) ou encore l'union (*ALU*). Une liste plus détaillée des principaux constructeurs, de leur signification et de leur nom de langage est présentée dans le Tableau 1.2 (Gómez-Pérez et al., 2004). Les moteurs d'inférence présentés dans le chapitre 2.4.3 (p.88) implémentent pour la plupart les constructeurs des extensions les plus complètes comme *SHIQ* ou *SHOIN*.

Constructeur	Syntaxe	Langage			
		FLO	FL-	AL	S
Concept	A	FLO	FL-	AL	S
Rôle	R				
Intersection	$C \cap D$				
Restriction de valeurs	$\forall R.C$				
Quantification existentielle	$\exists R$				
Universel	T				
Vide	\perp				
Négation atomique	$\neg A$				
Négation	$\neg C$	C			
Union	$C \cup D$	U			
Restriction existentielle	$\exists R.C$	E			
Restriction numérique	$(\geq n R), (\leq n R)$	N			
Nominaux	$\{a_1, \dots, a_n\}$	O			
Hierarchie de rôles	$R \subseteq S$	H			
Rôle inverse	R^{-}	I			
Restriction numérique qualifiée	$(\geq n R.C), (\leq n R.C)$	Q			

Tableau 1.2: Exemples de constructeurs et leur correspondance dans les langages d'après (Gómez-Pérez et al., 2004)

L'inférence dans les logiques de description

Dans les logiques de description, l'inférence peut se faire tant au niveau terminologique (TBOX), qu'assertionnel (ABOX) (Baader and Nutt, 2003) (Graphique 2.5).

Au niveau terminologique, quatre principaux problèmes d'inférences peuvent être identifiés :

- **la satisfiabilité** : identifier les inconsistances de la base de connaissances
- **la subsumption** : classification des concepts (calcul de la taxonomie et des hiérarchies)
- **l'équivalence** : permet de trouver les concepts équivalents
- **la disjonction** : permet de trouver les concepts disjoints

Au niveau assertionnel, quatre principaux problèmes d'inférences peuvent être identifiés :

- **la cohérence** : les assertions définies dans la ABOX sont cohérents avec la TBOX
- **la vérification d'instance** : vérifie que chaque instance respecte la définition de son concept
- **la vérification de rôle** : vérifie que les rôles sont correctement utilisés
- **les problèmes de récupération** : trouve le concept le plus spécifique pour chaque instance

2.3. Formalismes et Langages de représentation d'ontologies

Sa plus grande richesse sémantique, associée à ses possibilités d'inférence font des logiques de description le formalisme de représentations des connaissances le plus utilisé pour le développement d'ontologies. Si les premiers travaux au début des années 80 se limitaient à la résolution de problèmes pour des logiques de description peu expressives, la deuxième génération (à partir des années 90) a permis d'effectuer des raisonnements sur des logiques de description plus expressives ouvrant ainsi la voie à de nouvelles applications comme le web sémantique (Fournier-Viger, 2005).

2.3.1.5 Synthèse sur les formalismes de représentation des connaissances

Si les différents formalismes qui ont été présentés permettent tous par leur sémantique formelle de formaliser des connaissances, ils ne possèdent pas néanmoins le même niveau d'expressivité ni les mêmes capacités de raisonnement. Par exemple, les graphes conceptuels privilégient la description de connaissances assertionnelles alors que les logiques de description sont orientées vers la description de connaissances ontologiques (Baneyx, 2007). Conçus à différentes périodes et pour des besoins spécifiques ces langages se sont mutuellement enrichis.

Aujourd'hui, le formalisme de représentation le plus couramment utilisé pour la création d'ontologies est celui des logiques de description. Sa forte expressivité, associée à ses capacités d'inférences, sont autant de caractéristiques qui ont conduit les nouveaux langages issus du web sémantique à s'appuyer sur ce formalisme.

2.3.2. Les langages ontologiques

Un langage de représentation d'ontologies permet d'exprimer une interprétation du monde en se basant sur une sémantique formelle et une structure syntaxique précise. Historiquement on assiste à partir des années 90 à une multiplication des langages dits traditionnels car basé sur le domaine de l'Intelligence Artificielle (Nguyen, 2011). Face à cela, un travail de standardisation était nécessaire favorisé par l'avènement du Web. Sous l'impulsion du groupe de travail OntoWeb du W3C, des langages standardisés ont été définis préfigurant ainsi une nouvelle façon de manipuler l'information : le web sémantique (Bechhofer, 2002).

Dans les prochains paragraphes, les langages traditionnels seront brièvement présentés pour s'attarder ensuite sur ceux issus du Web Sémantique qui sont aujourd'hui majoritairement utilisés et cela même pour la construction d'ontologies sans relation avec le monde du web.

2.3.2.1 Les langages traditionnels

Au début des années 1990, différents langages ontologiques s'appuyant sur des représentations basées sur la logique, sur des représentations non logiques ou alors un mélange des deux ont été créés. Parmi les plus connus, citons notamment (Gómez-Pérez, 1999):

- **Ontolingua** : basé sur le langage Knowledge Interchange Format (KIF), il utilise une syntaxe LISP pour représenter la connaissance qui est alors modélisée sous la forme de schémas et de logiques du 1er ordre. C'est notamment le langage utilisé par Ontolingua Server (Farquhar et al., 1997)
- **Cycl** : Syntactiquement proche de Lisp, Cycl est basé sur la logique de premier ordre auquel il ajoute des extensions permettant par exemple d'exprimer la notion d'égalité. Cycl est notamment utilisé pour le développement de l'ontologie Cyc (Fensel, 2000).
- **LOOM** : est un langage de haut niveau basé sur la logique de 1er ordre et appartenant à la famille de KL-One. Il offre un modèle déclaratif expressif ainsi qu'un moteur efficace et une

vérification automatique de l'ontologie (Corcho and Gómez-Pérez, 2000).

- **Flogic** (Frame Logic) : tout comme ontolingua, Flogic combine à la fois les schémas et la logique de premier ordre. Il inclut la notion d'objet, d'héritage ou encore d'encapsulation (Fensel, 2000).

Une grande partie des efforts menés dans le domaine de l'intelligence artificielle à été portée sur les capacités de raisonnements des langages traditionnels. Mais l'avènement du Web et sa popularité croissante ont mis en évidence des besoins différents qui ont abouti à la création de nouveaux langages.

2.3.2.2 Les langages issus du Web Sémantique

Architecture du Web sémantique

L'expression Web Sémantique est due à Tim Berners-Lee qui fait référence par cette notion à la vision du web de demain comme étant un vaste espace d'échange de données entre humains et machines (Berners-Lee, 1998). Contrairement aux travaux issus de l'intelligence artificielle essentiellement axés sur la logique, le web sémantique se focalise sur les données (Gandon, 2008). Tel qu'il est actuellement conçu, le web est principalement syntaxique car destiné à être lu par des humains. Il ne peut donc être facilement manipulable par des agents logiciels car il lui manque une dimension sémantique permettant de relier entre elles les différentes informations pour permettre la réalisation de traitements automatiques. Comme le souligne Tim Berners-Lee, la notion du web sémantique n'est pas un web à part, mais plutôt une évolution de l'existant dans lequel les données seront interconnectées et renseignées (Berners-Lee and Fischetti, 1999) :

« The semantic Web is an extension of the current web in wich information is given well-defined meaning, better enabling computers and people to work in cooperation »

Pour cela, il propose une nouvelle infrastructure basée sur des standards technologiques et s'intégrant dans l'architecture du web actuelle de manière transparente (Figure 2.4). Les couches les plus basses de la pyramide assurent l'interopérabilité syntaxique entre les différents composants.

Tout d'abord, le protocole Universal Ressource Identifier (URI³⁷) qui permet une identification unique et universelle de n'importe quelle ressource. Le principe des URI est l'un des piliers du web sémantique ce qui explique sa place à la base de l'architecture.

S'ajoute ensuite la couche eXtensible Markup Language (XML) au sens large (XML, Name Space, XML Schema XML Query) comme format standard d'échange et de manipulation des ressources. Le XML, est un méta-langage proposé par le W3C permettant de structurer les informations sous une forme logique et selon un format standard. Néanmoins, il ne permet pas d'exprimer la sémantique d'une notion.

C'est pourquoi sont ensuite intégrés les langages Ressource Description Framework (RDF) et RDF Schema (RDFS) avec lesquels il devient possible d'annoter les ressources du Web. C'est à cette étape que le sens des informations transmises devient interprétable par des agents informatiques. RDF et RDFS sont ainsi considérées comme les premières fondations de l'interopérabilité sémantique. Mais, afin d'être le plus générique possible, le langage RDF offre volontairement une sémantique limitée. C'est pourquoi s'ajoute une couche ontologique aux possibilités étendues. Celle-ci se base sur un langage standard, le Ontology Web Language (OWL) auquel s'ajoutent des langages de requêtes comme SPARQL et des langages de règles comme le Semantic Web Rule

37 Décrit par le standard RFC2396 - http://www.w3.org/2002/11/dbooth-names/rfc2396-numbered_clean.htm

Language (SWRL).

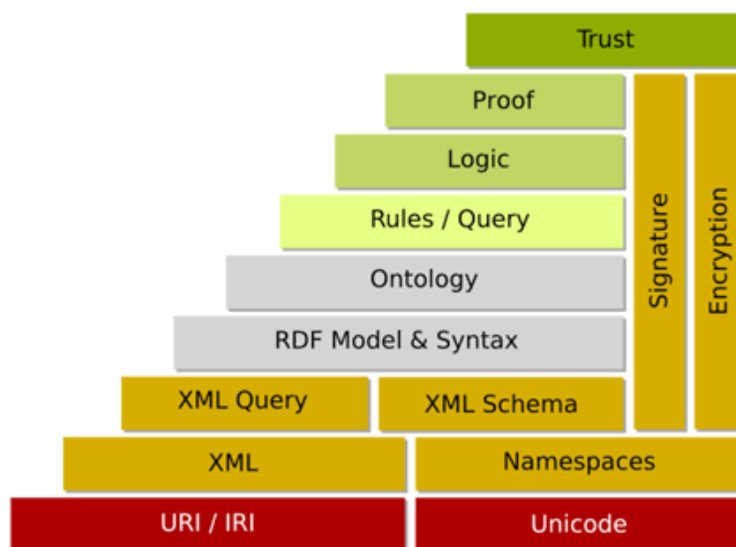


Figure 2.4: Architecture du Web Sémantique selon le W3C

Si les couches précédentes ont déjà été standardisées et recommandées par le W3C³⁸ il en est tout autrement pour celles de plus haut niveau (logique et preuve) qui permettent de justifier la pertinence et la fiabilité de l'information. Celles-ci font partie d'une vision prospective en cours d'élaboration.

La description des ressources du Web et plus généralement du monde réel peut être modélisée à l'aide des deux langages RDF et OWL issus du Web Sémantique. Ces derniers seront présentés dans les prochains paragraphes.

Ressource Description Framework (RDF)

Principes généraux

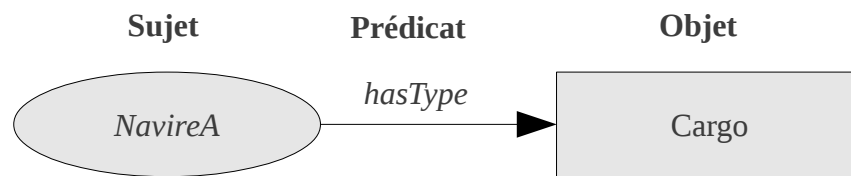
Le Ressource Description Framework (RDF) est le premier langage standardisé proposé par le W3C pour la description des ressources du web (Lassila and Swick, 1999). Il est selon Tim Berners-Lee (Berners-Lee, 1999) à la base du web sémantique pour qui le RDF est :

« the necessary foundation and infrastructure to support the description and management of (Web) data »

Créé pour représenter l'information de la manière la plus souple et la plus générique possible, RDF s'appuie sur un modèle fixe composé d'assertions³⁹. Ces assertions sont exprimées sous la forme d'un triplet composé d'un sujet, d'un prédicat et d'un objet et peuvent être ensuite regroupées pour former un graphe RDF. Par exemple, l'information suivante « le navireA est un cargo » traduite sous la forme d'un triplet RDF donnera (Graphique 2.6) :

38 WW3C : World Wide Web Consortium - <http://www.w3.org/>

39 Appelées aussi déclaration



Graphique 2.6: Exemple d'assertion RDF

Le sujet d'un prédicat précise l'entité à laquelle il est fait référence. Celle-ci peut être une ressource identifiée à l'aide d'un Universal Resource Identifier (URI) ou d'un nœud anonyme. Dans notre cas il s'agit du *NavireA*. Le prédicat d'une assertion permet de décrire une propriété applicable à cette ressource. Cette propriété peut être aussi bien relative à un attribut ou à une relation. Ici, le type du navire (*hasType*) est spécifié. Enfin, l'objet représente la valeur de la propriété. Il peut s'agir, comme cela est illustré dans notre exemple, d'une valeur littérale (*cargo*) ou alors d'une autre ressource. Dans ce dernier cas, l'objet est également le sujet d'une ressource et le résultat obtenu est un graphe comme cela est illustré dans le graphique ci-dessous (Graphique 2.7) :



Graphique 2.7: Exemple de schéma RDF

Les graphes créés ont pour vocation d'être interconnectés. Or un même sujet peut faire référence à différentes notions. Dans ce cas, comment s'assurer de l'unicité d'une ressource ? Afin de lever toute ambiguïté, chaque ressource RDF est identifiée au moyen d'un Universal Resource Identifier (URI). Beaucoup plus large que les Universal Resource Locator (URL), le concept d' URI permet de définir de manière unique n'importe quelle ressource électronique ou non.

Langage RDF

Si les graphiques présentés sont facilement compréhensibles par un humain, il en est tout autrement pour des machines. Ces dernières ont besoin que l'information soit encodée dans un format facilement manipulable. C'est ce que l'on nomme le processus de sérialisation. Bien que différentes syntaxes existent (N-Triples, N3, etc.), celle qui est actuellement la plus utilisée est RDF/XML. L'exemple du Graphique 2.6 traduit au format RDF/XML donnerait :

```
<rdf:Description about="NavireA">
  <hasType>Cargo</hasType>
</rdf:Description>
```

Cette thèse n'ayant pas pour objectif d'explorer toutes les subtilités du RDF/XML, l'exemple proposé est volontairement simplifié. Néanmoins, une description plus complète du langage est disponible dans (Beckett, 2004).

RDF Schema (RDFS)

Publié par le W3C en même temps que le RDF, le Resource Description Framework Schema (RDFS) (Brickley and Guha, 2004) peut être vu comme un méta-modèle qui spécifie les ressources utilisables dans un triplet RDF tout en lui associant une sémantique formelle. Pour cela, il enrichit le RDF en fournissant les primitives ontologiques nécessaires à la description de classes d'objets et de propriétés (*rdfs:Class*, *rdfs:Property*, *rdfs:subPropertyOf*, etc.).

Par exemple, il permet d'organiser hiérarchiquement les ressources grâce à des liens de subsomption (*rdfs:subClassOf*, *rdfs:subPropertyOf*) ou encore de spécifier des contraintes sur les valeurs d'une propriété (*rdfs:domain*, *rdfs:range*). Ainsi, les classes spécifiques d'un domaine seront déclarées comme étant des instances de la Ressource *Class* et ses propriétés comme des instances de la ressource *Property*. Ensuite, la notion de hiérarchie (de classes ou de propriétés) sera réalisée grâce à l'utilisation des propriétés *subClassOf* ou *subPropertyOf* (Figure 2.5).

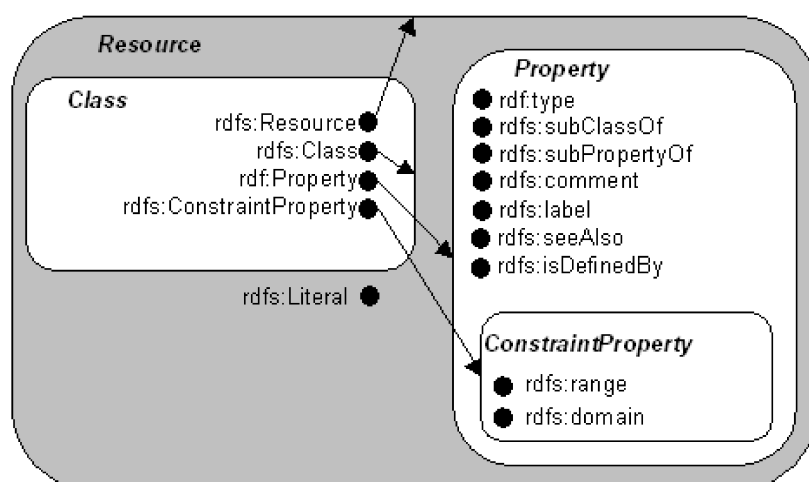


Figure 2.5: Classes et propriétés définies par le RDFS (source W3C)

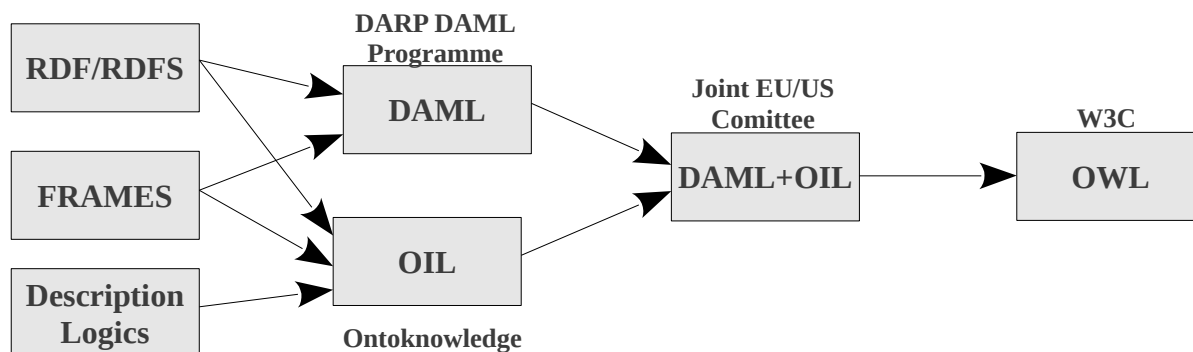
Limites du RDF

Si RDF et RDFS ont été conçus pour être les plus génériques possible, cette simplicité du langage s'accompagne également d'une expressivité insuffisante pour la description de situations complexes. Par exemple, il est impossible de spécifier des propriétés d'unicité, d'exprimer que deux classes sont disjointes ou encore de définir des restrictions de cardinalité (Antoniou and Harmelen, 2004).

Afin de pallier à ces limites, le W3C a proposé un nouveau langage plus expressif construit au dessus du RDF : le Web Ontologie Language.

Ontology Web Language (OWL)

A la fin des années 90, les langages OIL et DAML-ONT ont été développés afin de combler les lacunes du RDF(S). Le premier issu d'un groupe de chercheurs essentiellement européens (EU ontoknowledge project) et le second d'un groupe de chercheurs majoritairement américains (DARPA DAML Programme). Face à cette situation et à l'évolution parallèle de ces deux langages, un projet de fusion a été conduit par le Joint EU/US Committee on Agent Markup Languages qui a abouti au langage OWL (Graphique 2.8).



Graphique 2.8: Les familles de langages qui ont conduit au OWL

Principes généraux

Le Ontology Web Language (OWL) est, depuis 2004, le standard défini par le W3C pour la création d'ontologies (Bechhofer et al., 2004). Basé sur le langage DAML+OIL, OWL se fonde sur les primitives de base définies par les schémas RDF. Néanmoins, loin de n'être qu'une simple extension du RDF, il apporte toute la sémantique nécessaire à la description des connaissances comme par exemple des mécanismes de comparaison de classes (concepts d'équivalence, de symétrie, etc.). Il est de ce fait beaucoup plus expressif que le RDF et permet ainsi une modélisation plus fine. Plutôt que de définir un langage de modélisation complexe et difficilement utilisable, le W3C a volontairement fait le choix de proposer trois sous-langages OWL à l'expressivité croissante : OWL-Lite, OWL DL et OWL Full. Chacun de ces sous langages étant lui même une extension de son prédécesseur.

Une expressivité croissante, de OWL Lite à OWL Full

En fonction des besoins de représentation ou d'inférence, le consortium W3C propose trois sous langages à l'expressivité croissante :

- **OWL Lite** : c'est le sous langage OWL le plus simple. Il reprend tous les constructeurs RDF (*rdfs:subClassOf*, *rdfs:Property*, etc.) et y ajoute la possibilité de définir une nouvelle classe (*owl:Class*), de spécifier des notions d'équivalence (*owl:sameIndividualAs*) entre individus ou encore d'exprimer des contraintes simples (*owl:minCardinality* et *owl:maxCardinality*) en utilisant des entiers égaux à 0 ou 1. Le grand avantage de ce langage est sa simplicité aussi bien pour les utilisateurs que pour les développeurs ainsi que ses procédures de raisonnement efficaces basées sur la logique de description SHIQ (Figure 2.6). Il est donc particulièrement adapté pour les cas de migration de thésaurus ou de taxonomies.
- **OWL DL** : OWL DL, qui signifie Ontology Web Language Description Logics offre une expressivité supérieure à OWL Lite tout en gardant une complétude des raisonnements et une décidabilité des calculs. Il étend notamment les contraintes de cardinalité en permettant l'utilisation de n'importe quel entier positif ou encore il permet d'affirmer que deux classes n'ont aucune instance commune (*owl:disjointWith*). Pour cela OWL DL propose tous les constructeurs du langage OWL, mais ces derniers ne peuvent être utilisés que sous certaines conditions. Il se base sur SHOIN (D) (Figure 2.6) et permet d'effectuer des raisonnements aussi bien au niveau de la Tbox que de la ABox (cf. chapitre 2.3.1.4 (p.72)).
- **OWL Full** : c'est la version la plus complexe d'OWL mais également celle qui offre le plus haut niveau d'expressivité. Contrairement à OWL-DL tous les constructeurs OWL peuvent

2.3. Formalismes et Langages de représentation d'ontologies

être utilisés sans restriction. Néanmoins, cette expressivité accrue est gagnée au détriment de la décidabilité des raisonnements. Il est destiné aux situations où un haut niveau d'expressivité est primordial quitte à ne pouvoir garantir la complétude des calculs.

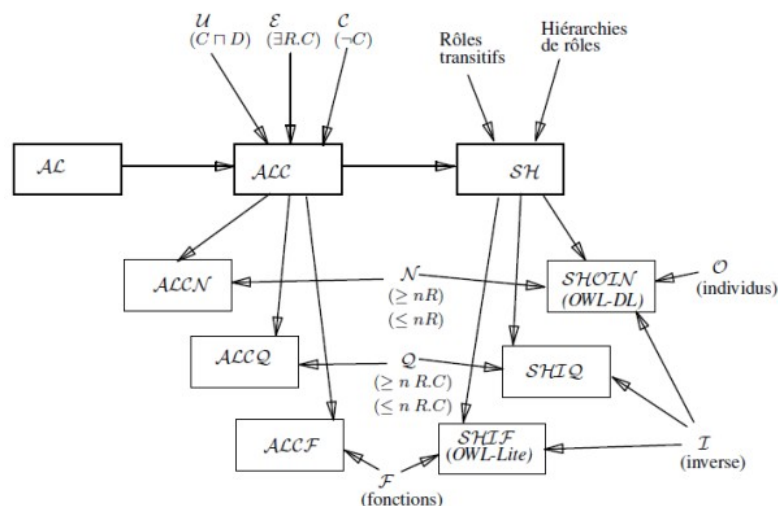


Figure 2.6: Implémentation des logiques de description par OWL

OWL 2

Développé par le Web Ontology Working Group (W3C), OWL 2 se veut être à la fois une extension et une révision du langage OWL. Les motivations qui ont conduit à l'élaboration de cette nouvelle version portent notamment sur l'expressivité limitée de OWL, sur la syntaxe du langage jugée trop complexe ou encore sur l'impossibilité d'annoter des axiomes (Grau et al., 2008). OWL 2 vient donc compléter les capacités initiales de OWL par l'ajout de nouveaux axiomes (*ObjectHasSelf*, *ObjectExactCardinality*, etc.) dont le besoin a été clairement identifié par la communauté des utilisateurs et pour lesquels ils existent des mécanismes de raisonnement efficaces (Patel-Schneider and Horrocks, 2006). Surtout, cette nouvelle version apporte la notion de profils OWL⁴⁰ (OWL 2 EL, OWL 2 QL, OWL 2 RL) qui peuvent être vus comme des sous-ensembles du langage permettant de répondre à des besoins applicatifs spécifiques. Enfin elle vise à apporter une touche de sucre syntaxique en facilitant l'écriture d'axiomes complexes.

2.3.2.3 Semantic Web Rule Language (SWRL)

Comme cela a été abordé dans le chapitre 2.3.2.2, au dessus de la couche ontologie du web sémantique s'ajoute un niveau supplémentaire permettant de spécifier des règles à effectuer dans le cas d'une situation spécifique. Ce processus va alors déduire de nouvelles informations à partir d'observations et de règles. Dans le domaine des ontologies, une règle est vue comme une instruction conditionnelle de type "si la condition est vérifiée alors une conséquence doit être exécutée". Une nouvelle connaissance est ainsi ajoutée dans le cas de la satisfaction de la règle donnée. Parmi les solutions actuellement existantes (Jena Rules, OWL 2 DL Axioms, etc.), SWRL apparaît comme la meilleure alternative entre l'expressivité, la standardisation et la décidabilité. De

40 <http://www.w3.org/TR/owl2-profiles>

plus, contrairement à d'autres solutions, celle-ci offre l'avantage d'être nativement utilisable dans les bibliothèques de programmation (API) existantes.

Le Semantic Web Rule Language (SWRL) est un langage de règles pour le Web sémantique. Bien qu'il soit encore à l'état de soumission dans le W3C⁴¹, il a déjà été implémenté dans de nombreux logiciels ou bibliothèques comme Protégé, Pellet ou encore OWL API.

Comparativement à d'autres alternatives, l'objectif du SWRL est d'étendre les potentialités du langage OWL-DL en permettant l'ajout de règles, tout en gardant une compatibilité maximale avec la syntaxe sémantique existante (Horrocks et al., 2004). Prévu pour supporter les raisonnements basés sur les logiques de description et les règles de Horn, la structure d'une règle SWRL est de la forme antécédent(s) -> conséquence(s) qui signifie : « si l'antécédent est vrai, alors la conséquence l'est aussi » (Graphique 2.9). Antécédents et conséquences sont eux-mêmes constitués d'atomes exprimés en utilisant des concepts OWL (classes, propriétés, individus, etc.). Dans une règle, les atomes peuvent être de la forme : C(x), P(x,y), ou SameAs(x,y) ; differentFrom(x,y) ou C est un élément OWL, P une propriété OWL, et x,y peuvent être des variables, des individus ou des valeurs.

```
rule ::= 'Implies(' [ URIreference ] { annotation } antecedent consequent ')'  
antecedent ::= 'Antecedent(' { atom } ')'  
consequent ::= 'Consequent(' { atom } ')'
```

Graphique 2.9: Structure d'une règle SWRL

Pour illustrer l'application d'une règle SWRL, prenons l'exemple d'un navire qui dépasserait la vitesse autorisée ce qui déclencherait l'ajout d'une alerte. La règle SWRL correspondante s'écrirait de la manière suivante :

```
Navire(?navireIndividu)  
^ DataProperty_Speed(?navireIndividu, ?navireSpeed)  
^ greaterThan(15, ?navireSpeed)  
→ OP_Vessel_hasAlert(?navireIndividu, Alert_Speed_HighSpeed)
```

Cette règle peut être interprétée comme : « Si la vitesse du navire (*?navireSpeed*) est supérieure (**greaterThan**) à la vitesse maximale autorisée (15) alors, une alerte est déclenchée (Alert_Speed_HighSpeed) ». Pour effectuer la comparaison entre la vitesse maximale autorisée et celle du navire, une fonction faisant partie des built-in SWRL est utilisée.

En effet, en plus des prédicats OWL, SWRL possède des fonctions supplémentaires nommées built-in⁴². Ces dernières peuvent être vues comme des points d'extensions permettant une approche modulaire. Certains built-in sont disponibles par défaut et s'appuient sur les potentialités de XQuery⁴³ et de XPath⁴⁴. Ils permettent par exemple de réaliser des opérations mathématiques (*swrlb:add*, *swrlb:subtract*, etc.), des comparaisons (*swrlb:equal*, *swrlb:greaterThan*, etc.) ou des manipulations de chaînes de caractères (*swrlb:contains*, *swrlb:replace*, etc.) ou encore des opérations sur les URI (*swrlb:anyURI*, *swrlb:resolveURI*, etc.).

41 <http://www.w3.org/Submission/SWRL/>

42 <http://www.w3.org/Submission/SWRL/#8>

43 <http://www.w3.org/TR/xquery/>

44 <http://www.w3.org/TR/xpath/>

2.3. Formalismes et Langages de représentation d'ontologies

Au-delà du SWRL, signalons également l'initiative Rule Interchange Format⁴⁵ du W3C. Face aux nombreux langages de règles existants, celle-ci vise à définir un format d'échange standard afin d'arriver à une meilleure interopérabilité. Plusieurs recommandations ont déjà été produites (RIF Basic logic Dialect, Rif Cor, etc.) mais elle s'avère complexe à implémenter. C'est pourquoi elle est encore actuellement assez peu utilisée.

45 <http://www.w3.org/TR/2007/WD-rif-core-20070330/>

2.4. Conception et exploitation d'ontologies

Formaliser la connaissance est une tâche complexe nécessitant de nombreuses interactions entre les différents acteurs. C'est pourquoi, des interfaces spécifiques facilitant la création d'ontologies ont été développées. Une fois formalisée, cette connaissance peut être alors manipulée et intégrée au sein d'un système informatique. Les principaux logiciels et bibliothèques permettant d'effectuer ces traitements sont présentés dans ce chapitre.

2.4.1. Éditeur d'ontologies

Les différents articles et rapports traitant du domaine de la gestion des connaissances listent un grand nombre d'éditeurs d'ontologie (Hepp et al., 2007; Staab and Studer, 2009). A titre d'exemple, la synthèse⁴⁶ la plus complète à ce jour recense plus de 90 éditeurs. Mais, en réalité, peu d'entre eux sont encore aujourd'hui disponibles ou maintenus. De ce fait, seuls les trois éditeurs les plus aboutis actuellement seront présentés⁴⁷. Tout d'abord le logiciel Open Source Protégé, TopBraid Composer qui dispose d'une double licence et enfin, OntoStudio complètement propriétaire. Ce choix volontaire de présenter ces trois logiciels aux licences différentes illustre également la capacité de l'ingénierie ontologique à proposer une solution adaptée à chaque situation.

2.4.1.1 Protégé

Protégé⁴⁸, dont le développement a été initié en 1995 par l'Université de Stanford, est l'un des projets les plus renommés (Khondoker, and Mueller, 2010). Son architecture modulaire et flexible permet l'intégration de nombreux langages (OWL, OWL 2, DAML+OIL, etc.) et plugins . Avant la version 4 de Protégé, deux modes de modélisation d'ontologies étaient disponibles :

- **Protégé-Frames** : il se base, comme son nom l'indique, sur le modèle de schéma (frames). Une ontologie se compose alors, d'un ensemble de classes organisé hiérarchiquement, de slots décrivant les attributs et les propriétés de ces classes et enfin d'instances de classes (Gennari et al., 2003).
- **Protégé-OWL** : spécialement adapté au format OWL, il permet notamment de visualiser, éditer ou ajouter des classes, des instances et des propriétés OWL mais aussi d'inférer de nouveaux faits grâce aux raisonneurs (Hermit, Pellet, etc.) directement intégrés (Knublauch et al., 2004) .

Seul l'éditeur Protégé-OWL a été porté vers la nouvelle version de Protégé (version 4). Développée dans le cadre du projet CO-ODE⁴⁹, Protégé 4 se base désormais sur l'API Open Source OWL API (voir section 2.4.2.1) et propose de nouvelles fonctionnalités comme la possibilité de charger plusieurs ontologies sur le même espace de travail ou encore un outil de fusion d'ontologies (Figure 2.7).

46 http://www.xml.com/2004/07/14/examples/Ontology_Editor_Survey_2004_Table_-_Michael_Denny.pdf

47 Une liste plus complète est également disponible sur le site du W3C :
http://www.w3.org/2001/sw/wiki/Category:Development_Environment

48 <http://protege.stanford.edu/>

49 <http://www.co-ode.org/>

2.4. Conception et exploitation d'ontologies

Le choix de la version 4 ou d'une version antérieure de Protégé est un critère à prendre en compte lors du développement d'une ontologie. En effet, du fait de son antériorité la précédente version (3.4) disposait d'un grand nombre de plugins et de fonctionnalités (version multi-utilisateurs, stockage des données dans une base relationnelle, etc) qui n'ont pas encore été portés vers Protégé 4. Néanmoins, cette situation devrait disparaître progressivement au fur et à mesure de l'adoption de cette nouvelle version par la communauté des utilisateurs.

Enfin, il est également nécessaire de souligner sa très large adoption tant par la communauté scientifique que par la communauté de développeur.

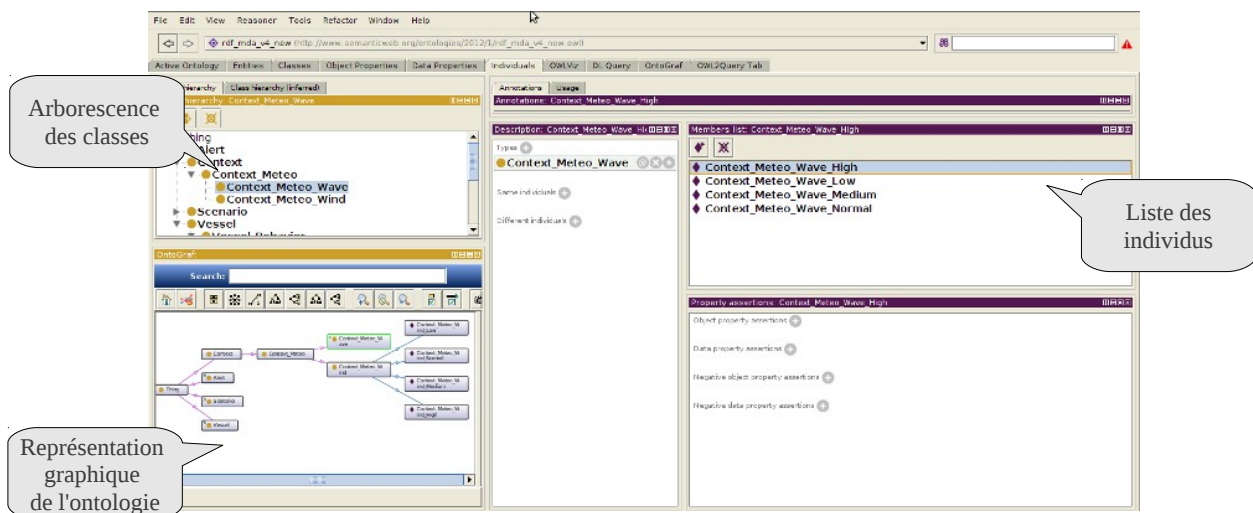


Figure 2.7: Interface du logiciel Protégé

2.4.1.2 TopBraid Composer

Développé par la compagnie TopQuadrant, TopBraid Composer⁵⁰ s'appuie sur la flexibilité de la plateforme de développement Eclipse⁵¹ et sur la librairie Jena (voir section 2.4.2.2) pour offrir un environnement d'édition d'ontologie (Figure 2.8). Trois versions sont proposées par l'éditeur : Free Edition, Standard Edition et Maestro Edition. Si la première peut être librement téléchargée, les fonctionnalités sont néanmoins très limitées. En effet, il est uniquement possible d'éditer (au format RDF/XML, N3 et N-Triples) et d'interroger (avec SPARQL) une ontologie. Les deux autres, plus complètes, nécessitent néanmoins l'acquisition d'une licence d'utilisation. En fonction des besoins, une comparaison des fonctionnalités est disponible sur le site de l'éditeur⁵². Une version web, nommée TopBraid Enterprise Vocabulary Net⁵³, est également proposée. Néanmoins, celle-ci est disponible uniquement pour les versions Standard et Maestro.

50 http://www.topquadrant.com/products/TB_Composer.html

51 <http://www.eclipse.org/>

52 http://www.topquadrant.com/products/TB_Composer.html#features

53 http://www.topquadrant.com/solutions/ent_vocab_net.html

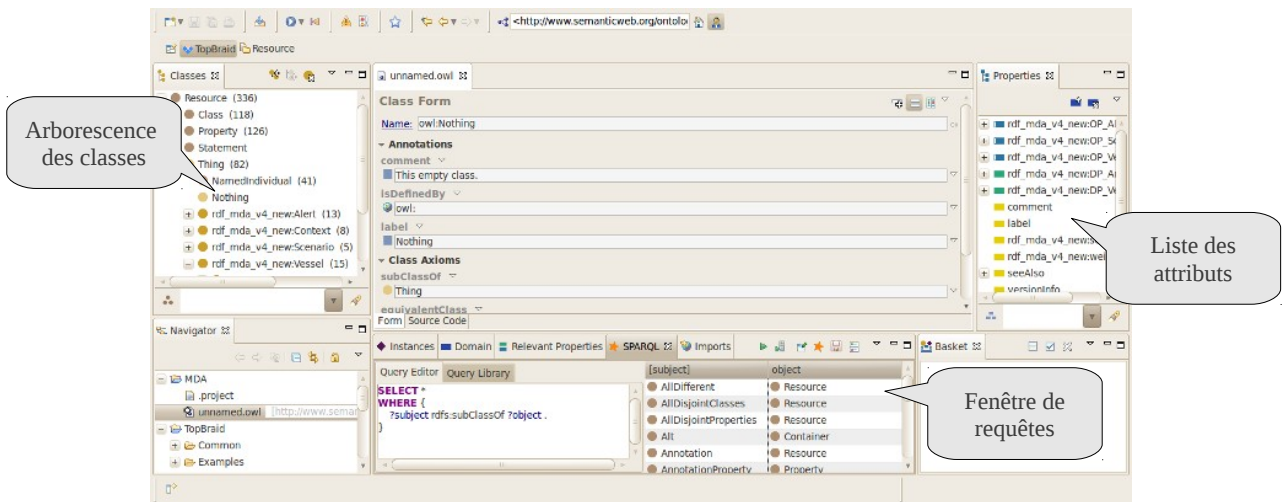


Figure 2.8: Interface du logiciel TopBraid Composer

2.4.1.3 OntoStudio

Également basé sur Eclipse et développé par la société Ontoprise⁵⁴, OntoStudio⁵⁴ est un logiciel propriétaire permettant le développement d'ontologies dans différents formats comme le OWL ou encore le RDF(S). De plus, il possède de nombreux connecteurs lui permettant de s'interfacer aussi bien avec des bases de données que des services web. Enfin, tout comme les précédents éditeurs, il permet l'édition collaborative d'ontologies. Signalons également qu'une version Open Source de OntoStudio, nommé NeOn Toolkit⁵⁵ existe.

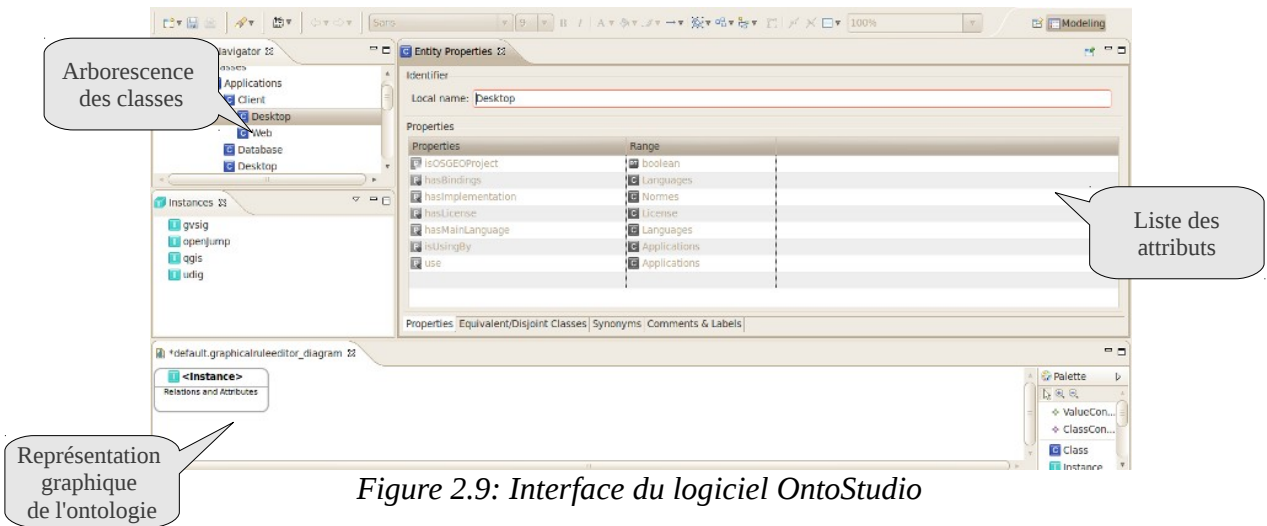


Figure 2.9: Interface du logiciel OntoStudio

54 <http://www.ontoprise.de/en/products/ontostudio/>

55 http://neon-toolkit.org/wiki/Main_Page

2.4.1.4 Conclusion sur les éditeurs d'ontologies

Comme cela a déjà été souligné, bien que l'offre d'éditeur d'ontologies soit riche, leur qualité est néanmoins inégale. En effet, les différents logiciels répertoriés dans les publications sont pour certains obsolètes ou introuvables. Par contre, d'autres ont continué à s'améliorer et sont pleinement opérationnels. Le triptyque de solution qui a été présenté allant du logiciel Open Source à sa déclinaison commerciale couvre les différents besoins que l'on peut rencontrer lors de la création d'ontologies.

2.4.2. Interfaces de programmation pour la manipulation d'ontologies

Une interface de programmation (API) peut être définie comme une bibliothèque de méthodes facilitant et automatisant le développement de programmes informatiques. Dans le domaine de l'ontologie, ces API permettent par exemple de manipuler différents formats (OWL, RDF, etc.) ou encore d'effectuer des raisonnements. Contrairement aux éditeurs présentés dans le chapitre ci-dessus, la manipulation d'une ontologie ne se fait pas au travers d'une interface homme-machine mais directement à partir de codes informatiques. Parmi les différentes bibliothèques existantes deux d'entre elles sont particulièrement populaires, il s'agit de OWL API et JENA qui seront présentées en détails dans les paragraphes suivants.

2.4.2.1 OWL API

OWL API⁵⁶ est une interface de programmation (API) Open Source écrite en langage Java. Initiée en 2003, elle est aujourd'hui activement maintenue par l'université de Ulm⁵⁷ et par l'entreprise Clark & Parsia⁵⁸ (qui est aussi à l'initiative du moteur d'inférence pellet). Elle a été ou est encore utilisée dans de nombreux projets comme Protégé (dans sa version 4), SWOOP ou encore NeOn Toolkit. Conçue pour la manipulation d'ontologies, ses fonctionnalités majeures sont (Bechhofer, Volz, et al., 2003):

- **sérialisation** : la sérialisation correspond au passage d'une représentation interne des données telle que modélisée par l'API à une syntaxe OWL
- **traitement** : le traitement (parsing) correspond au processus inverse de la sérialisation. Il consiste à traduire un document OWL en une représentation interne à OWL API.
- **modélisation** : si la représentation d'informations sous la forme d'un document XML est pertinente dans une ontologie, cela n'est pas le cas au sein d'un programme informatique. Il est donc nécessaire d'offrir un niveau d'abstraction élevé permettant de manipuler et d'encoder des documents OWL.
- **manipulation** : OWL API doit fournir les mécanismes nécessaires à la manipulation de document OWL.
- **inférence** : le processus d'inférence doit permettre la représentation et l'analyse de la sémantique des données contenues dans le document OWL.

Dans OWL-API, ces fonctionnalités sont traduites sous la forme d'interfaces permettant de manipuler facilement une ontologie. Il est par exemple possible d'accéder en lecture et en écriture à

56 <http://owlapi.sourceforge.net/>

57 <http://www.uni-ulm.de/in/ki/noppens>

58 <http://clarkparsia.com/>

des ontologies écrites dans différents formats comme le RDF/XML, Turtle, OWL/XML, ou encore Manchester OWL Syntax. OWL API permet également de raisonner à partir d'une ontologie grâce à l'intégration de différents moteurs d'inférence comme Fact++, Pellet, Racer Pro, etc. Concernant les possibilités de requêtes, celles-ci sont encore très limitées. Néanmoins, des implémentations sont en cours et devraient permettre d'utiliser nativement différents langages de requête comme SPARQL (Horridge and Bechhofer, 2011). Enfin, bien que le mode de stockage de l'ontologie se fasse sous la forme d'un fichier il est également possible d'ajouter des extensions afin par exemple de stocker les informations dans une base de données relationnelle ou une base de données pour des triplets (Henß et al., 2009; Redmond, 2010).

2.4.2.2 JENA

Également écrit en langage Java, Jena⁵⁹ est un framework Open Source dont le développement a débuté en 2000 dans le cadre du projet HP Labs Semantic Web Program (McBride, 2001; Carroll et al., 2004). Il vise à offrir les éléments nécessaires à la construction d'applications pour le web sémantique. Pour cela, il permet notamment la création et la manipulation de graphes dans différents formats (OWL, RDF, etc.) ou encore l'utilisation d'un moteur de règles (Jena Rules) et d'un moteur d'inférences. Concernant ce dernier, il est également possible d'utiliser des moteurs plus complets à l'exemple de Pellet ou de Racer.

2.4.2.3 Conclusion sur les interfaces de programmation

OWL API, tout comme Jena, propose une interface de programmation de haut niveau pour la manipulation d'ontologies. L'objectif ici n'est pas de comparer ces deux API mais plutôt de dégager les principaux avantages de ces deux solutions.

Bien que Jena affiche des résultats de performance supérieurs (Garcia-Castro et al., 2010), c'est pourtant OWL API qui est aujourd'hui majoritairement utilisé. De nombreux projets se basent sur cette librairie et le choix de l'équipe du logiciel Protégé de passer de Jena à OWL API est une preuve concrète de ce dynamisme.

2.4.3. Reasonner à partir d'une ontologie

Un moteur d'inférence, que l'on retrouve également dans la littérature sous le terme de raisonneur, a notamment pour objectif de : vérifier la consistance de l'ontologie, déduire de nouvelles informations à partir de la base de connaissances ou encore d'interroger l'ontologie. La plupart des moteurs existant actuellement sont conçus pour raisonner à partir des logiques de description et peuvent utiliser en entrée un fichier exprimé selon le standard OWL ou RDF(S).

Plus d'une vingtaine de moteurs d'inférence est actuellement recensée⁶⁰. Néanmoins, ces derniers se démarquent par le langage de programmation utilisé ou par leurs fonctionnalités (Mishra and Kumar, 2010). Plutôt que d'énumérer chacun d'entre eux séparément, il apparaît plus intéressant de ne présenter que les quatre (faCT++, Kaon2, Pellet et racerPro) implémentant le protocole Description Logic Implementation Group (DIG) (Bechhofer, Möller, et al., 2003). Celui-ci permet de s'absoudre des particularités des différents moteurs de raisonnement en proposant une interface de communication unifiée basée sur des requêtes HTTP.

59 <http://incubator.apache.org/jena/>

60 http://www.w3.org/2001/sw/wiki/Category:OWL_Reasoner et <http://www.cs.man.ac.uk/~sattler/reasoners.html>

Les différentes caractéristiques de ces chacun des quatre moteurs sont présentées ci-dessous et une vision synthétique de cette analyse est également disponible dans le Tableau 1.3 en fin de section :

- **Fact++** : Initialement développé par Ian Horrocks, Fact++⁶¹ est une implémentation en langage C++ du moteur de raisonnement Fact (Horrocks, 1998) écrit en Lisp. Il se base sur des algorithmes de vérification de satisfiabilité à base de tableaux et propose une interface Java (Tsarkov and Horrocks, 2006).
- **KAON2** : Successeur de Kaon (KArlsruhe ONtology) qui utilisait un langage de description propriétaire, Kaon2⁶² se base quant à lui sur OWL-DL. Contrairement aux autres raisonneurs, KAON2 s'appuie sur un processus d'inférence hybride basé sur Datalog (Hustadt et al., 2004, 2007).
- **Pellet** : Pellet⁶³ dispose d'un grand nombre de fonctionnalités comme une compatibilité avec le langage OWL 2, le support des interfaces de programmations Jena et OWL API ainsi que la possibilité de créer de nouvelles fonctions SWRL (built-in) (Sirin et al., 2007).
- **RacerPro** : Commercialisé par la société Racer Systems GmbH & Co. KG, RacerPro⁶⁴ est un moteur d'inférence écrit en Lisp. Il propose néanmoins une interface Java et se base sur des algorithmes de vérification de satisfiabilité à base de tableaux.

61 <http://owl.man.ac.uk/factplusplus/>

62 <http://kaon2.semanticweb.org/>

63 <http://clarkparsia.com/pellet>

64 <http://www.racer-systems.com/products/racerpro/>

Moteur	FaCT++	Kaon2	Pellet	RacerPro
Expressivité	SROIQ(D)	SHIQ(D)	SROIQ(D)	SHIQ(D)-
Implémentation	C++	Java	Java	Lisp
Interface de programmation	Java	/	/	Java
Interface avec OWL API/ Jena	OWL API	/	OWL API Jena	OWL API
Inférence	TBox/ABox	TBox/ABox	TBox/ABox	TBox/ABox
Algorithme de raisonnement	Tableau	Resolution & Datalog	Tableau	Tableau
OWL	OWL-DL	OWL-DL	OWL-DL	OWL-DL
DIG interface	Oui	Oui	Oui	Oui
Support des règles	Non	Oui (SWRL et DL Safe Rules)	Oui (SWRL et DL Safe Rules)	Oui (mais support incomplet de SWRL)
Licence	Free & Open Source	Free & Closed Source / Commerciale (OntoBroker)	Free & Open Source / Commerciale	Commerciale

Tableau 1.3: Analyse comparative des principaux moteurs d'inférence pour les logiques de description

Conclusion du chapitre

Modéliser la connaissance implique de disposer de structures permettant de traduire les schémas cognitifs des individus en une représentation informatique riche et complète. À l'heure actuelle, c'est le le domaine des ontologies qui offre les formalismes et les langages les plus aboutis.

Plus particulièrement les principaux travaux et avancées se focalisent autour d'un concept central le Web Sémantique. Bien que l'intitulé de cette notion renvoie au monde de l'internet, son utilisation est bien plus large. En effet, les langages issus du Web Sémantique tels que le OWL ou le RDF sont exploités dans de multiples domaines comme la médecine, l'informatique ou encore la géomatique. Ces langages offrent à la fois les composants nécessaires à la modélisation des connaissances, de multiples moteurs d'inférences ainsi que différentes applications annexes.

Dans le cadre de ce travail de thèse, c'est le langage OWL qui a été choisi pour la construction de l'ontologie. Néanmoins, celui-ci ne permet pas la représentation des notions géographiques et ne dispose pas non plus de mécanismes d'inférence adaptés au domaine spatial.

Différents travaux sont en cours pour combler ce manque. Ces derniers sont décrits et analysés dans le chapitre suivant.

CHAPITRE 3. REPRÉSENTATION ET RAISONNEMENT SPATIAL DANS LES ONTOLOGIES

Chapitre 3.Représentation et raisonnement spatial dans les ontologies.....	93
3.1. De la perception de l'espace à sa représentation ontologique.....	95
3.1.1. <i>De la difficulté à représenter l'espace.....</i>	<i>95</i>
3.1.2. <i>Différents niveaux de représentation de l'espace.....</i>	<i>96</i>
3.2. Modélisation des concepts géographiques et spatiaux à l'aide d'une ontologie ..	98
3.2.1. <i>Des ontologies géographiques aux ontologies géospatiales.....</i>	<i>98</i>
3.2.2. <i>Domaine d'application des ontologies géospatiales.....</i>	<i>100</i>
3.2.3. <i>Typologie des ontologies géospatiales.....</i>	<i>100</i>
3.2.4. <i>Ontologies géospatiales existantes.....</i>	<i>102</i>
3.2.5. <i>Architecture du Web Sémantique géospatial.....</i>	<i>105</i>
3.3. Raisonnement spatial dans les ontologies.....	109
3.3.1. <i>Le raisonnement spatial qualitatif, le cas de la topologie.....</i>	<i>109</i>
3.3.2. <i>Raisonnement spatial topologique dans les ontologies.....</i>	<i>116</i>

Introduction du chapitre

Tout comme dans le domaine de l'Intelligence Artificielle (IA), on retrouve dans l'utilisation des ontologies appliquées aux Systèmes d'Information Géographique (SIG) cette orientation à la fois philosophique et informatique.

Du côté philosophique, c'est avant tout le besoin d'explorer, de caractériser et de formaliser le sens profond des termes, notions et concepts utilisés. L'objectif est d'arriver, grâce à des ontologies conceptuelles, à une meilleure interopérabilité sémantique aussi bien entre les personnes qu'entre les systèmes.

D'un autre côté, c'est l'application directe des ontologies dans un SIG tel que définie par Studer. C'est à dire, « une spécification formelle et explicite d'une conceptualisation partagée » (Studer et al., 1998). Dans ce cas, il s'agit de créer une abstraction logique de la réalité à l'aide de langages et de raisonnements spécifiques.

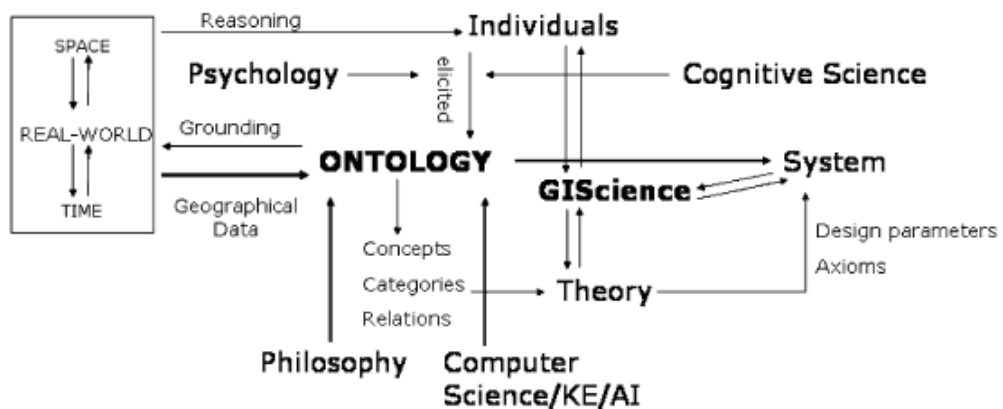


Figure 3.1. Connexions entre le domaine des ontologies et celui des SIG d'après (Agarwal, 2005)

Bien que distinctes, ces deux visions n'en sont pas pour autant complémentaires. En effet, pour être cohérente, une ontologie ne doit pas uniquement être consistante d'un point de vue logique, mais également représenter avec une certaine précision la manière dont les personnes perçoivent leur univers.

Dans les paragraphes ci-dessous, ces deux visions des ontologies seront présentées en commençant tout d'abord par l'aspect philosophique (§ 3.1). Cela permettra alors de comprendre les problèmes de perception géographiques et d'identifier les différents types d'ontologies existantes. Ensuite, la partie informatique relative aux ontologies spatiales (§ 3.2) sera abordée. Pour cela, les initiatives en cours concernant la standardisation des langages sémantiques pour la représentation d'informations géographiques seront détaillées. Enfin, les différents travaux concernant le raisonnement spatial dans les ontologies (§ 3.3) seront décrits.

3.1. De la perception de l'espace à sa représentation ontologique

L'espace, tout comme le temps, sont des notions manipulées quotidiennement. Pourtant, elles n'en sont pas moins subjectives et personnelles. En effet, l'information perçue et analysée passe au travers d'un certain nombre de filtres cognitifs et socioculturels personnels à chacun. Ces filtres sont fonction notamment des connaissances et de l'expérience de chaque personne. De ce fait, se pose alors le problème de la subjectivité de l'espace forcément et fortement dépendant à la fois de la situation et de l'interprétation de l'environnement lui-même. Cette problématique et les approches possibles afin de prendre en compte ces différences cognitives sont développées dans les paragraphes suivants.

3.1.1. De la difficulté à représenter l'espace

Dans le domaine spatial, quatre principaux facteurs de complexité de compréhension ont été identifiés (Chakhar, 2006): (1) la complexité liée aux données, (2) la complexité conceptuelle, (3) la complexité ontologique et sémantique, (4) la complexité technique.

Illustrons cela par l'exemple donné par Noucher (Noucher, 2009) dans le cadre d'une étude réalisée lors de sa thèse. Le contexte porte sur la création d'un SIG unique au sein d'une firme multinationale dans le domaine de la gestion de réseau. Ce déploiement nécessite la création d'un référentiel cartographique commun. Or, il advient que les acteurs impliqués dans ce processus n'ont pas la même vision de ce que représente un tronçon. Une partie d'entre eux n'arrive pas à établir un niveau d'abstraction suffisant pour passer d'un mode multifilaire 3D, proche de la réalité, à une représentation informatique en mode unifilaire 2D. Il en résulte alors un conflit cognitif qui aboutira à un rejet complet des négociations.

Ce cas est particulièrement révélateur des facteurs qui entrent en compte dans notre perception de l'espace. En effet, il a été démontré qu'il n'existe pas de concept universel associé aux entités géographiques, ce qui implique que la représentation de l'espace n'est pas universelle (Smith and Mark, 2003). En effet, les personnes, selon leur culture ou leurs origines utilisent ou pensent l'espace de différentes façons (Mark and Turk, 2003). Or, les SIG se basent sur une approche objective de la création des connaissances, ce qui signifie que ces systèmes représentent le « monde réel » indépendamment de la perception et de la cognition des individus.

Bien que Noucher (Noucher, 2009) porte un regard critique sur les capacités des ontologies à réaliser cette synchronisation cognitive, nous sommes au contraire persuadés qu'elles peuvent participer à harmoniser les points de vues des individus par rapport à un concept géographique. En effet, Laurini (Laurini, 2007) a notamment démontré que les ontologies pouvaient être utilisées pour aboutir à la fois à un consensus entre les acteurs d'un domaine, mais aussi à une interopérabilité entre systèmes d'information. Pour cela, il propose d'affiner sémantiquement les informations et concepts utilisés en se basant sur deux ontologies. La première, qu'il associe au processus de pré-consensus, consiste à identifier tous les concepts, attributs et points de vue des acteurs. Cela permet alors d'aboutir à une multiplicité des définitions à partir desquelles sera construite la seconde ontologie supérieure issue du consensus des acteurs. Néanmoins, afin que ces ontologies soient utiles, il faut qu'elles se basent sur les activités qu'ont les personnes de cet espace géographique (Kuhn, 2001). Ainsi, sans pour autant remettre en cause la perception subjective de l'espace, les ontologies apparaissent comme un moyen à la fois de synchronisation cognitive entre les individus mais aussi d'interopérabilité entre les systèmes.

3.1.2. Différents niveaux de représentation de l'espace

Il semble alors légitime d'envisager une perception et une cognition spatiales non plus uniques mais dépendantes des différentes visions que peuvent projeter les individus. Ces visions ont été synthétisées par Fonseca (Fonseca et al., 2002) sous la forme de quatre univers (Figure 3.2) :

- **l'univers physique** : c'est l'univers tel qu'il existe. Il est constitué des objets et phénomènes du monde réel.
- **l'univers logique** : qui représente d'une manière formelle les objets et phénomènes de l'univers physique.
- **l'univers des représentations** : dans lequel une description symbolique et mathématique des éléments est réalisée en vue de leur implémentation dans un langage informatique.
- **l'univers cognitif** : qui contient la perception des personnes à propos de l'univers physique

Afin d'éviter une trop grande dissonance cognitive, tout l'enjeu est alors de réussir à créer une abstraction de l'univers physique sous la forme d'un langage de représentation (l'univers des représentations) qui soit le plus proche possible de l'univers cognitif des individus.

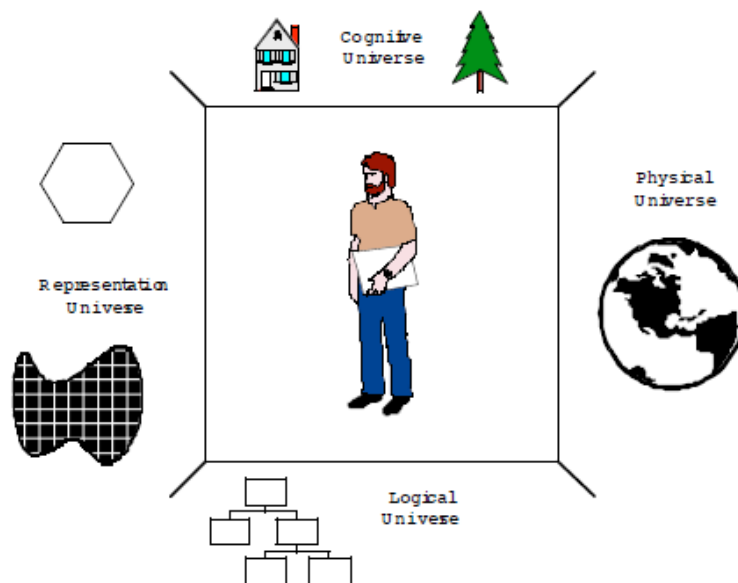


Figure 3.2: Les quatre paradigmes de l'information spatiale. Source (Fonseca et al., 2002)

Parallèlement à cette représentation sous la forme de quatre univers, Clementini et Laurini (Clementini and Laurini, 2008) ont proposé une vision ternaire des représentations spatiales. Celle-ci s'appuie sur trois niveaux de représentation : le niveau géométrique, le niveau informatique et le niveau utilisateur.

Au niveau géométrique, les objets spatiaux sont présentés sous une forme géométrique abstraite (points, régions, etc.) et les relations spatiales sont modélisées grâce à des propriétés géométriques spécifiques. Par exemple, dans le modèle des 9 intersections, que sera étudié dans la section 3.3.1.2, les relations topologiques sont définies en fonction des valeurs vides ou non vides des intersections entre la frontière, l'intérieur et l'extérieur de deux objets. Le niveau géométrique peut être considéré

3.1. De la perception de l'espace à sa représentation ontologique

comme le niveau de base pour l'étude des relations spatiales. En effet, c'est sur celui-ci que s'appuieront les deux autres niveaux.

Dans le niveau informatique, les objets du monde réel sont représentés sous la forme de types de données spatiaux. De plus, les relations spatiales doivent être calculées par des opérateurs spatiaux spécifiques (intersection, contient, etc.). Comme cela a déjà été souligné, ces types de données et ces relations se basent sur les modèles définis au niveau géométrique. Néanmoins, le niveau informatique introduit une part d'approximation en raison d'une représentation forcément simplifiée de la réalité. Par exemple, la précision des données raster est définie par la taille des cellules et pour les données vecteurs l'approximation sera fonction du nombre de nœuds utilisé pour représenter un objet. Les SIG, spécialisés dans le traitement de l'information géographique, sont pour la plupart capables de manipuler indifféremment le mode vecteur et le mode raster. Mais certains d'entre eux sont spécialisés dans l'un de ces modes comme Idrisi⁶⁵ ou Grass⁶⁶ pour le mode raster. Dans le domaine des bases de données géographiques, celles-ci ont été pendant longtemps limitées à la manipulation de données vecteur (Baumann, 2007). Néanmoins, les nouvelles versions de Postgis⁶⁷ (2.0 ou supérieur) et d'Oracle⁶⁸ (10g ou supérieur) possèdent la particularité de gérer les données raster et de proposer des fonctions spatiales spécialisées (ré-échantillonnage, rotation, etc.).

Enfin, au niveau utilisateur, la représentation spatiale dépend essentiellement du contexte spécifique d'application. Ce contexte peut être alors défini à l'aide d'une ontologie du domaine dans laquelle seraient précisées les relations spatiales possibles entre les différents objets. En effet, en reprenant l'exemple précédent donné par Noucher (Noucher, 2009), on voit bien qu'un même objet peut avoir différentes représentations. Par exemple, le tronçon pourrait être représenté au niveau géométrique sous la forme : d'une ligne simple, complexe ou même d'une région bidimensionnelle.

La prise en compte de la dimension spatiale dans les ontologies doit être capable de satisfaire la vision ternaire qui a été présentée. Si le niveau géométrique est aujourd'hui formalisé grâce aux nombreux travaux notamment dans le domaine du raisonnement spatial qualitatif (voir chapitre 3.3), il en est tout autrement pour les deux autres. En effet, bien que des travaux soient actuellement en cours, il n'existe aucun standard pour la représentation d'objets spatiaux sous une forme ontologique.

65 <http://www.clarklabs.org/products/idrisi.cfm>

66 <http://grass.fbk.eu/>

67 http://postgis.refractive.net/docs/RT_reference.html

68 http://docs.oracle.com/cd/B19306_01/appdev.102/b14254/geor_intro.htm

3.2. Modélisation des concepts géographiques et spatiaux à l'aide d'une ontologie

Comme cela a été mis en évidence dans le chapitre précédent, la notion d'espace est à la fois polymorphe et polysémique. En effet, celle-ci est dépendante des visions et des projections internes que se font chaque individu. Comment résoudre alors ces différences afin d'arriver à un consensus premier pas vers la collaboration ? Pour y répondre, l'approche choisie dans cette thèse est de s'appuyer sur les potentialités descriptives des ontologies. Néanmoins, celles-ci ne disposent pour le moment d'aucune fonction et d'aucun type spatial standardisés. Les différents travaux de recherche sur ce domaine ainsi que les limites actuelles sont présentés dans les paragraphes ci-dessous.

3.2.1. Des ontologies géographiques aux ontologies géospatiales

Le terme ontologie géographique ou également ontologie spatiale est la mise en commun de deux disciplines, de deux mondes aux concepts différents. Tout d'abord, celui des ontologies qui a déjà été présenté et celui de la géographie au sens large dans laquelle on retrouve cette tradition de description de l'espace mais aussi une application plus directe au sein des Systèmes d'Information Géographique (SIG). Mais, comme le souligne Agarwal (Agarwal, 2005) les différents travaux scientifiques alliant SIG et ontologies se sont essentiellement appuyés sur les potentialités de chacun des domaines sans pour autant former une véritable discipline commune. Par exemple, bien que ce projet soit à l'étude par le W3C⁶⁹, il n'existe actuellement aucun standard permettant la représentation de données spatiales dans les ontologies ni de procédures de raisonnement spatial dans les moteurs d'inférence.

« Currently, there is no comprehensive ontology for the geo-spatial domain, and it is recognized that there are no definitive methods for ontology derivation available to the geographic community » (Agarwal, 2005)

Ce constat est notamment dû aux particularités du domaine spatial qui nécessite de pouvoir (Cullot et al., 2003) :

- **définir des caractéristiques spatio-temporelles pour les concepts de l'ontologie** : cela implique la définition de types d'objets spatiaux (points, lignes, polygones, etc.) mais aussi des relations spatiales, notamment topologiques.
- **définir des concepts spatiaux à l'aide d'axiomes contenant des prédicats spatiaux** : par exemple, dans le domaine maritime, il pourrait être utile de définir dans l'ontologie une zone dangereuse comme étant l'intersection d'une zone non balisée et de toute autre zone possédant un risque pour la navigation (rochers affleurants, courants forts, etc.).
- **raisonner sur des objets spatiaux** : ces raisonnements doivent être à minima ceux implémentés actuellement dans les SIG, comme le raisonnement spatial qualitatif.

Mais, des différents travaux et projets de recherche il ressort que l'on retrouve indifféremment le terme ontologie géographique, ontologie spatiale, voire même parfois ontologie géo-spatiale. Si ces notions semblent similairement proches, elles font référence à des réalités différentes. A partir de l'analyse proposée par Spaccapietra (Spaccapietra et al., 2004) une première distinction entre ces

69 W3C Geospatial Incubator Group : <http://www.w3.org/2005/Incubator/geo/>

3.2. Modélisation des concepts géographiques et spatiaux à l'aide d'une ontologie

termes peut être opérée :

- **Les ontologies de l'espace** : elles font référence à une vision mathématique de l'espace et modélisent les objets existants sous une forme géométrique (points, lignes, polygones, etc.). La représentation de ces entités se fait sous une forme manipulable de manière informatique. Elle contient également des mécanismes d'inférence spécifiques au domaine spatial. À l'heure actuelle, aucune ontologie de ce type n'existe.
- **Les ontologies géographiques** : qui modélisent des concepts ayant une réalité géographique mais qui n'ont pas forcément d'entités spatiales rattachées. Ce sont essentiellement des ontologies métier comme celles notamment développées par l'Institut National de l'Information Géographique et Forestière (anciennement l'Institut Géographique National – IGN) (Abadie and Mustière, 2010).
- **Les ontologies spatialisées** : ces ontologies sont l'union des deux précédentes ontologies. Elles possèdent le plus haut niveau de formalisation. En effet, les entités du monde réel sont modélisées et possèdent chacune une référence spatiale explicite définie selon l'un des types d'objets spatiaux existants (point, ligne, polygone, etc.)

À partir de cette analyse et des différents projets existants (ex : Sweet⁷⁰, Ordnance Survey⁷¹, SWING⁷², etc.), cette classification peut être alors étendue en fonction du degré de formalisation de représentation des concepts spatiaux. Pour cela une catégorie supplémentaire nommée ontologie géospatiale hybride a été ajoutée (Figure 3.3). Cette ontologie contient une description d'éléments géographiques à laquelle est associée une localisation. Néanmoins, ce type d'ontologie est dite hybride car la référence spatiale ne peut être utilisée directement. Celle-ci est simulée en définissant par exemple la localisation d'une manière textuelle. De plus, les fonctionnalités spatiales et les mécanismes d'inférence devront faire appel à des applications tierces. C'est actuellement le fonctionnement adopté par la majorité des ontologies intégrées à un Système d'Information Géographique.

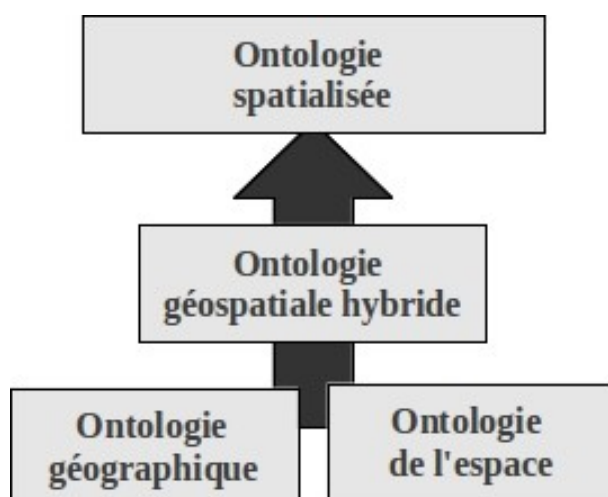


Figure 3.3: Typologie des ontologies selon le niveau de conceptualisation géospatiale

70 <http://sweet.jpl.nasa.gov/ontology/>

71 <http://www.ordnancesurvey.co.uk/oswebsite/ontology/>

72 <http://swing.brgm.fr/repository/ontologies>

3.2.2. Domaine d'application des ontologies géospatiales

Au début des années 2000, la communauté des Systèmes d'Information Géographique a porté un intérêt significatif à l'utilisation et l'application des ontologies. Les principaux travaux alors réalisés ont notamment porté sur l'utilisation des ontologies pour (Agarwal, 2005) :

- la gestion et l'acquisition des connaissances. Il s'agit dans ce cas de définir les termes d'un domaine et de proposer une modélisation ontologique des relations afin de les utiliser ensuite dans d'autres applications. Les travaux menés dans le cadre du projet Towntology (Roussey et al., 2004) ou encore ceux réalisés par l'Institut National de l'Information Géographique et Forestière (anciennement l'Institut Géographique National - IGN) (Abadie and Mustière, 2010) en sont un exemple.
- La spécification d'un domaine afin de favoriser l'interopérabilité aussi bien syntaxique que structurelle ou sémantique. Cette interopérabilité peut être réalisée grâce à un processus d'alignement sémantique qui consiste à mesurer la similarité entre des concepts proches ou similaires provenant de différentes ontologies. Un exemple d'alignement sémantique entre des bases de données SOLAP (Spatial on Line Analytical processing) a notamment été réalisé par (Sboui et al., 2009; Sboui and Bedard, 2010; Brodeur, 2012).
- le développement de systèmes d'informations comme cela a notamment été réalisé par Fonseca (Fonseca et al., 2002) pour la création d'un système dirigé par une ontologie (ODGIS – Ontology Driven Geographic Information Systems).

Ce vaste champ d'application a conduit à l'élaboration de nombreuses ontologies. Comme cela a déjà souligné, certaines d'entre elles portent uniquement sur la description de concepts géographiques ou d'autres proposent l'implémentation de types spatiaux.

3.2.3. Typologie des ontologies géospatiales

Les ontologies, tout comme leur rôle peuvent prendre différentes formes. Afin d'opérer une première distinction, il convient donc de définir une typologie de celles-ci. D'un point de vue plus conceptuel, Kolas (Kolas et al. 2005) a mis en évidence les éléments nécessaires à la construction d'un système géographique pour le web sémantique. Pour cela, il dénombre cinq ontologies :

- **une ontologie géospatiale de haut niveau** (*base geospatial ontology*) qui décrit le vocabulaire et la structure de la connaissance géographique.
- **une ontologie des entités** (*Feature data source ontology*) qui fournit une vision ontologique des entités géographiques. Cela est réalisé par la traduction du format Geography Markup Language (GML) sous un formalisme ontologique.
- **une ontologie de services géospatiaux** (*geospatial service ontology*) qui permet l'exploration et l'utilisation de tous les services géospatiaux (WMS, WFS, WCS, etc)
- **une ontologie des filtres spatiaux** (*geospatial filter ontology*) qui offre la possibilité de définir les relations potentielles entre les entités.
- **une ontologie du domaine** (*domain ontology*) qui représente la connaissance en fonction d'un domaine ou d'un groupe d'utilisateur spécifique.

3.2. Modélisation des concepts géographiques et spatiaux à l'aide d'une ontologie

Plus récemment, le groupe de travail Geospatial Incubator Group⁷³ du W3C a proposé une classification en sept catégories des ontologies spatiales pour le web sémantique (Lieberman et al., 2007a). Ce groupe de travail a pour objectif de définir les futures orientations à suivre en vue de l'implémentation de standards pour les ontologies géospatiales. On retrouve dans la classification du Geospatial Incubator Group certains éléments identifiés par Kolas :

- **ontologie géospatiale pour la description des caractéristiques spatiales des entités géographiques** : l'objectif de ces ontologies est de fournir une représentation formelle des caractéristiques spatiales des éléments (point, ligne, polygone, etc.) définies par les normes ISO⁷⁴ et OGC⁷⁵.
- **ontologie pour la définition de types d'éléments géographiques** : ces ontologies peuvent être vues comme des ontologies du domaine. Elles sont constituées de notions reconnues par une communauté étendue. Un exemple de ce type d'ontologies⁷⁶ est celle réalisée par l'Ordnance Survey, l'équivalent anglais de l'Institut National de l'Information Géographique et Forestière (IGN), qui a défini une ontologie des limites administratives⁷⁷.
- **ontologie pour la définition de relations spatiales** : l'utilisation des relations spatiales est l'une des caractéristiques majeures des SIG. De nombreux logiciels se basent notamment sur celles définies par le formalisme RCC8 ou encore sur celles d'Egenhofer. Néanmoins, l'utilisation de celles-ci demande des compétences spécifiques. Il serait plus simple d'imaginer une ontologie des relations utilisant un vocabulaire familier comme « derrière », « face à », « perpendiculaire à », etc. Cela permettrait ainsi une utilisation plus intuitive de ces relations. Un exemple d'ontologie de ce type a été réalisé par l'Ordnance Survey⁷⁸.
- **ontologie de noms de lieux** : de nombreuses ressources disponibles sur le Web sont identifiées et géolocalisées uniquement de par leur nom. Or celui-ci peut changer (Lutèce, Paris), faire référence à des endroits différents (Paris en France et Paris au Texas) ou s'écrire de différentes manières selon la langue. La création d'une ontologie des noms de lieux, à l'exemple de celle de Geonames⁷⁹, serait une solution aux limites identifiées. Par exemple, la recherche de la ville de Paris sur Geonames nous apprend les informations suivantes (en raison de la taille fichier XML seules quelques informations ont été gardées) :

```
<rdf:RDF>
  <gn:Feature rdf:about="http://sws.geonames.org/2988507/">
    <gn:name>Paris</gn:name>
    <gn:alternateName xml:lang="af">Parys</gn:alternateName>
    <gn:countryCode>FR</gn:countryCode>
    <gn:population>2138551</gn:population>
    <wgs84_pos:lat>48.85341</wgs84_pos:lat>
    <wgs84_pos:long>2.3488</wgs84_pos:long>
  </gn:Feature>
</rdf:RDF>
```

73 <http://www.w3.org/2005/Incubator/geo/>

74 International Organization for Formalisation - <http://www.iso.org/iso/home.html>

75 Open Geospatial Consortium - <http://www.opengeospatial.org/>

76 <http://www.ordnancesurvey.co.uk/oswebsite/ontology/>

77 <http://www.ordnancesurvey.co.uk/oswebsite/ontology/v1/AdministrativeGeography.owl>

78 <http://www.ordnancesurvey.co.uk/oswebsite/ontology/SpatialRelations/v0.2/SpatialRelations.owl>

79 <http://www.geonames.org/>

- **ontologie des systèmes de coordonnées** : bien que le système géodésique WGS84 soit suffisant au niveau mondial, d'autres systèmes locaux mieux adaptés existent également. Ces systèmes sont déjà décrits par l'OGC et par l'ISO qui ont définis des normes d'implémentations au format XML. Parmi les projets de traductions de ces normes vers le format OWL, citons notamment l'ontologie réalisée par l'Université de Drexel⁸⁰.
- **ontologie pour les métadonnées** : la qualité des données ainsi que leur provenance sont essentielles afin d'exploiter efficacement les ressources géographiques disponibles sur le Web. Bien que dans le domaine spatial des normes (ISO 19115 et ISO 19139) permettant la description de ces métadonnées existent déjà, ce n'est pas encore le cas pour le Web sémantique. Des questions essentielles comme la durée de validité d'une ressource géospatiale ou encore la représentation de ses propriétés n'ont pas encore été résolues.
- **ontologie pour les services Web spatiaux** : la mise en place et l'évolution des différentes normes comme le OWL-S (OWL pour les services Web) ont mis en évidence la nécessité de disposer de descriptions formelles et utilisables pour les services Web. Dans le cas du géospatial des précisions sont encore nécessaires concernant la manière dont ces services Web interagissent. Une description plus complète de ce dernier point est abordée dans le rapport réalisé par l'OGC (Lieberman, 2006).

Dans le cadre de cette thèse, les deux types d'ontologies géospatiales qui ont été utilisées sont celles portant sur la description des caractéristiques spatiales et celles sur les relations spatiales.

3.2.4. Ontologies géospatiales existantes

Bien qu'il existe déjà des projets modélisant la dimension spatiale sous une forme ontologique (ex : Sweet⁸¹, Ordnance Survey⁸², SWING⁸³, etc.) ces initiatives varient significativement quant à l'usage, la formalisation de représentation et la rigueur philosophique employée. En effet, l'étude réalisée par Ressler (Ressler, Dean, and Kolas, 2007) montre que plus d'une quarantaine d'ontologies proposent aujourd'hui une gestion complète ou partielle des différentes catégories identifiées dans la section 3.2.2 (p.100). Néanmoins, seules sept d'entre-elles offrent un support complet pour la description d'entités géographiques.

Afin de représenter concrètement les différences de perception et de représentation qui peuvent exister dans la modélisation de l'espace, deux ontologies de haut niveau intégrant cette dimension seront présentées.

3.2.4.1 Représentation spatiale dans l'ontologie SUMO

Suggested Upper Merged Ontology (Niles and Pease, 2001), plus connu sous l'acronyme SUMO⁸⁴, est une ontologie de haut niveau développée par le groupe de travail IEEE Standard Upper Ontology (SUO). L'objectif de SUMO est de constituer un standard afin de favoriser l'interopérabilité, la communication et la recherche dans le domaine du Web sémantique.

80 <http://loki.cae.drexel.edu/%7Ewbs/ontology/2004/09/iso-19111.owl>

81 <http://sweet.jpl.nasa.gov/ontology/>

82 <http://www.ordnancesurvey.co.uk/oswebsite/ontology/>

83 <http://swing.brgm.fr/repository/ontologies>

84 <http://www.ontologyportal.org/>

3.2. Modélisation des concepts géographiques et spatiaux à l'aide d'une ontologie

Développée en utilisant le langage KIF⁸⁵ (mais aussi disponible en OWL⁸⁶), SUMO a été réalisée en fusionnant différentes ontologies de haut niveau comme celle de John Sowa, de Russell et Norvig ou encore celle provenant du serveur ontolingua (Pease et al., 2002). Au niveau de la logique de conception, SUMO se divise en deux grandes catégories de classes à savoir celles définissant des concepts de très hauts niveaux et celles définissant des notions plus élémentaires.

Représentation de l'espace

Parmi les 25 000 termes et 80 000 axiomes, on retrouve de nombreuses catégories relatives aux entités géographiques comme une description des relations, attributs et fonctions (Bateman and Farrar, 2006). Dans SUMO, la représentation spatiale s'appuie sur les notions de méréologie et de topologie développées notamment par Casati et Varzi (Casati and Varzi, 1995; Varzi, 2007) par Smith (Smith, 1996) et Borgo (Borgo et al., 1996).

Conceptuellement, l'ontologie SUMO, distingue les objets dits physiques (*Physical*), de ceux dits abstraits (*Abstract*). La première catégorie fait référence à toutes les entités du monde réel qui existent à la fois dans le temps et dans l'espace. Ces entités possèdent donc automatiquement une relation de localisation et de temporalité. Cela est illustré par l'axiome ci-dessous qui définit un objet physique dans SUMO :

```
(<=> (instance ?PHYS Physical)
      (exists (?LOC ?TIME)
        (and
          (located ?PHYS ?LOC)
          (time ?PHYS ?TIME))))
```

La position ou la forme d'un objet peuvent également être précisées grâce aux classes *ShapeAttribute* et *PositionalAttribute*. Par exemple, *PositionalAttribute* peut prendre l'une des valeurs suivantes : *Vertical*, *Above*, *Adjacent*, *On*, etc. Les relations de cardinalité (nord, sud, etc.) sont également prises en compte par la classe *DirectionalAttribute* sous classe de *PositionalAttribute*.

Relations spatiales

Comme cela a déjà été abordé, SUMO contient également un grand nombre de relations spatiales. Dans l'ontologie, celles-ci sont toutes rattachées à la classe *SpatialRelation*. À cela s'ajoutent également différentes propriétés d'objets permettant de spécifier un lien spatial entre les entités. Par exemple, l'individu *connected* possède deux propriétés d'objets *overlapsPartially* et *meetsSpatially*. Grâce à ces relations, il est possible de spécifier que deux objets se chevauchent comme cela est illustré par l'axiome ci-dessous :

```
(<=>
  (overlapsSpatially ?OBJ1 ?OBJ2)
  exists (?OBJ3)
  (and
    (part ?OBJ3 ?OBJ1)
    (part ?OBJ3 ?OBJ2))))
```

85 Knowledge Interchange Format

86 <http://www.ontologyportal.org/translations/SUMO.owl.txt>

Bien que SUMO offre un inventaire riche des relations spatiales entre des entités, on note néanmoins qu'il n'existe aucune théorie unifiée de l'espace. L'une des raisons invoquées serait notamment les différentes influences philosophiques qui ont servi de base à sa construction. De plus, bien que la notion d'espace soit présente, aucun type spatial formel n'a été défini (Ressler, Dean, and Kolas, 2007).

3.2.4.2 Représentation spatiale dans l'ontologie Basic Formal ontology

Développé par l'Institute for Formal Ontology and Medical Information Sciences (IFOMIS), Basic Formal ontology (BFO) est une ontologie formelle de haut niveau modélisant les structures de base du monde réel. Son objectif est de proposer un cadre conceptuel qui puisse être utilisé pour la construction d'ontologie de n'importe quel domaine. La conception de BFO a été guidée en suivant les quatre principes suivant : *realist*, *perspectivalist*, *fallibilist* et *adequatist*. *Realist* implique que la réalité ainsi que ses constituants existent indépendamment de nos différentes représentations (linguistique, conceptuelle, théorique, culturelle, etc.). *Perspectivalist* signifie qu'il existe ou peut exister différentes perspectives de la réalité même si celles-ci restent toujours contraintes par le précepte *réaliste*. *Fallibilist* indique que les théories et classifications exprimées peuvent être sujettes à révision. Enfin, *adequatist* qui s'oppose au réductionnisme spécifie que les différentes visions alternatives ne peuvent se résumer à une vision unique (Grenon et al., 2004).

Représentation de l'espace

Dans BFO, la description du monde réel est séparée en deux catégories : les objets de type SPAN (par analogie au concept d'occurant) et les objets de type SNAP (par analogie au concept de continuant). SNAP est une théorie ontologique qui soutient que les entités ont une existence continue et capable de persister dans le temps. A l'inverse, SPAN fait référence aux entités, événements et activités sensibles au changement ou bien qui n'ont d'existence qu'un laps de temps donné. Par exemples, à l'échelle humaine, les montagnes ou encore les villes correspondent à des ontologies de type SNAP. A l'inverse des entités spatiales en mouvement seront modélisées par une ontologie de type SPAN (Grenon and Smith, 2004). Bien que distinctes, ces ontologies n'en sont pas moins complémentaires. C'est pourquoi, afin de faire le lien entre les différents types, BFO définit plusieurs relations formelles dont les principales sont *Intra-Ontological*, *Trans-ontological*, *Meta-Ontological* et *Ontological Indice of Relation* (Temal, 2008).

De par ses particularités, la dimension spatiale est à la fois présente dans la catégorie SNAP et SPAN. Dans la catégorie SNAP, l'espace est représenté comme une entité identifiable grâce à la classe *spatial_region* qui fait partie de l'une des trois classes de plus haut niveau. Cette position lui vaut d'être considérée comme une entité unique possédant ses propres particularités. Ainsi, toute instance de la classe *spatial_region* est donc nécessairement rattachée au concept d'espace. Cela peut être traduit par l'axiome ci-dessous (Grenon and Smith, 2004) :

$$\text{spatial_region}(x) \equiv \text{Part}(x, \text{space})$$

Dans la catégorie SPAN, il n'existe pas de concept spatial unique mais une classe nommée *spatiotemporal_region*. Ainsi, en accord avec le concept d'éternité temporelle définie par BFO, une région spatio-temporelle peut être décrite par l'axiome suivant :

$$\text{spatiotemporal_region}(x) \equiv \text{Part}(x, \text{spacetime})$$

3.2. Modélisation des concepts géographiques et spatiaux à l'aide d'une ontologie

BFO possède la particularité d'offrir une théorie ontologique fortement liée à l'aspect temporel. Cette orientation se traduit également dans la définition des concepts et des entités spatiaux, ce qui lui permet de prendre en compte les évolutions potentielles d'un objet géographique (Donnelly and Smith, 2003).

Relations spatiales

Contrairement à SUMO, BFO ne possède qu'un nombre limité de relations spatiales. Les concepteurs ont volontairement fait ce choix afin de laisser le soin aux modélisateurs de s'appuyer sur des ontologies spécialisées si cela est nécessaire. Ainsi, seules trois relations spatiales de très haut niveau sont disponibles : *SpatialLocation*, *SpatioTemporalLocation* et *Occupies*. *SpatialLocation* est considérée comme une primitive de SNAP. De ce fait, chaque entité SNAP possède dans l'ontologie une localisation spatiale (Grenon and Smith, 2004). La relation *SpatioTemporalLocation*, qui est également une relation primitive, décrit quant à elle la localisation d'une entité de type SPAN. Enfin, *Occupies* est un cas spécifique permettant de spécifier une relation spatiale entre des entités (*SubstantialEntity*) et un site (*Site*).

Ces différentes approches de la notion d'espace dans les ontologies ont mis en évidence la nécessité de disposer d'une théorie unifiée sur lesquels s'appuieront les futurs standards. À l'heure actuelle, ce travail est majoritairement réalisé par le W3C en raison de son implication dans le domaine du Web Sémantique.

3.2.5. Architecture du Web Sémantique géospatial

3.2.5.1 Intégration de la dimension spatiale dans l'architecture du Web Sémantique

Comme cela a déjà été souligné, l'intégration de la dimension spatiale au sein des ontologies est forcément liée à un critère de subjectivité. Si cette subjectivité est légitime, il est néanmoins nécessaire de disposer de standards sur lesquels s'appuyer afin de construire les prochaines applications géo-sémantiques. Ce travail est réalisé au sein du W3C par le Geospatial Incubator Group (GeoXG) dont l'objectif est de spécifier les futures orientations à suivre concernant la définition des propriétés géographiques des ressources du Web. Comme le montre la figure ci-dessous (Figure 3.4), cette intégration transversale de la dimension spatiale se fait aussi bien au niveau des capacités de raisonnement (Sparql et SWRL) que de la création de types géographiques dans les langages ontologiques (OWL) (Karmacharya, 2011; Karmacharya et al., 2011).

Dans la section suivante, l'intégration de la dimension spatiale dans le langage OWL sera abordé. En raison de sa complexité, les travaux portant sur le raisonnement spatial dans les ontologies feront l'objet d'un paragraphe spécifique (§ 3.3 p.109).

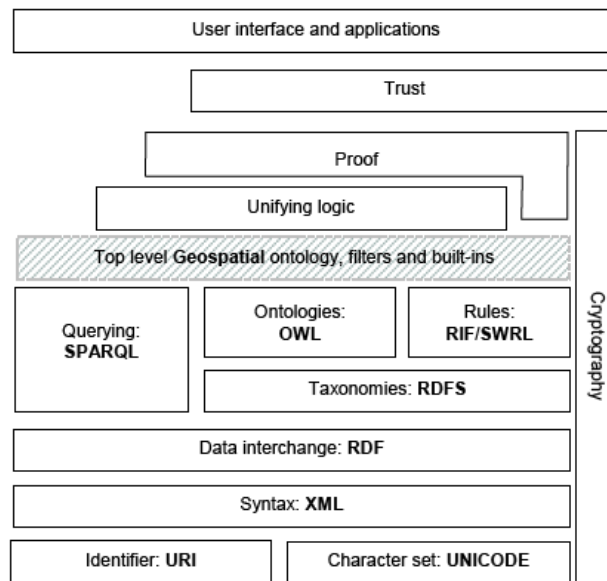


Figure 3.4: Intégration de la dimension spatiale dans l'architecture du Web Sémantique source : (Karmacharya, 2011)

3.2.5.2 Représentation ontologique des objets spatiaux

Avant de raisonner sur les relations spatiales dans une ontologie, il est nécessaire de pouvoir représenter et manipuler les entités géographiques. Si cela a déjà été formalisé dans le domaine des SIG, ce n'est pas le cas pour celui des ontologies. Par exemple, l'étude réalisée par Ressler (Ressler, Dean, and Kolas, 2007) montre que si plus d'une quarantaine d'ontologies offrent aujourd'hui une gestion complète ou partielle des différentes catégories d'ontologies géographiques identifiées dans la section 3.2.2 (p.100), seules sept d'entre-elles offrent un support complet pour la description d'entités géographiques. De plus, même si des travaux proposent déjà une ontologie pour la description des éléments, le standard Ontology Web Language (OWL) ne dispose à l'heure actuelle d'aucun type spatial (points, lignes, etc.).

Néanmoins cela pourrait changer avec l'adoption du GeoRSS comme recommandation du consortium W3C pour la description des propriétés géospatiales des ressources du Web. Dans sa forme la plus complète le GeoRSS s'appuie sur le standard Geography Markup Language (GML). Ces deux langages seront présentés dans les paragraphes suivants.

Geography Markup Language (GML)

Défini par l'Open Geospatial Consortium (OGC) mais également adopté comme standard ISO (ISO 19136), le GML⁸⁷ est un langage standardisé basé sur une grammaire XML. Il permet l'encodage, le stockage et l'échange de la géométrie et des attributs d'entités géographiques (Cox et al., 2004). Dans la philosophie de l'OGC, une entité géographique doit être comprise comme « une abstraction d'un phénomène du monde réel ; Une entité peut être qualifiée de géographique si une localisation relative à la terre lui est associée »⁸⁸.

87 <http://www.opengeospatial.org/standards/gml>

88 « A feature is an abstraction of a real world phenomenon; it is a geographic feature if it is associated with a location relative to the Earth »

3.2. Modélisation des concepts géographiques et spatiaux à l'aide d'une ontologie

Le format GML permet notamment de décrire des entités, des géométries, des systèmes de coordonnées, des unités de mesure ou encore des symbologies. Chacun de ces éléments est décrit dans un schéma GML⁸⁹ qui spécifie les normes de notation et de nommage. Par exemple, la définition d'une entité géométrique de type point devra respecter le schéma ci-dessous :

```
<complexType name="PointType">
  <complexContent>
    <extension base="gml:AbstractGeometricPrimitiveType">
      <sequence>
        <choice>
          <element ref="gml:pos"/>
          <element ref="gml:coordinates"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Ce schéma spécifie que l'encodage des coordonnées du point pourra se faire en utilisant l'élément *pos* ou *coordinates*. La différence entre ces deux éléments est très mince puisqu'il s'agit simplement du format de séparation entre la première et la seconde coordonnée. Pour *pos* la séparation sera réalisée par un espace et dans le cas de *coordinates* par une virgule. Par exemple, le stockage au format GML un point dont les coordonnées sont 3.11 et 42.51, cela se fera de la manière suivante :

```
<gml:Point>
  <gml:coordinates>3.11,42.51</gml:coordinates>
</gml:Point>
```

A ses débuts le standard GML ne permettait que la représentation de primitives géographiques simples (lignes, points, polygones, etc.). Mais, la version actuelle du standard GML (version 3) autorise la prise en compte de primitives plus complexes (courbes, surfaces, grilles raster, etc.), des éléments temporels et de la topologie.

GeoRSS

Conçu initialement pour ajouter une dimension géographique au flux RSS (Really Simple Syndication ou RDF Site Summary), GeoRSS a par la suite été adopté par le W3C comme vocabulaire de référence pour la description des propriétés géospatiales des ressources Web (Lieberman et al., 2007b). Issus des travaux du Geospatial Incubator Group (GeoXG) GeoRSS s'inspire du GML tout en le simplifiant afin d'être le plus générique possible.

Le schéma adopté (Figure 3.5) repose sur la relation générique *_featureproperty* qui peut être utilisée pour définir la localisation spatiale de n'importe quelle ressource sur le Web (*_content*). La relation *where* permet d'associer à la ressource à localiser une géométrie (*_geometry*) définie par l'un des quatre types de base : *point*, *line*, *box* et *polygon*. Cette localisation peut être encodée de trois façons. Soit en utilisant le W3C Geo qui se limite à la représentation de points dans le système de référence WGS84. Avec le format GeoRSS Simple qui supporte un plus grand nombre de géométries de base. Ou alors, avec le GeoRSS GML qui permet d'utiliser toutes les géométries

89 <http://schemas.opengis.net/gml/3.1.1/>

Chapitre 3. Représentation et raisonnement spatial dans les ontologies

définies dans le GML. Enfin, il est également possible de spécifier des informations complémentaires mais non géographiques comme le nom de l'entité (*featurename*), son type (*featuretype*), l'élévation (*elev*), ou encore la précision de la donnée (*radius*).

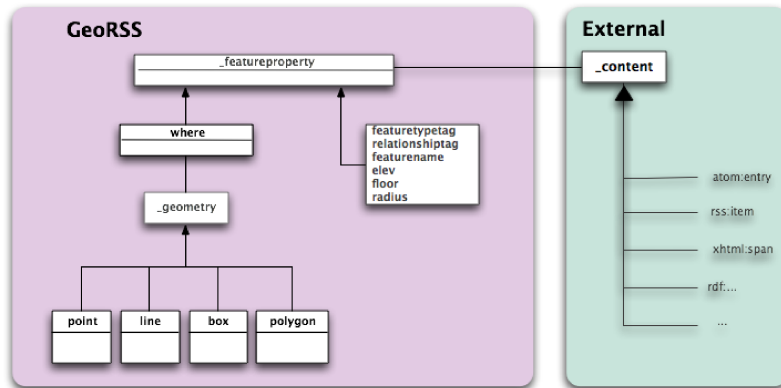


Figure 3.5: Représentation schématique du modèle GeoRSS (Source W3C)

Afin d'être le plus générique possible, GeoRSS se décline sous différentes formes : Atom, GML, RDF et OWL. L'encodage OWL du modèle GeoRSS Simple⁹⁰ reprend les éléments mis en évidence dans la Figure 3.5. En effet, on retrouve au centre de l'ontologie la classe *gml:_feature*. Celle-ci peut être reliée à une description géométrique (*gml:_Geometry*) au moyen de la propriété d'objets *where*. Les types géométriques disponibles (*gml:Point*, *gml:LineString*, *gml:Polygon*, *gml:LinearRing*, *gml:Envelope*) sont définis comme étant des spécialisations de la classe *gml:_Geometry* (Figure 3.6) (Dia Miron, 2009).

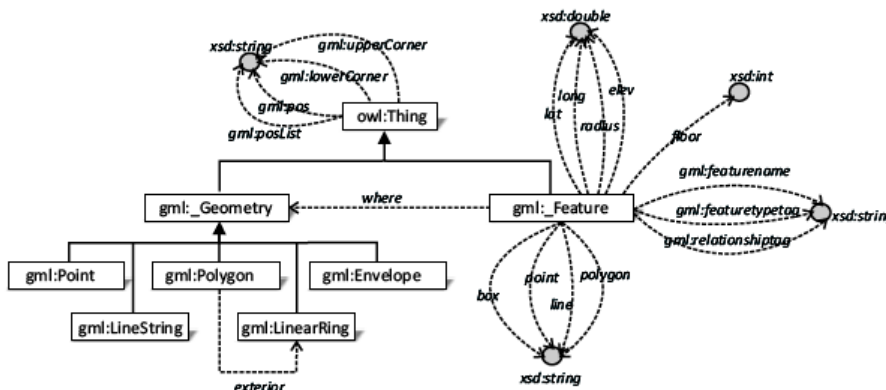


Figure 3.6: Représentation du GeoRSS au format OWL Source : (Dia Miron, 2009)

L'utilisation du GeoRSS dans le domaine des ontologies offre l'avantage d'une représentation spatiale simplifiée et de ce fait bénéficie d'une plus grande adoption par la communauté. Son adoption en tant que recommandation par le W3C est une grande avancée pour le domaine des ontologies spatiales. Néanmoins, il souffre d'une sémantique parfois trop limitée (Ressler, Dean, and Kolas, 2007). De plus, au-delà de cette formalisation des entités spatiales, très peu de travaux ont été initiés concernant l'implémentation de procédures de raisonnements spatiaux dans les ontologies.

90 <http://mapbureau.com/neogeo/neogeo.owl>

3.3. Raisonnement spatial dans les ontologies

Qu'il s'agisse d'indiquer son chemin à un passant ou de se rendre dans un lieu précis, nous manipulons au quotidien des structures spatiales à partir desquelles nous sommes capables d'effectuer des raisonnements. Cette manipulation intuitive de l'espace, que certains qualifient de naïve (Egenhofer and Mark, 1995), est un savant mélange d'intuitions géométriques reposant sur des raisonnements cognitifs spatiaux.

Appliqué à l'Intelligence Artificielle (IA) et aux Systèmes d'Information Géographique (SIG) se pose alors le problème de la transcription des connaissances entre les humains et le système informatique. Il s'agit ainsi de comprendre comment représenter et raisonner à partir d'entités géographiques ? Répondre à cette question nécessite d'identifier les principaux composants de l'espace à savoir les notions de topologies (contact et connexité), d'orientation (projection, positionnement, etc.), de distance et de forme (convexité, courbure, etc.). (Balbiani and Muller, 2000; Cohn and Hazarika, 2001; Renz, 2002).

Dans le domaine des mathématiques, l'espace peut être classifié selon une dimension quantitative ou qualitative. Dans cette dernière, la géométrie du monde repose sur des évocations spatiales sans représentation numérique précise ainsi que sur les connaissances intuitives que nous avons. Le raisonnement alors opéré est qualifié de Raisonnement Spatial Qualitatif (RSQ) (Cohn and Renz, 2008). Comparativement au raisonnement spatial quantitatif, le RSQ présente l'avantage de réduire la complexité des algorithmes, de diminuer les effets de bords et de permettre l'utilisation d'informations partielles ou incertaines (Cotteret, 2005).

Dans les paragraphes suivant le raisonnement spatial qualitatif et plus précisément les relations topologiques seront étudiés. Ce choix se justifie par le fait que ce sont celles qui ont été majoritairement utilisées dans ce travail de thèse. Seules les principales théories et les raisonnements qui peuvent être appliqués seront présentées. Néanmoins, une revue complète des différentes relations (topologie, orientation, etc.) est disponible dans (Dia Miron, 2009).

3.3.1. Le raisonnement spatial qualitatif, le cas de la topologie

Également appelée géométrie de la feuille de caoutchouc en raison de son invariance aux transformations linéaires (translation, rotation, agrandissement, etc.), la topologie est l'une des disciplines étudiant les relations spatiales la plus largement étudiée. Ainsi, on retrouve cette notion dans des domaines aussi variés que les SIG, les mathématiques, la robotique ou encore la linguistique. Celle-ci permet de modéliser les relations existantes entre des objets spatiaux au travers par exemple des notions de connectivité, d'inclusion ou encore de chevauchement.

Plus globalement, celle-ci est incluse dans une représentation beaucoup plus large de l'espace composée de trois catégories géométriques : topologiques, projectives et métriques. Ainsi, les relations spatiales peuvent être classifiées en fonction d'une représentation de l'espace sous une forme topologique, projective ou euclidienne. Ces trois espaces mathématiques sont liés hiérarchiquement ce qui signifie que toute propriété géométrique évaluée dans l'espace topologique reste valide dans l'espace projectif et euclidien. Par contre, l'inverse n'est pas forcément vrai (Clementini and Laurini, 2008).

La représentation de ces informations topologiques fait l'objet d'une abondante littérature scientifique et de nombreux modèles mathématiques ont été proposés (9-Intersections, Region

Connection Calculus, Dimension Extended Model, Calculus-Based Model, etc.). Généralement, tous ces modèles permettent au minimum de distinguer huit relations topologiques fondamentales identifiées par le Region Connection Calculus (RCC) entre des entités spatiales. C'est pourquoi cette théorie sera tout d'abord décrite pour ensuite présenter celle des 9-Intersections ainsi que son extension.

3.3.1.1 Region Connection Calculus (RCC)

Basé initialement sur les travaux de Clarke (Clarke, 1981), Randell *et al* (Randell and Cohn, 1989; Randell et al., 1992) ont développé un formalisme similaire au modèle temporel d'Allen (Allen, 1983) : le Region Connection Calculus (RCC). Ce formalisme décrit en logique du 1er ordre des relations spatiales entre des entités dont les primitives sont des régions.

Néanmoins, pour comprendre le RCC, il est nécessaire auparavant d'expliquer la théorie développée par Clarke. Celle-ci se base sur une primitive C spécifiant un lien de connexion entre deux régions. Ce lien peut être schématisé par la relation symétrique et réflexive suivante : $C(x,y)$. Cette relation doit être comprise comme « x est en connexion avec y ». Dans le formalisme RCC, l'interprétation de la relation $C(x,y)$ est légèrement modifiée afin d'exprimer le fait que la fermeture topologique entre deux régions partage un point. Bien que la notion de « connexion » propre au formalisme RCC soit plus limitée que celle définie par Clarke, elle reste néanmoins suffisante pour permettre un raisonnement spatial commun (Randell et al., 1992). Ainsi, la primitive $C(x,y)$ peut être utilisée pour définir huit relations élémentaires (Figure 3.7):

- **DisConnected** (DC) : x est déconnecté de y
- **Externally Connected** (EC) : x est connecté à y par sa frontière
- **Equal to** (EQ) : x et y sont identiques
- **Partial Overlaps** (PO) : x et y ont une partie commune
- **Tangential Proper Part** (TPP) : la région x est une partie tangentielle de y
- **Non Tangential proper Part** ($NTPP$) : la région x est une partie non tangentielle de y
- **Tangential Proper Part inverse** ($TPPi$) : la région y est une partie tangentielle de x
- **Non Tangential proper Part inverse** ($NTPPi$) : la région y est une partie non tangentielle de x

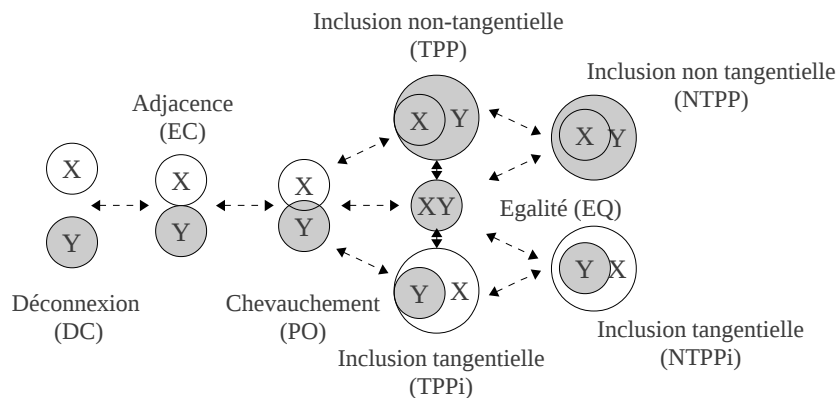


Figure 3.7: Illustration des 8 relations de base définies par le RCC (RCC8)

3.3. Raisonnement spatial dans les ontologies

En plus de ces huit relations élémentaires, signalons également l'existence d'algèbres simplifiées ou au contraire étendues comprenant une, deux, trois, cinq, quinze et vingt-trois relations. Ces sous-ensembles sont respectivement notés : RCC1, RCC2, RCC3, RCC5 et RCC15 et RCC23. Pour des raisons de clarté et parce que ce n'est pas l'objet de cette thèse les particularités de chacune de ces versions ne seront pas décrites. L'important est de retenir que ces autres algèbres sont toutes construites à partir des relations de base définies dans le RCC-8. Par exemple, la relation de *partie propre* (PP) qui est l'union des relations *Tangential Proper part of* (TPP) et *Non-Tangential Proper Part of* (NTPP) (Hogenboom et al., 2010).

Toutes les relations définies par Randell (Randell et al., 1992) sont également représentées sous la forme d'axiomes utilisant la logique de 1er ordre permettant ainsi d'effectuer des raisonnements. Pour cela, une table de composition a été définie pour chaque sous-ensemble des relations définies par le RCC (Tableau 2.1) (Gotts, 1996; Giritli, 2003). Cette table associe en entrée une relation spatiale entre deux régions x et y à une relation spatiale entre les régions y et z et donne en sortie un ensemble de relations spatiales entre les régions x et z .

A partir des relations définies par le RCC ainsi que des connaissances sur la position des objets, il peut être nécessaire de déduire de nouvelles positions ou relations. Pour cela, le RCC permet également de réaliser des raisonnements à partir d'une table de composition. Chaque sous-ensemble RCC dispose de sa propre table dans laquelle est spécifiée les relations possibles entre deux régions x et z à partir des relations entre deux régions x et y et des relations entre deux autres régions y et z . Le tableau ci-dessous (Tableau 2.1) représente les différentes relations topologiques qui peuvent être déduites à partir de deux relations initiales pour l'algèbre RCC-8. Illustrons cela par l'exemple suivant traduisant les relations entre Paris, la France et l'Europe. Les relations obtenues seraient alors les suivantes : NTPP(Paris, France) correspondant à $S(x,y)$ et NTPP(France, Europe) correspondant à $T(y,z)$. La table de composition de l'algèbre RCC-8 (Tableau 2.2) permet alors d'établir la relation NTPP(Paris, Europe) spécifiant qu'il existe bien une relation entre Paris et l'Europe sans pour autant partager de frontières communes.

Relation	Définition axiomatique de $R(x,y)$	Nom de la relation
$DC(x,y)$	$\neg C(x, y)$	DisConnected From
$EC(x,y)$	$C(x, y) \wedge \neg O(x, y)$	Externally Connected to
$O(x,y)$	$\exists z [P(z, x) \wedge P(z, y)]$	Overlaps
$P(x,y)$	$\forall z [C(z, x) \rightarrow C(z, y)]$	Part of
$PP(x,y)$	$P(x, y) \wedge \neg P(y, x)$	Proper Part of
$PO(x,y)$	$O(x, y) \wedge \neg P(x, y) \wedge \neg P(y, x)$	Partially Overlaps
$EQ(x,y)$	$P(x, y) \wedge P(y, x)$	EQual to
$TPP(x,y)$	$P(x, y) \wedge \exists z [EC(z, x) \wedge EC(z, y)]$	Tangential Proper part of
$NTPP(x,y)$	$P(x, y) \wedge \exists z [EC(z, x) \wedge EC(z, y)]$	Non-Tangential Proper Part of
$DR(x,y)$	$\neg O(x, y)$	DiscRete from
$ONE(x,y)$	$O(x, y) \wedge \neg EC(x, y)$	Overlaps not Equal

Tableau 2.1: Définition axiomatique des relations topologiques du RCC d'après (Randell et al., 1992)

°	DC	EC	PO	TPP	NTPP	TPPi	NTPPi	EQ
DC	°	DC, EC, PO, TPP, NTPP	DC, EC, PO, TPP, NTPP	DC, EC, PO, TPP, NTPP	DC, EC, PO, TPP, NTPP	DC	DC	DC
EC	DC, EC, PO, TPPi, NTPPi	DC, EC, PO, TPPi, NTPPi, EQ	DC, EC, PO, TPP, NTPP	EC, PO, TPP, NTPP	PO, TPP, NTPP	DC, EC	DC	EC
PO	DC, EC, PO, TPPi, NTPPi	DC, EC, PO, TPPi, NTPPi	°	PO, TPP, NTPP	PO, TPP, NTPP	DC, EC, PO, TPPi, NTPPi	DC, EC, PO, TPPi, NTPPi	PO
TPP	DC	DC, EC	DC, EC, PO, TPP, NTPP	TPP, NTPP	NTPP	DC, EC, PO, TPP, TPPi, EQ	DC, EC, PO, TPPi, NTPPi	TPP
NTPP	DC	DC	DC, EC, PO, TPP, NTPP	NTPP	NTPP	DC, EC, PO, TPP, NTPP	°	NTPP
TPPi	DC, EC, PO, TPPi, NTPPi	EC, PO, TPPi, NTPPi	PO, TPPi, NTPPi	PO, TPP, TPPi, EQ	PO, TPP, NTPP	TPPi, NTPPi	NTPPi	TPPi
NTPPi	DC, EC, PO, TPPi, NTPPi	PO, TPPi, NTPPi	PO, TPPi, NTPPi	PO, TPPi, NTPPi	PO, TPP, NTPP, TPPi, NTPPi, EQ	NTPPi	NTPPi	NTPPi
EQ	DC	EC	PO	TPP	NTPP	TPPi	NTPPi	EQ

Tableau 2.2: Tableau de composition pour les relations de l'algèbre RCC-8.

En ligne la relation spatiale entre deux régions x et y tel que $- S(x,y)$;

et en colonne la relation spatiale entre y et z tel que $- T(y,z)$.

Le résultat est l'ensemble des relations possibles entre x et z . source : (Dia Miron., 2009)

L'un des apports majeurs de cette théorie est d'expliquer grâce à des relations binaires primitives comment sont connectées deux entités spatiales. Elle est de ce fait très largement utilisée dans les Systèmes d'Information Géographique actuels (SIG). Néanmoins, le système RCC ne permet pas de distinguer les géométries ouvertes des fermées. Pour les auteurs cette distinction est contre-intuitive car aucun objet du monde réel n'est un ouvert. Même si cette pensée est partagée par de nombreux chercheurs, il est également prouvé que conceptuellement la pensée humaine est capable de manipuler des notions d'ouverture plus ou moins abstraites comme l'intérieur d'une pièce, d'un bâtiment, etc. (Larvor, 2004). De plus, comme le souligne Dia Miron (Dia Miron, 2009), les procédures d'inférences basées sur cette formalisation ne sont pas les plus efficaces et les raisonnements peuvent s'avérer parfois incomplets ou indécidables.

3.3.1.2 Modèle des 9 intersections et son extension DE-9IM

Proposé par Egenhofer et Herring (Egenhofer et al., 1994), le modèle des 9-intersections (9-IM) permet l'analyse des relations entre deux objets géographiques et cela pour toutes les primitives spatiales (point, ligne, polygone). L'approche utilisée consiste à déterminer les relations valides

entre l'intérieur (A°), l'extérieur (A^-) et la frontière (∂A) de deux objets (Figure 3.8) :

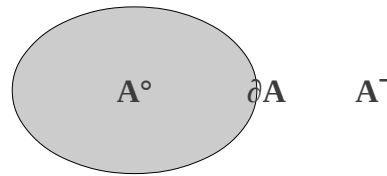


Figure 3.8: La décomposition d'une région selon le modèle des 9-Intersections

Ainsi, la relation topologique (notée R) existante entre les deux objets se base sur la comparaison entre l'intérieur (A°), la frontière (∂A) et l'extérieur (A^-) de l'objet A et l'intérieur (B°), la frontière (∂B) et l'extérieur (B^-) de l'objet B. Ce paradigme peut être représenté grâce à la matrice ci-dessous (Figure 3.9) :

$$R(A,B) = \begin{Bmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{Bmatrix}$$

Figure 3.9: Matrice du modèle des 9-intersections

Chacune des neuf comparaisons donnera une valeur non nulle en cas d'intersection (notée 0) et une valeur nulle (notée -0). Comme chaque cellule de la matrice peut prendre l'une ou l'autre de ces deux valeurs, il existe alors 2^9 (512) configurations possibles et donc autant de relations spatiales potentielles. Néanmoins, dans le cas de régions spatiales simples, c'est à dire de régions uniquement constituées d'une frontière externe sans frontière interne (autrement dit une région qui ne possède pas de trou), cet ensemble peut être ramené à huit possibilités correspondantes aux relations de l'algèbre RCC-8. La figure ci-dessous (Figure 3.10), est une représentation de ces huit relations sous la forme d'une matrice ainsi que son motif (pattern) associé (Kurata and Egenhofer, 2007). Ces motifs sont utilisés afin de rendre la visualisation des relations spatiales plus facilement compréhensibles par les utilisateurs (Noyon et al., 2007).

Chapitre 3. Représentation et raisonnement spatial dans les ontologies

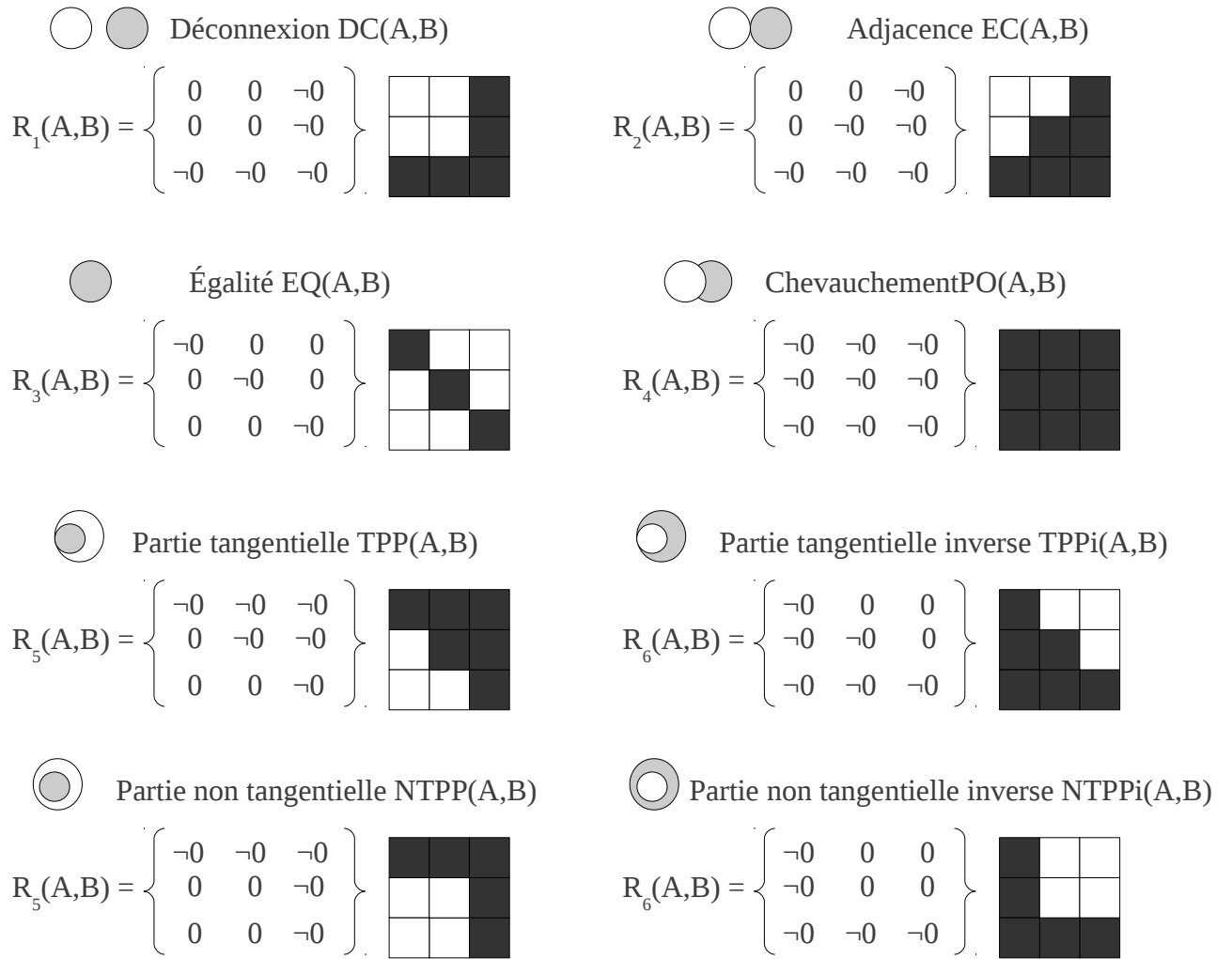


Figure 3.10: Représentation des huit relations topologiques en utilisant la méthode des 9-intersections et leur équivalence dans leur formalisme du RCC8

De nombreux travaux ont étendu le champ d'application de la méthode des 9-intersections, afin par exemple d'inclure les régions avec des trous (Egenhofer and Mark, 1994) ou de prendre en compte des entités de différentes dimensions (ligne, point, etc.) (Clementini and Di Felice, 1995). Ce dernier cas est particulièrement important car il est très largement utilisé dans les logiciels SIG actuels. Ainsi, l'extension proposée par Clementini, nommée Dimensionally Extended 9 Intersection Matrix (DE+9IM), permet de définir la relation spatiale entre des géométries de types et de dimensions différents. La dimension d'une intersection peut alors prendre les valeurs -1, 0, 1 ou 2 selon que l'intersection n'existe pas, qu'elle soit constituée d'un point, d'une ligne ou d'un polygone (Strobl, 2008). La matrice étendue des 9-Intersection est alors représentée de la manière suivante :

$$R(A,B) = \left\{ \begin{array}{ccc} \dim(A^\circ \cap B^\circ) & \dim(A^\circ \cap \partial B) & \dim(A^\circ \cap B^-) \\ \dim(\partial A \cap B^\circ) & \dim(\partial A \cap \partial B) & \dim(\partial A \cap B^-) \\ \dim(A^- \cap B^\circ) & \dim(A^- \cap \partial B) & \dim(A^- \cap B^-) \end{array} \right\}$$

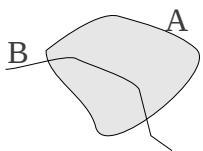
3.3. Raisonnement spatial dans les ontologies

La prise en compte du type et de la dimension des objets permet au modèle DE+9IM de représenter un plus grand nombre de relations. En effet, le modèle DE+9IM est capable de distinguer 87 relations topologiques réelles contre 68 pour le modèle 9IM. Le tableau ci-dessous (Tableau 2.3) résume le nombre de relations topologiques possibles pour chacun des deux modèles en fonction du type d'entités géographiques (R pour région, L pour ligne et P pour point) (Clementini and Laurini, 2008):

Modèle	R/R	L/R	P/R	L/L	P/L	P/P	Total
9IM	8	19	3	33	3	2	68
DE+9IM	12	31	3	36	3	2	87

Tableau 2.3: Résumé des relations topologiques pour les modèles 9IM et DE+9IM et pour toutes les dimensions entre objets simples d'après (Clementini and Laurini, 2008)

Comme chaque intersection peut prendre 4 valeurs différentes (-1, 0, 1, 2), il existe alors 4⁹ combinaisons possibles soit 262144 matrices différentes. Prenons par exemple le cas d'une ligne traversant un polygone. En se basant sur le modèle DE+9IM, La matrice résultante serait alors la suivante :



		B		
		Intérieur	Limite	Extérieur
A	Intérieur	1	-1	2
	Limite	0	-1	1
	Extérieur	1	0	2

Tableau 2.4: Exemple de relation entre deux objets géographiques et leur traduction dans le modèle des 9 Intersections. Les valeurs du tableau correspondent respectivement à :
-1 n'existe pas, 0 point, 1 polyligne et 2 surface.

Pour simplifier son écriture, la matrice DE+9IM est très souvent représentée sous la forme d'une chaîne de neuf caractères correspondant aux valeurs contenues dans les cellules de la matrice. Cette chaîne peut prendre comme valeur 0, 1 ou 2 correspondant à la dimension de l'intersection mais aussi le caractère T désignant une intersection non vide, F ou -1 une intersection vide et le signe * correspondant à valeur générique pour l'ensemble des autres valeurs. Par exemple, le prédicat d'égalité entre deux géométries sera noté de la manière suivante : *TFF FTF FFT*.

Le modèle DE+9IM est aujourd'hui très largement utilisé dans les solutions SIG existantes. On le retrouve notamment dans la base de données PostGIS⁹¹, dans la librairie Java Topology Suite (JTS)⁹² ou encore dans ArcGis⁹³.

91 http://postgis.org/documentation/manual-svn/using_postgis_dbmanagement.html#DE-9IM

92 <http://www.vividsolutions.com/jts/JTSHome.htm>

93 http://webhelp.esri.com/arcgisserver/9.3/java/index.htm#geodatabases/spatial_relationships.htm

3.3.2. Raisonnement spatial topologique dans les ontologies

Différentes approches ont été proposées afin d'effectuer des raisonnements spatiaux dans une ontologie. Certaines d'entre-elles se sont concentrées sur le formalisme utilisé et d'autre sur la modification du système de représentation de connaissance. Chacune de ces approches sont présentées dans les sections ci-dessous.

3.3.2.1 Ajout de la dimension spatiale dans les langages ontologiques

Il n'existe actuellement aucun standard permettant de représenter des informations géographiques ou des relations spatiales dans les logiques de description. Cependant, de nombreuses extensions ont été proposées. Parmi celles-ci, l'une des principales est le langage **ALCRP(D)** proposé par (Haarslev et al., 1998). Basé sur le langage **ALC(D)**, **ALCRP(D)** permet d'effectuer des raisonnements spatiaux qualitatifs entre des régions spatiales. Pour cela, l'extension propose de gérer les objets spatiaux et leurs relations à l'aide de prédicats (composante P de la logique) et sont exprimés sur un domaine concret de région de l'espace (S_2). Au nombre de huit, les relations topologiques implémentées sont celles définies par le Region Connection Calculus (RCC-8). Cette extension a notamment été mise en application au sein du système VISCO (Wessel et al., 2000) ainsi que dans le système MADS (Cullot et al., 2003). Notons également l'existence du langage **GeoSPARQL**⁹⁴ qui permet l'interrogation des données provenant de GeoNames en utilisant des prédicats spatiaux. Le soutien de l'OGC⁹⁵ à ce projet, préfigure certainement le futur du monde sémantique géographique.

Plus récemment, Stocker et Sirin (Stocker and Sirin, 2009) ont proposé d'intégrer au moteur de raisonnement Pellet⁹⁶ un extension spatiale⁹⁷ (PelletSpatial) permettant : (1) de vérifier la consistance d'un ensemble de relations spatiales exprimées avec le formalisme RCC8, (2) d'inférer de nouvelles relations, (3) de répondre à des requêtes spatiales exprimées en utilisant le langage SPARQL. Dans le cadre de leurs travaux et en raison des limites observées, Stocker et Sirin ont proposé deux mécanismes d'inférences. Le premier consiste à traduire les relations RCC-8 en axiomes de classes OWL et le second est basé sur la table de composition RCC-8.

Le premier raisonneur de PelletSpatial utilise une traduction des relations RCC-8 en axiomes OWL-DL comme cela a été initialement proposé par Katz et Grau (Katz and Grau, 2005) (Tableau 2.5). Pour cela, chaque relation RCC-8 est traduite sous la forme d'un axiome de classe (propriété d'objet) et chaque région RCC correspond à un concept valide en logique de description. Cette validité est soumise à deux conditions : (1) un concept ne peut être vide et (2) tous les points de la région doivent être contenus. Bien que fonctionnelle, cette approche souffre de mauvaises performances en raison du processus de traduction d'une logique vers une autre. En effet celui-ci s'avère être coûteux en temps de calcul et cela, même pour un nombre réduit de régions (Stocker and Sirin, 2009). Du fait de ces problèmes de performances, une seconde approche, qualifiée d'hybride, a été proposée. Dans celle-ci, les raisonnements spatiaux sont séparés des raisonnements OWL-DL par l'utilisation d'un raisonneur RCC spécialisé. Dans ce cas, la consistance des données spatiales est assurée par un algorithme de *path-consistency* basé sur la table de composition RCC-8 (cf Tableau 2.2). On retrouve d'ailleurs ici une logique similaire à celle que exposée dans la section suivante (3.3.2.2).

94 <http://www.geosparql.org/>

95 <http://www.opengeospatial.org/projects/groups/geosparqlswg>

96 <http://clarkparsia.com/pellet>

97 <http://clarkparsia.com/pellet/spatial>

Relation	OWL DL
EC(X,Y)	$\forall R.X \subseteq \exists R.\neg Y ; Z_1 \equiv X \cap Y$
EQ(X,Y)	$X \equiv Y$
DC(X,Y)	$X \subseteq \neg Y$
PO(X,Y)	$\forall R.X \cap \forall R.Y ; Z_3 \equiv X \cap \neg Y ; Z_4 \equiv \neg X \cap Y ;$
TPP(X,Y)	$X \subseteq Y ; Z_5 \equiv X \cap \exists R.\neg Y ;$
NTPP(X,Y)	$X \subseteq \forall R.Y$
TPPi(X,Y)	$Y \subseteq X ; Z_5 \equiv Y \cap \exists R.\neg X ;$
NTPPi(X,Y)	$Y \subseteq \forall R.X$

Tableau 2.5: Traduction des 8 relations du formalisme RCC8 en OWL DL d'après (Katz and Grau, 2005)

En plus de ce raisonneur spatial, PelletSpatial implémente également un langage de requête basé sur SPARQL. Il est ainsi possible d'interroger la base de connaissances en utilisant des critères de sélection spatiaux et non spatiaux. Par exemple, il est possible de demander à mettre en évidence tous les navires se trouvant dans une zone de pêche et qui sont enregistrés sous un pavillon de complaisance. En fonction du mécanisme d'inférence choisi, les algorithmes utilisés ainsi que les processus seront différents.

Pour celui effectuant une traduction des relations RCC-8 en OWL-DL, la requête SPARQL peut être directement traitée car les relations et les axiomes sont déjà stockés dans la base de connaissances. Néanmoins, toutes les relations RCC ne sont pas vérifiables à l'exemple de Externally Connected (EC). Cette limitation est due à l'architecture de Pellet qui ne permet pas l'utilisation de variable portant sur la restriction de rôle dans une requête SPARQL. Bien qu'une autre formulation de la requête soit possible cela affecte néanmoins les performances du raisonneur. Dans la seconde approche, du fait de son caractère hybride, la requête SPARQL doit être scindée en une partie spatiale et l'autre non spatiale. La première requête est alors réalisée par un moteur spatial spécifique et les résultats sont ensuite retournés pour être exécutés par la partie non spatiale.

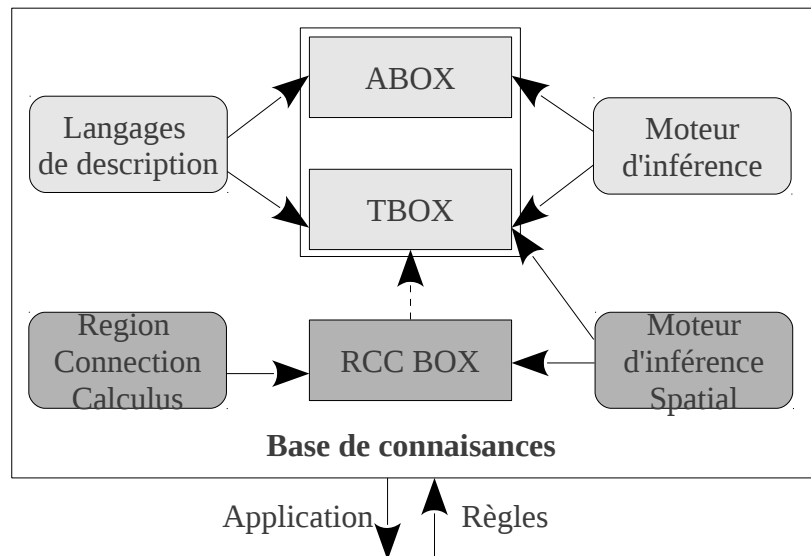
Pour le moment, PelletSpatial est avant tout une preuve de concept. Les performances du système ne sont pas optimales même dans le cas où peu de régions sont analysées. Il apparaît donc nécessaire de proposer une architecture alternative capable de s'adapter à une plus forte volumétrie de données.

3.3.2.2 Une approche hybride pour la représentation des connaissances spatiales

Comme cela a été abordé, la traduction des relations RCC8 en axiomes OWL pose d'importantes contraintes en terme de performances. C'est pourquoi, Grütter et Bauer-Messmer (Grütter and Bauer-Messmer, 2007) ont proposé une approche qualifiée d'hybride afin d'inclure des capacités de raisonnement spatial dans une base de connaissances. Dans ce cas, la combinaison entre RCC et OWL ne se fait pas au niveau du formalisme mais directement dans le système de représentation des connaissances. Pour cela, à partir de la représentation classique d'un système à base de connaissances telle que définie par (Baader and Nutt, 2003), les auteurs ajoutent une couche spatiale comprenant un composant principal nommé *RCC Box* (Graphique 3.1). Celui-ci contient les définitions des différentes relations ainsi que les tables de composition (cf Tableau 2.2) pour les différents sous-ensembles RCC implémentés (de RCC1 à RCC8). Le moteur d'inférence spatial

Chapitre 3. Représentation et raisonnement spatial dans les ontologies

utilise à la fois ces définitions ainsi que les assertions présentes dans la ABOX afin de calculer les relations spatiales existantes. Ensuite, les tables de composition sont utilisées pour s'assurer de la consistance de la ABOX. Afin d'optimiser le processus d'inférence, les résultats peuvent être stockés dans la ABOX comme cela est représenté dans le Graphique 3.1 par la ligne en pointillés. Bien que cela permette de ne pas avoir à calculer de nouveau le résultat de requêtes identiques cela entraîne par contre une augmentation de la taille de la base de connaissances. De plus, le stockage des inférences déjà réalisées soulève également la question de la véracité des résultats dans le temps. En effet, des entités peuvent être fusionnées ou détruites et les relations doivent être alors calculées de nouveau afin d'éviter que la base de connaissances soit inconsistante.



Graphique 3.1: Architecture hybride d'une base de connaissances incluant des raisonnements spatiaux basés sur le formalisme RCC8 d'après (Grütter and Bauer-Messmer, 2007)

Cette approche a été notamment démontrée afin de déterminer les relations spatiales existantes entre plus d'une quarantaine d'objets géographiques (Grütter et al., 2008). Ces objets correspondaient aux districts, communes et biotopes du canton de Zurich qui ont été intégrés à la ABOX en tant qu'individus. Dans le cadre de cette expérimentation 262 relations de type *connectsWith* – $C(x,y)$ ont été identifiées. Il est intéressant de noter que la définition et le nombre d'axiomes définis dans la base de connaissances influent considérablement sur la véracité des résultats et sur la rapidité des calculs. Néanmoins, il n'est fait aucunement mention des performances du système proposé.

Bien que toutes fonctionnelles, les différentes solutions proposées varient grandement en terme de performances et de capacité de raisonnement. Si l'intégration de concepts spatiaux directement au sein du formalisme de représentation semble être l'alternative la plus pertinente elle souffre néanmoins d'importantes limites notamment en termes de calcul. De ce fait, l'approche proposée par Grütter consistant à modifier le système de représentation des connaissances apparaît comme la solution la plus adaptée à l'heure actuelle. Mais, ce système hybride n'est capable de raisonner qu'à partir de relations déjà spécifiées dans la ABOX. Autrement dit, cela signifie qu'il n'est pas possible d'inférer des relations topologiques entre des objets en se basant uniquement sur leur position. Or, cela est une des fonctionnalités les plus basiques dans le domaine des SIG. Dans la suite de ce travail de thèse, les solutions qui ont été proposées afin de permettre l'interrogation d'objets spatiaux à partir de leurs positions et de leurs trajectoires seront détaillées.

Conclusion du chapitre

L'espace et le temps, bien que notions universelles, sont avant tout des visions personnelles. En effet, la représentation de celles-ci dépend de schéma cognitifs individuels. La question est alors de savoir comment traduire ces représentations sous une forme exploitable et partagée aussi bien entre les individus qu'entre les systèmes.

Les précédents chapitres ont montré que si les ontologies permettaient d'effectuer cette traduction, elles manquaient néanmoins des composants nécessaires à la représentation des notions géographiques. De ce fait, il est également impossible d'effectuer des raisonnements prenant en compte cette composante spatiale.

Face à cela, différents travaux issus d'organismes comme le W3C ou du monde de la recherche ont permis d'identifier les différents besoins et de proposer des pistes pour l'intégration de la dimension spatiale. Néanmoins, aucun consensus n'a pour le moment été établi et de ce fait il n'existe aucun formalisme standardisé.

La partie suivante explique l'approche adoptée dans le cadre de cette thèse pour décrire la composante spatiale nécessaire à l'analyse du comportement des navires au sein d'une ontologie. Les différents éléments ainsi que les modules nécessaires sont détaillés. Enfin, quelques cas d'utilisation sont expérimentés.

PARTIE 2 APPROCHE PROPOSÉE

CHAPITRE 4. UNE APPROCHE ONTOLOGIQUE POUR L'ANALYSE DES COMPORTEMENTS DE NAVIRES

Chapitre 4. Une approche ontologique pour l'analyse des comportements de navires.....	123
4.1. Modélisation du contexte sémantique	125
4.1.1. <i>Conceptualisation du modèle.....</i>	<i>125</i>
4.1.2. <i>Ontologie géométrique des trajectoires.....</i>	<i>126</i>
4.1.3. <i>Ontologie géographique</i>	<i>130</i>
4.2. Ontologie du domaine : l'analyse de comportements anormaux de navires. .	131
4.2.1. <i>Principales classes de l'ontologie du domaine.....</i>	<i>131</i>
4.2.2. <i>Modélisation des alertes.....</i>	<i>132</i>
4.2.3. <i>Modélisation des scénarios.....</i>	<i>134</i>
4.2.4. <i>Modélisation du contexte maritime</i>	<i>135</i>
4.3. Modélisation des règles d'inférence et enrichissement du SWRL.....	137
4.3.1. <i>Le langage SWRL.....</i>	<i>137</i>
4.3.2. <i>Modélisation des règles classiques.....</i>	<i>137</i>
4.3.3. <i>Modélisation des règles spatiales.....</i>	<i>138</i>

Introduction du chapitre

Modéliser les comportements anormaux de navires nécessite au préalable de formaliser la connaissance du domaine. Pour cela, l'approche adoptée a été d'utiliser les potentialités des ontologies. En effet, celles-ci offrent les éléments nécessaires à la description des notions mais aussi des liens qui les unissent tout en permettant d'effectuer des raisonnements.

Une infrastructure basée sur trois principales composantes a été développée. Ces composantes sont tout d'abord, une ontologie géométrique des trajectoires, une ontologie géographique et une ontologie du domaine. L'union de ces différentes ontologies permet alors de disposer des notions nécessaires à la description d'un comportement. Cette description a été réalisée à l'aide de la connaissance des experts puis intégrée au sein de différentes ontologies qui forment le cœur de la plate-forme développée dans cette thèse. Bien qu'il existe de nombreux types d'objets mobiles (sous-marins, ouragan, etc.) inhérent à la surveillance maritime, il est important de préciser que le modèle proposé dans le cadre de cette thèse s'applique uniquement aux relations potentielles entre des navires.

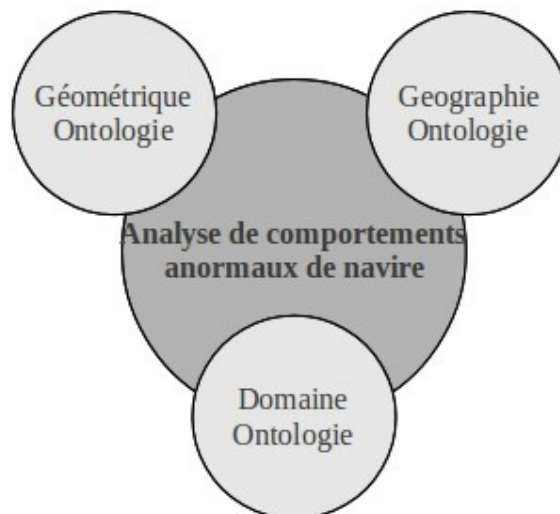


Figure 4.1: Les trois principales composantes ontologiques pour l'analyse de comportements anormaux de navires

Ce chapitre présente la démarche et l'approche adoptée. Dans un premier temps, les éléments constitutifs de l'ontologie créée seront détaillés (§4.1). Ensuite, ceux relatifs à l'analyse des comportements anormaux seront décrits plus spécifiquement (§4.2). Enfin, les règles d'inférences et plus spécifiquement l'intégration de la dimension spatiale au sein de ces règles seront étudiées (§4.3).

4.1. Modélisation du contexte sémantique

Afin de comprendre le comportement d'un navire, sa position seule ne suffit pas. En effet, ce comportement s'analyse au travers d'une multitude d'informations (condition météorologique, navires à proximité, etc.) qu'il est nécessaire de prendre en compte. C'est là toute l'idée du concept de comportement sémantique. Ainsi, à chaque position est attaché un ensemble d'informations jugé pertinent qui sera ensuite utilisé lors du processus d'inférence.

4.1.1. Conceptualisation du modèle

Créer une représentation sémantique des trajectoires et par extension une représentation des comportements nécessite d'enrichir les positions des navires par différentes informations contextuelles.

Pour cela, en s'appuyant sur le modèle initialement développé par Yan (Yan, 2009, 2011) trois principales ontologies ont été créées (Figure 4.2). La raison de ce découpage en différentes ontologies modulaires s'explique notamment par une plus grande facilité de création et de maintenance des ontologies ainsi que par une meilleure optimisation des requêtes potentielles. Ces ontologies sont :

- une ontologie des trajectoires contenant les différents concepts spatio-temporels nécessaires à la description géométrique des trajectoires. Ces concepts géométriques peuvent être donnés de manière absolue sous la forme d'objets géographiques (points, lignes, etc.) ou relative sous la forme d'adresses.
- une ontologie géographique contenant les concepts spécifiques à la description du territoire (routes, ports, baies, etc.). Cette ontologie est reliée à la fois à l'ontologie des trajectoires par les concepts géométriques qu'elle utilise et à l'ontologie du domaine pour les thématiques spécifiques qui peuvent être ajoutées en fonction des besoins.
- une ontologie du domaine qui, comme son nom l'indique, est relative au domaine étudié. Celle-ci contient les concepts permettant d'enrichir les informations initialement contenues et d'apporter ainsi une connaissance spécifique au sein de l'application.

Bien que l'ontologie pour l'analyse des comportements anormaux s'appuie sur ces trois différents modules, l'essentiel des travaux réalisés a porté sur celle ayant trait au domaine. En effet, conformément aux préceptes de ré-utilisabilité, celles concernant les trajectoires et la géographie ont été créées en s'appuyant sur des ontologies déjà existantes. Les paragraphes ci-dessous décrivent la démarche adoptée ainsi que les différents composants utilisés.

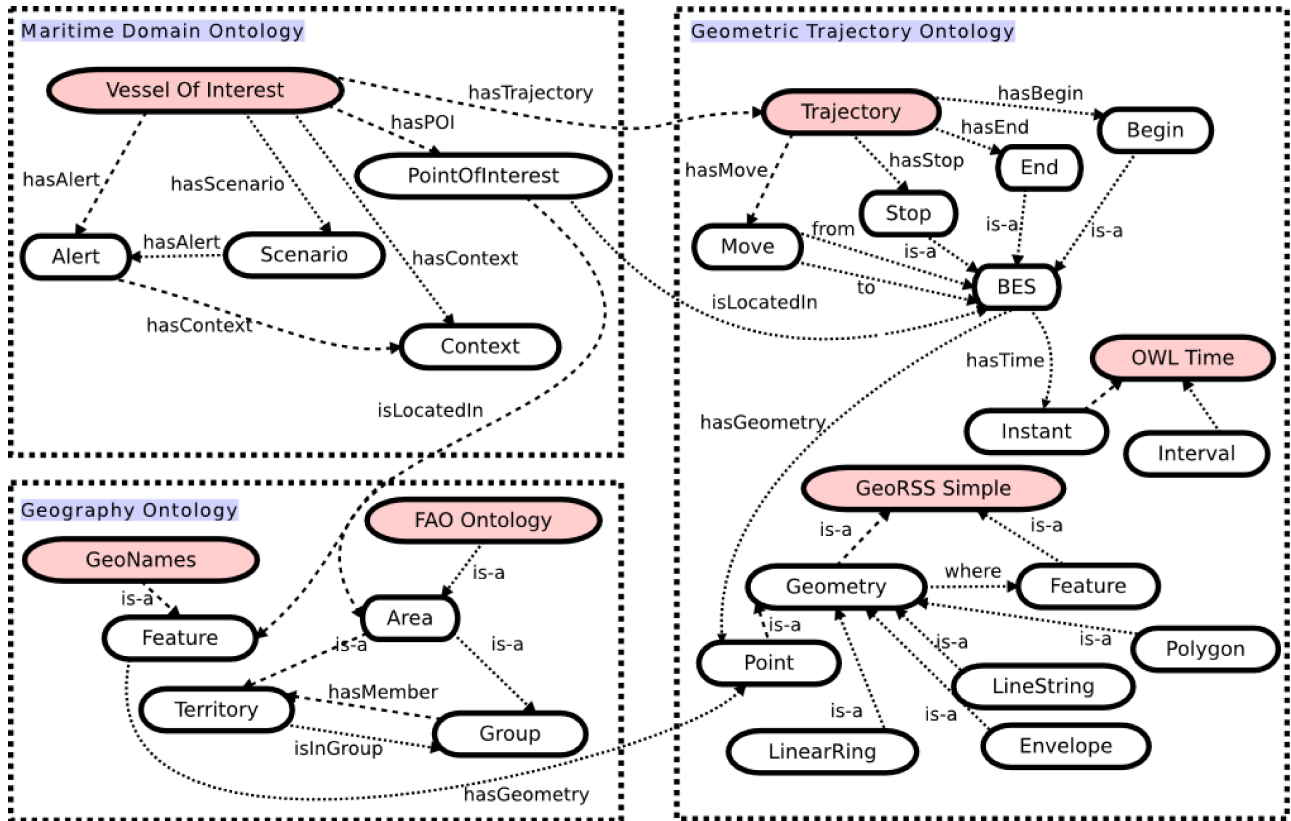


Figure 4.2: Modèle ontologique pour la représentation des trajectoires

4.1.2. Ontologie géométrique des trajectoires

L'ontologie géométrique des trajectoires est elle-même composée de différentes ontologies fournissant les notions essentielles à la description sémantique d'une trajectoire (Figure 4.2). Pour cela une approche à la fois spatiale et temporelle est nécessaire.

4.1.2.1 Ontologie spatio-temporelle

Comme cela a déjà été souligné dans le chapitre 3.2.4 (p.102), différentes ontologies spatio-temporelles existent. Celles-ci varient à la fois en fonction du formalisme utilisé et de la précision sémantique des notions. Lors de la phase de conception de l'ontologie, deux d'entre elles ont été retenues. La première est la représentation OWL de GeorSS Simple⁹⁸. La seconde, plus complète, est une déclinaison au format OWL du standard Geography Markup Language (GML⁹⁹).

Ontologie GeorSS Simple

S'appuyant sur la modélisation réalisée initialement par la société MapBureau, le GeoSpatial Incubator Group (GeoXG) du W3C propose en téléchargement une ontologie au format OWL du standard GeorSS Simple. Celle-ci permet une représentation géométrique des entités géographiques.

98 http://www.w3.org/2005/Incubator/geo/XGR-geo-20071023/W3C_XGR_Geo_files/geo_2007.owl

99 <http://www.opengeospatial.org/standards/gml/>

Pour cela deux principales classes sont définies. La classe *_Feature* qui définit l'objet lui-même et la classe *_Geometry* qui spécifie ses propriétés géométriques (Figure 4.3). Pour cette dernière, différentes sous-classes comme *Polygon*, *LineString* ou encore *Point* permettant de préciser le type géométrique de l'objet sont disponibles. Les classes *_Feature* et *_Geometry* sont ensuite reliées entre elles par la propriété d'objet *where*. Enfin, à cela s'ajoutent également les propriétés de données directement rattachées à l'entité *_Feature* qui permettent par exemple de renseigner sa longitude, sa latitude ou encore son nom.

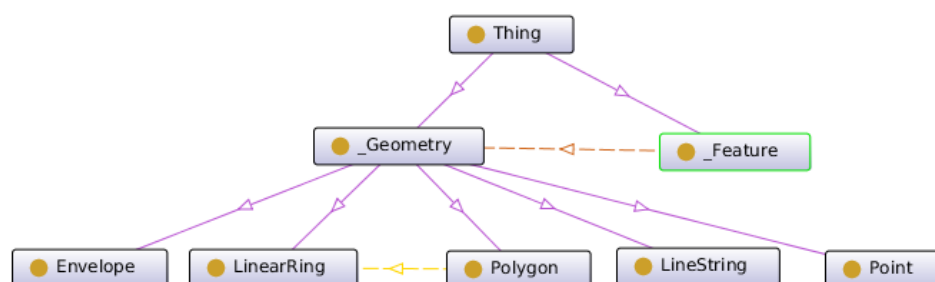


Figure 4.3: Représentation graphique de la classe *_Geometry* de l'ontologie GeorSS Simple

Bien que simple, cette ontologie contient les principaux éléments nécessaires à la description d'une entité géographique. Par exemple, le fragment OWL ci-dessous définit les coordonnées géographiques d'un navire en rade au port de Toulon :

```

<owl:NamedIndividual rdf:about="&gml;navireA">
  <rdf:type rdf:resource="&gml;_Feature"/>
  <rdfs:label xml:lang="fr">navireA</rdfs:label>
  <geo:point rdf:datatype="&xsd:string">43.109 5.911</geo:point>
</owl:NamedIndividual>
  
```

Ontologie GML

Réalisée à l'initiative de l'Université de Drexel, cette ontologie est une traduction du standard GML (en version 3.0) sous une forme OWL. Bien plus complète que l'ontologie précédente, celle-ci contient une quarantaine de classes, autant de propriétés de données et plus de quatre-vingts propriétés d'objets. La création d'une entité se fait par l'intermédiaire de la classe *_Object* qui contient les différentes propriétés spatio-temporelles nécessaires.

Représentation spatiale

Dans cette ontologie, la représentation spatiale d'une entité se fait par l'intermédiaire de la classe *_Geometry* (Figure 4.4). Celle-ci contient trois autres sous-classes dépendantes du type géométrique de l'objet. Par exemple, la classe *_GeometricPrimitive* contient différentes primitives géométriques permettant la création d'entité spatiale. Ces entités sont notamment : *Point*, *_Curve*, *_Solid* et *_Surface*. Dans certains cas, il peut être souhaitable de disposer d'agrégation d'objets géographiques. L'ontologie dispose pour cela d'une classe *_GeometricAggregate* contenant les sous classes : *MultiCurve*, *MultiGeometry*, *MultiLineString*, *MultiPoint*, *MultiPolygon*, *MultiSolid* et *MultiSurface*.

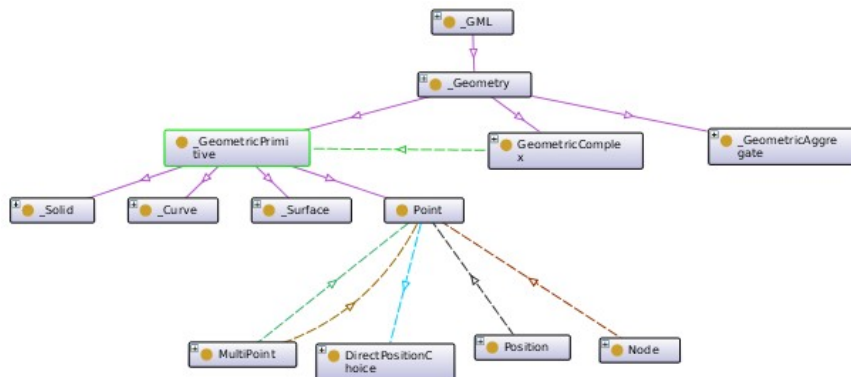


Figure 4.4: Représentation graphique de la classe *_Geometry* de l'ontologie GML

Représentation temporelle

Dans le cas d'un objet mobile, la dimension temporelle est un élément primordial. Celle-ci est prise en compte dans l'ontologie GML par la classe *TemporalPosition* (Figure 4.5). Plusieurs modes de représentation (année, mois, jours, etc.) sont disponibles.

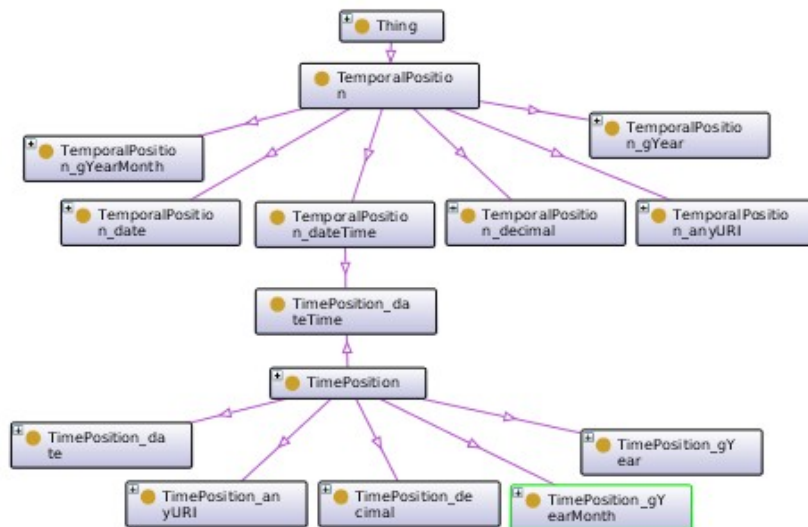


Figure 4.5: Représentation graphique de la classe *TemporalPosition* de l'ontologie GML

Ontologie Spatio-Temporelle sélectionnée

Si les deux ontologies présentées permettent de définir spatialement un objet géographique, celles-ci varient quant à la précision de la description. Bien que plus complète, l'ontologie GML souffre d'un manque de documentation et d'une certaine complexité. De plus, celle-ci ne semble actuellement plus maintenue.

De ce fait, dans le cadre de ce travail de recherche, c'est l'ontologie GeorSS Simple qui a été choisie pour la représentation des objets spatiaux. Néanmoins, celle-ci ne propose nativement

aucune notion temporelle. C'est pourquoi, l'ontologie *OWL Time*¹⁰⁰ a également été ajoutée. Recommandation officielle du W3C, *OWL Time* considère deux types d'entités temporelles : les instants et les intervalles. Un instant est un événement ponctuel contrairement à un intervalle qui représente une durée (Hobbs and Pan, 2004; Dia Miron, 2009).

L'union de ces deux ontologies offre ainsi tous les éléments nécessaires à la description d'objet géographique. Le choix de séparer conceptuellement la composante spatiale et temporelle permet une plus grande modularité. En effet, chacune des ontologies peut alors évoluer de manière individuelle et cela sans interférer avec les autres ontologies. De plus, ces ontologies étant soutenues par le W3C cela garantit à la fois une stabilité et une pérennisation des concepts.

4.1.2.2 Ontologie des trajectoires

Cette ontologie contient les concepts nécessaires à la description d'une trajectoire. Plusieurs modèles existent mais dans le cadre de cette thèse, c'est celui proposé par Spaccapietra (Spaccapietra et al., 2008) qui a été choisi. Celui-ci définit une trajectoire comme étant composée d'un élément *Begin* et d'un élément *End* ainsi qu'un ou plusieurs éléments *Move* qui peuvent être entrecoupés de *Stop*.

À ces primitives de trajectoires, il a été nécessaire dans le cadre du domaine maritime d'ajouter une information complémentaire. En effet, se pose le cas où la position de l'objet mobile étudié n'est plus connue pendant un certain laps de temps. Cette absence de localisation peut être intentionnelle (un navire qui coupe son AIS) ou involontaire (un navire en perdition). Or cette information est particulièrement importante dans le cadre de l'analyse des comportements. C'est pourquoi la notion *Lost* a été ajoutée afin de traduire sémantiquement cet événement (Error: Reference source not found).

Le concept *Lost* ne peut être connu au préalable. Celui-ci est fonction du laps de temps écoulé depuis la dernière position connue de l'objet mobile. Ainsi, une position *Begin* ou *End* sera transformée en *Lost* si le navire n'a plus émis de signal AIS depuis un certain temps ou s'il y a une absence de données importante entre deux positions.

Concept	Définition en logique de description
TimePoint	$\subseteq \exists \text{ hasGeometry.Point} \cap \exists \text{ hasTime.Instant}$
Begin	$\subseteq \text{TimePoint}$
End	$\subseteq \text{TimePoint}$
Lost	$\subseteq \text{TimePoint}$
Stop	$\subseteq \exists \text{ hasGeometry.Point} \cap \exists \text{ hasTime.Interval}$
B.E.S	$\equiv \text{Begin} \cup \text{End} \cup \text{Stop}$
Move	$\subseteq \text{TimeVaryingPoint} \cap \exists \text{ from.D.F.A}$
Trajectory	$\equiv \exists \text{ hasBegin.Begin} \cap \neq 1 \text{ hasBegin} \cap \exists \text{ hasEnd.End} \cap \neq 1 \text{ hasEnd} \cap \exists \text{ hasMove.Move} \cap \forall \text{ hasStop.Stop}$

Tableau 3.1: Représentation des éléments constitutifs d'une trajectoire avec ajout du concept *Lost*.
Source (Yan, 2008)

¹⁰⁰<http://www.w3.org/2006/time>

4.1.3. Ontologie géographique

Cette ontologie contient les concepts géographiques nécessaires à la compréhension d'une trajectoire. Elle est composée d'éléments aussi bien génériques que spécifiques à un domaine applicatif donné. Là encore, le principe de réutilisabilité a été appliqué avec l'utilisation de différentes ontologies déjà existantes.

Par exemple, la description des pays et de leurs relations a été obtenue en utilisant l'ontologie proposée par la Food and Agriculture Organization (FAO) des Nations Unies¹⁰¹. Celle-ci propose une classification des pays en fonction de différents critères aussi bien économiques que géographiques. De plus différentes informations complémentaires sont disponibles comme les pays frontaliers, le code ISO ou encore le nom du pays en différentes langues (Figure 4.6).

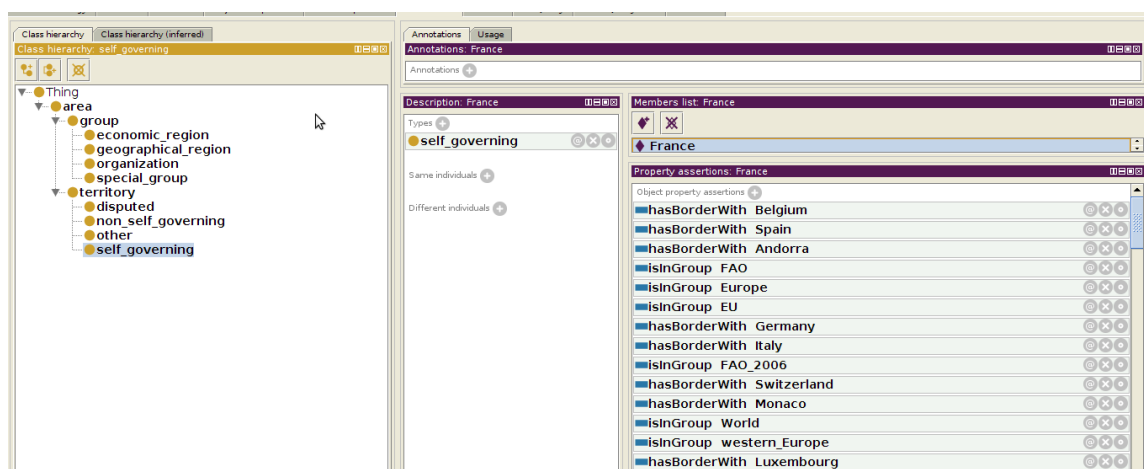


Figure 4.6: Les différentes propriétés du pays "France" dans l'ontologie de la FAO

Cette ontologie géographique a également été enrichie par l'ajout des concepts issus du projet GeoNames¹⁰². Celui-ci contient plus de 6 millions de toponymes accessibles soit à partir d'une extraction complète des données¹⁰³ ou alors par l'intermédiaire d'un service web¹⁰⁴.

Afin de garder une meilleure compatibilité avec l'évolution potentielle de l'ontologie GeoNames, l'accès par service web a été préféré. Celui-ci permet au moyen d'une simple requête *HTTP* d'accéder à la totalité des informations répertoriées. Tout comme l'ontologie précédente, le nom du pays en différentes langues est disponible mais aussi la population totale ou encore sa localisation. Le fragment OWL ci-dessous représente le résultat d'une requête vers le service web de GeoNames avec pour pays la Somalie :

```
<rdf:RDF><gn:Feature rdf:about="http://sws.geonames.org/51537/">
  <rdfs:isDefinedBy>http://sws.geonames.org/51537/about.rdf</rdfs:isDefinedBy>
  <gn:name>Somalia</gn:name>
  <gn:alternateName xml:lang="af">Somalië</gn:alternateName>
```

101 <http://aims.fao.org/geopolitical.owl>

102 <http://www.geonames.org/ontology/documentation.html>

103 <http://download.geonames.org/all-geonames-rdf.zip>

104 <http://www.geonames.org/export/geonames-search.html>

```

<gn:alternateName xml:lang="ak">Somalia</gn:alternateName>
<gn:featureClass rdf:resource="http://www.geonames.org/ontology#A"/>
<gn:featureCode rdf:resource="http://www.geonames.org/ontology#A.PCLI"/>
<gn:countryCode>SO</gn:countryCode>
<gn:population>10112453</gn:population>
<wgs84_pos:lat>6</wgs84_pos:lat><wgs84_pos:long>48</wgs84_pos:long>
<gn:childrenFeatures rdf:resource="http://sws.geonames.org/51537/contains.rdf"/>
<gn:locationMap rdf:resource="http://www.geonames.org/51537/somalia.html"/>
</gn:Feature>

```

4.2. Ontologie du domaine : l'analyse de comportements anormaux de navires

L'ontologie du domaine est l'un des apports majeurs de cette thèse. En effet, si l'ontologie des trajectoires et celle géographique s'appuient sur des travaux existants, celle relative au domaine est créée en fonction d'un besoin et d'un objectif précis. En accord avec la typologie donnée par Guarino (Guarino, 1998), celle-ci peut être alors considérée comme une ontologie d'application. Les paragraphes suivants décrivent la méthodologie adoptée ainsi que l'organisation de l'ontologie.

4.2.1. Principales classes de l'ontologie du domaine

Quatre principales classes structurent l'ontologie du domaine permettant l'analyse de comportements des navires :

- **la classe *alert*** : cette classe contient les différentes alertes relatives à un navire. Ces dernières ont été définies en collaboration avec les experts du domaine. Différentes sous-classes ont été créées en fonction de la thématique des alertes. Sont par exemple répertoriées, celles relatives à l'état de la mer, à la localisation ou à la cinématique du navire. Certaines de ces alertes nécessitent la prise en compte de la dimension spatiale. Dans ce cas, il est nécessaire d'ajouter aux individus la propriété de donnée *DP_Analysis_DataPath* indiquant la localisation des données géographiques à utiliser.
- **la classe *context*** : elle permet d'enrichir les alertes avec des éléments issus de l'environnement immédiat dans lequel évolue le navire. Les notions définies sont par exemple mer forte ou vent fort.
- **la classe *vessel*** : cette classe contient toutes les notions relatives aux navires. Elles permettent de décrire un navire en précisant par exemple son type ou encore son pavillon d'origine.
- **la classe *scenario*** : dans cette classe les experts du domaine définissent le comportement d'un navire. Un scénario est composé de notions provenant des trois classes précédentes. Les experts construisent alors un scénario en fonction des alertes, des éléments du contexte et des caractéristiques du navire ou des navires.

Les classes *alert*, *scenario* et *vessel* étant au cœur de l'ontologie, celles-ci sont décrites en détail dans les paragraphes suivants.

4.2.2. Modélisation des alertes

Dans l'ontologie du domaine, la classe *alert* permet de spécifier un événement inhabituel ou suspect. Ces événements ont été définis à la fois avec les experts du domaine mais aussi en s'appuyant sur la littérature scientifique existante (Nilsson et al., 2008; Van Laere and Nilsson, 2009; Roy and Davenport, 2010). Deux grands types d'alertes peuvent être distingués. Les alertes à caractère spatial et celles à caractère non spatial (Figure 4.7). Les concepts définis dans cette classe seront ensuite utilisés pour la description des scénarios et pour l'enrichissement de la base de connaissance.

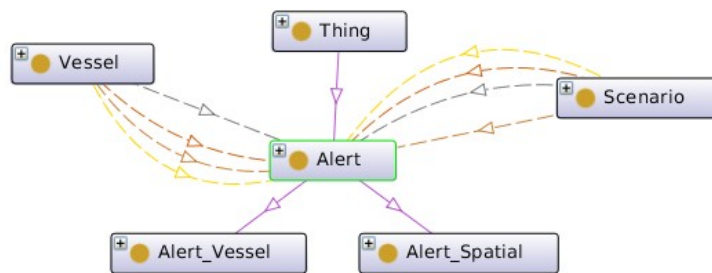


Figure 4.7: Organisation et principales sous-classes de la classe *Alert*

4.2.2.1 Modélisation des alertes spatiales

Comme son nom l'indique, les alertes spatiales s'appuient directement sur les éléments géographiques. Il s'agit par exemple de signaler l'entrée d'un navire dans un espace soumis à une réglementation spécifique, d'identifier une trajectoire particulière ou encore de détecter une zone où les conditions météorologiques sont mauvaises (Figure 4.8).

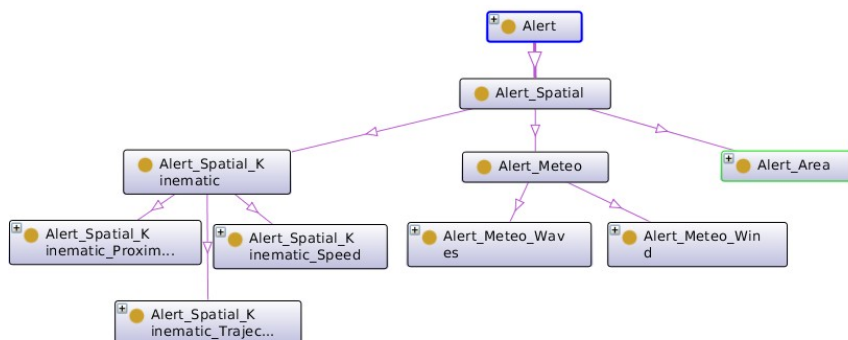


Figure 4.8: Sous-classes de la classe *Alert_Spatial*

Les données et les analyses spatiales à réaliser peuvent être définies par l'utilisateur au moyen des propriétés d'objet *DP_Analysis_Spatial* et *DP_Analysis_DataPath*. La première permet de spécifier l'analyse spatiale à réaliser. La seconde définit les données à utiliser.

Par exemple, comme cela est illustré par le fragment OWL ci-dessous, l'alerte *Alert_Area_Fishing*

4.2. Ontologie du domaine : l'analyse de comportements anormaux de navires

contient la propriété de données *DP_Analysis_DataPath*. La valeur définie (*FSHZNE_FR3_A.shp*), sera ensuite utilisée par le moteur d'inférence lors de l'analyse spatiale.

```
<owl:NamedIndividual rdf:about="http://mda/rdf_mda_v4_new.owl#Alert_Area_Fishing">
  <rdf:type rdf:resource="http://mda/rdf_mda_v4_new.owl#Alert_Area"/>
  <rdfs:label xml:lang="fr">Alert_Area_Fishing</rdfs:label>
  <DP_Analysis_DataPath
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    ./gis_data/mda/ENC/FSHZNE_FR3_A.shp
  </DP_Analysis_DataPath>
</owl:NamedIndividual>
```

Ces alertes spatiales s'appliquent aussi bien à des événements ponctuels que linéaires. C'est pourquoi la classe *Alert_Spatial_Kinematic_Trajectory* décrivant la trajectoire des navires a été définie. Celle-ci contient trois types de trajectoires suspectes (Figure 4.9). La première (*Alert_Trajectory_Slalom*) correspond à un navire effectuant de nombreux changements de cap. La seconde (*Alert_Trajectory_Circle*), décrit un navire effectuant des cercles au même endroit. Cela peut être par exemple le signe d'un point de rendez-vous afin d'effectuer un échange de marchandises illicites. Enfin, la dernière (*Alert_Trajectory_Parallel*), est relative à une navigation en parallèle entre deux navires. Cela permet par exemple de détecter des navires pêchant en collaboration.

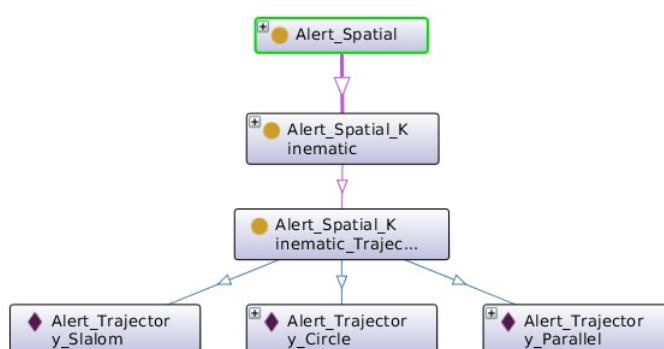


Figure 4.9: Individus de la classe correspondant à l'analyse de trajectoire

4.2.2.2 Modélisation des alertes non spatiales

Les alertes non spatiales s'appuient essentiellement sur les caractéristiques du navire. Il s'agit par exemple de détecter un navire trop vieux ou faisant partie d'une liste noire définie (Figure 4.10).

Ces alertes sont bien évidemment spécifiques à chaque type de navires. Ainsi, la limite théorique de vitesse maximale sera différente pour un navire de pêche et un cargo.

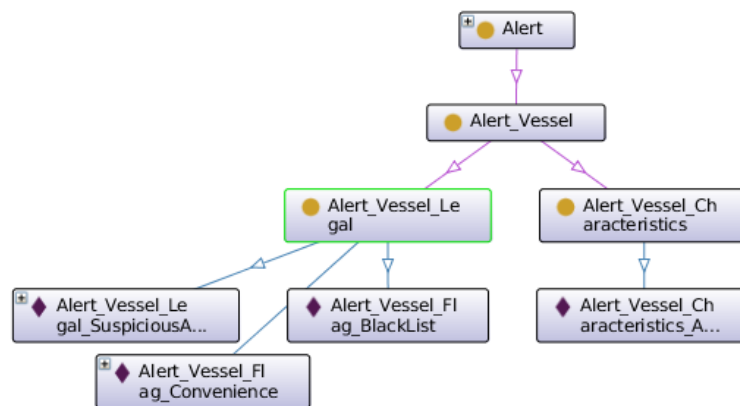


Figure 4.10: Liste des différentes alertes potentielles liées aux caractéristiques du navire.

4.2.3. Modélisation des scénarios

La notion de scénario, telle qu'elle a été définie dans l'ontologie, est composée d'une suite d'événements qui peut être enrichie par des éléments contextuels. Les événements sont les alertes détectées et les éléments contextuels portent aussi bien sur les conditions de navigation que sur le navire lui-même. Les scénarios renseignés par les experts du domaine permettront ensuite au module de Raisonnement À Partir de Cas (cf. chapitre 5.4 p.160) d'identifier le comportement d'un navire.

4.2.3.1 Principales classes de la notion de scénario

La modélisation des scénarios utilise les différentes notions de l'ontologie. Celle-ci s'appuie aussi bien sur des éléments issus de la classe *Alert*, *Context* ou *Vessel*. Ces scénarios sont par la suite utilisés par le moteur de Raisonnement À Partir de Cas pour identifier les comportements anormaux (cf §5.4.1 p.160).

Tout comme les alertes, les scénarios potentiels ont été divisés par thématique. A l'heure actuelle, quatre sous-classes ont été définies (Figure 4.11) :

- ***Scenario Attack*** : il décrit les cas d'attaques potentiels. Il s'agit par exemple d'une attaque terroriste ou encore d'une attaque à l'aide d'un tanker.
- ***Scenario Fishing*** : il s'agit dans ce cas de scénarios liés à la pêche. Sont notamment définis ceux en rapport au transbordement ou encore la pêche illégale.
- ***Scenario Traffic*** : ce scénario est relatif à tous les types de trafic potentiel. Il peut s'agir d'un trafic de marchandises, d'armes ou de drogues.
- ***Scenario Trajectory*** : il a trait aux scénarios prenant en compte la trajectoire du navire. Cela peut être par exemple nécessaire pour détecter une future collision.

4.2. Ontologie du domaine : l'analyse de comportements anormaux de navires

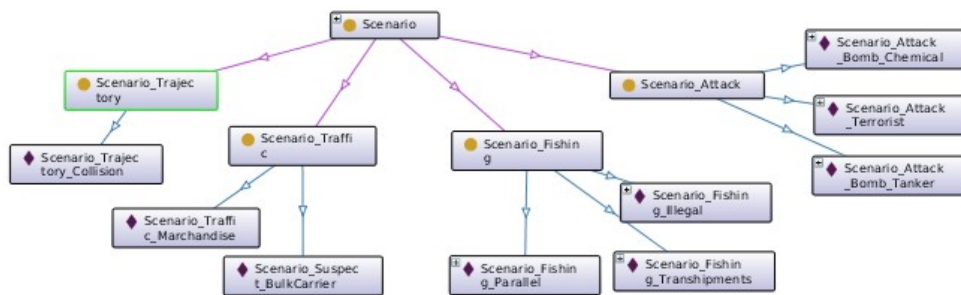


Figure 4.11: Sous-classes et instances pour la description des scénarios

4.2.3.2 Description d'un scénario

Un scénario est composé des éléments provenant des différentes classes de l'ontologie du domaine. Il s'agit alors de décrire une situation type qui sera ensuite analysée lors de la phase de Raisonnement À Partir de Cas.

Par exemple, la description d'une situation de transbordement se définit de la manière suivante :

- la météo est propice au transbordement
- les navires impliqués sont de type pêche et cargo
- les deux navires ont une alerte arrêt en pleine mer ainsi qu'une alerte de proximité
- l'appartenance à un pavillon de complaisance peut être un facteur supplémentaire

Cette situation est alors représentée dans l'ontologie par le fragment OWL ci-dessous :

```
<owl:NamedIndividual rdf:about="&rdf_mda_v4_new;Scenario_Fishing_Transhipments">
<rdf:type rdf:resource="&rdf_mda_v4_new;Scenario_Fishing"/>
  <DP_Analysis_Criticality rdf:datatype="&xsd;float">2</DP_Analysis_Criticality>
  <OP_Alert_hasAlert rdf:resource="&rdf_mda_v4_new;Alert_Area_Fishing"/>
  <OP_Alert_hasAlert rdf:resource="&rdf_mda_v4_new;Alert_Proximity_toClose"/>
  <OP_Alert_hasAlert rdf:resource="&rdf_mda_v4_new;Alert_Speed_StopInOpenSea"/>
  <OP_Alert_hasAlert rdf:resource="&rdf_mda_v4_new;Alert_Vessel_Flag_Convenienc"/>
  <OP_Vessel_hasContext rdf:resource="&rdf_mda_v4_new;Context_Meteo_Wave_Low"/>
  <OP_Alert_isToCloseWithType rdf:resource="&rdf_mda_v4_new;Vessel_Type_Cargo"/>
  <OP_Vessel_hasVesselType rdf:resource="&rdf_mda_v4_new;Vessel_Type_Fishing"/>
</owl:NamedIndividual>
```

En plus des différentes propriétés d'objet, la propriété de données *DP_Analysis_Criticality* est ajoutée. Celle-ci permet de préciser le niveau de dangerosité du scénario et de prioriser ainsi les interventions.

4.2.4. Modélisation du contexte maritime

Au cœur de l'analyse de comportement, il y a bien évidemment l'objet à étudier, le navire lui-même. Celui est représenté dans l'ontologie par la classe *Vessel* qui regroupe l'ensemble des notions importantes à sa caractérisation (Figure 4.12).

Chapitre 4. Une approche ontologique pour l'analyse des comportements de navires

Différentes sous-classes ont été définies et permettent par exemple d'annoter son comportement, son historique ou encore son type.

Les navires identifiés comme suspects sont stockés dans l'ontologie sous la forme d'instance de la classe *VOI* (Vessel Of Interest).

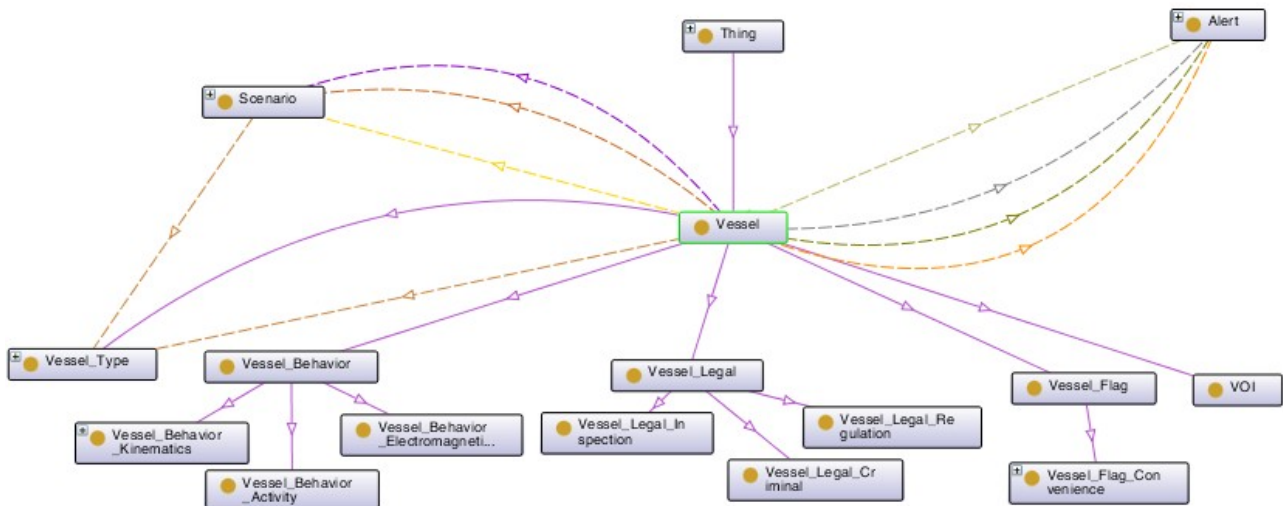


Figure 4.12: Sous-classes de la classe *Vessel*

4.3. Modélisation des règles d'inférence et enrichissement du SWRL

Une fois le domaine modélisé, il est alors possible d'ajouter des règles permettant l'analyse du comportement des navires. Celles-ci peuvent être vues comme des déclencheurs qui vont modifier l'ontologie en fonction de la satisfaction ou non de la règle. Pour cela le langage utilisé a été le Semantic Web Rule Language (SWRL). Bien que nombreux opérateurs (mathématiques, de comparaison de chaînes, etc.) sont disponibles par défaut, la dimension spatiale ne peut être actuellement prise en compte. Or celle-ci est un élément important lors du processus de décision. Les paragraphes suivants présentent l'intégration de cette dimension au sein du langage SWRL.

4.3.1. Le langage SWRL

La représentation et l'intégration des règles dans l'ontologie ont été réalisées à l'aide du langage SWRL. Celui-ci enrichit la sémantique d'une ontologie en combinant le langage OWL-DL et RuleML (Horrocks et al., 2004). Basée sur les logiques de description et les règles de Horn, la structure d'une règle SWRL est de la forme antécédent(s) -> conséquence(s) qui signifie : « si l'antécédent est vrai, alors la conséquence l'est aussi ».

Contrairement à OWL, SWRL permet uniquement l'ajout de relations ou de propriétés déjà existantes en fonction de la satisfaction ou non de la règle. En plus des prédicats OWL, SWRL possède également des fonctions supplémentaires (mathématiques, booléennes, etc.) nommées 'built-in' (Grau et al., 2008). Ces derniers étendent les fonctions initialement proposées par le langage OWL en permettant notamment d'effectuer des comparaisons et des calculs.

De la même manière, cette approche sous la forme de 'built-in' a été adoptée afin d'intégrer des fonctionnalités spatiales aux SWRL. Définie en collaboration avec les experts du domaine et en s'appuyant également sur la littérature scientifique existante, la base de règles distingue celle utilisant les potentialités initiales du SWRL et celles prenant en compte la dimension spatiale.

4.3.2. Modélisation des règles classiques

Les règles classiques exploitent directement les potentialités du langage SWRL. En effet, différents opérateurs permettant par exemple la gestion des dates, des chaînes de caractères ou la comparaison de valeurs numériques sont déjà disponibles.

Imaginons que l'une des règles porte sur la détection d'une vitesse excessive. La requête, traduite en langage SWRL est présentée ci-dessous et se lit de la manière suivante : « Si la vitesse du navire (?vesselType) est supérieure (*greaterThan*) à la vitesse maximale en fonction du type du navire (?speedTypeVessel) alors une alerte est déclenchée (*Alert_Speed_HighSpeed*) ». Le mot clé *greaterThan* est un opérateur mathématique permettant d'effectuer la comparaison entre deux valeurs numériques.

```
Vessel(?vessel), OP_Vessel_hasVesselType(?vessel, ?vesselType),
DP_Vessel_hasSpeed(?vessel, ?speedVessel),
DP_Vessel_hasSpeed(?vesselType, ?speedTypeVessel),
greaterThan(?speedTypeVessel, ?speedVessel) ->
OP_Vessel_hasAlert(?vessel, Alert_Speed_HighSpeed)
```

De la même manière, il peut être intéressant de mettre en évidence les navires trop âgés. Cette limite est à mettre en corrélation avec le type du navire. Cette analyse peut notamment servir lorsque les conditions météorologiques sont mauvaises afin de mettre en évidence les navires dont la probabilité d'échouage est forte.

La modélisation de cette règle sous la forme SWRL présentée ci-dessous se lit de la manière suivante : « Si l'âge du navire (*?ageVessel*) est supérieur à l'âge maximale en fonction du type du navire (*?ageTypeVessel*) alors une alerte est déclenchée (*Alert_Vessel_Characteristics_Age*) ».

```
Vessel(?vessel),
OP_Vessel_hasType(?vessel, ?vesselType),
DP_Vessel_hasAge(?vessel, ?ageVessel),
DP_Vessel_hasAge(?vesselType, ?ageTypeVessel),
greaterThan(?ageVessel, ?ageTypeVessel) ->
OP_Vessel_hasAlert(?vessel, Alert_Vessel_Characteristics_Age)
```

4.3.3. Modélisation des règles spatiales

Comme cela a déjà été souligné, la prise en compte de la dimension spatiale est essentielle au processus décisionnel. En effet, celle-ci participe pleinement à l'analyse du comportement des navires. L'intégration de cette composante était donc nécessaire au sein de la plate-forme développée.

Afin de respecter une compatibilité maximale avec le SWRL, l'approche adoptée s'est faite en s'appuyant sur les mécanismes d'extension proposés par les 'built-in'. Ainsi, tout comme les opérateurs mathématiques ou ceux portant sur les chaînes de caractères, l'intégration de la dimension spatiale se fait au moyen de mots clé. Ces mots-clés, que l'on retrouve également dans le domaine des bases de données spatiales, sont par exemple : *touches*, *intersects*, etc. Cette approche permet ainsi de bénéficier de toute la richesse sémantique proposées par les ontologies tout en intégrant la composante géographique.

Dans l'ontologie du domaine plusieurs règles spatiales ont été définies. Certaines peuvent être qualifiées de simple (*touches*, *intersects*, etc.) car elles ne comparent que la position du navire et d'autres sont plus complexes car elles prennent également en compte sa trajectoire. Des fonctions spécifiques au domaine maritime ont également été ajoutées comme celles portant sur la cinématique du navire.

4.3.3.1 Règle spatiale simple

L'entrée d'un navire dans une zone réglementée doit automatiquement déclencher une alerte. La règle SWRL correspondante doit donc prendre en compte sa position (*?position*) ainsi que les données correspondant aux zones réglementaires (*Alert_Area_Restricted*). Celles-ci seront ensuite croisées afin de vérifier si il y a ou non une intersection (*intersects(?geoData, ?position)*). Si le résultat est positif, alors une alerte (*Alert_Area_Restricted*) est automatiquement attribuée au navire concerné.

Comme cela est illustré par la règle SWRL ci-dessous, le simple ajout du mot-clé « *intersects* » est suffisant pour prendre en compte la dimension spatiale des éléments de l'ontologie.

```
Vessel(?vessel),
OP_Vessel_hasPosition(?vessel, ?OWLposition),
DP_Vessel_hasPosition(?OWLposition, ?position),
DP_Analysis_DataPath(Alert_Area_Restricted, ?geoData),
intersects(?geoData, ?position) ->
OP_Vessel_hasAlert(?vessel, Alert_Area_Restricted)
```

4.3.3.2 Règle spatiale simple avec un constructeur géométrique

La règle précédente vérifie l'intersection entre deux objets. Néanmoins, dans certains cas, il peut être nécessaire de construire une entité géométrique avant d'effectuer la comparaison. C'est le cas notamment lors de la création de zones tampons quand un navire arrive à proximité d'un endroit où la navigation est dangereuse.

Cette situation est illustrée par le fragment SWRL ci-dessous. Dans celui-ci, un paramètre (?geoBuffer) indiquant la taille de cette zone tampon est spécifié à l'alerte concernée (Alert_Area_UnderwaterRoc). Ensuite, l'intersection entre la position du navire (?position) et les données (?geoData) est réalisée en fonction du paramètre précédent. Si la position du navire augmentée de la zone tampon touche la zone concernée alors une alerte est ajoutée (Alert_Area_UnderwaterRoc).

```
Vessel(?vessel),
DP_Analysis_BufferValue(Alert_Area_UnderwaterRoc, ?geoBuffer),
DP_Analysis_DataPath(Alert_Area_UnderwaterRoc, ?geoData),
OP_Vessel_hasPosition(?vessel, ?OWLposition),
DP_Vessel_hasPosition(?OWLposition, ?position),
touches(?position, ?geoData, ?geoBuffer) ->
OP_Vessel_hasAlert(?vessel, Alert_Area_UnderwaterRoc)
```

4.3.3.3 Règle spatiale simple avec utilisation des 'built-in'

Les règles spatiales peuvent également être utilisées conjointement avec les 'built-in' disponibles nativement en SWRL. Par exemple, la détection d'une pêche illégale nécessite de connaître le type du navire (?vesselType) ainsi que son pavillon d'origine (?country). Si c'est un navire de pêche (equal(?vesselTypeName, "Fishing"^^string)), qu'il n'est pas français (notEqual(?country, "France"^^string)) et qu'il se trouve dans une zone de pêche (intersects(?position, ?geoData)), alors une alerte doit être automatiquement déclenchée (?vessel, Alert_Area_Fishing).

```
Vessel(?vessel),
OP_Vessel_hasType(?vessel, ?vesselType),
DP_Analysis_DataPath(Alert_Area_Fishing, ?geoData),
DP_Vessel_hasCountry(?vessel, ?country),
OP_Vessel_hasPosition(?vessel, ?OWLposition),
DP_Vessel_hasPosition(?OWLposition, ?position),
DP_Vessel_hasTypeName(?vesselType, ?vesselTypeName),
equal(?vesselTypeName, "Fishing"^^string),
notEqual(?country, "France"^^string),
```

```
intersects(?position, ?geoData) ->  
OP_Vessel_hasAlert(?vessel, Alert_Area_Fishing)
```

Dans le cas ci-dessus, les ‘built-in’ SWRL natifs *equal* et *notEqual* sont utilisés conjointement avec la fonction spatiale *intersects*. Si tous sont vérifiés alors l'alerte est ajoutée.

4.3.3.4 Règle spatiale portant sur la trajectoire

Bien que les règles présentées ci-dessus permettent l'intégration de la dimension spatiale, celles-ci ne sont pas suffisantes pour l'analyse de comportements anormaux. En effet, en plus de la position du navire il est nécessaire de prendre en compte également sa trajectoire. C'est pourquoi des fonctions spécifiques ont été ajoutées.

L'une d'entre elles porte notamment sur la détection de trajectoires similaires. Cela peut être le cas lorsque deux navires pêchent en parallèle. Pour cela, la trajectoire de chacun des navires (*?trajA*, *?trajB*) est comparée grâce au mot-clé *parallelTrajectory*. Si ces trajectoires sont similaires alors une alerte est ajoutée (*Alert_Trajectory_Parallel*) ainsi qu'une propriété d'objet liant les deux navires (*OP_Alert_hasParallelTrajWith*).

La similitude des trajectoires est obtenue en utilisant la fonction d'Haverine qui compare deux objets géométriques et retourne le coefficient de ressemblance. Bien que cette fonction ne s'adapte pas à tous les cas potentiels, ses limites sont connues et peuvent donc être prises en compte (Devogele, 2009).

```
Vessel(?vesselA),  
Vessel(?vesselB),  
DP_Vessel_hasIMO(?vesselA, ?imoA),  
DP_Vessel_hasIMO(?vesselB, ?imoB),  
OP_Vessel_hasTraj(?vesselA, ?OWLtrajA),  
OP_Vessel_hasTraj(?vesselB, ?OWLtrajB),  
DP_Vessel_hasTraj(?OWLtrajA, ?trajA),  
DP_Vessel_hasTraj(?OWLtrajB, ?trajB),  
notEqual(?imoA, ?imoB),  
parallelTrajectory(?trajA, ?trajB) ->  
OP_Vessel_hasAlert(?vesselA, Alert_Trajectory_Parallel),  
OP_Alert_hasParallelTrajWith(?vesselA, ?vesselB)
```

4.3.3.5 Règle spatiale maritime

Enfin, le dernier type de règle concerne celles portant spécifiquement sur le domaine maritime. Celles-ci répondent à un besoin métier spécifique et sont donc implémentées au cas par cas.

Il s'agit par exemple de détecter la collision entre deux navires. Dans ce cas plusieurs paramètres sont nécessaires. En effet, il est nécessaire de connaître la position du navire mais également sa position future. C'est ce vecteur de position qui est ensuite utilisé pour vérifier la collision.

La traduction de cette situation en langage SWRL a été réalisée de la manière suivante. Tout d'abord le cap (*?headingA*, *?headingB*), la position (*?positionA*, *?positionB*) ainsi que la vitesse (*?speedA*, *?speedB*) de chaque navire sont enregistrés. Ensuite, le numéro IMO (*notEqual(?imoA, ?imoB)*) est comparé afin de vérifier que l'analyse ne porte pas sur le même navire. Enfin, la fonction spatiale

4.3. Modélisation des règles d'inférence et enrichissement du SWRL

spatialToClose calculant les futures positions est réalisée. Si le résultat de celle-ci est positif alors une alerte de proximité est ajoutée (*Alert_Proximity_toClose*) ainsi qu'une propriété d'objet entre les deux navires (*OP_Alert_isToCloseFrom*).

```
Vessel(?vesselA),
Vessel(?vesselB),
DP_Vessel_hasHeading(?vesselA, ?headingA),
DP_Vessel_hasHeading(?vesselB, ?headingB),
DP_Vessel_hasIMO(?vesselA, ?imoA),
DP_Vessel_hasIMO(?vesselB, ?imoB),
OP_Vessel_hasPosition(?vesselA, ?OWLpositionA),
DP_Vessel_hasPosition(?OWLpositionA, ?positionA),
OP_Vessel_hasPosition(?vesselB, ?OWLpositionB),
DP_Vessel_hasPosition(?OWLpositionB, ?positionB),
DP_Vessel_hasSpeed(?vesselA, ?speedA),
DP_Vessel_hasSpeed(?vesselB, ?speedB),
notEqual(?imoA, ?imoB),
spatialToClose(?positionA, ?positionB, ?headingA,
                ?headingB, ?speedA, ?speedB, ?imoA, ?imoB) ->
OP_Vessel_hasAlert(?vesselA, Alert_Proximity_toClose),
OP_Alert_isToCloseFrom(?vesselA, ?vesselB)
```

Conclusion du chapitre

L'architecture proposée sous la forme de trois grandes composantes permet de séparer les différents éléments nécessaires à la modélisation des trajectoires sémantiques et à l'analyse des comportements anormaux de navires.

Si l'ontologie géométrique et l'ontologie géographique sont issues de la fusion d'ontologies déjà existantes, l'ontologie du domaine a été créée spécifiquement pour la surveillance maritime. Celle-ci offre les notions nécessaires à la description d'un navire et de ses alertes. Ainsi, les experts du domaine peuvent facilement implémenter au sein de l'ontologie les alertes qu'ils souhaitent en spécifiant simplement l'un des mots clés définis (*touches*, *intersects*, etc.). Pour cela, le langage SWRL a été enrichi par des procédures d'inférence spécifiques au raisonnement spatial.

Cette connaissance modélisée, il est maintenant possible de l'exploiter au sein d'un système d'analyse de comportements anormaux de navires. Les éléments constituant ce système ainsi que la logique d'implémentation sont décrits dans le chapitre suivant.

CHAPITRE 5. STRUCTURATION DE LA PLATE-FORME D'ANALYSE DES COMPORTEMENTS ANORMAUX DE NAVIRES

Chapitre 5. Structuration de la plate-forme d'analyse des comportements anormaux de navires.....	143
5.1. Analyse de comportements dirigé par une ontologie : application à la surveillance maritime	145
5.1.1. <i>Présentation générale du prototype</i>	<i>145</i>
5.1.2. <i>Vue du prototype.....</i>	<i>145</i>
5.2. Module de données.....	147
5.2.1. <i>Présentation des données.....</i>	<i>147</i>
5.2.2. <i>Préparation des données.....</i>	<i>149</i>
5.2.3. <i>Enrichissement des données et création de la trajectoire sémantique.....</i>	<i>152</i>
5.3. Module d'inférence	154
5.3.1. <i>Architecture du module d'inférence.....</i>	<i>154</i>
5.3.2. <i>Processus d'inférence.....</i>	<i>156</i>
5.3.3. <i>Résultats du processus d'inférence</i>	<i>159</i>
5.4. Module de calcul de similarité des comportements anormaux.....	160
5.4.1. <i>Raisonnement à partir de cas.....</i>	<i>160</i>
5.4.2. <i>Architecture du module de calcul de similarité.....</i>	<i>163</i>
5.4.3. <i>intégration de l'ontologie et calcul de similarité.....</i>	<i>165</i>
5.5. Module cartographique.....	168
5.5.1. <i>Architecture du module.....</i>	<i>168</i>
5.5.2. <i>Interface et présentation des comportements.....</i>	<i>168</i>

Introduction du chapitre

Une fois les connaissances modélisées, il est alors possible de les exploiter. Celles-ci sont alors intégrées au sein d'une plate-forme dédiée à l'analyse de comportements anormaux de navires. L'objectif de cette plate-forme est d'identifier un navire suspect et son comportement potentiellement à risque.

Pour cela quatre étapes sont nécessaires. Tout d'abord, le nettoyage et l'optimisation des données en entrée. En effet, le volume à traiter étant particulièrement important cela nécessite au préalable d'effectuer un certain nombre de traitements. Il est ensuite possible de déclencher le processus d'inférence qui va analyser les positions et les trajectoires sémantiques des navires. À partir de ces analyses, les comportements seront catégorisés et le niveau de dangerosité déterminé. Enfin, les résultats obtenus seront affichés au sein de l'interface cartographique.

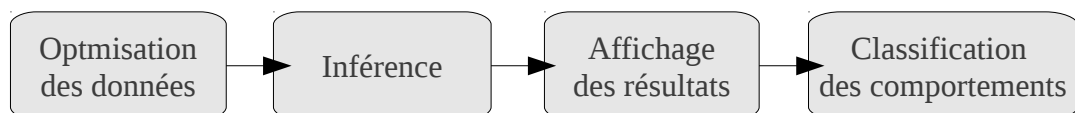


Figure 5.1: Principales étapes nécessaires à l'analyse de comportements de navires

Les différents modules permettant d'accomplir cet objectif sont décrits dans ce chapitre. Dans un premier temps, une présentation générale du prototype est effectuée (§5.1). Puis les différents modules sont détaillés. Tout d'abord le module de données permettant d'obtenir une trajectoire sémantique à partir de positions initiales (§5.2). Ensuite le module d'inférence qui va détecter les alertes potentielles (§5.3). Puis le module de calcul de similarité permettant d'établir la correspondance entre les alertes et les comportements anormaux (§5.4). Enfin, le module cartographique dédié à l'affichage des résultats (§5.5).

5.1. Analyse de comportements dirigé par une ontologie : application à la surveillance maritime

La plateforme *OntoMap* présentée dans ce chapitre permet d'analyser le comportement des navires. Celle-ci s'appuie sur l'ontologie précédemment décrite ainsi que sur différents modules. Ces modules, ainsi que l'architecture générale du prototype, sont détaillés dans les paragraphes suivants.

5.1.1. Présentation générale du prototype

Les différentes approches étudiées dans ce travail de thèse ont été mises en application au sein du prototype *OntoMap* qui est une plate-forme dédiée à l'étude des comportements anormaux de navires basée sur une ontologie. *OntoMap* est utilisé à la fois dans les phases de modélisation afin de capitaliser les connaissances expertes et dans les phases d'analyse afin d'aider à diagnostiquer les causes d'un comportement anormal.

Dans son architecture actuelle, cette plate-forme s'adresse principalement à des experts du domaine ou à des modélisateurs ayant une bonne connaissance du langage OWL. En effet, l'essentiel des efforts a été porté sur la construction d'un modèle générique d'analyse et sur l'intégration de la dimension spatiale dans le SWRL.

Les différents modules ainsi que les étapes nécessaires sont présentés dans ce chapitre. Le rôle de chacun d'eux est décrit ainsi que son implémentation technique.

5.1.2. Vue du prototype

Le prototype *OntoMap* a été développé sous la forme de modules et en s'appuyant sur différentes bibliothèques disponibles. Cette architecture permet ainsi d'assurer une meilleure évolution de la plate-forme. En raison des bibliothèques utilisées, le choix du langage s'est porté sur Java. Pour la partie base de données, c'est le couple PostgreSQL/Postgis qui a été sélectionné. Une partie des traitements, notamment ceux sur les données, a été réalisée par l'implémentation de procédures stockées et de fonctions spécifiques écrites en PL/pgSQL.

Au final, l'architecture générale du prototype d'analyse de comportements de navires se compose de quatre principaux modules (Figure 5.2). Afin d'avoir une vue générale de la plate-forme, le rôle de chacun d'eux est présenté succinctement dans les paragraphes ci-dessous.

5.1.2.1 Module de filtrage et d'optimisation des données

La première étape consiste à acquérir et traiter les flux AIS émis par les navires. Mais, les capteurs placés sur les objets mobiles génèrent une quantité importante de données. Une étape de filtrage est donc nécessaire avant le stockage des positions dans une base de données spatio-temporelle. Par la suite, ces informations ponctuelles sont ensuite formatées sous la forme de trajectoires en reprenant le modèle conceptuel de trajectoires défini par Spaccapietra (Spaccapietra et al., 2008).

5.1.2.2 Module d'ontologie et d'inférence

Une fois les données stockées et correctement formatées, il est possible de les intégrer au sein de la

l'ontologie d'analyse des comportements anormaux. Cette étape permet de relier sémantiquement les informations contenues dans la base avec les liens modélisés par les experts et d'aboutir à de véritables trajectoires sémantiques (Baglioni et al., 2008). Ce sont ces trajectoires qui seront ensuite utilisées lors des phases d'analyses.

5.1.2.3 Module de classification de comportements anormaux

À partir des informations sémantiquement enrichies, le module de classification analyse les comportements des navires et les compare aux scénarios définis par les experts du domaine. Pour cela, la distance sémantique entre chaque concept est évaluée. Puis le scénario ou les scénarios les plus probables sont déterminés. Ce processus s'appuie sur des méthodes issues du Raisonnement à Partir de Cas (RàPC)

5.1.2.4 Module de visualisation cartographique

Enfin, il est possible de visualiser le résultat des analyses effectuées au sein d'une plate-forme cartographique. Ce mode de visualisation offre un accès direct et intuitif à l'information favorisant ainsi une meilleure interactivité.

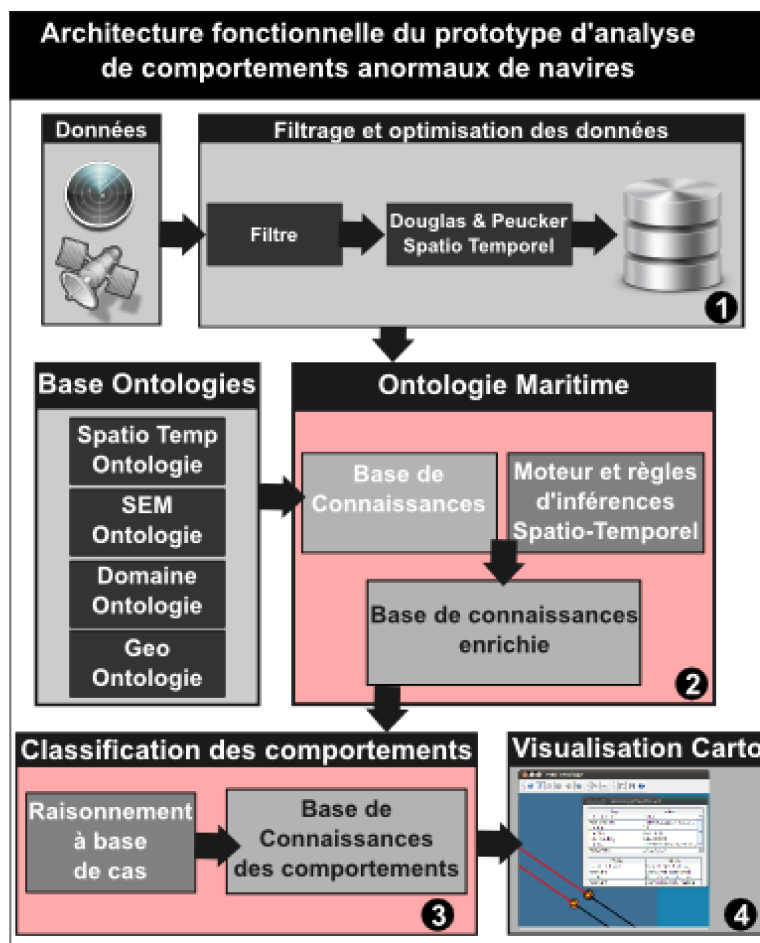


Figure 5.2: Architecture fonctionnelle du prototype d'analyse de comportements anormaux de navires

5.2. Module de données

Les objectifs du module de données sont multiples. Tout d'abord, il s'agit de supprimer les données aberrantes issues par exemple d'une mauvaise calibration du GPS lors de l'initialisation de l'AIS. Ces données erronées pouvant entraîner des biais lors des analyses futures, il est donc nécessaire de réaliser auparavant une étape de filtrage. Ensuite, les données sont simplifiées afin de ne garder que celles significatives d'une trajectoire, puis modélisées sous la forme de *move* et de *stop* tel que défini par Spaccapietra (Spaccapietra et al., 2008). Enfin, en fonction des différentes informations disponibles, ces trajectoires seront enrichies afin d'arriver à une trajectoire sémantique (Baglioni et al., 2008).

Pour cela, une partie des différentes phases abordées lors du chapitre 1.2 (p.40) ont été mises en application. Néanmoins, le processus complet n'a pas encore été réalisé. En effet, cette thèse vise à fournir les éléments nécessaires à la formalisation des connaissances. Des recherches portant sur des algorithmes de détection à partir d'une approche orientée Data-Mining (Idiri and Napoli, 2012) ou à l'aide de réseaux bayésiens (Chaze et al., 2012) sont actuellement menées par notre équipe et seront ajoutées par la suite.

5.2.1. Présentation des données

L'enrichissement des trajectoires est réalisé grâce à différentes sources de données. Celles-ci proviennent de Services Web mais aussi de données locales. Quelques exemples sont décrits dans les paragraphes ci-dessous.

5.2.1.1 Localisation des navires

Systeme AIS

Les données brutes qui sont utilisées par le système proviennent des messages AIS émis par tous les navires soumis au chapitre V¹⁰⁵ de la convention SOLAS¹⁰⁶. Rendu obligatoire dès 2002 par l'Organisation Maritime internationale (OMI), l'AIS a été conçu afin d'éviter les risques de collision. Pour cela, chaque navire de plus de 300 tonnes, ceux transportant des passagers ou encore ceux effectuant des trajets dans les eaux internationales sont obligés de disposer de ce type de transpondeur. Celui-ci est composé de deux éléments. Premièrement, d'un récepteur GPS permettant d'obtenir la position du navire. Deuxièmement, d'un émetteur VHF afin de transmettre sur des fréquences dédiées les différentes informations aux autres navires et aux stations côtières. Ces informations peuvent être classées en trois grandes catégories (Le Guyader et al., 2012) :

- **des informations statiques** : elles permettent l'identification du navire (N°IMO, N°MMSI) et apporte des renseignements complémentaires comme son nom, son type ou encore sa longueur.
- **des informations dynamiques** : elles sont relatives à la position et à la cinématique du navire. Ces données sont par exemple sa vitesse, son cap, son rayon de giration ou encore son statut de navigation (au mouillage, en cours de navigation).

105Convention SOLAS Chapitre V, règlement 19

106Safety of Life At Sea

- **des informations relatives au trajet** : elles sont spécifiques au trajet effectué par le navire. Sont notamment mentionnés la nature et la dangerosité de la cargaison, le port de départ et le port de destination du navire.

En fonction de la vitesse et du type du navire les informations statiques et dynamiques sont envoyées toutes les 2 à 30 secondes. Celles-ci sont ensuite complétées toutes les 6 minutes par les informations relatives au trajet. Bien qu'originellement destiné à l'anti-collision, le système AIS peut être également utilisé à des fins de surveillance. Pour cela, des stations terrestres équipées de récepteurs ont été déployées et permettent le suivi des navires. En France, ce type d'application a notamment été mis en place dans le cadre du projet SPATIONAV (Michoux, 2007).

Si l'AIS apporte une plus value certaine en terme de navigation et de surveillance sa portée reste limitée à la zone des 15 à 20 miles nautiques (sauf à utiliser un système de stations de réception AIS par satellite). Afin de pallier à cela, l'OMI a proposé l'adoption d'un système complémentaire nommé Long Range Identification and Tracking of ships (LRIT). Celui-ci étend considérablement la portée d'émission en offrant une couverture mondiale à l'aide de satellites. Mais, les échanges et les informations sont moins régulières (une émission toutes les 6 heures) et moins complètes.

Volumétrie initiale de la base de données

Les données ont été acquises pendant une période de 4 mois principalement autour de la Méditerranée. Au total, cela représente un volume de près de 14 millions de points. Bien évidemment et même si les moyens et les performances informatiques évoluent constamment, il n'est pas pertinent de traiter l'ensemble de ces données. Seules les positions significatives d'une trajectoire doivent être conservées. De plus, comme cela est illustré sur la Figure 5.3, cela permet également de supprimer les données aberrantes provenant d'une erreur de positionnement GPS. Cette étape sera décrite en détail dans la §5.2.2.

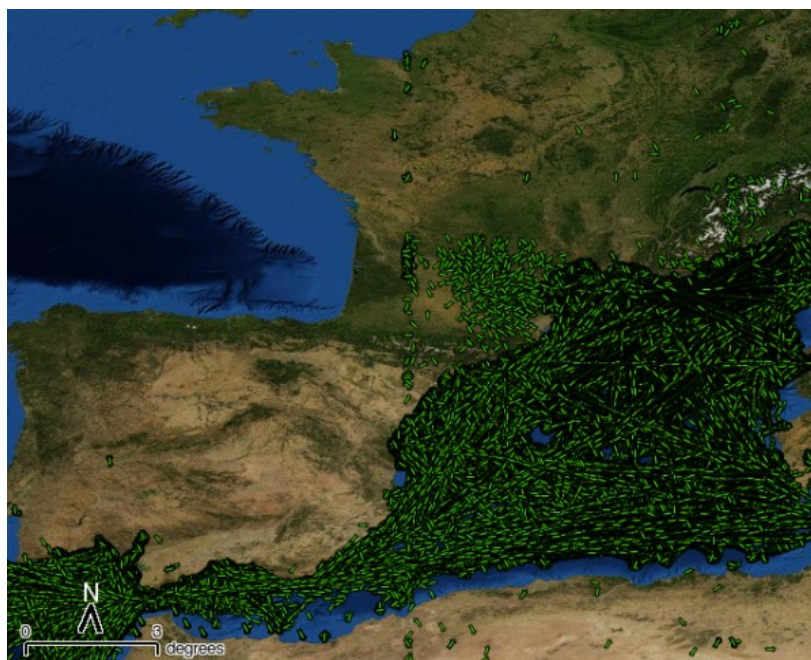


Figure 5.3: Position des navires en méditerranée pour une période de 4 mois.

5.2.1.2 Données annexes

Au-delà des positions des navires, différentes sources de données institutionnelles ou privées ont été utilisées. Deux types peuvent être distingués.

Premièrement, celles provenant de Web Services comme cela est notamment proposé par le site internet Marine trafic¹⁰⁷. Ces informations sont essentiellement non spatiales et permettent d'enrichir le contenu informationnel lié au navire.

Deuxièmement, les données géographiques. Cela concerne notamment la météorologie, les zones maritimes ou encore les ports. Ces données sont stockées dans une base de données PostGis¹⁰⁸ et serviront aussi bien lors du processus d'inférence spatiale que lors de l'affichage des données par l'interface cartographique.

5.2.2. Préparation des données

L'important volume de données recueilli a nécessité une première étape de nettoyage et de filtrage. L'objectif est double. Premièrement, supprimer les enregistrements GPS aberrants qui peuvent apparaître lors de l'initialisation du transpondeur AIS. Deuxièmement, ne sélectionner que les positions significatives d'une trajectoire. Cela permettra ainsi de diminuer le nombre total de données par le moteur d'inférence géographique. Les paragraphes ci-dessous expliquent en détail la procédure qui a été adoptée.

5.2.2.1 Extraction et filtrage des données

Comme cela est illustré sur la Figure 5.3, des erreurs peuvent apparaître lors de l'initialisation du GPS. Certaines positions de navires se retrouvent alors en pleine terre ou à plusieurs milliers de kilomètres de leur position réelle. Sans traitement adéquat, cela entraînerait obligatoirement des biais dans l'analyse. C'est pourquoi une étape de filtrage est nécessaire.

La première vérification qui a été effectuée porte directement sur le message AIS lui-même. En effet, lors de la transmission un paramètre décrivant la qualité du signal GPS est automatiquement ajouté à la trame. Ainsi, seules les données dont le paramètre de précision est inférieur à 10 mètres sont insérées en base. Néanmoins, ce paramètre ne semble pas toujours exact car malgré ce filtre de nombreuses positions aberrantes sont tout de même enregistrées. Une seconde vérification à partir des données antérieures est alors nécessaire. Cela a été réalisé grâce à une procédure stockée directement dans la base de données. Lors de l'insertion d'une nouvelle position, celle-ci est comparée à la position précédente en fonction du temps passé et de la distance. Si la vitesse obtenue est incompatible avec la vitesse maximale par type de navires, alors la position est classifiée comme aberrante et l'insertion en base est alors annulée.

La seconde étape a consisté à ne garder que les données d'entrée et de sorties de port. En effet, à l'intérieur d'un port, l'AIS fonctionnant toujours, un grand nombre de positions sont enregistrées. Celles-ci représentent au total près de 70% du volume total et ne sont pas pertinentes pour l'analyse. Cette étape de nettoyage s'avérait donc nécessaire afin d'optimiser les algorithmes et les traitements futurs.

¹⁰⁷<http://www.marinetraffic.com/ais/>

¹⁰⁸<http://www.postgis.org/>

5.2.2.2 Création des trajectoires

L'objectif de cette étape est de créer la trajectoire d'un navire à partir des données initiales de position. Les différentes étapes sont décrites dans les paragraphes ci-dessous.

Création du graphe de zones

Les données issues des transpondeurs AIS fournissent la position du navire mais non leur trajectoire qu'il est alors nécessaire de modéliser. Or, comme cela a été abordé dans le chapitre 1, différents modèles existent. De plus, contrairement aux objets mobiles se déplaçant sur un réseau, les navires évoluent dans un espace ouvert. Toute la problématique consiste alors à passer d'une représentation ponctuelle à une véritable trajectoire. Pour cela, l'approche adoptée dans ce travail de recherche s'appuie sur le concept de graphes de zones. Celui-ci consiste à sectoriser l'espace en fonction de points d'intérêts (ports, zone de mouillage, etc.) identifiés comme pertinents pour l'analyse (Devogele, 2009; Etienne, 2011). Le passage d'un navire dans l'une de ces zones permet alors d'identifier des points de départ ou d'arrivée nécessaire à la construction d'une trajectoire. Par exemple, la figure ci-dessous (Figure 5.4) représente la position des navires au niveau du détroit de Gibraltar. Cinq zones d'intérêts principales peuvent être identifiées. Quatre sont des ports (A, D, E, C) et la dernière (B) est un goulet d'étranglement. Pour plus de précision, il aurait même été possible de diviser cette dernière en fonction du Dispositif de Séparation de Trafic.

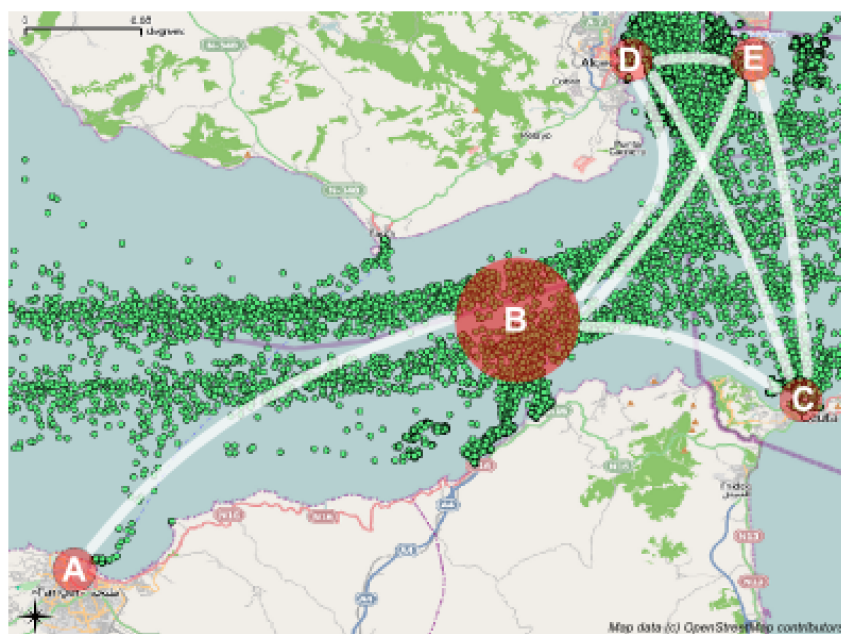


Figure 5.4: Exemple de graphe de zones pour la région de Gibraltar

Actuellement, les données utilisées pour créer ce graphe de zone sont notamment celles des ports mondiaux ainsi que certaines entités (zone de mouillage, zone de séparation de trafic, etc.) contenues dans les cartes électroniques de navigation¹⁰⁹ produites en France par le SHOM¹¹⁰. Le graphe résultant est donc fortement lié aux données spécifiées en entrée et souffre donc d'une certaine limite en terme d'évolutivité. Néanmoins, des travaux sont en cours au sein de notre équipe de recherche afin de rendre ce graphe dynamique (Idiri and Napoli, 2012). Celui-ci pourra alors être

109 Plus connu sous l'acronyme ENC : Electronic Navigational Charts

110 SHOM : Service Hydrographique et Océanographique de la Marine

automatiquement enrichi par de nouvelles zones grâce un processus de fouille de données basé notamment sur des analyses de densité (Giannotti et al., 2009; Pelekis et al., 2009).

Création des trajectoires

À partir des données filtrées et du graphe de zone, il est désormais possible de construire la trajectoire des navires. Pour cela, toutes les positions sont ordonnées temporellement puis reliées entre elles. Quand un navire entre dans une zone d'intérêt, la trajectoire est alors découpée. Le résultat final est un ensemble de segments reliant les éléments du graphe de zones.

Même si après l'étape de filtrage le volume de données à analyser a considérablement diminué, il est encore possible d'optimiser le processus en ne gardant que les positions significatives. En effet, l'AIS du navire ayant été conçu à des fins d'anti-collision, la position du navire est envoyée selon un pas de temps régulier. Or, ces positions ne sont pas nécessaires pour modéliser la trajectoire à analyser. De ce fait, seules les plus significatives (changement de cap, de vitesse, etc.) sont gardées. Pour cela, un filtre de Douglas et Peucker Spatio-Temporel (DPST) a été utilisé (Bertrand et al., 2007; Etienne, 2011). Comme son nom l'indique, ce filtre utilise la même logique que l'algorithme initialement proposé par Douglas et Peucker (Douglas and Peucker, 1973) auquel a été ajoutée une composante temporelle. Plus concrètement, il s'agit de déterminer en fonction de la vitesse du navire le point le plus éloigné entre : (1) la trajectoire réelle et (2) la trajectoire entre le point de départ et le point d'arrivée. Le point identifié forme alors le sommet d'une nouvelle trajectoire et l'opération est effectuée jusqu'à n'avoir plus que des points dont la tolérance est inférieure au seuil fixé (Figure 5.5). En raison de la précision d'un GPS, ce seuil a été fixé à 10m. Les différentes étapes du filtre DPST ainsi que l'algorithme sont disponibles dans la thèse d'Etienne (Etienne, 2011).

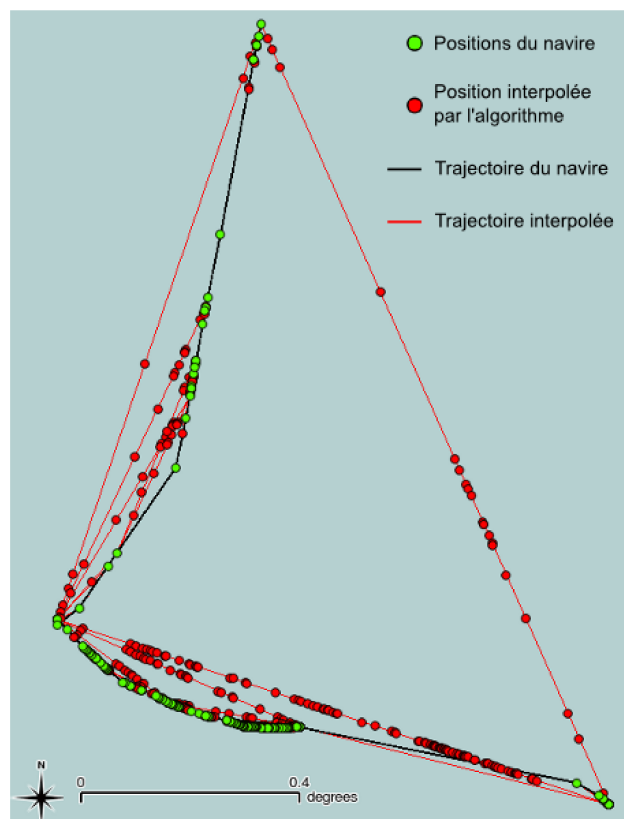


Figure 5.5: Filtre de Douglas et Peucker Spatio-Temporel

Au final, le filtre DPST aura permis de supprimer les erreurs locales de positionnement et de réduire considérablement le volume de données. Les trajectoires ainsi obtenues ont été intégrées dans la base de données avant leur traitement par le moteur d'inférence spatiale.

5.2.3. Enrichissement des données et création de la trajectoire sémantique

Par la suite, les données AIS et les trajectoires initialement obtenues ont été enrichies à partir de différentes sources d'informations. Certaines sont accessibles directement depuis des web services et d'autres sont issues de données internes. Ces différentes données sont le socle informationnel permettant de passer d'une trajectoire à une trajectoire sémantique.

5.2.3.1 Enrichissement des données

L'enrichissement des données initiale a été réalisé de deux façons. La première est basée sur les Services Web accessibles au moyen d'une simple requête HTTP. La seconde utilise différentes données locales stockées sous forme de fichiers ou directement dans la base.

Enrichissement des données avec des sources externes

Ces sources proviennent pour certaines de sites internet tels que MarineTraffic¹¹¹ qui permettent l'interrogation de leurs données à partir de simples requêtes Http. Par exemple, la requête Http pour obtenir la liste des navires (au format XML) des navires passés à proximité de la station de réception 1234567890 ces deux dernières minutes est la suivante :

```
http://www.marinetraffic.com/ais/exportraw.aspx?id=1234567890&protocol=xml
```

Le résultat retourné est une liste de navires au format XML. Pour des raisons de lisibilité, seule la version allégée est présentée ci-dessous. Néanmoins, il est possible d'obtenir un plus grand nombre d'informations (destination, port de départ, nom du navire, etc.) en ajoutant le paramètre *msgtype* avec comme valeur *extended* à la requête.

```
<POS>
  <row MMSI="271001101" LAT="37.46283" LON="25.327271" SPEED="1" COURSE="265"
  STATUS="15" TIMESTAMP="2012-06-19T09:28:00"/>
  <row MMSI="237327000" LAT="37.432339" LON="24.943821" SPEED="0" COURSE="353"
  STATUS="15" TIMESTAMP="2012-06-19T09:28:00"/>
  <row MMSI="240869000" LAT="37.31905" LON="25.21933" SPEED="60" COURSE="338"
  STATUS="0" TIMESTAMP="2012-06-19T09:28:00"/>
</POS>
```

Enrichissement des données avec des sources internes

Différentes autres sources de données, stockées en local, ont également été utilisées. Il s'agit notamment de celles provenant du SHOM pour la partie réglementaire ou de producteur de données

¹¹¹<http://www.marinetraffic.com/ais/>

météorologique pour les conditions de navigation. Les formats hétérogènes (*netcdf*, *grib*, etc.) ont nécessité la création de connecteurs spécifiques qui traduisent ces données pour les insérer ensuite dans la base PostgreSQL/PostGIS.

5.2.3.2 Création de la trajectoire sémantique

Maintenant que les positions ont été regroupées sous la forme de trajectoires cohérentes à l'aide du graphe de zone, il est possible d'enrichir celles-ci. Pour cela les différentes positions sont classifiées sous la forme de *stop* et de *move* tel que défini par Spaccapietra (Spaccapietra et al., 2008). Cet état est défini directement dans la base de données comme attribut du navire.

Mais, comme le souligne Baglioni (Baglioni et al., 2008), la modélisation des trajectoires proposée par Spaccapietra est volontairement générique. De ce fait et en fonction du domaine d'étude, celle-ci peut être enrichie par de nouvelles notions. Dans le cadre du domaine d'étude sur lequel porte ce travail de thèse, il a été nécessaire d'enrichir cette modélisation à l'aide d'une troisième valeur permettant de gérer la perte de localisation. En effet, il peut arriver qu'un navire coupe volontairement ou involontairement son AIS. C'est le cas par exemple des navires de pêches qui ne souhaitent pas être suivis, ou d'un navire qui pourrait potentiellement être en perdition. Pour l'étude de comportements anormaux, cette information est particulièrement importante. De ce fait, un paramètre *lost* a été également défini. Celui-ci est automatiquement calculé par rapport à la dernière position et au laps de temps écoulé. Conceptuellement, cela signifie qu'une position anciennement classifiée comme *stop* ou *move* peut évoluer vers un *lost* (Figure 5.6).

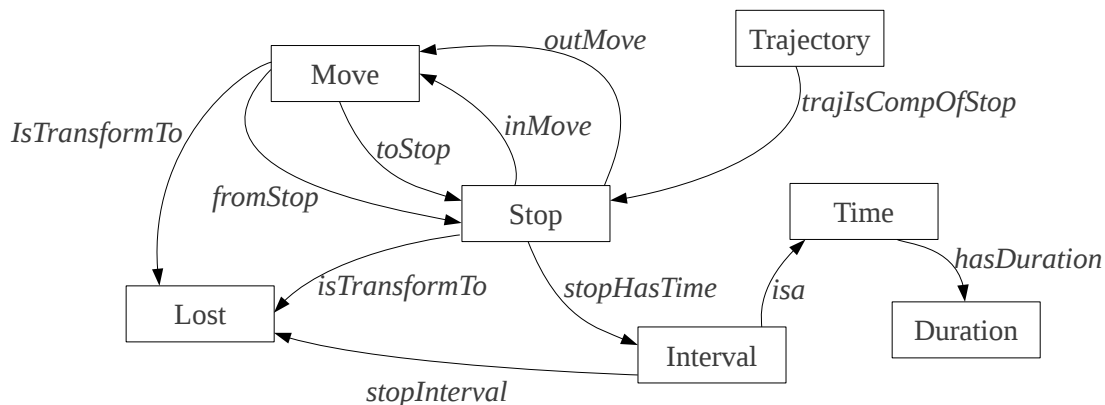


Figure 5.6: Représentation ontologique de la notion de trajectoire sémantique avec ajout de la notion *Lost*

Après avoir catégorisé les positions du navire, celles-ci sont enrichies à l'aide des différentes sources d'informations présentées précédemment. Cela permet ainsi d'obtenir des trajectoires sémantiques sur lesquelles porteront les analyses de comportements

5.3. Module d'inférence

Le module d'inférence a pour objectif d'interpréter et d'exécuter les règles SWRL définies par les utilisateurs afin de produire une base de connaissances enrichie. Ce processus appelle différents composants en fonction du caractère spatial ou non spatial des informations et des règles. Les étapes nécessaires sont décrites plus en détail dans les prochains paragraphes.

5.3.1. Architecture du module d'inférence

Le moteur d'inférence s'appuie sur l'ontologie créée mais aussi sur des composants informatiques qui vont interpréter, analyser et produire de nouvelles connaissances à partir de celles déjà présentes. L'architecture adoptée ainsi que les caractéristiques du moteur d'inférence choisi.

5.3.1.1 Architecture générale

Les différentes étapes nécessaires à la création de l'ontologie maritime enrichie sont illustrées par la Figure 5.7.

Tout d'abord, les fichiers contenant les règles sont importés par le module. Comme cela a déjà été abordé dans le paragraphe 4.3 (p.137), deux types de règles peuvent être identifiés. Si les règles classiques peuvent être directement interprétées, les règles spatiales nécessitent quant à elles un processus de traduction. En effet, le langage SWRL ne proposant aucun type ni fonction spatiale, il a été nécessaire de réaliser une étape préalable d'interprétation. Celle-ci permet de reconnaître la fonction géographique utilisée (*intersects*, *touches*, etc.) ainsi que les arguments passés en paramètre avant de l'envoyer au module de raisonnement spatial.

Ensuite, le module d'inférence va pour chaque individu (navires) exécuter les différentes règles contenues dans la base de connaissances.

Enfin, le résultat final de ce processus d'inférence est stocké dans une base de connaissances enrichie qui servira lors du calcul de similarité de comportements.

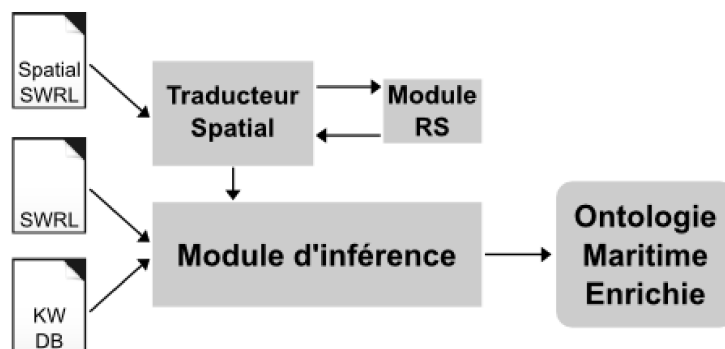


Figure 5.7: Étapes et flux du module d'inférence

Chacune des étapes nécessaires à l'obtention de cette ontologie maritime enrichie est expliquée dans les paragraphes suivants.

5.3.1.2 Caractéristique du moteur d'inférences

Afin d'exploiter l'ontologie et les caractéristiques spatiales de celles-ci il est nécessaire de l'intégrer à un moteur de raisonnement. Ce moteur a pour objectif d'inférer de nouveaux faits à partir des données existantes. Dans le cas de l'ontologie maritime, celui-ci aura à charge d'évaluer la situation en fonction des positions et des caractéristiques des navires.

La première étape a consisté dans le choix du moteur d'inférence à utiliser. En effet, comme cela a déjà été abordé dans le paragraphe 2.4.3 (P.88) l'offre dans ce domaine est riche. À l'heure actuelle plus d'une vingtaine de moteurs existent et ces derniers varient en fonction des algorithmes implémentés, des langages de programmation utilisés ou encore par les fonctionnalités proposées (Mishra and Kumar, 2010). Différents tests ont donc été effectués afin de sélectionner le moteur le plus adapté au développement du prototype d'analyse de comportements anormaux de navires. Ces derniers ont permis d'identifier Pellet comme étant celui répondant au mieux à la problématique mise en évidence.

Développé initialement par le MINDSWAP Group¹¹², Pellet est aujourd'hui principalement maintenu par la société Clark & Parsia¹¹³. Concernant le processus d'inférence, Pellet s'appuie sur un algorithme de vérification de satisfiabilité à base de tableaux et il permet d'effectuer des raisonnements sur des ontologies au format OWL-DL et OWL-Lite au travers de nombreuses interfaces (API Java, en ligne de commande, via le protocole DIG, etc.). De plus, il offre une compatibilité avec le langage OWL 2, un très bon support du langage SWRL et un interfaçage avec les bibliothèques JENA et API OWL. Enfin, et c'est l'une des raisons principales du choix d'utiliser Pellet, il offre la possibilité d'ajouter de nouvelles fonctions built-in (Sirin et al., 2007).

Contrairement à d'autres raisonneurs, l'architecture de Pellet a été conçue dès le départ pour fonctionner avec le langage OWL. Ainsi, il est par exemple possible de raisonner avec des instances de la Abox sans pour autant nécessiter une unicité des noms (*Unique Name Assumption*). De plus, chaque composant a été implémenté avec un objectif de modularité ce qui permet de facilement créer des points d'extensions. Plus concrètement, l'architecture de Pellet s'articule autour de trois composants principaux (Figure 5.8) :

- **le moteur d'inférence** (tableau `reasoner`) : module essentiel de Pellet, la fonctionnalité principale de ce composant est de s'assurer de la consistance de l'ontologie. Une ontologie est dite consistante si celle-ci satisfait tous les axiomes qui ont été définis. Conçu de manière extensible, il est possible d'implémenter différents algorithmes de complétion.
- **L'interface à la base de connaissances** (Knowledge Base Interface) : ce composant offre les éléments nécessaires à l'interprétation des requêtes qui peuvent être effectuées sur l'ontologie. Ces requêtes peuvent être relatives aux classes (`getSubClasses`, `getDisjoints`, etc.), aux propriétés (`getSubProperties`, `isFunctional`, etc.) ou encore aux individus (`getTypes`, `getPropertyvalues`, etc.).
- **Le moteur de requête Abox** (Abox Query Engine) : l'interface à la base de connaissance est couplée au moteur de requête Abox afin de répondre aux requêtes potentielles. Ce composant supporte aussi bien le langage Sparql que RDQL.

112 <http://www.mindswap.org/>

113 <http://clarkparsia.com/>

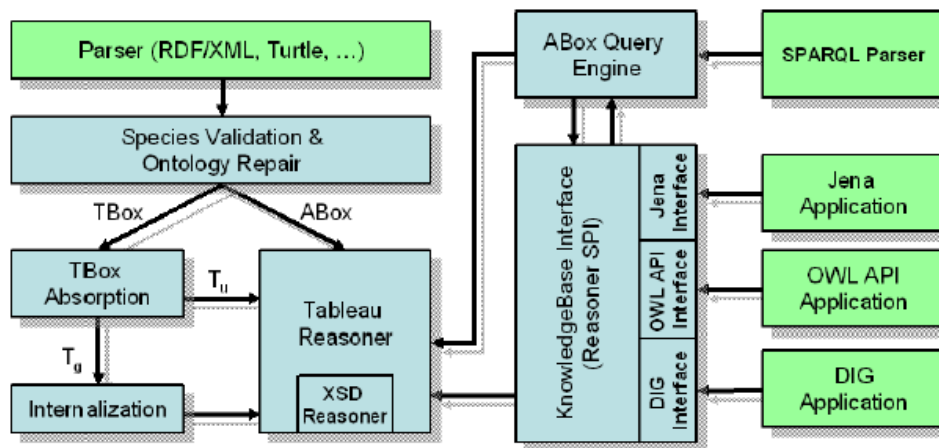


Figure 5.8: Principaux composants du raisonneur Pellet. Source (Sirin et al., 2007)

A partir des données contenues dans l'ontologie, le raisonneur va exécuter toutes les règles SWRL afin de déterminer si un navire possède ou non un comportement anormal. En fonction du résultat de l'inférence, de nouveaux attributs et de nouvelles relations seront automatiquement ajoutés et un niveau de criticité sera attribué au navire concerné.

5.3.2. Processus d'inférence

Le processus d'inférence utilise directement les potentialités du langage SWRL. Pour cela, le fichier contenant les règles est tout d'abord interprété par OWL API¹¹⁴. Cela permet d'obtenir une représentation informatique facilement manipulable par le moteur d'inférence. En fonction du type de règles, deux processus distincts sont appelés. Dans le premier cas, il s'agit de règles utilisant les opérateurs classiques du langage SWRL. Dans le second cas, il est nécessaire de passer au préalable par un processus de traduction et d'interprétation.

5.3.2.1 Processus d'inférence classique

Ce processus utilise directement les potentialités du langage SWRL. En effet, le SWRL propose en un certain nombre d'opérateurs (*built-in*) permettant d'effectuer des comparaisons sur des chaînes de caractères, des nombres ou encore des dates.

Par exemple, la règle pour contrôler la vitesse du navire utilise le *built-in* mathématique *greaterThan* qui permet de comparer deux valeurs. Si cette valeur est supérieure à la vitesse maximale autorisée du navire (*speedTypeVessel*), alors une alerte (*Alert_HighSpeed*) lui est automatiquement affectée.

```
Vessel(?vessel),
OP_Vessel_hasVesselType(?vessel, ?vesselType),
DP_Vessel_hasSpeed(?vessel, ?speedVessel),
DP_Vessel_hasSpeed(?vesselType, ?speedTypeVessel),
greaterThan(?speedVessel, ?speedTypeVessel) ->
OP_Vessel_hasAlert(?vessel, Alert_HighSpeed)
```

114 <http://owlapi.sourceforge.net/>

5.3.2.2 Processus d'inférence spatio-temporel

Fonctions spatiales

Les fonctions spatio-temporelles définies dans l'ontologie permettent une utilisation plus intuitive de la composante géographique tout en exploitant au mieux les informations sémantiques. Pour le moment, seules celles issues du formalisme RCC8 ainsi que certaines fonctions spécifiques au domaine maritime ont été implémentées.

Ces différentes fonctions sont stockées et décrites dans l'ontologie. La classe principale *spatialOperation* contient l'ensemble des opérations possibles. Celle-ci est ensuite divisée en trois sous-classes (*spatialRelational*, *spatialConstructor* et *spatialMaritime*) dépendantes du type de fonctions spatiales (Figure 5.9) :

- ***spatialRelational*** : ces fonctions s'appuient sur le formalisme RCC8 et permettent de comparer deux objets géographiques entre eux. Par exemple la fonction *intersects*, vérifie si deux géométries s'intersectent.
- ***spatialConstructor*** : ces fonctions vont construire une nouvelle géométrie à partir d'une géométrie existante. C'est le cas notamment de *buffer* qui crée une zone tampon autour d'un objet géographique en fonction d'une distance donnée. Elles sont essentiellement utilisées en complément de fonction de type *spatialRelational* ou *spatialMaritime*. Ce type de fonction peut être utile par exemple pour créer une zone de danger initialement constituée de point (ex : rochers affleurants).
- ***spatialMaritime*** : ces fonctions sont spécifiques au domaine maritime et sont définies en fonction des besoins des experts du domaine. Par exemple, il était nécessaire d'identifier les trajectoires parallèles. De ce fait, comme cela a été notamment abordé par Devogele (Devogele, 2009), une fonction s'appuyant sur l'algorithme de la distance de Hausdorff a été implémentée (Hausdorff, 1918).

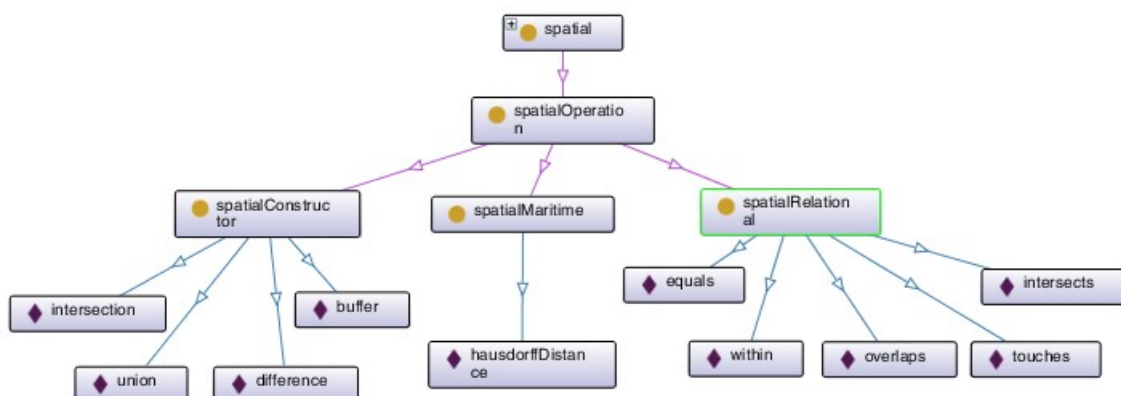


Figure 5.9: Exemple de fonctions spatiales disponibles et utilisables par le moteur d'inférence

Il est important de préciser que ces différentes classes ne sont pas directement utilisées par les experts. Elles sont surtout présentées à titre informatif pour expliquer le comportement de la fonction. En effet, l'architecture du module d'inférence spatial s'appuie sur l'utilisation de mots clés

spécifiques directement intégrés à la règle SWRL. Ce choix a été fait notamment pour garder une compatibilité maximale avec l'écriture de fonctions spatiales telle que définie par les spécifications de l'OpenGIS et par le standard SQL/MM¹¹⁵.

Module de traduction, du spatial vers le sémantique

Ce module a pour objectif de traduire les requêtes spatiales initialement écrites en SWRL afin de pouvoir ensuite les exécuter. Pour cela, il interprète la fonction ainsi que les objets géographiques et applique l'opération nécessaire. Les résultats sont ensuite traduits sous la forme de propriétés de données ou d'objets dans la base de connaissances.

Processus de traduction

Le processus de traduction interprète les règles SWRL qui ont été définies et sélectionne celles qui possèdent un caractère spatial. Cela est notamment rendu possible par les points d'extensions offerts par le langage SWRL sous la forme de *built-in* et par les caractéristiques du moteur d'inférence sélectionné (Pellet). Par rapport à cette fonction, la classe java correspondante est ensuite appelée. Par contre, cela nécessite au préalable que soient définies dans le code de l'application les fonctions à identifier. Cela se fait grâce à la classe *BuiltInRegistry* de la bibliothèque Pellet. Celle-ci prend deux paramètres, tout d'abord le nom du *built-in* puis le nom de la méthode à exécuter :

```
BuiltInRegistry.instance.registerBuiltIn(  
    "geo:builtIn#intersects",  
    new GeneralFunctionBuiltIn( new inference.RuleIntersects() )  
);
```

Ainsi, quand le moteur d'inférence trouvera une règle portant la signature *intersects*, il appellera alors automatiquement la méthode *new inference.RuleIntersects()* avec les arguments passés en paramètre. Ensuite, l'opération spatiale est réalisée grâce à la bibliothèque Java Topology Suite¹¹⁶ qui possède de nombreux algorithmes relatifs au domaine topologique.

Plus concrètement, dans le cas d'une règle concernant l'intersection d'un navire et d'une zone réglementaires, l'atome spatial est défini de la manière suivante :

```
BuiltInAtom(  
    geo:intersects  
    Variable(var:position)  
    Variable(var:geoData)  
);
```

Cet atome prend nécessairement deux variables à savoir la position du navire et les données qui seront utilisées pour l'intersection. Les données disponibles sont directement définies et décrites dans l'ontologie. Cela apporte aux utilisateurs une plus grande facilité car ils n'ont pas besoin de connaître les schémas parfois complexes des données. Pour le moment, celles-ci sont principalement au format shapefile. Mais, il n'est pas exclu d'intégrer d'autres connecteurs (postgis, mapinfo, etc.) afin de couvrir une plus grande gamme de données.

115 SQL/MM Part 3 : Spatial

116 <http://www.vividsolutions.com/JTS>

5.3.3. Résultats du processus d'inférence

Les résultats issus du processus d'inférence sont ajoutés aux individus sous la forme de propriétés d'objets ou de données. La principale propriété, nommée "*OP_Alert_hasAlert*", permet de spécifier qu'un navire fait, ou à fait l'objet d'une alerte. Celle-ci relie alors le navire, à l'une des alertes spécifiées dans l'ontologie.

5.3.3.1 Résultats de l'inférence avec le SWRL classique

Comme cela a déjà été abordé, il peut être nécessaire de vérifier qu'un navire ne dépasse pas une vitesse moyenne en fonction de son type. Dans le cas contraire, une alerte doit donc être automatiquement levée. Cela se traduit par la création d'une propriété d'objet "*OP_Alert_hasAlert*" reliant le navire identifié et l'individu "*Alert_Speed_HighSpeed*" de la classe *Alert* (Figure 5.12).

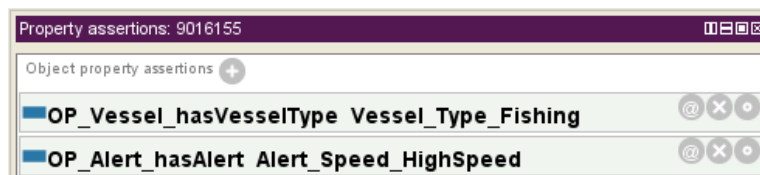


Figure 5.10: Alerte de type HighSpeed

5.3.3.2 Résultats de l'inférence avec le SWRL spatial

Tout comme pour les règles SWRL classiques, les règles SWRL spatiales et par extension maritime utilisent la propriété d'objet "*OP_Alert_hasAlert*".

La Figure 5.11, est le cas d'une alerte spatiale obtenue à l'aide de la fonction SWRL *intersects*. Le navire a été détecté comme naviguant dans une zone réglementaire et une alerte "*Alert_Area_Restricted*" lui est alors automatiquement ajoutée.

La navigation en parallèle de deux bateaux de pêche peut être le signe d'une pêche en collaboration. Il est donc intéressant d'identifier ce type d'acte. Pour cela, une fonction évaluant la similarité des trajectoires de navires a été implémentée. Si un comportement de ce type est détecté alors une alerte *Alert_Trajectory_Parallel* est ajoutée au navire concerné (Figure 5.12).

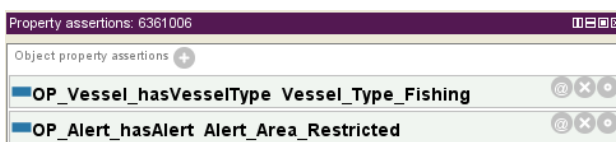


Figure 5.11: Alerte spatiale

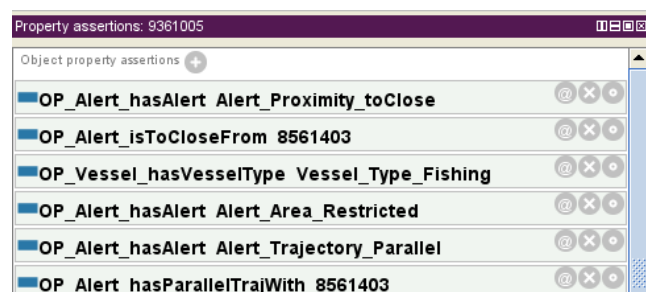


Figure 5.12: Alerte maritime

5.4. Module de calcul de similarité des comportements anormaux

Le module de calcul de similarité permet de déterminer le comportement d'un navire par rapport aux scénarios définis dans l'ontologie. Pour cela, des algorithmes spécifiques au raisonnement à partir de cas appliqués aux ontologies ont été utilisés. Les paragraphes ci-dessous présentent la démarche ainsi que les étapes qui ont été appliquées.

5.4.1. Raisonnement à partir de cas

Issu de travaux à la fois en sciences cognitives et en intelligence artificielle, le Raisonnement à Partir de Cas (RàPC)¹¹⁷ est un paradigme de résolution de problème fondé sur la réutilisation d'expériences passées pour résoudre de nouveaux problèmes. L'origine de ce domaine trouve notamment ses racines dans les travaux de Schank sur la modélisation de la mémoire dynamique (Schank and Abelson, 1977; Schank, 1983). Dans ces derniers, il met en évidence que la capacité d'une personne à réagir face à une situation connue ou inédite est fortement liée aux expériences passées sur lesquels il pourra fonder son raisonnement.

Ces travaux précurseurs ont par la suite trouvé un écho plus général notamment grâce aux conférences organisées par la DARPA à la fin des années 80 aux États-Unis et en Europe au début des années 90.

Les paragraphes suivants présentent les principes généraux, les éléments constitutifs et les cycles d'un RÀPC.

5.4.1.1 Principes généraux

Conceptuellement, le RàPC s'appuie sur l'existence de deux espaces : l'espace de problèmes et l'espace de solutions. Il existe entre eux une relation binaire nommée (*Sol*) signifiant « a pour solution ». Ces différents éléments forment ce que Mille (Mille et al., 1996) nomme le carré d'analogie qui exprime le lien entre la description d'un cas source, d'un cas cible et leurs solutions respectives. Ainsi, d'une manière générale, raisonner à partir d'un cas, c'est résoudre le problème cible en faisant appel à une base de cas. Ces éléments peuvent être alors représentés de la manière suivante (Lieber et al., 2004; Lieber, 2008) :

$\text{RàPC} : (\text{cible}, \text{BaseDeCas}) \mid \rightarrow \text{Sol}(\text{cible}) \in \text{Solutions}$

Contrairement à d'autres approches (systèmes experts, à base de règles, etc.), le RàPC offre de nombreux avantages. En effet, celle-ci est plus simple à mettre en œuvre, elle permet d'éviter les problèmes d'acquisition de connaissances (« Knowledge Bottleneck ») et il n'est pas nécessaire de disposer d'une connaissance exhaustive du domaine pour résoudre un problème. De plus, elle est particulièrement adaptée aux applications dont les tâches sont accomplies par des experts du domaine et où la recherche de solution optimale n'est pas une nécessité (Lamontagne and Lapalme, 2002).

Mais, ce paradigme s'appuie également sur un certain nombre d'hypothèses fortes qui doivent être satisfaites. Tout d'abord, le RàPC suppose que le monde est régulier. Cela signifie que ce qui est vrai

117 Plus connu sous l'acronyme anglais CBR – Case Based Reasoning

aujourd'hui le sera également demain. De plus, il est nécessaire que les situations se répètent, sinon il n'y a pas d'utilité à les remémorer et à les réutiliser. Enfin, on présuppose également que les problèmes similaires ont des solutions similaires. La satisfaction de ces éléments est une condition essentielle au bon fonctionnement et à la pertinence du RàPC.

5.4.1.2 Éléments constitutifs d'un système de raisonnement à partir de cas

Comme cela est illustré par la figure ci-dessous (Figure 5.13), un système RàPC est composé de quatre éléments principaux nommés conteneurs de connaissances : (1) un vocabulaire d'indexation, (2) une base de cas, (3) des mesures de similarité et enfin (4) des règles d'adaptation.

Le conteneur vocabulaire est l'élément central au sein du système. Il représente les connaissances nécessaires à la description d'un domaine d'application et contient l'ensemble des représentations utilisées pour représenter les cas dans la base de cas. En fonction des besoins, celui-ci peut prendre différentes formes allant d'une représentation peu formalisée (glossaire, taxonomie, etc.) à une représentation très formalisée (ontologie). La base de cas constitue la mémoire d'un raisonneur. Ces cas peuvent être homogènes c'est-à-dire qu'ils partagent la même structure de données ou hétérogènes. Les mesures de similarité permettent de calculer les liens potentiels entre le nouveau problème (problème cible) et le ou les cas sources similaires (problèmes sources). Différentes mesures existent dans la littérature (prenant en compte les structures complexes, l'espace, le temps, etc.) mais les plus connus et également les plus utilisés sont celles de similarité et de distance. Enfin, les règles d'adaptation qui sont des heuristiques du domaine permettant de modifier les solutions et d'évaluer leur applicabilité à de nouvelles situations (Abou Assali, 2010).

Tous ces conteneurs de connaissances sont construits à l'aide de différentes sources pouvant provenir d'une base de données, de documents ou encore de l'expertise de spécialiste du domaine. Contrairement au processus de résolution de problèmes, la phase d'élicitation des connaissances peut se faire de manière off-line et permet la création de la base de cas qui servira lors des phases d'analyses (Lamontagne and Lapalme, 2002).

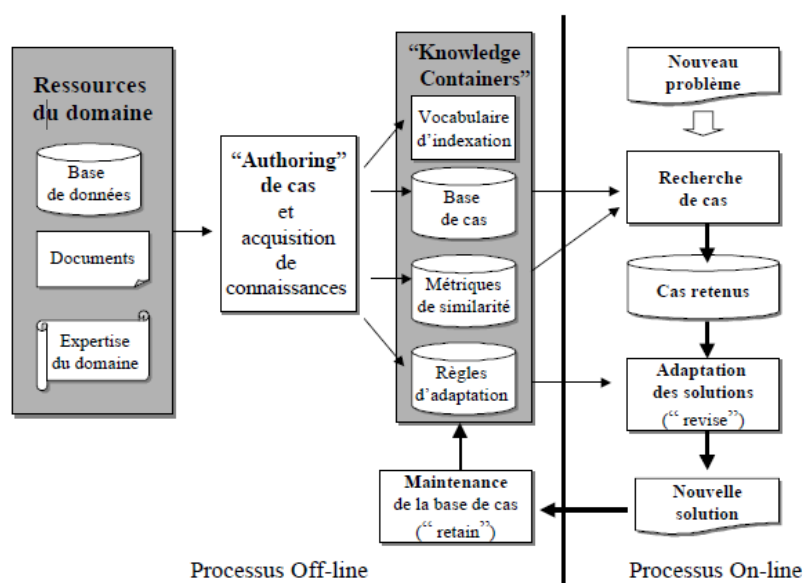


Figure 5.13: Modèle générique d'un système de raisonnement à partir de cas. Source (Lamontagne et Lapalme, 2002)

Dans cette section, les éléments du processus off-line ont été présentés. Ceux correspondant au processus On-Line, relatif à la résolution d'un problème seront abordé dans le paragraphe suivant.

5.4.1.3 Cycle d'un Raisonnement à partir de cas

Bien que le nombre de phases correspondant au cycle d'un RàPC varie en fonction des auteurs, les travaux les plus récents réalisés par Cordier (Cordier et al., 2007) distinguent cinq principales étapes : (1) élaboration d'un nouveau cas, (2) remémoration des cas passés pertinents, (3) adaptation du ou des cas sources, (4) révision de la solution obtenue et enfin (5) mémorisation de cette solution en vue d'une utilisation future (Figure 5.14).

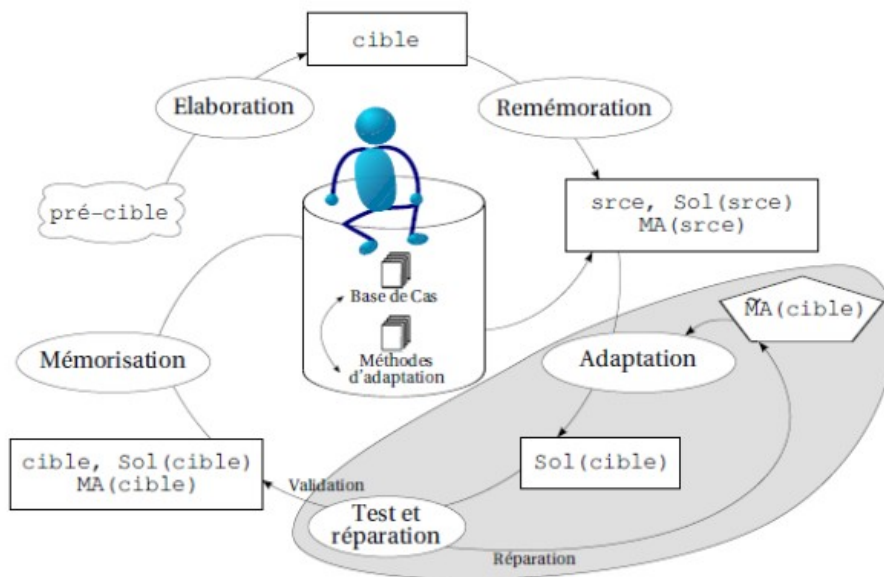


Figure 5.14: Le cycle RàPC re-revisité. Source (Cordier et al., 2007)

Durant la phase d'élaboration, les informations concernant un nouveau problème sont agencées ce qui conduit à l'élaboration d'un problème cible pour lequel il sera nécessaire de trouver une solution. Vient ensuite la phase de remémoration qui vise à chercher les cas sources similaires correspondant complètement ou partiellement au cas cible. La troisième phase consiste à adapter la ou les solutions trouvées en fonction des spécificités du nouveau problème. La phase de réparation que l'on retrouve également sous le terme de révision vise à valider ou au contraire refuser la solution proposée. Cette étape peut se faire en s'appuyant sur les connaissances d'un expert ou à l'aide d'un processus automatique. Enfin, la phase de mémorisation qui stocke le cas nouveau cas résolu dans la base de cas.

5.4.1.4 Raisonnement à partir de cas et ontologies

Comme cela a déjà été partiellement abordé, les ontologies peuvent être utilisées dans des systèmes de RàPC. Dans ce type de cas, qualifiée par Aamodt (Aamodt, 1991) de *knowledge-intensive*, les connaissances sont alors structurées à l'aide d'ontologies qui peuvent servir de (Recio-García, Díaz-Agudo, González-Calero, et al., 2006) :

- structure de stockage et d'indexation des cas en utilisant des instances ou des concepts de l'ontologie

5.4. Module de calcul de similarité des comportements anormaux

- vocabulaire pour définir la structure des cas et cela même si ces derniers ne sont pas stockés dans l'ontologie elle-même mais sur un autre support comme une base de données.
- vocabulaire pour une meilleure définition des requêtes. Grâce à l'ontologie, l'utilisateur dispose d'une terminologie plus riche permettant de ce fait une meilleure description.
- support lors phases de remémoration, de calcul de similarité, d'adaptation et de mémorisation

Cette approche unissant les ontologies et le RàPC a notamment été mise en application au sein du système Creek développé par Aamodt (Aamodt, 1991, 2004); dans les travaux d'Aquin (d' Aquin et al., 2006) qui a proposé une extension du langage OWL pour la représentation de connaissances dans un RàPC; ou encore dans le modèle CBROnto réalisé par Díaz-Agudo et González-Calero (Díaz-Agudo and González-Calero, 2002). Ces derniers travaux sont particulièrement intéressants car l'architecture proposée est indépendante du domaine d'application. En plus de cette structure, les auteurs ont également développé une bibliothèque Java nommée jColibri permettant la création d'un système RàPC à l'aide de différents supports de stockage dont les ontologies (Recio-García, Díaz-Agudo, González-Calero, et al., 2006; Díaz-Agudo et al., 2007). Dans le cadre de ce travail de recherche, c'est cette bibliothèque qui a été utilisée. L'approche adoptée est décrite dans les paragraphes suivants.

5.4.2. Architecture du module de calcul de similarité

Le module de similarité a pour objectif de comparer une situation observée aux scénarios déjà enregistrés dans la base de connaissances. Pour cela une approche s'appuyant sur le Raisonnement À Partir de Cas (RÀPC) est proposée. Les paragraphes suivants présentent la démarche adoptée.

5.4.2.1 Architecture générale

Le module de calcul de similarité repose principalement sur la bibliothèque jColibri. Celle-ci offre les connecteurs et les mesures de similarité permettant de calculer les scénarios potentiels par rapport à une situation donnée. Pour cela différentes étapes et différents sous-modules sont nécessaires (Figure 5.15). Tout d'abord, il s'agit à partir d'un cas source de récupérer les cas cibles correspondants contenus dans l'ontologie. Ensuite, pour chacun d'entre eux il sera nécessaire d'effectuer tout d'abord un calcul de similarité local qui servira ensuite lors du calcul de similarité globale. Les différentes étapes et méthodes sont expliquées en détail dans les paragraphes suivants.

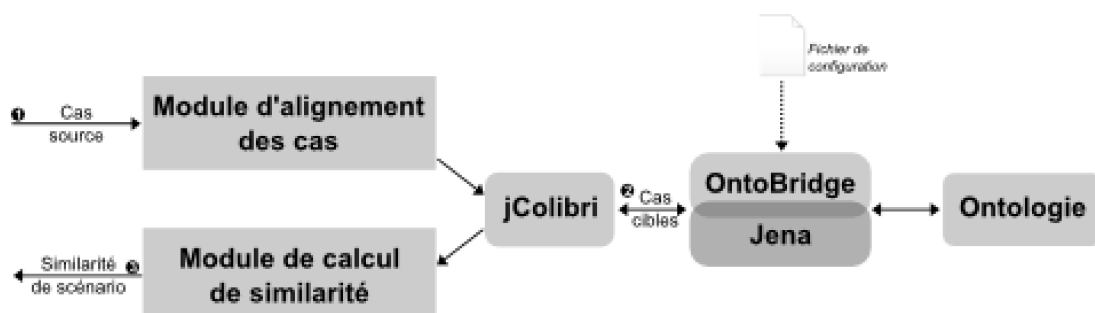


Figure 5.15: Étapes et flux du module de calcul de similarité

5.4.2.2 Présentation générale de jColibri

JColibri¹¹⁸ est une bibliothèque Open Source écrite en langage Java permettant la création de systèmes RàPC. Les choix qui ont conduit à son utilisation sont principalement :

- ses connecteurs natifs permettant d'accéder à des connaissances stockées aussi bien dans une base de données que dans des fichiers texte ou encore dans une ontologie. L'accès à l'ontologie se fait par un connecteur spécifique, nommé Ontobridge¹¹⁹, basé sur la bibliothèque Jena¹²⁰.
- son architecture modulaire
- l'intégration des différentes phases du RàPC (remémoration, évaluation, mémorisation, etc.)
- une documentation complète et exhaustive ainsi que la disponibilité de nombreux exemples

La version utilisée pour le développement du prototype est jColibri 2.0. Celle-ci offre notamment : une meilleure séparation entre les classes principales et la description des cas ; une représentation simplifiée des cas grâce à l'utilisation de composants logiciels réutilisables (*Java Beans*) ; ou encore la possibilité de gérer la persistance des objets avec hibernate (Figure 5.16) (Recio-García, Díaz-Agudo, Sanchez-Ruiz, et al., 2006).

Bien que seule la bibliothèque jColibri ait été utilisée dans ce travail de recherche, il est important de signaler qu'il existe également une interface graphique spécifiquement dédiée à la création d'un système RàPC. Nommée Colibri Studio¹²¹, celle-ci permet de ne pas avoir à manipuler le code source et offre un accès intuitif pour des non-spécialistes du domaine.

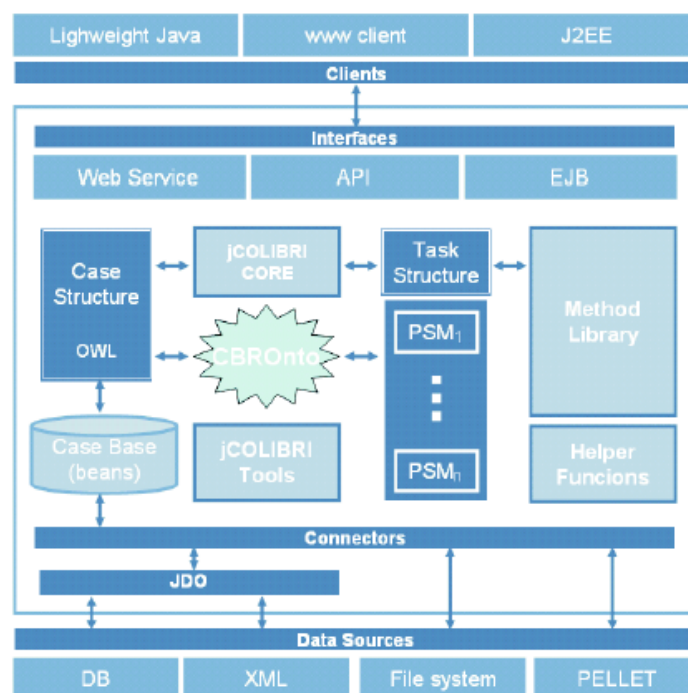


Figure 5.16: Architecture de jColibri 2.0. Source (Recio-garcía et al., 2006)

118 <http://gaia.fdi.ucm.es/research/colibri/jcolibri>

119 <http://gaia.fdi.ucm.es/research/ontobridge>

120 <http://jena.apache.org/>

121 <http://gaia.fdi.ucm.es/research/colibri/colibrstudio>

5.4. Module de calcul de similarité des comportements anormaux

Malgré l'intérêt de cette bibliothèque pour notre architecture, il faut souligner que celle-ci ne permet pas le traitement de cas dynamiques ou hétérogènes. En effet, dans sa configuration actuelle, elle nécessite que les attributs des cas soient précisés dans le code source de l'application. Dans le cadre du prototype développé, cela n'a que peu d'incidence. Mais, si une évolution future devait être envisagée, il serait intéressant d'intégrer les améliorations réalisées par Abou Assali permettant la gestion de ce type de configuration (Abou Assali, 2010).

5.4.3. intégration de l'ontologie et calcul de similarité

Le calcul de similarité permet de déterminer les scénarios correspondants à une situation observée. Les étapes et l'approche adoptées sont présentées dans les paragraphes ci-dessous.

5.4.3.1 Configuration de jColibri et intégration de l'ontologie

La configuration générale du module de RàPC se fait simplement à l'aide d'un fichier XML. Comme cela est illustré par l'extrait du fichier de configuration ci-dessous, celui-ci contient les informations nécessaires permettant de faire le lien entre l'ontologie et le calcul de similarité. Les paramètres à définir sont notamment l'emplacement de l'ontologie contenant les cas ou encore les attributs permettant de faire le lien entre l'ontologie et les classes Java du RàPC.

Le paramètre *Map* est particulièrement important car il assure l'interopérabilité entre l'ontologie et le module RàPC. Les paramètres *Property* et *Concept* permettent de spécifier les éléments de l'ontologie sur lesquels portera l'analyse et le paramètre *Attribute* définit l'attribut qui sera utilisé par la classe Java. Ce fichier de configuration associé aux composants réutilisables (*Java Beans*) forment le socle logiciel nécessaire aux calculs de similarité.

```
<OntologyConfiguration>
  <MainOntology>
    <URL>http://www.semanticweb.org/ontologies/2012/1/rdf_mda_v4_new.owl</URL>
    <LocalCopy>/home/arnaud/ontologies/mda_v4/rdf_mda_v4_new.owl</LocalCopy>
  </MainOntology>
  <CaseMainConcept>Scenario</CaseMainConcept>
  <DescriptionClassName>cbr.VesselCBRdescription</DescriptionClassName>
  <DescriptionMappings>
    <Map>
      <Property>OP_Alert_hasAlert</Property>
      <Concept>Alert_Area</Concept>
      <Attribute>AlertArea</Attribute>
    </Map>
  </DescriptionMappings>
  <SolutionClassName>cbr.VesselCBRresponse</SolutionClassName>
  <SolutionMappings></SolutionMappings>
</OntologyConfiguration>
```

5.4.3.2 Méthodes de calcul de similarité

Cette partie permet d'aligner un nouveau problème cible représenté par un navire ainsi que ses

alertes à un ou des cas sources représentés par les scénarios définis dans l'ontologie. Pour cela, deux étapes principales sont nécessaires :

1. tout d'abord, récupérer et aligner les cas sources correspondant.
2. Ensuite, calculer la similarité locale entre chaque propriété puis la similarité globale du cas en fonction des poids définis

Alignement de cas

L'étape d'alignement communique directement avec l'ontologie dans laquelle sont enregistrés à la fois les scénarios mais aussi les navires possédant potentiellement une ou plusieurs alertes. Cette communication se fait à l'aide de la bibliothèque Java Jena et du connecteur Ontobridge.

Les propriétés d'objets et de données sont alors analysées et enregistrées afin de construire la requête type qui permettra de mettre en correspondance les cas sources potentiels.

À partir des cas cibles et des cas sources, il est possible de calculer la distance sémantique les séparant. Cette phase fait intervenir deux concepts. Tout d'abord celui de distance qui consiste à mesurer la similarité entre les objets. Cette similarité se définit par l'intersection de l'ensemble d'éléments qui décrivent ces objets. En d'autres termes, plus ces objets partagent de points communs et plus ils sont similaires (Rodríguez and Egenhofer, 1999, 2003). Le second concept est directement rattaché au sens des objets manipulés. En effet, quand cette mesure de similarité est couplée à une ontologie, il est possible d'y ajouter une dimension sémantique. Dans ce cas, la similarité ne se fait plus uniquement sur un simple attribut ou un mot, mais aussi sur les propriétés unissant les concepts. L'utilisation d'une ontologie participe ainsi à l'amélioration de l'analyse effectuée par le RàPC en offrant un niveau d'abstraction supérieur capable de prendre en compte le sens des objets manipulés (Abou Assali, 2010).

Cette mesure de similarité sémantique s'appuie sur les propriétés intrinsèques d'une ontologie à savoir les nœuds et les arcs. Ce sont ces éléments qui sont alors utilisés pour mesurer la distance entre deux concepts. Pour cela, deux approches peuvent être distinguées.

La première, issue du champ de la théorie de l'information, est basée sur le contenu informationnel des nœuds. Il s'agit alors de mesurer la quantité d'information partagée par les deux objets comparés. Différentes méthodes peuvent être utilisées comme celle proposée par Resnik (Resnik, 1995) s'appuyant sur un corpus de connaissances ou encore celle de Seco (Seco et al., 2004) se basant sur le plus petit ancêtre commun.

La seconde, considérée souvent comme plus intuitive, s'appuie sur les arcs du graphe et consiste à mesurer la distance entre les concepts. Plus le chemin à parcourir est court et plus ces objets sont similaires. Là encore, différentes métriques existent, comme le calcul du nombre d'arcs nécessaires pour aller d'un concept à un autre (Rada et al., 1989) ou encore celui du plus petit généralisant commun prenant en compte à la fois le nombre d'arcs et la profondeur (Wu and Palmer, 1994).

Calcul de similarité locale et globale avec spécification des poids

Adapté à notre problématique, il s'agit alors de trouver en fonction des caractéristiques d'un navire de l'ontologie le ou les scénarios les plus significatifs. Au niveau informatique, cela consiste à comparer les propriétés d'objets et de données du navire à celles des scénarios. Imaginons que le navire de mon problème source soit un navire de pêche, qu'il possède une alerte de type *Fishing_area*, que la mer soit calme et qu'il ait une vitesse de 12 nœuds. Quel sera alors le scénario le plus probable par rapport à cette situation ? (Figure 5.17).

5.4. Module de calcul de similarité des comportements anormaux

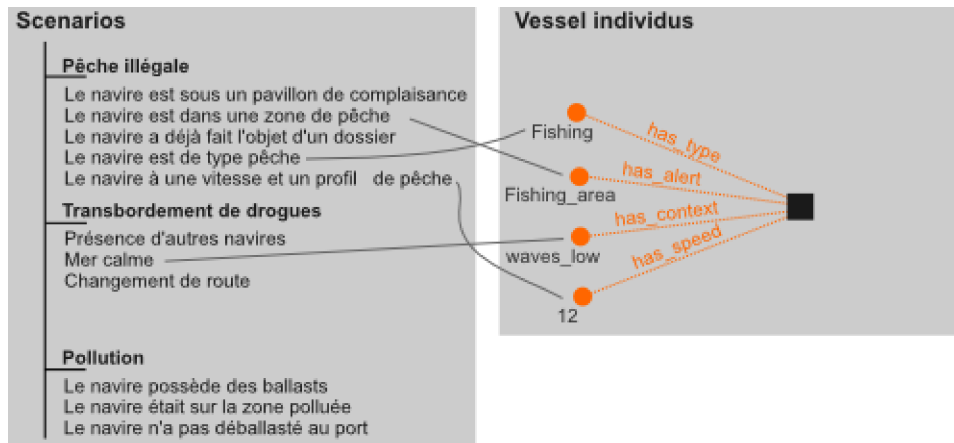


Figure 5.17: Illustration du calcul de similarité sémantique

Pour effectuer le calcul de similarité, différentes métriques (*OntCosine*, *OntDeep*, *OntDeepBasic*, *OntDetail*) sont disponibles dans jColibri. Chacune d'entre elles s'appuie sur des caractéristiques spécifiques qu'il est nécessaire de bien comprendre afin de calculer la distance la plus réaliste possible (Recio-García, Díaz-Agudo, González-Calero, et al., 2006). Par exemple, comme cela est illustré par les formules ci-dessous, *OntoDeepBasic* et *OntDeep* se basent toutes les deux sur le concept commun le plus proche. Mais le calcul de profondeur est différent. Dans le premier cas, c'est la profondeur totale des concepts de l'ontologie qui est pris en compte alors que dans le second cas ce sont les individus :

$$OntoDeepbasic(i1, i2) = \frac{\max(\text{prof}(LCS(i1, i2)))}{\max(\text{prof}(Ci)), Ci \in CN}$$

Où :
 LCS est le concept commun le plus proche
 Prof(Ci) est la profondeur du concept Ci
 Prof(i) est la profondeur de l'individu i

$$OntoDeep(i1, i2) = \frac{\max(\text{prof}(LCS(i1, i2)))}{\max(\text{prof}(i1), \text{prof}(i2))}$$

Dans le cas du prototype développé, c'est la méthode *OntoDeep* qui a été utilisée. Ce choix se justifie notamment par le fait qu'elle offre une meilleure finesse d'analyse dans le cas d'une ontologie contenant différents niveaux de profondeur.

Une fois le niveau de similarité de chaque élément déterminé, il est possible de calculer le niveau de similarité global. Néanmoins, il était nécessaire de pouvoir fixer pour chacun des éléments définis dans les scénarios des poids positifs ou négatifs. En effet, ces derniers n'interviennent pas de la même manière dans l'analyse et doivent donc être différenciés. Par exemple, dans le cas d'un scénario de pêche illicite, le type du navire est d'une grande importance. Celui-ci aura donc un poids très fort.

De ce fait, dans l'ontologie il est possible d'ajouter pour chaque individu de la classe scénario une annotation indiquant son poids. Celle-ci est ensuite automatiquement prise en compte par le module de calcul de similarité qui applique alors les paramètres nécessaires.

5.5. Module cartographique

Le module cartographique offre un accès visuel et simplifié à l'information. L'utilisateur n'a accès qu'aux données importantes et pertinentes pour son analyse. Les alertes et scénarios potentiels concernant les navires sont directement issus de l'ontologie maritime enrichie. Pour cela un connecteur spécifique a été développé assurant la connexion entre les différents éléments.

5.5.1. Architecture du module

L'architecture du module se compose de deux éléments principaux à savoir la partie des données et la partie interface cartographique (Figure 5.18).

Pour la partie des données, différentes sources d'informations peuvent être exploitées. Certaines sont classiques aux Systèmes d'Information Géographique comme des fichiers plats (Shapefile, GML, etc.) ou des web service (WMS, WFS) et d'autres sont spécifiques à l'application comme l'ontologie maritime enrichie.

La lecture des fichiers classiques s'effectue à l'aide de la bibliothèque Geotoolkit¹²². Celle-ci offre un grand nombre de connecteurs natifs vers la plupart des formats de données existants.

Concernant l'ontologie, un connecteur spécifique a été développé. Il permet d'accéder aux individus de l'ontologie ainsi qu'aux propriétés de données et d'objets qu'ils possèdent. Les attributs géographiques sont traduits puis affichés sur la carte. Les autres informations sont ensuite disponibles dans la fenêtre d'information.

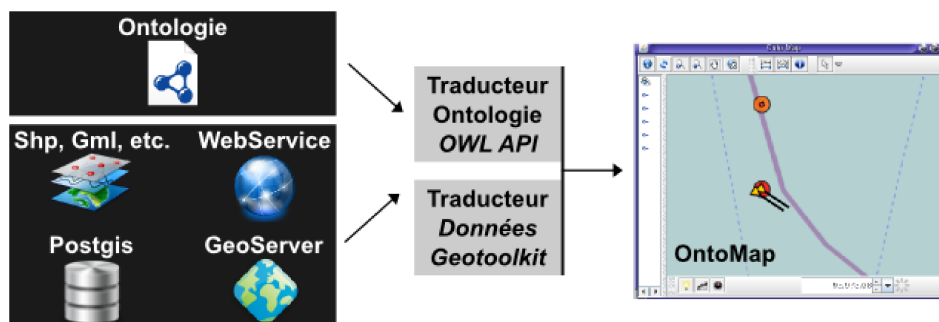


Figure 5.18: Etapes et flux du module cartographique

5.5.2. Interface et présentation des comportements

L'interface générale de l'application s'articule principalement autour de la carte. Ce choix volontaire s'explique par le fait que ce média offre un accès facilité et intuitif à l'information. Différents outils sont également implémentés afin d'aider à l'interprétation (Figure 5.19).

Chaque navire identifié comme potentiellement suspect est mis en évidence par une pastille colorée dont la couleur dépend du niveau de dangerosité. Ce niveau est directement issu de l'interprétation effectuée par le RàPC. Les différentes alertes ainsi que les informations sémantiquement enrichies

¹²² <http://www.geotoolkit.org/>

sont ensuite consultables depuis la fiche du navire. Cette fiche est construite en se connectant à l'ontologie.

Enfin, un curseur temporel permet de visualiser l'évolution du navire et l'ordonnancement des alertes enregistrées. L'utilisateur peut alors se déplacer temporellement et comprendre ainsi les causes qui ont amené aux scénarios potentiels.

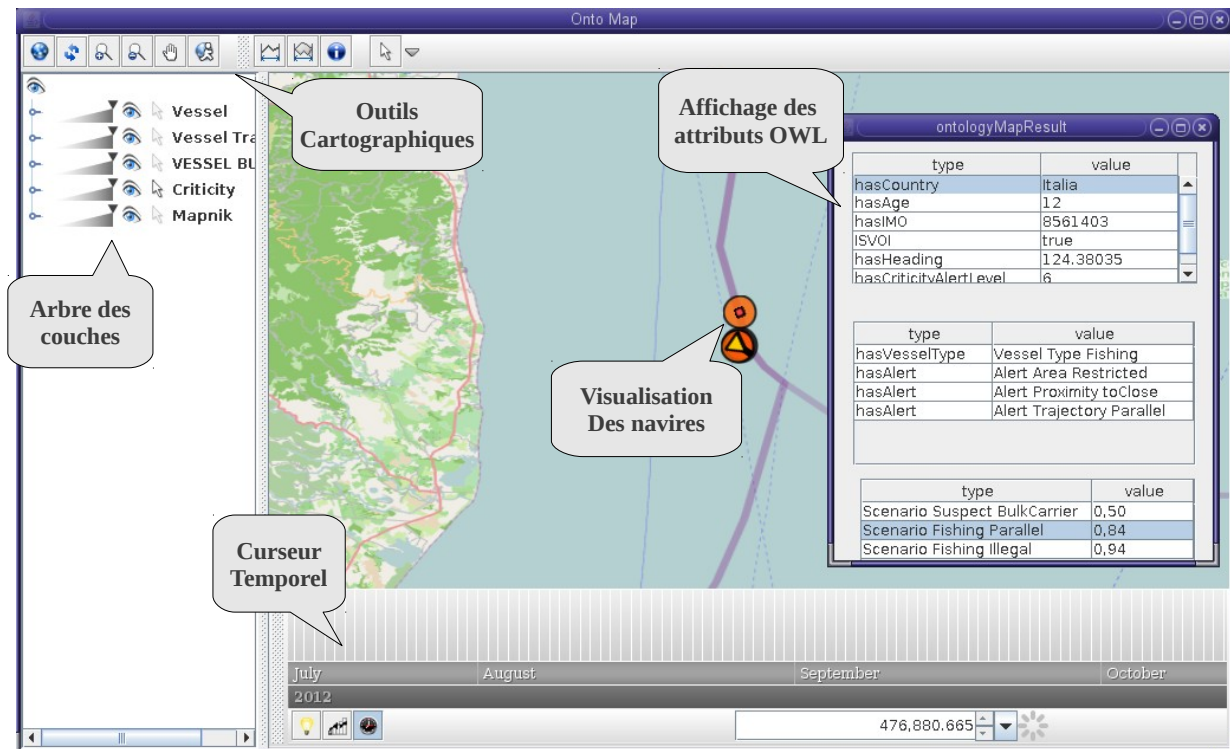


Figure 5.19: Interface d'OntoMap

Ce module, cartographique bien que fonctionnel, n'est pas forcément le plus adapté, ni le plus ergonomique par rapport aux besoins des experts du domaine maritime. C'est pourquoi, celui-ci sera à terme remplacé par le portail cartographique *FishEye* développé par le Centre de Recherche sur les risques et les Crises (CRC).

Conclusion du chapitre

Dans ce chapitre, les différents modules constituant la plate-forme *OntoMap* ont été présentés. Ces modules forment les éléments nécessaires à la compréhension et à l'analyse d'un comportement anormal.

Trois éléments sont particulièrement importants. Tout d'abord l'ontologie à partir de laquelle sont effectués les différents traitements. Puis le module d'inférence qui analyse les données en fonction des règles définies par les experts du domaine. Grâce à celui-ci, il est alors possible de déterminer les alertes relatives à un navire. Enfin, à partir de ces alertes, le module en charge de la classification des comportements a pour objectif de relier les navires aux scénarios dont la distance sémantique est la plus faible.

Les essais effectués sur plate-forme afin de valider l'approche proposée ont été réalisés avec un jeu de données réduit. Mais, afin d'identifier les limites potentielles liées à la performance du système, il apparaît nécessaire de vérifier son fonctionnement en faisant varier de manière croissante la volumétrie des données. Le résultat de ces tests ainsi que quelques cas d'utilisations sont présentés dans le chapitre suivant.

CHAPITRE 6. EXPÉRIMENTATIONS ET ÉVALUATIONS

Chapitre 6. Expérimentations et évaluations.....	171
6.1. Exemples de scénarios type.....	173
6.1.1. <i>Scénario d'un navire dangereux dans une zone réglementée.....</i>	<i>173</i>
6.1.2. <i>Scénario de transbordement de pêche.....</i>	<i>176</i>
6.1.3. <i>Autres scénarios.....</i>	<i>178</i>
6.2. Évaluation du système.....	179
6.2.1. <i>Évaluation des performances du système.....</i>	<i>179</i>
6.2.2. <i>Synthèse des résultats.....</i>	<i>181</i>
6.3. Perspectives d'amélioration.....	182
6.3.1. <i>Amélioration fonctionnelle</i>	<i>182</i>
6.3.2. <i>Amélioration du processus de modélisation des connaissances.....</i>	<i>183</i>

Introduction du chapitre

Les approches proposées dans ce travail de thèse ont été mises en application au sein de la plate-forme *OntoMap*. Celle-ci offre aux experts du domaine un ensemble d'éléments pour la modélisation et l'analyse de comportements anormaux de navires.

Afin de juger de la pertinence de cette plate-forme quelques cas sont présentés dans ce chapitre (§6.1). Contrairement à ceux définis dans l'ontologie, les scénarios utilisés ont été volontairement simplifiés afin de se concentrer sur les différentes étapes du processus plutôt que sur une exhaustivité des attributs. L'objectif est de vérifier le comportement du système dans des conditions proches d'une utilisation réelle. Ensuite, une analyse de la plate-forme en terme de performance sera réalisée (§ 6.2). Enfin, les perspectives d'amélioration potentielles sont détaillées (§ 6.3).

6.1. Exemples de scénarios type

Afin de vérifier la validité des approches proposées ainsi que le fonctionnement de la plate-forme *OntoMap*, une série de scénarios et de jeux de données a été intégrée. Seuls quelques cas d'expérimentations sont présentés dans les paragraphes ci-dessous.

6.1.1. Scénario d'un navire dangereux dans une zone réglementée

Certaines zones sont, pour des questions de sécurité, interdites à la navigation. L'entrée d'un navire dans l'une d'entre elles doit donc générer une alerte afin d'informer l'opérateur en charge de la surveillance.

Ce type d'analyse fait appel à une fonction spatiale de type intersection (*intersects*) et aux données du SCHOM¹²³ dans lesquelles sont décrites ces zones. Dans le cas présenté, il s'agit plus précisément d'identifier les navires ayant été potentiellement détournés afin de servir comme arme potentielle. Bien que cela ne soit encore jamais arrivé, ce type de scénario a déjà été envisagé dans les experts en charge du Maritime Domain Awareness (Medalia, 2004).

6.1.1.1 Modélisation du cas et du scénario

Ce cas nécessite de disposer d'informations concernant notamment le type du navire, les conditions de navigation ainsi que les zones dans lesquelles il est allé. Pour cela, deux requêtes spatiales sont nécessaires. La première qui va permettre de savoir si un navire s'est rendu dans le port d'un pays identifié comme étant à risque et la seconde qui signale si ce même navire entre dans une zone réglementée. Ces deux requêtes font appel à la même fonction spatiale *intersects* mais avec des jeux de données différents.

Comme cela est illustré par la règle SWRL ci-dessous, l'identification de l'entrée dans une zone réglementée nécessite de connaître le navire (*vessel*) ainsi que sa position (*position*), la couche de données sur laquelle portera l'analyse (*Alert_Area_Restricted*) et enfin la fonction spatiale à réaliser (*intersects*). Si cette analyse est positive alors une alerte est automatiquement ajoutée au navire correspondant (*Alert_Area_Restricted*) :

```
Vessel(?vessel),
DP_Analysis_DataPath(Alert_Area_Restricted, ?geoData),
DP_Vessel_hasPosition(?vessel, ?position),
intersects(?position, ?geoData) ->
OP_Alert_hasAlert(?vessel, Alert_Area_Restricted)
```

123 Service Hydrographique et Océanographique de la Marine (<http://www.shom.fr/>)

Chapitre 6. Expérimentations et évaluations

La seconde règle concernant le pays à risque est similaire à celle présentée précédemment. Seules les données ainsi que l'alerte générée (*Alert_Area_SuspiciousCountry*) sont différentes.

```
Vessel(?vessel),
DP_Analysis_DataPath(Alert_Area_SuspiciousCountry, ?geoData),
DP_Vessel_hasPosition(?vessel, ?position),
intersects(?position, ?geoData) ->
OP_Alert_hasAlert(?vessel, Alert_Area_SuspiciousCountry)
```

Maintenant que les différentes alertes ont été définies, il est possible de spécifier les scénarios correspondants. En fonction du type du navire, trois scénarios, constitués de sept propriétés d'objets (contexte dans lequel évolue le navire, son type, les zones, etc.), ont été créés. L'illustration ci-dessous (Figure 6.1), présente les paramètres nécessaires à l'identification du scénario par l'attaque d'un tanker. Les scénarios de type terroriste et chimique varient assez peu. En effet, seul le type du navire a été modifié.

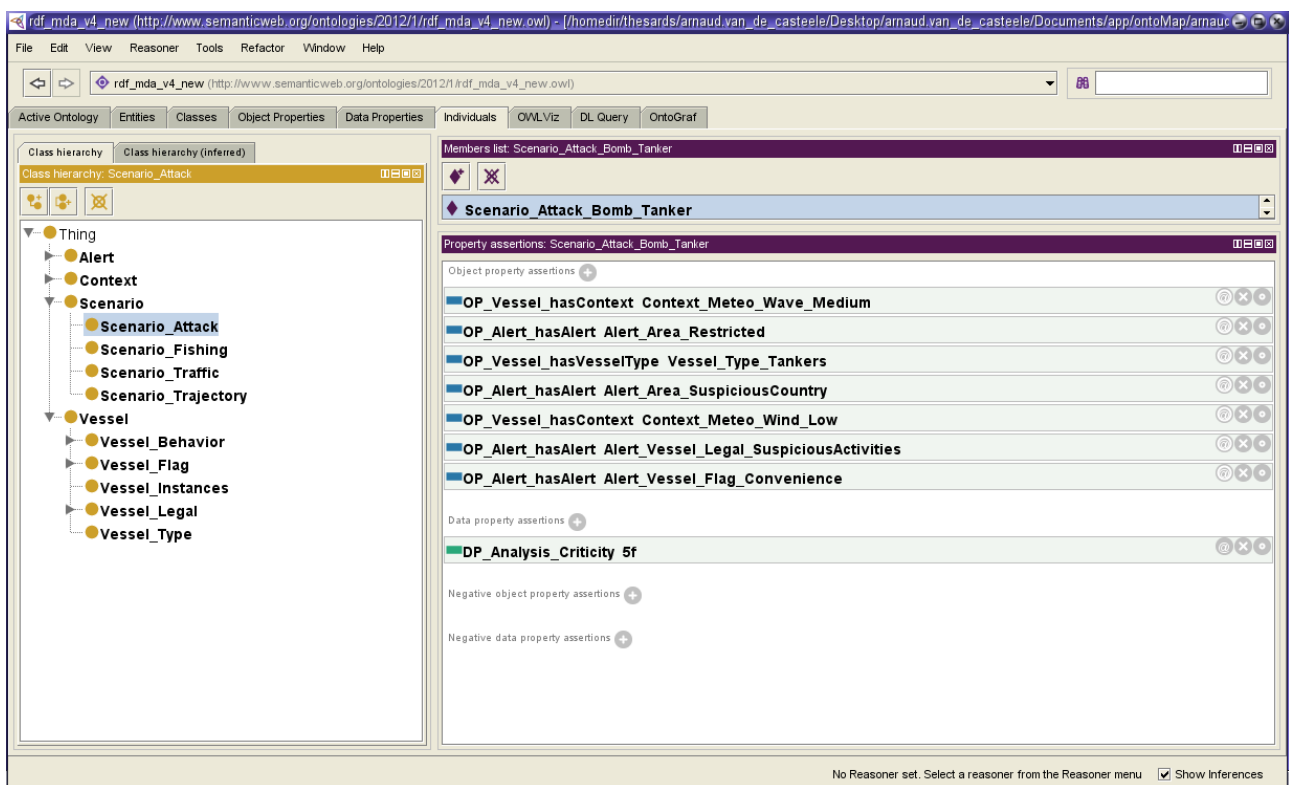


Figure 6.1: Description du scénario de type attaque de tanker

6.1.1.2 Analyse du comportement

Une fois le cas modélisé, il est alors possible d'alimenter l'ontologie et de vérifier la concordance des règles et des scénarios. Pour cela deux étapes sont nécessaires. Dans un premier temps, le moteur d'inférence spatiale va détecter les alertes potentielles (*Alert_Area_Restricted*,

6.1. Exemples de scénarios type

Alert_Area_SuspiciousCountry, etc.). Puis, dans un second temps, le module de Raisonnement à Partir de Cas (RàPC) va mettre en correspondance les scénarios préalablement définis et la situation observée.

Ce cas est illustré par la figure ci-dessous (Figure 6.2), où le navire 6361006 (pour des raisons légales, son numéro IMO a été volontairement modifié) possède certaines propriétés similaires à celles définies dans les scénarios précédents.



Figure 6.2: Liste des propriétés du navire pour le scénario d'attaque

Les informations automatiquement calculées par le moteur d'inférence peuvent être maintenant analysées par le module de RàPC. Dans le cas présenté, c'est le scénario d'une attaque de tanker transformé en bombe flottante qui est le plus probable (Figure 6.3). En effet, la similarité entre le navire et le scénario est de 96%. Les deux autres scénarios (*Scenario_Attack_terrorist*, *Scenario_Attack_Bomb_Chemical*) correspondant à la classe *Scenario_Attack* apparaissent également du fait de la distance sémantique faible entre les différents concepts. Enfin, l'indice de criticité du scénario ayant la plus forte probabilité d'occurrence est automatiquement attribué au navire.

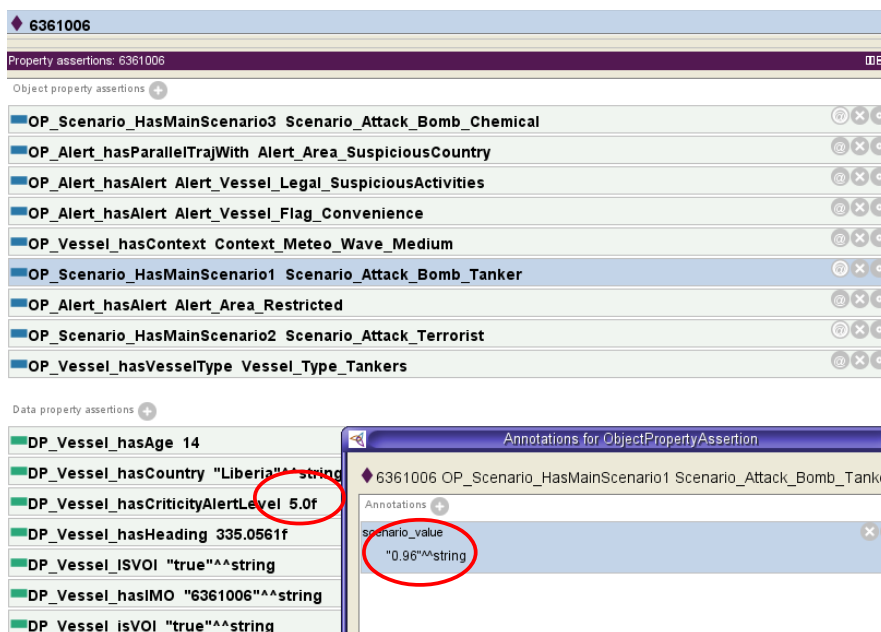


Figure 6.3: Scénario d'attaque potentielle et indice de criticité

6.1.2. Scénario de transbordement de pêche

Afin de lutter contre la pêche illicite et l'exploitation abusive des ressources halieutiques, l'Europe a défini des directives spécifiques¹²⁴ concernant le transbordement du produit de la pêche. Dans le cas d'un navire ayant atteint ses quotas, celui-ci peut être tenté d'écouler une partie de sa cargaison directement auprès d'un navire cargo réfrigéré.

Pour analyser ce type de comportement, il est nécessaire d'utiliser une fonction spatiale de type *touches*. Mais, comme cette analyse porte sur deux objets ponctuels, un opérateur de construction (*buffer*) est ajouté. Celui-ci transforme la position du navire en surface qui sera ensuite utilisée lors du calcul de proximité géographique.

6.1.2.1 Modélisation du cas et du scénario

L'identification de ce cas nécessite de disposer d'informations concernant le type du navire et les conditions météorologiques mais surtout de connaître les navires à proximité. Pour cela une requête spatiale de type *touches* a été implémentée. Celle-ci prend comme paramètre la géométrie des deux entités ainsi qu'un rayon de recherche qui étend la distance de calcul. La règle SWRL correspondante est alors la suivante :

```
Vessel(?vesselA),  
Vessel(?vesselB),  
DP_Vessel_hasPosition(?vesselA, ?positionA),  
DP_Vessel_hasPosition(?vesselB, ?positionB),  
touches(?positionA, ?positionA, 10) ->  
OP_Alert_IsToCloseFrom(?vesselA, ?vesselB)
```

Cette règle calcule pour chaque paire de navires (*vesselA*, *vesselB*), la satisfiabilité de la règle *touches* en fonction du rayon de recherche spécifié (ici 10). Si cela est vérifié, alors une alerte transitive de proximité (*OP_Alert_IsToCloseFrom*) est définie entre les deux navires.

Maintenant que les différentes alertes ont été spécifiées, le scénario peut alors être modélisé. Celui-ci est constitué de six propriétés d'objets et d'une propriété de données (Figure 6.4). Les propriétés d'objets décrivent les différents éléments nécessaires à la validation du scénario. Par exemple, pour le cas ci-dessous, il est précisé que la mer doit être calme (*Context_Meteo_Wave_Low*), que le navire possède un pavillon de complaisance (*Alert_Vessel_Flag_Convenience*), ou encore qu'il est situé dans une zone de pêche (*Alert_Area_Fishing*).

124 Voir notamment les directives publiées dans le Journal Officiel n°302 (30/12/2009)

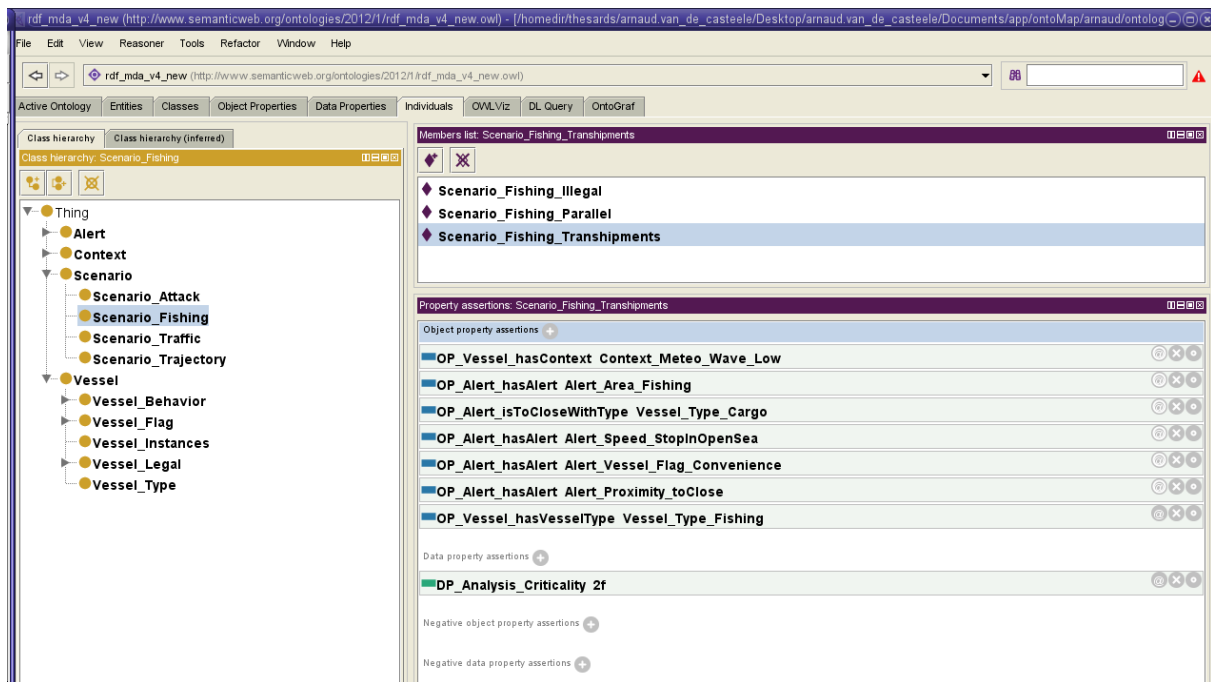


Figure 6.4: Description du scénario de type transbordement de pêche

6.1.2.2 Analyse du comportement

Tout comme le cas précédent, il est maintenant possible de vérifier la concordance des règles et des scénarios. Une fois les données analysées par le moteur d'inférence, il apparaît que le navire dont le numéro est le 7922934 possède des attributs pouvant correspondre à un transbordement de pêche (Figure 6.5).

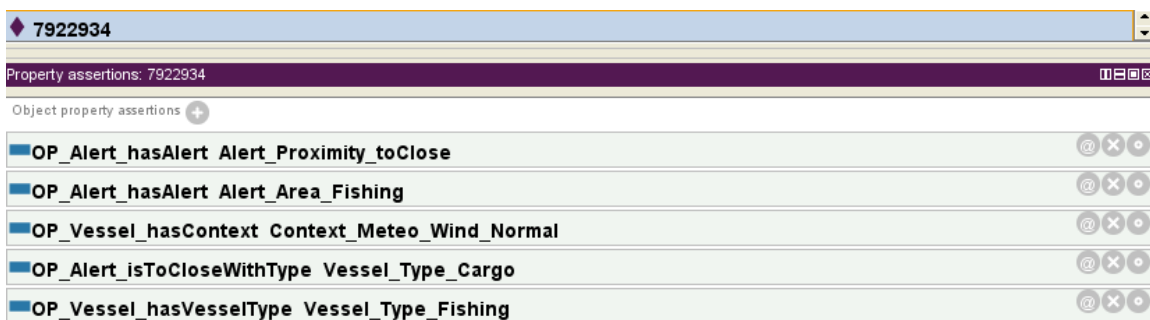


Figure 6.5: Liste des propriétés du navire pour le scénario de transbordement de pêche

Le navire possède cinq propriétés d'objets, dont deux alertes. Ces dernières sont dues à sa proximité avec un autre navire et à sa présence dans une zone de pêche. La question est maintenant de savoir quelle est la probabilité pour que le scénario défini précédemment corresponde ?

Pour cela, le module de RàPC effectue le calcul de distance sémantique entre les concepts. Celui-ci met en évidence les scénarios *Scenario_Fishing_Transhipments* et *Scenario_Fishing_Illegal* comme étant les plus probables (Figure 6.6).

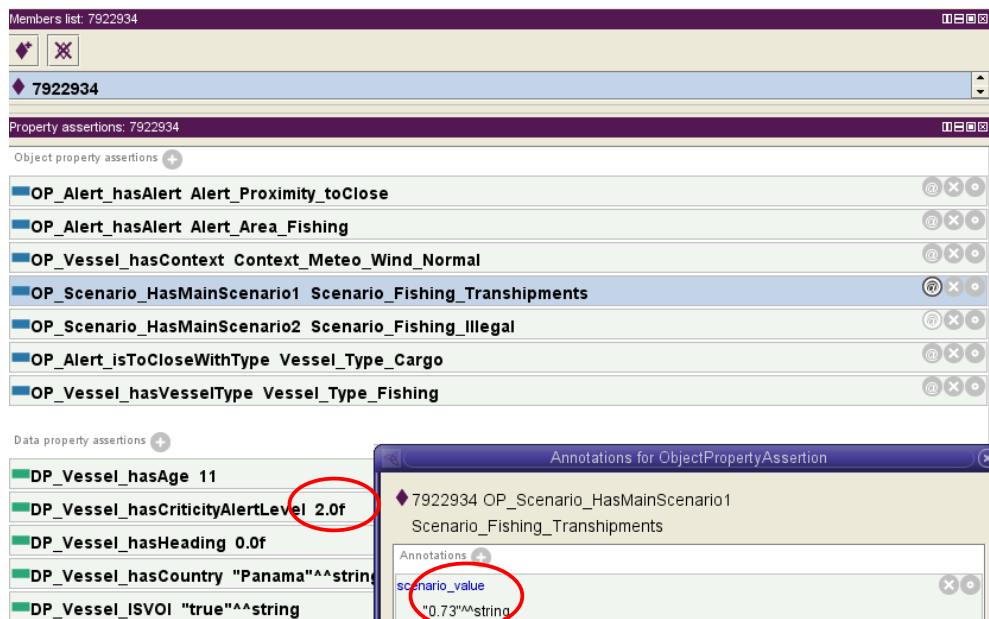


Figure 6.6: Scénario de transbordement potentiel et indice de criticité

6.1.3. Autres scénarios

Comme cela a déjà été souligné, les deux cas présentés ne reflètent pas la totalité des scénarios implémentés. Néanmoins, l'objectif n'était pas de faire un catalogue exhaustif de ces derniers mais plutôt de montrer les étapes et les potentialités de l'approche proposée.

Grâce à la plate-forme *OntoMap*, les experts du domaine disposent des composants nécessaires à la modélisation et à la compréhension d'une situation. La complémentarité entre d'une part l'ontologie et d'autre part le Raisonnement À Partir de Cas permet de disposer de capacités descriptives fortes pouvant être ensuite exploitées à des fins d'analyse.

6.2. Évaluation du système

Dans les paragraphes précédents, les avantages de la plate-forme *OntoMap* ont été mis en évidence. Néanmoins, les différents cas d'utilisation ont été réalisés sur des jeux de données restreints et dans des conditions d'utilisation bornées. Bien que cela ne soit qu'un prototype et que les moyens informatiques évoluent constamment, une analyse des performances de *OntoMap* apparaît nécessaire. Celle-ci permettra d'identifier les limites actuelles ainsi que les axes d'amélioration potentiels. De cette analyse, dépendra également l'échelle de surveillance à laquelle il est actuellement possible de répondre. En effet, les besoins et le volume de données ne sont pas les mêmes si cette surveillance s'effectue au niveau du port, d'un rail de navigation ou d'une préfecture maritime.

6.2.1. Évaluation des performances du système

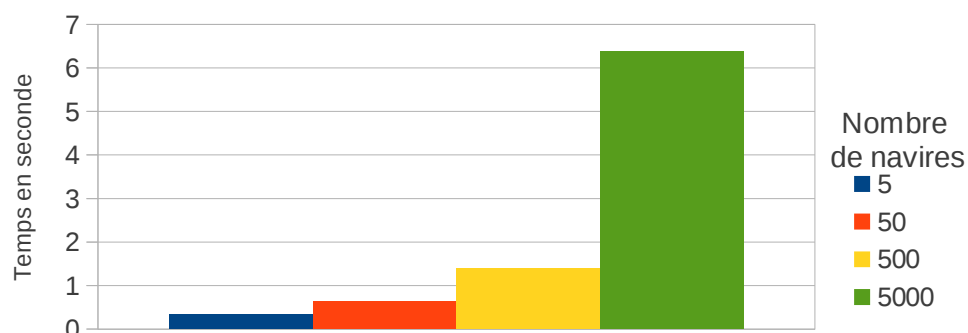
L'évaluation réalisée a consisté à analyser les temps de réponse de la plate-forme *OntoMap* en faisant varier de manière croissante le nombre de navires ainsi que le nombre de règles. Ces tests de performance ont été réalisés sur un serveur linux disposant de 8 CPU (Intel(R) 2.50GHz) et de 8Go de mémoire. Pour les navires, les données utilisées sont celles issues du module de filtrage et d'optimisation (Chapitre 5.2.2 p.149). Concernant la règle, celle utilisée est un cas d'intersection. Afin de garder une même métrique, le passage de 5 à 100 règles a été réalisé en la dupliquant autant de fois que nécessaire. Les temps de réponse ont été obtenus en plaçant un marqueur temporel avant et après le processus d'inférence et de la même manière avant et après le processus de Raisonnement à Partir de Cas.

Les résultats obtenus pour chacune des phases seront discutés dans les paragraphes suivants.

6.2.1.1 Évaluation de la phase de mise à jour de l'ontologie

La phase de mise à jour de l'ontologie consiste, comme son nom l'indique, à supprimer, modifier ou ajouter des données issues à la fois de la base mais aussi des différentes sources d'informations annexes (ex. *MarineTraffic*). Pour cette phase, seuls le temps et le nombre de navires ont été conservés. En effet, aucune règle n'est utilisée.

Logiquement, le temps d'exécution de ce processus croît avec l'augmentation du nombre de navires (Graphique 6.1). Néanmoins, si à chaque itération le nombre de navires est multiplié par 10, le temps nécessaire lui n'augmente que d'un facteur de 2 ou 5. Pour 5000 navires, le temps constaté est d'à peine plus de 6 secondes.

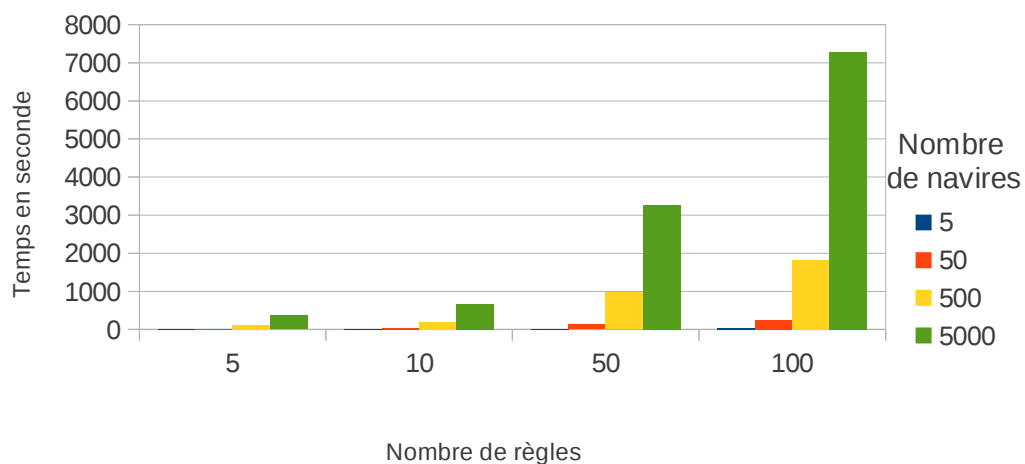


Graphique 6.1: Répartition du temps de calcul pour la phase de mise à jour

6.2.1.2 Évaluation de la phase d'inférence

La phase d'inférence fait directement intervenir les règles spatiales. Elle consiste à valider les règles implémentées en fonction de la position de chaque navire. Si les temps de calcul sont assez faibles quand les navires sont peu nombreux, ceux-ci augmentent ensuite fortement (Graphique 6.2). La dernière phase du processus avec 5000 navires a nécessité un peu moins de deux heures. Cela est bien évidemment incompatible avec une application en temps réel. Mais, dans ce cas la plate-forme *OntoMap* peut être utilisée pour analyser une situation post-crise.

L'hypothèse formulée est que le temps d'inférence est directement lié au nombre de navires et non au nombre de règles. En effet, en l'absence d'un index spatial, la règle va devoir vérifier tous les navires de la base de connaissance. Plus ce nombre augmente et plus les vérifications nécessaires aussi. L'une des solutions pour diminuer les temps de calcul serait alors d'ajouter un index GIST comme cela se fait dans les bases de données spatiales.

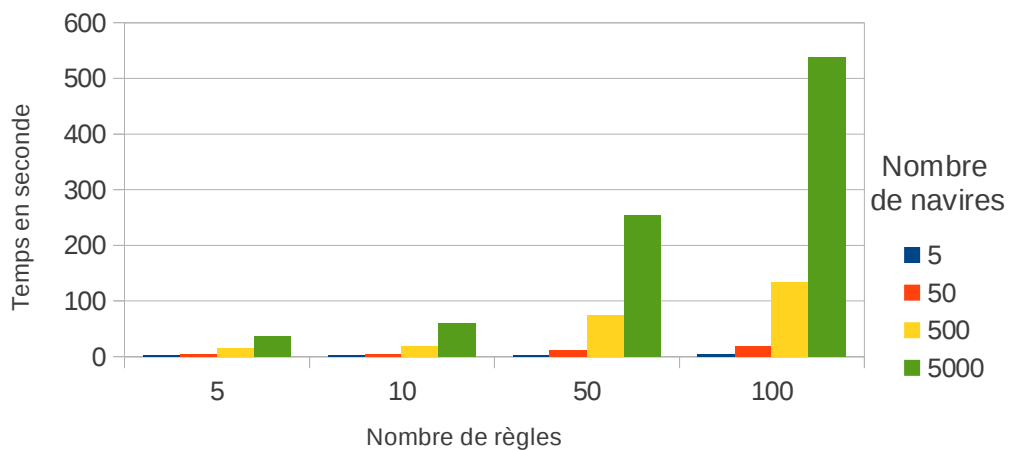


Graphique 6.2: Répartition du temps de calcul pour la phase d'inférence

6.2.1.3 Évaluation de la phase de Raisonnement à Partir de Cas

La dernière phase correspond au calcul de similarité sémantique entre un navire et les scénarios définies dans l'ontologie. Tout comme la phase de mise à jour, l'hypothèse de départ était que le nombre de règles importait peu. Mais celle-ci s'est avérée fautive, car comme cela est démontré par le graphique ci-dessous (Graphique 6.3), ce nombre fait varier significativement les temps de calcul. Pour 5 navires et 5 règles le temps constaté a été de 1.7 seconde alors que pour 100 règles celui-ci passe à 4.5 secondes. Ce constat s'explique notamment par le fait que *OntoBridge*, le connecteur ontologique de *jColibri*, utilise en interne les potentialités d'inférence du raisonneur implémenté (Pellet). Hors ce dernier, pré-calculait auparavant certaines inférences.

De plus, il aurait été intéressant d'effectuer un calcul de performance similaire mais en faisant cette fois varier le nombre total de scénarios. Malheureusement, le nombre de cas n'était pas suffisamment significatif pour permettre cette mesure. Néanmoins, une prochaine rencontre avec les experts maritimes devrait permettre d'enrichir la base de connaissance et à terme le résultat de cette métrique pourra également être calculée.



Graphique 6.3: Répartition du temps de calcul pour la phase de RàPC

6.2.2. Synthèse des résultats

Les expérimentations réalisées ont consisté à faire varier le nombre de navires et le nombre de règles afin d'observer les temps de calcul. Si les résultats sont satisfaisants pour un nombre réduit de navires et/ou de règles, les performances se dégradent ensuite avec leur augmentation. De ces résultats, plusieurs remarques peuvent être énoncées.

Premièrement, ces chiffres sont incompatibles avec une surveillance en temps réel du trafic maritime. La solution *OntoMap* ne peut donc pour le moment prétendre se substituer aux actuels *Vessel traffic System* (VTS). Néanmoins, *OntoMap* peut être utilisée conjointement afin d'analyser une situation et comprendre le déroulement des événements. À terme, l'amélioration de la qualité du code ainsi que l'évolution de l'infrastructure technique pourrait permettre une amélioration des performances observées.

Deuxièmement, les principaux efforts devront être portés sur la phase d'inférence. Comme cela a été mise en évidence lors de l'étude des performances, c'est elle qui prend aujourd'hui le plus de temps. Il est nécessaire de comprendre les causes et de trouver les moyens nécessaires pour y remédier.

Enfin, une meilleure intégration de la dimension spatiale directement dans les ontologies apparaît nécessaire. En effet, le parti pris dans ce travail a été de simuler cette dimension en dehors de l'ontologie. Si cela offre une plus grande souplesse, cela entraîne également des performances moindres. L'implémentation de types, de fonctions et d'index spatiaux spécialisés pour les ontologies apparaît alors comme obligatoire pour la modélisation des connaissances.

6.3. Perspectives d'amélioration

Même si cette plate-forme est fonctionnelle, des améliorations peuvent être apportées. Celles-ci portent autant sur la modélisation des connaissances que sur les performances. Quelques perspectives sont présentées dans les paragraphes ci-dessous.

6.3.1. Amélioration fonctionnelle

Comme cela a été mise en évidence dans la section précédente, l'augmentation du nombre de navires entraîne une diminution des performances du système. De ce fait, l'une des solutions pourrait être d'enregistrer les individus dans une base de données (relationnelle, ou rdf). De plus, cela permettrait de séparer complètement la partie ontologie servant à la modélisation des connaissances de la partie base de connaissances.

Différentes solutions existent actuellement et devront être étudiées. Néanmoins, deux semblent correspondre aux besoins évoqués.

La première consiste à conserver les potentialités des bases de données relationnelles et plus particulièrement des bases de données spatiales. Le passage du relationnel au RDF se faisant alors par l'ajout d'un composant tiers. Dans ce cas, les données sont stockées en tant que *tuples* mais sont, par l'intermédiaire de ce composant, représentées sous forme de triplets. C'est par exemple ce que propose la plate-forme D2RQ¹²⁵ développée par Cyganiak et Bizer (Bizer and Cyganiak, 2007) dont le fonctionnement est illustré dans la figure ci-dessous (Figure 6.7). D2RQ supporte de nombreuses bases de données relationnelles comme Oracle, MySQL ou encore PostgreSQL. Néanmoins, il n'est pas fait mention de PostGIS. Si cela est confirmé, un connecteur spécifique permettant la prise en charge de la dimension spatiale devra être alors développé.

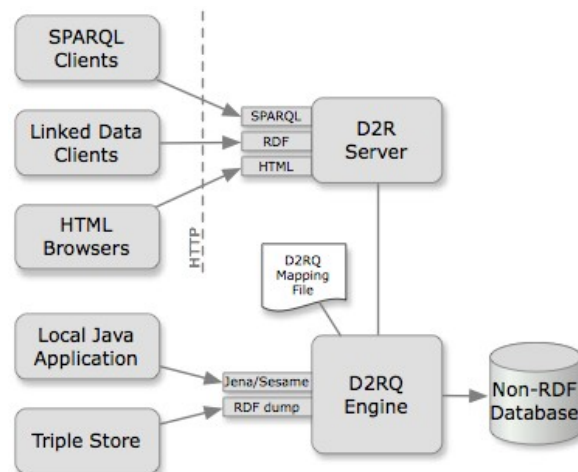


Figure 6.7: Fonctionnement général de D2RQ.
Source site internet de D2RQ (<http://d2rq.org/>)

La seconde solution consiste à stocker directement les individus sous la forme de triplets RDF. La liste actuelle¹²⁶ compte plus d'une dizaine de solutions potentielles. Au niveau commercial, l'offre la

125 <http://d2rq.org/>

126 <http://www.w3.org/wiki/LargeTripleStores>

plus intéressante est celle proposée par Oracle¹²⁷. C'est d'ailleurs l'approche utilisée par Mefteh pour modéliser les trajectoires de mammifères marins (Mefteh et al., 2012). D'autres alternatives Open Source existent également. L'une d'entre elles, AllegroGraph¹²⁸ permet le stockage de plusieurs centaines de milliards de triplets. Bien que dans notre cas, une telle volumétrie ne soit pas nécessaire cela offre néanmoins un aperçu des potentialités de ce type de solution.

6.3.2. Amélioration du processus de modélisation des connaissances

Si les langages ontologiques offrent les éléments nécessaires à la modélisation des connaissances, leur création et leur formalisation restent difficiles pour des personnes qui ne sont pas du domaine. C'est pourquoi, la création d'interfaces spécifiques apparaît comme une solution potentielle.

Création des règles SWRL

À l'heure actuelle, la définition des règles permettant de spécifier les alertes ainsi que la création des scénarios se font en utilisant le langage OWL et SWRL. Bien que le logiciel Protege offre une interface conviviale, celui-ci s'adresse à des utilisateurs connaissant déjà les concepts inhérents aux ontologies. Il est alors difficile de demander à un expert maritime (CROSS, gendarmerie maritime, etc.) de disposer de ce type de compétences. C'est pourquoi la création d'interfaces spécialement adaptées au domaine étudié pourrait être une solution. Pour illustrer cette hypothèse, deux applications ont été choisies.

La première, nommée Snoogle¹²⁹, a été développée par Ressler et son équipe (Ressler, Dean, Benson, et al., 2007). L'objectif de cette application est de proposer une interface visuelle et intuitive pour la création de lien (*mapping*) entre les ontologies (Figure 6.8). Ces liens permettent d'identifier qu'un concept est présent dans les deux ontologies et cela même si les termes sont différents.

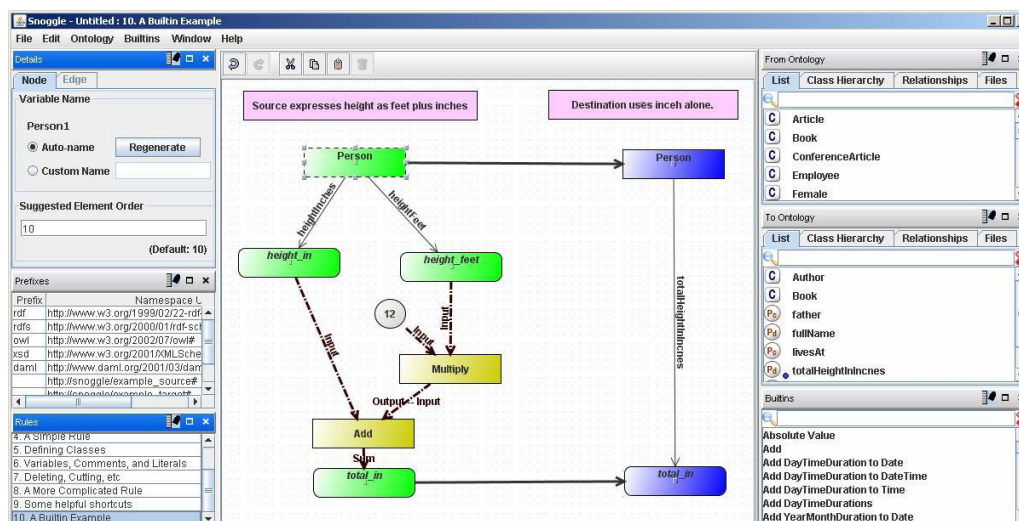


Figure 6.8: Interface du logiciel Snoogle. Source (Ressler, Dean, Benson, et al., 2007)

De la même manière, une interface de ce type pour la construction des alertes serait intéressante. Il

127 <http://www.oracle.com/technetwork/database/options/semantic-tech/index.html>

128 <http://www.franz.com/agraph/allegrograph/>

129 <http://snoggle.projects.semwebcentral.org/>

Chapitre 6. Expérimentations et évaluations

serait alors possible de créer une nouvelle règle SWRL, puis à l'aide de liens représentant les conditions de spécifier les paramètres nécessaires. Une pré-visualisation des règles spatiales pourrait même être apportée afin que l'utilisateur puisse visualiser le résultat potentiel de l'opération. Cette approche s'appuyant sur une composante visuelle de l'information semble plus intuitive que les solutions offertes actuellement et de ce fait plus adaptée pour des personnes qui ne sont pas du domaine.

Création des scénarios

Concernant les scénarios, ils sont actuellement créés directement au sein du logiciel Protégé. Comme cela a déjà été souligné, bien que parfaitement fonctionnelle, cette approche nécessite de connaître le logiciel et surtout les concepts ontologiques. C'est pourquoi la plate-forme Cobra développée par Abou Assali dans le cadre de sa thèse apparaît comme une solution potentielle (Abou Assali, 2010).

Cette plate-forme qui s'appuie sur le framework Rich Client Platform¹³⁰, offre les éléments nécessaires à l'exploration de l'ontologie et à la construction des cas (Figure 6.9). Celle-ci dispose notamment d'un explorateur d'ontologie, d'un éditeur de la cas ou encore d'un éditeur de requête. Au sein de cette interface, il est alors possible de créer de nouveaux cas qui seront ensuite sauvegardés dans l'ontologie. Pour la partie raisonnement, celle-ci est directement reliée à la bibliothèque jColibri.

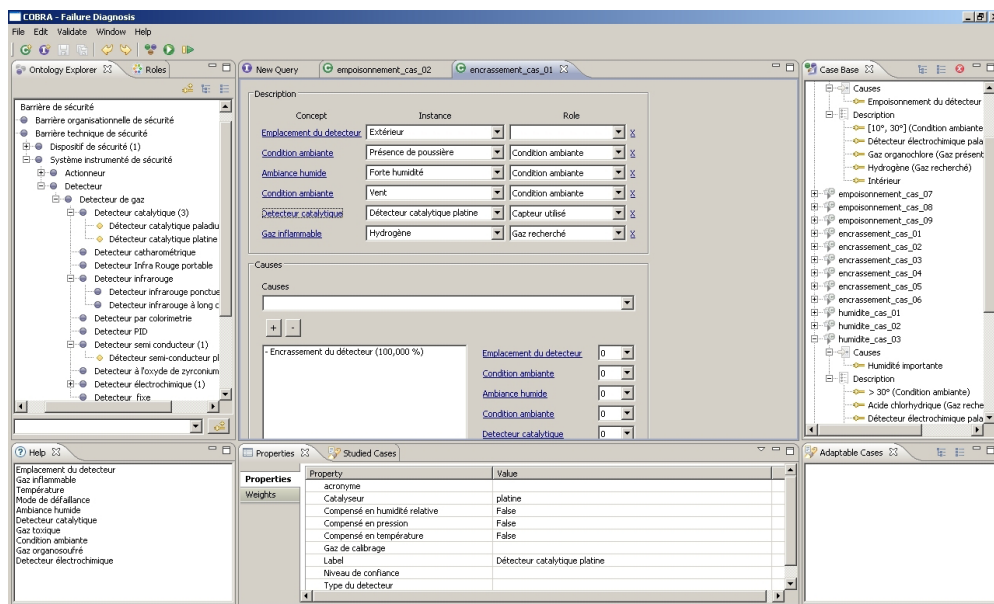


Figure 6.9: interface de la plate-forme Cobra. Source (Abou Assali, 2010)

De la même manière, il serait possible de s'appuyer sur le travail réalisé et de l'adapter pour la plate-forme *OntoMap*. Les avantages de cette interface pour la création des scénarios sont évidents. En effet, grâce à elle l'utilisateur dispose de tous les éléments nécessaires pour implémenter son raisonnement au sein de l'ontologie. Comme cela a été proposé par Abou Assali (Abou Assali, 2010) mais pas encore réalisé, la création de différents profils (expert, utilisateur occasionnel) permettrait de répondre encore mieux aux différentes attentes des utilisateurs.

130 <http://www.eclipse.org/home/categories/rcp.php>

Représentation spatiale des concepts

Enfin, l'un des aspects qui pourraient être améliorés porte sur la représentation des concepts spatio-temporels. En effet, à l'heure actuelle, cette représentation se fait à l'aide de mots clés exprimés sous la forme d'un langage naturel contrôlé. Mais, cette approche ne permet pas une visualisation rapide et intuitive des types et des relations spatio-temporelles potentielles entre les objets. De ce fait, l'ajout d'un outil de modélisation visuelle apparaît comme particulièrement pertinent.

Plusieurs formalismes permettant la modélisation d'objets géographiques existent (MADS, CONGOO, OMEGA, Perceptory, etc.). Actuellement les deux principaux outils pour réaliser ce type de modélisation sont *Perceptory* et *MADS*. Une description succincte des potentialités de chacun d'eux est réalisée dans les paragraphes ci-dessous. Néanmoins, une revue plus complète est disponible dans la thèse de Salehi (Salehi, 2009).

Développé comme une extension UML, *Perceptory* (Figure 6.10) et un outil de modélisation visuelle permettant de construire des modèles conceptuels d'objets à composante spatio-temporelle (Larrivée and Bédard, 2002; Bédard et al., 2004). Les entités peuvent être décrites sous la forme de géométries simples ou être combinées afin de former des géométries alternatives, complexes ou multiples. Concernant la partie temporelle, *Perceptory* permet de représenter des temporalités uniques (instant) ou plus longues (intervalle).

Tout comme la géométrie, ces notions temporelles peuvent être associées afin de former des temporalités alternatives. Plus récemment, l'équipe de *Perceptory* a développé un nouvel outil nommé *PictograF* qui se veut plus universel et polyvalent.

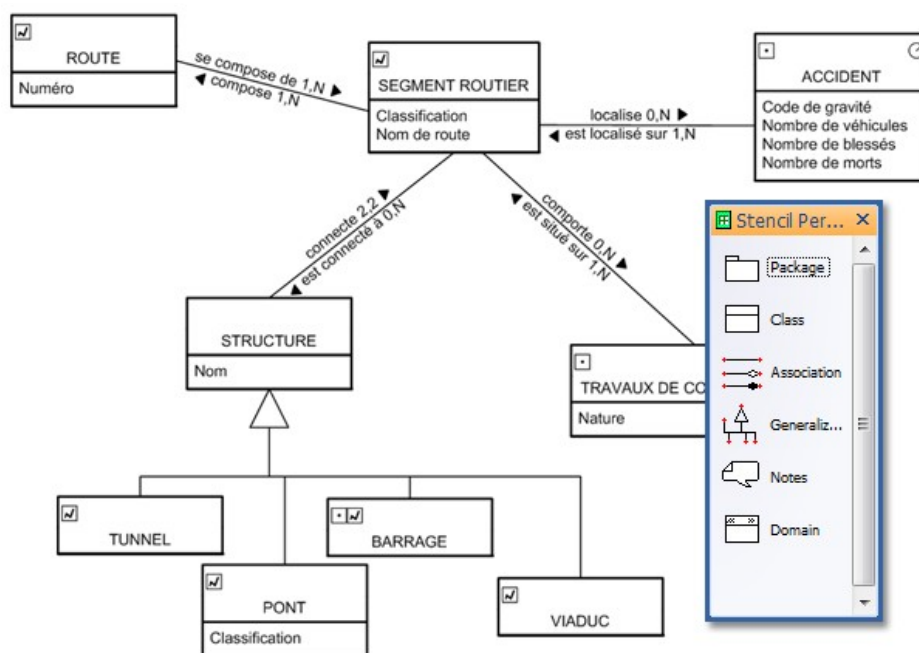


Figure 6.10: Exemple de représentation de concepts spatiaux à l'aide du logiciel Perceptory (Source : site internet Perceptory)

Basé sur un modèle de type objet-relation, *MADS*¹³¹ permet également de décrire des entités spatio-temporelles (Parent et al., 2006). Les géométries disponibles sont notamment les géométries

¹³¹ Model for Application Data with Spatio-temporal features

Chapitre 6. Expérimentations et évaluations

simples, complexes ou composées mais aussi des types plus particuliers comme une collection de points (semis), de lignes (graphe) ou encore de lignes orientés. Différentes contraintes topologiques (*disjoint*, *overlap*, *cross*) entre les objets spatiaux peuvent également être exprimées.

Ce type d'approche pourrait être intégré aux outils de modélisation ontologique existants et permettrait ainsi une visualisation facilitée des concepts, contraintes et relations géographiques.

Conclusion du chapitre

Les différents cas présentés dans ce chapitre ont permis de vérifier l'adéquation de la plate-forme OntoMap aux besoins des experts du domaine. Bien que les résultats montrent qu'une utilisation en temps réel est pour le moment incompatible avec les exigences de la surveillance maritime, celle-ci peut être néanmoins d'une grande utilité pour rejouer, analyser et comprendre un événement passé.

Plusieurs pistes peuvent être explorées afin d'améliorer l'utilisabilité et les performances de la plate-forme. Celles-ci portent notamment sur la création d'interfaces spécifiques et adaptées aux besoins des experts. En effet, la création et la manipulation des concepts ne sont pas intuitives pour une personne extérieure au domaine de l'ingénierie des connaissances. De plus, il apparaît essentiel de séparer la partie modélisation de la partie bases de connaissances. Pour cela, le stockage des faits et des individus dans une base de données RDF semble être une solution pertinente. Différentes expérimentations devront être néanmoins effectuées afin de valider les faisabilités techniques de cette approche.

CONCLUSION

Les travaux présentés dans ce manuscrit se situent au croisement de la gestion des connaissances et du raisonnement spatial. Il s'agit en effet d'étudier les défis et les avantages qui découlent d'une exploitation maîtrisée des informations spatiales à des fins décisionnelles. Le terrain d'expérimentation qui a servi de support à ce travail de recherche est le domaine maritime. L'objectif est de proposer une modélisation des connaissances expertes pour l'analyse des comportements anormaux de navires.

Pour cela, une approche basée sur les ontologies a été étudiée. Les ontologies offrent un cadre unificateur permettant une conceptualisation partagée et compréhensible de la connaissance grâce à la richesse sémantique qu'elles autorisent. Plus concrètement, une ontologie des comportements anormaux de navires a été créée puis intégrée à des fins décisionnelles au sein d'un système de surveillance maritime. Mais, la prise en compte de la dimension spatiale par les formalismes de représentation actuels est aujourd'hui insuffisante. Les capacités d'analyse et par conséquent la prise de décision sont, de ce fait, incomplètes. Ce travail de recherche vise à réduire cette limite en proposant une formalisation de la connaissance spatiale sous la forme de mots clés spécifiques (*intersects*, *touches*, etc.) pouvant être ensuite utilisés comme paramètre afin d'exprimer des règles de comportements.

Deux axes structurent l'organisation de cette étude. Premièrement, la modélisation de la connaissance experte et deuxièmement son implémentation au sein d'un système de surveillance maritime. Ce chapitre présente les principaux apports et le bilan de l'approche proposée pour ensuite conclure sur les perspectives potentielles.

BILAN DE LA THÈSE

L'hypothèse de départ qui a soutenu ce travail de recherche était qu'une **formalisation enrichie à l'aide de la composante spatiale favorise la description et l'analyse du comportement des navires**. Afin de vérifier celle-ci, trois principaux objectifs ont dû être au préalable atteints. Le bilan de chacun d'eux est présenté dans les paragraphes ci-dessous.

INTÉGRATION DE LA DIMENSION SPATIALE AU SEIN D'UN LANGAGE DE MODÉLISATION

La modélisation des connaissances nécessite de disposer d'un réceptacle disposant de capacités descriptives suffisamment riches pour traduire la complexité du monde réel. Le domaine qui regroupe aujourd'hui les fonctionnalités les plus avancées est celui des ontologies. En effet, ce dernier dispose notamment de langages à l'expressivité croissante ainsi que des procédures et des moteurs d'inférence.

En raison de leurs caractéristiques, les ontologies peuvent alors servir de support à la création d'une mémoire organisationnelle. Celles-ci permettent la structuration et la capitalisation des connaissances,

l'amélioration des interactions potentielles d'humain à humain, de système à système ou d'humain à système mais aussi l'implémentation de raisonnements spécifiques. De ce fait, les ontologies apparaissent comme une approche privilégiée pour la représentation du raisonnement spatial.

Mais, si les ontologies et la dimension géographique sont complémentaires, il n'existe à l'heure actuelle aucun formalisme de représentation standardisé ou de procédures de raisonnement permettant d'unir ces deux domaines. Tout l'enjeu a alors été de proposer une nouvelle démarche capable d'utiliser à la fois les potentialités descriptives des ontologies et les spécificités du raisonnement spatial.

Cette démarche a été réalisée en permettant aux utilisateurs de décrire les concepts spatiaux à l'aide de mots-clés exprimés en langue naturelle contrôlée. Ces mots-clés peuvent être alors utilisés pour décrire des règles de comportements qui seront utilisées lors des phases d'analyses.

CONCEPTION D'UN MODÈLE GLOBAL POUR LA SURVEILLANCE MARITIME

Cette approche sur la représentation et l'inférence des connaissances spatiales a été confrontée à un domaine applicatif réel, celui de la surveillance maritime. Pour cela une ontologie des comportements anormaux de navires a été réalisée. Celle-ci contient les éléments nécessaires à l'analyse d'un comportement. Différentes ontologies déjà existantes (temps, événement, etc.) ont été intégrées et servent de support à l'ontologie du domaine.

A partir de cette ontologie et en s'appuyant à la fois sur la connaissance des experts maritimes et sur différentes sources d'informations, un ensemble de règles a été défini. Ces règles ont pour objectif d'identifier des alertes potentielles en fonction notamment de la cinématique du navire, de la proximité de zones géographiques spécifiques (rochers affleurants, pays en guerre, etc.) ou encore des conditions météorologiques. Ces règles s'appuient sur les potentialités initiales du langage Semantic Web Rule Language (SWRL) auquel a été ajoutée la dimension spatiale.

CONCEPTION D'UN SYSTÈME D'ANALYSE AUTOMATIQUE DES COMPORTEMENTS ANORMAUX

Une fois la connaissance experte modélisée, il est alors possible de l'intégrer au sein d'un système d'analyse et d'aide à la décision. Ce système s'appuie sur le modèle préalablement défini et sur les informations contenues dans la base de connaissance afin de classifier le comportement des navires. Chaque nouvelle position est alors comparée aux règles qui ont été définies par les experts afin de déterminer si le navire a un caractère anormal.

Pour cela, un moteur d'inférence spécifique a été intégré. Celui-ci est capable de raisonner à la fois à l'aide d'opérateurs SWRL déjà standardisés (opérateurs mathématiques, opérateurs pour des dates, etc.) mais aussi à l'aide d'opérateurs spatiaux. Les experts du domaine maritime ont ainsi une plus grande richesse descriptive pour la modélisation de comportements de navires. Cela constitue l'un des apports principaux de ce travail de recherche.

Une fois les alertes potentielles détectées par le moteur d'inférence, il est alors nécessaire de classifier le comportement du navire. Pour cela, le paradigme choisi est celui du raisonnement à partir de cas

Conclusion

(RàPC). En effet, le RàPC est particulièrement adapté dans le cas d'applications dont les tâches sont accomplies par des experts du domaine et où la recherche de solution optimale n'est pas une nécessité. La classification s'opère alors en comparant la situation actuelle d'un navire aux cas déjà enregistrés afin de déterminer les plus similaires. Un indice de ressemblance est ensuite calculé à partir de la similarité sémantique entre les individus.

PERSPECTIVES

Cette section liste quelques-unes des perspectives potentielles de ce travail de recherche. Les perspectives applicatives sont tout d'abord présentées, puis celles liées au domaine maritime pour enfin terminer sur celles relatives à la gestion des connaissances.

PERSPECTIVES APPLICATIVES

Bien que fonctionnel, le prototype présenté peut encore être amélioré. Ces améliorations portent aussi bien sur l'architecture elle-même que sur la base de connaissances.

Par exemple, à l'heure actuelle, l'implémentation des règles de détection a été réalisée manuellement avec l'aide des connaissances expertes. Ce processus, bien que valide, nécessite un engagement humain important. Une approche basée sur l'exploration des données (*Data mining*) apparaît alors comme particulièrement pertinente. Celle-ci permettrait un enrichissement automatique des scénarios à partir d'analyses statistiques spécifiques au domaine maritime. Une thèse sur ce sujet est en cours au laboratoire et devrait permettre de faire le lien entre l'approche orientée expert qui a été privilégiée dans ce travail et une approche orientée données .

Avec l'augmentation du nombre de règles et du nombre de cas, une attention particulière devra être portée aux temps de calcul et d'inférences. Pour cela, plusieurs alternatives peuvent être proposées. L'une des plus importantes est notamment l'enregistrement des individus dans une base de données. Trois propositions méritent d'être explorées. Premièrement, l'enregistrement de ces individus dans une base de données relationnelle auquel serait ajouté un module de traduction permettant d'obtenir une représentation en RDF des informations. Deuxièmement, l'intégration d'une base de données stockant directement les informations au format RDF comme cela est proposé par Oracle et AllegroGraph. Enfin, une troisième option pourrait être d'utiliser les potentialités des bases de données NOSQL et notamment celles orientées graph qui offre un formalisme de représentation proche de la logique ontologique. Pour cela, le choix de Neo4J apparaît comme particulièrement intéressant. En effet, bien que limitée, une cartouche spatiale comprenant des objets et des fonctions spatiales est disponible. De plus, il existe également un connecteur au serveur cartographique GeoServer. Cela permettrait ainsi une intégration plus forte entre les différents modules.

PERSPECTIVES DANS LE DOMAINE DES SYSTÈMES DE SURVEILLANCE MARITIME

Les besoins en terme de surveillance maritime sont dépendants de l'échelle d'analyse. En effet, les

analyses spatiales à utiliser peuvent être différentes en fonction de la zone à observer (un port, un rail de navigation ou une préfecture maritime). La création d'ontologies spécifiques par type de zones permettrait d'apporter une meilleure finesse dans l'exploration du domaine. Dans le même ordre d'idée, il serait également intéressant de proposer une ontologie en fonction du profil de l'utilisateur. En effet, un expert du CROSS, de la Gendarmerie Maritime ou encore le préfet maritime n'ont pas les mêmes besoins, missions et leviers d'actions. Dans l'ontologie actuelle, cette distinction n'a pas été explorée mais celle-ci nous semble nécessaire pour arriver à une plus grande généricité du modèle.

De plus, afin de couvrir l'ensemble du processus de gestion des risques, la phase de retour d'expérience doit également être prise en compte. Cette remarque est d'ailleurs à mettre en correspondance avec les phases du raisonnement à partir de cas et plus particulièrement celle de mémorisation. En effet, la validation d'un cas par les experts pourrait permettre ce retour d'expérience et participer ainsi à une amélioration générale du processus de gestion des risques.

A cela s'ajoute également la possibilité d'étudier non plus uniquement les navires mais également les alertes. Celles-ci seraient vues alors comme des objets géographiques localisés spatialement et temporellement. La mise à disposition de la plate-forme et son intégration dans un système plus global peut également être envisagées. Cela pourra notamment être réalisés dans le cadre des différents projets dans lesquels nous sommes impliqués. Au niveau du laboratoire, cette intégration au portail cartographique *Fish Eye* est en cours de réalisation. Ce portail réalisé en interne est une interface web au sein de laquelle seront intégrés les différents travaux de recherche effectués.

Enfin, bien que l'approche proposée ait été appliquée à la surveillance maritime il serait intéressant de généraliser celle-ci à d'autres domaines. Cela permettrait ainsi de vérifier la généricité du modèle et de voir si les ajustements à réaliser sont importants.

PERSPECTIVES DANS LE DOMAINE DE LA MODÉLISATION DES CONNAISSANCES

En raison de l'absence de prise en compte de la dimension spatiale dans les ontologies, une adaptation s'appuyant sur les potentialités du Semantic Web Rule Language a été proposée. Même si le travail élaboré dans le cadre de cette thèse répond à la problématique initiale, certaines limites et questions restent ouvertes.

Tout d'abord, le prototype proposé a été conçu pour une manipulation spécifique et limitée des relations spatiales. Quelles seraient alors les étapes nécessaires pour rendre celui-ci le plus générique possible ? A cela s'ajoute également la nécessité d'une meilleure intégration de la dimension spatiale afin que celle-ci ne soit plus gérées comme une extension, mais directement intégrée dans les ontologies. Cela nécessite que soient créées des types spatiaux à partir desquels pourront être définies des fonctions spatiales.

De plus, pour le moment, seule la partie concernant les règles SWRL a été abordée. Il serait intéressant et pertinent de conduire une analyse similaire mais cette fois en proposant une extension du langage de requête Sparql.

En raison des contraintes temporelles, uInne rencontre finale avec les experts de la surveillance maritime n'a malheureusement pas pu être réalisée. Mais, celle-ci a été programmée afin de présenter les travaux effectués et d'identifier les améliorations futures à apporter. Cette rencontre permettra

Conclusion

également d'enrichir la base de connaissances, les règles d'inférence et de compléter les mesures de similarité du module de raisonnement à partir de cas.

Enfin, bien que cela ne soit encore qu'une hypothèse, la création d'un outil visuel pour la gestion des connaissances apparaît comme particulièrement intéressante. Celui-ci permettrait la création des scénarios et des règles et cela sans qu'il soit nécessaire aux experts du domaine maritime de connaître les langages OWL ou SWRL.

RÉFÉRENCES

- Aamodt, A., 1991. A knowledge-intensive, integrated approach to problem solving and sustained learning (Phd Thesis).
- Aamodt, A., 2004. Knowledge-intensive case-based reasoning in creek, in: In Proceedings of the Seventh European Conference on Case-Based Reasoning. Springer, pp. 1–15.
- Abadie, N., Mustière, S., 2010. Constitution et exploitation d'une taxonomie géographique à partir des spécifications de bases de données. *Revue Internationale de Géomatique* 20, 145–174.
- Abou Assali, A., 2010. Acquisition des connaissances d'adaptation et traitement de l'hétérogénéité dans un système de RàPC basé sur une ontologie (Thèse).
- Agarwal, P., 2005. Ontological considerations in GIScience. *International Journal of Geographical Information Science* 19, 501–536.
- Allen, J.F., 1983. Maintaining knowledge about temporal intervals. *Commun. ACM* 26, 832–843.
- Alt, H., 2009. The Computational Geometry of Comparing Shapes, in: Albers, S., Alt, H., Näher, S. (Eds.), *Efficient Algorithms, Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, pp. 235–248.
- Anderson, J.R., Bower, G.H., 1980. *Human associative memory: a brief edition*. Routledge.
- Andrienko, N., Andrienko, G., 2007. Designing Visual Analytics Methods for Massive Collections of Movement Data. *Cartographica: The International Journal for Geographic Information and Geovisualization* 42, 117–138.
- Antoniou, G., Harmelen, F. van, 2004. Web Ontology Language: OWL, in: Staab, S., Studer, R. (Eds.), *Handbook on Ontologies*. Springer, Berlin, Heidelberg, pp. 67–92.
- Ashish, N., Sheth, A., 2011. *Geospatial Semantics and the Semantic Web: Foundations, Algorithms, and Applications*. Springer, p. 176.
- Baader, F., 2003. *The description logic handbook: theory, implementation, and applications*. Cambridge University Press.
- Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., 2007. Relationships with other Formalisms, in: *The Description Logic Handbook*. Cambridge University Press.
- Baader, F., Nutt, W., 2003. Basic description logics, in: *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, pp. 47–100.
- Baader, F., Nutt, W., 2003. Basic description logics, in: *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, pp. 47–100.
- Bachimont, B., 2000. Engagement Sémantique et Engagement Ontologique : Conception et Réalisation D'ontologies En Ingénierie Des Connaissances, in: *Ingénierie Des Connaissances Évolutions Récentes Et Nouveaux Défis*. J. Charlet, M. Zacklad, G. Kassel & D. Bourigault (Eds.), p. 505.
- Baglioni, M., Macedo, J., Renso, C., Wachowicz, M., 2008. An Ontology-Based Approach for the

Semantic Modelling and Reasoning on Trajectories, in: *Advances in Conceptual Modeling – Challenges and Opportunities*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 344–353.

Baglioni, M., Masserotti, M.V., Renso, C., Spinsanti, L., 2011. Improving Geodatabase Semantic Querying Exploiting Ontologies, in: *GeoSpatial Semantics*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 16–33.

Balbani, P., Muller, P., 2000. Le raisonnement spatial, in: *Le temps, l'espace et l'évolutif en Sciences du Traitement de l'Information*. Cépaduès éd., pp. 33–53.

Baneyx, A., 2007. Construire une onthologie de la pneumologie : aspects théoriques, modèles et expérimentations.

Bateman, J., Farrar, S., 2006. Spatial Ontology Baseline (No. SFB/TR 8:I1-[OntoSpace]). University of Bremen.

Baumann, P., 2007. Raster databases, in: *Proceedings of the 33rd International Conference on Very Large Data Bases, VLDB '07*. VLDB Endowment, pp. 1428–1428.

Bechhofer, S., 2002. Ontology Language Standardisation Efforts (No. Deliverable 4.0). University of Manchester.

Bechhofer, S., Goble, C., Horrocks, ian, 2002. Requirements of Ontology Languages.

Bechhofer, S., Möller, R., Crowther, P., 2003. The DIG Description Logic Interface, in: *Proceedings of DL2003*. Presented at the International Workshop on Description Logics.

Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A., 2004. OWL Web Ontology Language Reference.

Bechhofer, S., Volz, R., Lord, P., 2003. *Cooking the Semantic Web with the OWL API*. Springer, pp. 659–675.

Beckett, D. (Ed.), 2004. *RDF/XML Syntax Specification (Revised)*.

Bédard, Y., Larrivée, S., Proulx, M.-J., Nadeau, M., 2004. 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: *Conceptual Modeling for Advanced Application Domains*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 17–30.

Berners-Lee, T., 1998. *The Semantic Web Roadmap*.

Berners-Lee, T., 1999. *W3C Issues Recommendation for Resource Description Framework (RDF)*, Introduces model for defining and organizing information.

Berners-Lee, T., Fischetti, M., 1999. *Weaving the Web : the original design and ultimate destiny of the World Wide Web by its inventor*. HarperSanFrancisco, San Francisco.

Bertrand, F., Bouju, A., Claramunt, C., Devogele, T., Ray, C., 2007. Web Architecture for Monitoring and Visualizing Mobile Objects in Maritime Contexts, in: Ware, J., Taylor, G. (Eds.), *Web and Wireless Geographical Information Systems*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 94–105.

- Bizer, C., Cyganiak, R., 2007. D2RQ - Lessons Learned. W3C Workshop on RDF Access to Relational Databases.
- Borgo, S., Guarino, N., Masolo, C., 1996. Qualitative Spatial Modelling Based on Parthood.
- Brachman, R.J., 1977. What's in a concept: structural foundations for semantic networks. *International Journal of Man-Machine Studies* 29, 127–152.
- Brachman, R.J., Levesque, H.J., 2004. Knowledge representation and reasoning. Morgan Kaufmann.
- Brakatsoulas, S., Pfoser, D., Tryfona, N., 2004. Modeling, storing and mining moving object databases, in: Database Engineering and Applications Symposium, 2004. IDEAS '04. Proceedings. International. Presented at the Database Engineering and Applications Symposium, 2004. IDEAS '04. Proceedings. International, pp. 68 – 77.
- Brax, C., Karlsson, A., Andler, S.F., Johansson, R., Niklasson, L., 2010. Evaluating precise and imprecise state-based anomaly detectors for maritime surveillance. *Information Fusion FUSION 2010 13th Conference on* 1–8.
- Brickley, D., Guha, R.V., 2004. RDF Vocabulary Description Language 1.0: RDF Schema (W3C Recommendation 10 February 2004).
- Brodeur, J., 2012. Geosemantic Interoperability and the Geospatial Semantic Web, in: Kresse, W., Danko, D.M. (Eds.), *Springer Handbook of Geographic Information*. Springer Berlin Heidelberg, pp. 291–310.
- Carloni, O., 2008. Introduction de raisonnement dans un outil industriel de gestion des connaissances (Thèse).
- Carroll, J.J., Dickinson, I., Dollin, C., Seaborne, A., Wilkinson, K., Reynolds, D., Reynolds, D., 2004. Jena: Implementing the Semantic Web Recommendations. Presented at the WWW2004, New York, pp. 74–83.
- Casati, R., Varzi, A.C., 1995. *Holes and Other Superficialities*. The MIT Press.
- Chakhar, S., 2006. *Cartographie décisionnelle multicritère : formalisation et implémentation informatique* (Phd).
- Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A survey. *ACM Comput. Surv.* 41, 1–58.
- Chaze, X., Bouejla, A., Guarnieri, F., 2012. The contribution of bayesian networks to manage risks of maritime piracy against oil offshore fields. Presented at the DASFAA, Korean.
- Claramunt, C., Devogele, T., Fournier, S., Noyon, V., Petit, M., Ray, C., 2007. Maritime GIS: From Monitoring to Simulation Systems, in: *Information Fusion and Geographic Information Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 34–44.
- Clarke, B.L., 1981. A Calculus of Individuals Based on Connection. *Notre Dame Journal of Formal Logic* Notre-Dame, Ind. 22, 204–219.
- Clementini, E., Di Felice, P., 1995. A comparison of methods for representing topological relationships. *Information Sciences - Applications* 3, 149–178.

- Clementini, E., Laurini, R., 2008. Un cadre conceptuel pour modéliser les relations spatiales. *Revue des Nouvelles Technologies de l'Information RNTI-E-14*, 1–17.
- Cohn, A., Renz, J., 2008. Qualitative Spatial Representation and Reasoning, in: *Handbook of Knowledge Representation*. Elsevier, pp. 551–596.
- Cohn, A.G., Hazarika, S.M., 2001. Qualitative Spatial Representation and Reasoning: An Overview.
- Collins, A., Quillian, R., 1969. Retrieval time from semantic memory. *Journal Of Verbal Learning And Verbal Behavior* 8, 240–247.
- Corcho, O., 2003. Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering* 46, 41–64.
- Corcho, O., Gómez-Pérez, A., 2000. *A Roadmap to Ontology Specification Languages*. Springer-Verlag, pp. 80–96.
- Cordier, A., Fuchs, B., Lieber, J., Mille, A., 2007. Acquisition interactive des connaissances d'adaptation intégrée aux sessions de raisonnement à partir de cas - Principes, architecture IakA et prototype KayaK. Presented at the 15ème atelier sur le raisonnement à partir de cas - RàPC-07, pp. 71–84.
- Cotteret, G., 2005. Extraction d'éléments curvilignes guidée par des mécanismes attentionnels pour des images de télédétection (Thèse).
- Cox, S., Daisez, P., Portele, C., Whiteside, A., 2004. OpenGIS Geography Markup Language. (GML) Implementation Specification (No. OGC 03-105r1). Open Geospatial Consortium.
- Cullot, N., Parent, C., Spaccapietra, S., Vangenot, C., 2003. Des ontologies pour données géographiques. *Revue internationale de géomatique, Les SIG sur le web* 13, 285–306.
- d' Aquin, M., Lieber, J., Napoli, A., 2006. Case-Based Reasoning Within Semantic Web Technologies, in: *Artificial Intelligence: Methodology, Systems, and Applications*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 190–200.
- De Nicola, A., Missikoff, M., Navigli, R., 2009. A software engineering approach to ontology building. *Information Systems* 34, 258–275.
- Dee, H., Velastin, S., 2008. How close are we to solving the problem of automated visual surveillance? *Machine Vision and Applications* 19, 329–343.
- Devogele, T., 2009. *Système d'information géographique temporelle maritime; Des distances linéaires à l'analyse temps réel des trajectoires (Habilitation à Diriger des Recherches Système d'information géographique temp)*.
- Dia Miron, A., 2009. *Découverte d'associations sémantiques pour le Web Sémantique Géospatial - le framework ONTOAST (Thèse)*.
- Díaz-Agudo, B., González-Calero, P.A., 2002. CBRonto: A Task/Method Ontology for CBR, in: *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference*. AAAI Press, pp. 101–105.
- Díaz-Agudo, B., González-Calero, P.A., Recio-García, J.A., Sánchez-Ruiz-Granados, A.A., 2007. Building CBR systems with jcolibri. *Sci. Comput. Program.* 69, 68–75.

- Donnelly, M., Smith, B., 2003. Layers: A New Approach to Locating Objects in Space, in: Kuhn, W., Worboys, M., Timpf, S. (Eds.), *Spatial Information Theory. Foundations of Geographic Information Science*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 46–60.
- Douglas, D., Peucker, T., 1973. Algorithms for the reduction of the number of points required to represent a digitized line of its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10, 112–122.
- Egenhofer, M.J., Mark, D.M., 1994. Categorizing binary topological relations between regions, lines, and points in geographic databases (No. Report 94-1). National Center for Geographic Information and Analysis.
- Egenhofer, M.J., Mark, D.M., 1995. Naive Geography. Presented at the COSIT'95, Lecture Notes in Computer Science, Semmering, Austria, pp. 1–15.
- Egenhofer, M.J., Mark, D.M., Herring, J., 1994. The 9-Intersection: Formalism And Its Use For Natural-Language Spatial Predicates (No. Report 94-1). National Center for geographic Information and Analysis.
- Ekman, J., Holst, A., 2008. Incremental Stream Clustering and Anomaly Detection (No. T2008:1). SICS.
- Etienne, L., 2011. Motifs spatio-temporels de trajectoires d'objets mobiles, de l'extraction à la détection de comportements inhabituels ; application au trafic maritime (Phd).
- Etienne, L., Devogele, T., Bouju, A., 2010. Spatio-temporal trajectory analysis of mobile objects following the same itinerary, in: *Proceedings of the International Symposium on Spatial Data Handling*.
- Farquhar, A., Fikes, R., Rice, J., 1997. The Ontolingua Server: a tool for collaborative ontology construction. *Int. J. Hum.-Comput. Stud.* 46, 707–727.
- Fensel, D., 2000. *Relating Ontology Languages and Web Standards*.
- Fernández-López, M., Gomez, A., Juristo, N., 1997. METHONTOLOGY: From Ontological Art Towards Ontological Engineering 33–40.
- Fernández-López, M., Gómez-Pérez, A., 2002. Overview and analysis of methodologies for building ontologies. *Knowl. Eng. Rev.* 17, 129–156.
- Fonseca, F.T., Egenhofer, M.J., 1999. Ontology-driven geographic information systems, in: *Proceedings of the 7th ACM International Symposium on Advances in Geographic Information Systems, GIS '99*. ACM, New York, NY, USA, pp. 14–19.
- Fonseca, F.T., Egenhofer, M.J., Agouris, P., Camara, G., 2002. Using Ontologies for Integrated Geographic Information Systems. *Transactions in GIS* 6, 231–257.
- Forlizzi, L., Güting, R.H., Nardelli, E., Schneider, M., 2000. A data model and data structures for moving objects databases, in: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. Presented at the SIGMOD '00, pp. 319–330.
- Fournier-Viger, P., 2005. Un modèle de représentation des connaissances à trois niveaux de sémantique pour les systèmes tutoriels intelligents.

- Fréchet, M., 1905. Sur l'écart de deux courbes et sur les courbes limites. *Transactions of the American Mathematical Society* 6, 435–449.
- Gandon, F., 2008. *Graphes RDF et leur Manipulation pour la Gestion de Connaissances (Habilitation à Diriger des Recherches)*.
- Garcia-Castro, R., Grimm, S., Schneider, M., Toma, I., Marte, A., Tymaniuk, S., 2010. Semantic Evaluation at Large Scale (No. SEALS/2010/D10.3/V1.0), Results of the first evaluation of ontology engineering tools. Union Européenne.
- Gennari, J.H., Musen, M.A., Ferguson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F., Tu, S.W., 2003. The evolution of Protégé an environment for knowledge-based systems development. *Int. J. Hum.-Comput. Stud.* 58, 89–123.
- Gensel, J., 2006. *Représentation de connaissances par objets pour la modélisation et l'adaptation d'informations multi-médias spatio-temporelles (Habilitation à Diriger des Recherches)*.
- Giannotti, F., Nanni, M., Pedreschi, D., Renso, C., Trasarti, R., 2009. Mining Mobility Behavior from Trajectory Data, in: *CSE (4)'09*. pp. 948–951.
- Giritli, M., 2003. Who Can Connect in RCC?, in: Günter, A., Kruse, R., Neumann, B. (Eds.), *KI 2003: Advances in Artificial Intelligence, Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, pp. 565–579.
- Gómez-Pérez, A., 1999. Ontological Engineering: A State Of The Art. *Expert Update: Knowledge Based Systems and Applied Artificial Intelligence* 2, 33–43.
- Gómez-Pérez, A., Fernández-López, M., Corcho, O., 2004. *Ontological engineering: with examples from the areas of knowledge management, e-commerce and the Semantic Web*. Springer.
- Gotts, N.M., 1996. *An Axiomatic Approach to Topology For Spatial Information Systems (No. 96.25)*. University of Leeds.
- Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U., 2008. OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web* 6, 309–322.
- Grenon, P., Smith, B., 2004. SNAP and SPAN: Towards Dynamic Spatial Ontology. *Spatial Cognition & Computation* 4, 69–104.
- Grenon, P., Smith, B., Goldberg, L., 2004. Biodynamic Ontology: Applying BFO in the Biomedical Domain, in: *Stud. Health Technol. Inform.* IOS Press, pp. 20–38.
- Gruber, T.R., 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition* 5, 199–220.
- Gruber, T.R., 1995. Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies* 43, 907–928.
- Grütter, R., Bauer-Messmer, B., 2007. Towards Spatial Reasoning in the Semantic Web: A Hybrid Knowledge Representation System Architecture, in: Fabrikant, S.I., Wachowicz, M. (Eds.), *The European Information Society, Lecture Notes in Geoinformation and Cartography*. Springer Berlin Heidelberg, pp. 349–364.

- Grütter, R., Scharrenbach, T., Bauer-Messmer, B., 2008. Improving an RCC-Derived Geospatial Approximation by OWL Axioms, in: Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (Eds.), *The Semantic Web - ISWC 2008, Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, pp. 293–306.
- Guarino, N., 1998. Formal Ontology and Information Systems, in: *Proceedings of FOIS'98*. IOS Press, Trento, Italy, pp. 3–15.
- Guarino, N., Giaretta, P., 1995. Ontologies and Knowledge Bases: Towards a Terminological Clarification. *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing* 25–32.
- Guarino, N., Welty, C., 2000. *A Formal Ontology of Properties*.
- Guerriero, M., Willett, P., Coraluppi, S., Carthel, C., 2008. Radar/AIS data fusion and SAR tasking for Maritime Surveillance, in: *Information Fusion, 2008 11th International Conference On*. pp. 1–5.
- Guizzardi, G., 2007. On Ontology, ontologies, Conceptualizations, Modeling Languages. *AND (META)MODELS, FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS, DATABASES AND INFORMATION SYSTEMS IV*, IOS.
- Güting, R.H., Almeida, V.T.D., Ding, Z., 2006. Modeling and Querying Moving Objects in Networks. *VLDB J* 15:2, 165–190.
- Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M., 2000. A foundation for representing and querying moving objects. *ACM Trans. Database Syst.* 25, 1–42.
- Haarslev, V., Lutz, C., Möller, R., 1998. Foundations of Spatioterminological Reasoning with Description Logics, in: *Proceedings of Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*. Morgan Kaufmann, pp. 112–123.
- Hausdorff, F., 1918. Dimension und äußeres Maß. *Mathematische Annalen* 79, 157–179.
- Henß, J., Kleb, J., Grimm, S., Bock, J., 2009. A Database Backend for OWL, in: *Proceedings of the 5th International Workshop on OWL: Experiences and Directions*. Presented at the OWLED 2009, United States.
- Hepp, M., 2008. Ontologies: State of the Art, Business Potential, and Grand Challenges, in: Hepp, M., Leenheer, P., Moor, A., Sure, Y. (Eds.), *Ontology Management*. Springer US, Boston, MA, pp. 3–22.
- Hepp, M., Leenheer, P.D., Moor, A. de, 2007. *Ontology management: semantic web, semantic web services, and business applications*. Springer.
- Hernandez, N., 2005. *Ontologies de domaine pour la modélisation de contexte en recherche d'information*.
- Hobbs, J.R., Pan, F., 2004. An Ontology of Time for the Semantic Web. *ACM Transactions on Asian Language Information Processing* 3, 66–85.
- Hodge, V.J., Austin, J., 2004. A survey of outlier detection methodologies. *Artificial Intelligence Review* 22, 2004.

- Hogenboom, F., Borgman, B., Frasincar, F., Kaymak, U., 2010. Spatial Knowledge Representation on the Semantic Web, in: Semantic Computing (ICSC), 2010 IEEE Fourth International Conference On. pp. 252–259.
- Hornsby, K., Egenhofer, M.J., 2002. Modeling Moving Objects over Multiple Granularities. *Annals of Mathematics and Artificial Intelligence* 36, 177–194.
- Horridge, M., Bechhofer, S., 2011. The OWL API: A Java API for OWL ontologies. *Semantic Web* 2, 11–21.
- Horrocks, I., 1998. The FaCT system, in: In Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX'98), Volume 1397 in Lecture Notes in Artificial Intelligence. Springer, pp. 307–312.
- Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosz, B., Dean, M., 2004. SWRL: A Semantic Web Rule Language Combining OWL and RuleML.
- Hu, W., Xiao, X., Fu, Z., Xie, D., Tan, T., Maybank, S., 2006. A system for learning statistical motion patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28, 1450–1464.
- Hustadt, U., Motik, B., Sattler, U., 2004. Reducing SHIQ – Description Logic to Disjunctive Datalog Programs. *AAAI Press*, pp. 152–162.
- Hustadt, U., Motik, B., Sattler, U., 2007. Reasoning in Description Logics by a Reduction to Disjunctive Datalog. *J. Autom. Reason.* 39, 351–384.
- IALA, 2010. IALA Technical clarifications on recommendation ITU-R M.1371-3. International Institution of Marine Aids to Navigation and Lighthouse Authorities.
- Idiri, B., Napoli, A., 2012. Découverte de règles d'association pour l'aide à la prévision des accidents maritimes. Presented at the Extraction et Gestion des Connaissances (EGC), Bordeaux.
- Jangal, F., Georgé, J.-P., Bonnot, A., Giraud, M.-A., Morel, M., Napoli, A., Littaye, A., 2009. Toward a complete system for surveillance of the whole EEZ: ScanMaris and associated projects, in: OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges. Presented at the OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges, pp. 1–4.
- Jangal, F., Giraud, M.-A., Littaye, A., Morel, M., Mano, J.-P., Napoli, A., 2008. Extraction of Suspicious Behavior of Vessels in the Exclusive Economic Zone, in: International Symposium on Antennas Et Propagation. Presented at the ISAP'08 - International Symposium on Antennas et Propagation.
- Johansson, F., Falkman, G., 2007. Detection of vessel anomalies - a Bayesian network approach. 2007 3rd International Conference on Intelligent Sensors Sensor Networks and Information 395–400.
- Karmacharya, A., 2011. Intégration d'une couche spatiale dans l'architecture du Web Sémantique: Une proposition via la plateforme ArchaeoKM.
- Karmacharya, A., Cruz, C., Boochs, F., Marzani, F., 2011. Integration of spatial processing and knowledge processing through the semantic web stack, in: Proceedings of the 4th

- International Conference on GeoSpatial Semantics, GeoS'11. Springer-Verlag, Berlin, Heidelberg, pp. 200–216.
- Katz, Y., Grau, B.C., 2005. Representing Qualitative Spatial Information in OWL-DL, in: OWL: Experiences and Directions Workshop. Galway, Ireland.
- Kaufman, L., Rousseeuw, P.J., 2005. Finding groups in data: an introduction to cluster analysis. Wiley.
- Keita, A.K., 2007. Conception coopérative d'ontologies pré-consensuelles : application au domaine de l'urbanisme.
- Khondoker, R., Mueller, P., 2010. Comparing Ontology Development Tools Based on an Online Survey, in: Proceedings of the World Congress on Engineering. Presented at the World Congress on Engineering, London.
- Knublauch, H., Fergerson, R.W., Noy, N.F., Musen, M.A., 2004. The Protégé OWL plugin: An open development environment for semantic web applications. Springer, pp. 229–243.
- Kokar, M.M., Matheus, C.J., Baclawski, K., 2009. Ontology-based situation awareness. *Information Fusion* 10, 83–98.
- Kotis, K., Papasalouros, A., Nikitakos, N., 2009. Supporting Decision Making in Maritime Environmental Protection with a Knowledge-based Education and Awareness Approach, in: Proceedings of the Workshops of the 5th IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI-2009), CEUR Workshop Proceedings. Presented at the AIAI Workshops'09, Nick Bassiliades, Thessaloniki, Greece, pp. 280–290.
- Kou, Y., Lu, C., 2006. Spatial weighted outlier detection, in: In Proceedings of SIAM Conference on Data Mining.
- Kristiansen, S., 2004. Maritime Transportation: Safety Management And Risk Analysis. Butterworth-Heinemann Ltd.
- Kuhn, W., 2001. Ontologies in support of activities in geographical space. *International Journal of Geographical Information Science* 15, 613–631.
- Kuhn, W., 2005. Geospatial Semantics: Why, of What, and How?, in: Spaccapietra, S., Zimányi, E. (Eds.), *Journal on Data Semantics III*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 587–587.
- Kurata, Y., Egenhofer, M.J., 2007. The 9 +-Intersection for Topological Relations between a Directed Line Segment and a Region, in: BMI'07. Osnabrück, Germany, University of Bremen.
- Lamontagne, L., Lapalme, G., 2002. Raisonement à base de cas textuels : Etat de l'art et perspectives. *Revue d'intelligence artificielle* 16, 339–366.
- Larrivée, S., Bédard, Y., 2002. Représentation multiple et généralisation avec UML et l'outil Perceptory. *Hermes, Lecture Notes in Computer Science Généralisation et représentation multiples*, 113–129.
- Larvor, Y., 2004. Notions de méréogéométrie: description qualitative de propriétés géométriques du mouvement et de la forme d'objets tridimensionnels.

- Lassila, O., Swick, R.R., 1999. Resource Description Framework (RDF) Model and Syntax Specification (W3C Recommendation).
- Laurini, R., 2007. Pre-consensus Ontologies and Urban Databases., in: *Ontologies for Urban Development, Studies in Computational Intelligence*. Springer, pp. 27–36.
- Laxhammar, R., 2011. Anomaly detection in trajectory data for surveillance applications (Phd).
- Le Guyader, D., Brosset, D., Gourmelon, F., 2012. Exploitation de données AIS pour la cartographie du transport maritime. *Mappemonde* 105, 15.
- Lieber, J., 2008. Contributions à la conception de systèmes de raisonnement à partir de cas (Habilitation à Diriger des Recherches).
- Lieber, J., D'Aquin, M., Brachais, S., Napoli, A., 2004. Une étude comparative de quelques travaux sur l'acquisition de connaissances d'adaptation pour le raisonnement à partir de cas. Presented at the 12ème Atelier de Raisonnement à Partir de Cas - RàPC'04, Université Paris.
- Lieberman, J., 2006. Geospatial Semantic Web Interoperability Experiment Report (No. OGC 06-002r1 V0.5). Open Geospatial Consortium.
- Lieberman, J., Singh, R., Goad, C., 2007a. W3C Geospatial Ontologies - W3C Incubator Group. W3C Incubator Group.
- Lieberman, J., Singh, R., Goad, C., 2007b. W3C Geospatial Vocabulary. W3C Incubator Group.
- Maedche, A., Staab, S., 2001. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems* 16, 72–79.
- Malki, J., Mefteh, W., Bouju, A., 2009. Une approche ontologique pour la modélisation et le raisonnement sur les trajectoires. Prise en compte des règles métiers, spatiales et temporelles, in: 3ème Édition Des Journées Francophones Sur Les Ontologies. Presented at the JFO 2009, France, pp. 157–168.
- Mark, D., Egenhofer, M., Bian, L., Rogerson, P., Vena, J., 1999. Spatio-temporal GIS analysis for environmental health using geospatial lifelines, in: 2nd International Workshop on Geography and Medicine. Presented at the GEOMED'99, Paris, France.
- Mark, D.M., Turk, A.G., 2003. Landscape categories in Yindjibarndi: Ontology, environment, and language. *LNCS* 2825, 28–45.
- Martineau, E., Roy, J., 2011. Maritime Anomaly Detection: Domain Introduction and Review of Selected Literature (No. DRDC-VALCARTIER-TM-2010-460). Defence R&D Canada, Canada.
- McBride, B., 2001. Jena: Implementing the RDF Model and Syntax Specification.
- MDA Report, 2005. National Plan to Achieve Maritime Domain Awareness for the National Strategy for Maritime Security.
- Medalia, J., 2004. Port and Maritime Security: Potential for Terrorist Nuclear Attack Using Oil Tankers.
- Mefteh, M., Bouju, A., Malki, J., 2012. Une approche ontologique pour la structuration de données

spatio-temporelles de trajectoires. Application à l'étude des déplacements de mammifères marins. *Revue Internationale de Géomatique* 22, 55–75.

- Michoux, L., 2007. SPATIONAV : Pierre angulaire de la protection et de la sauvegarde maritime. *REE. Revue de l'électricité et de l'électronique* 81–84.
- Mille, A., Fuchs, B., Herbeaux, O., 1996. A unifying framework for Adaptation in Case-Based Reasoning, in: *In Workshop on Adaptation in Case-Based Reasoning, 12th European Conference on Artificial Intelligence, ECAI-1996*. John Wiley & Sons, Ltd, pp. 22–28.
- Minsky, M., 1974. *A Framework for Representing Knowledge*. Massachusetts Institute of Technology.
- Mishra, R.B., Kumar, S., 2010. Semantic web reasoners and languages. *Artificial Intelligence Review* 35, 339–368.
- Mizoguchi, R., 2003. Part 1: introduction to ontological engineering. *New Gen. Comput.* 21, 365–384.
- Mizoguchi, R., Bourdeau, J., 2000. Using Ontological Engineering to Overcome Common AI-ED Problems. *Journal of Artificial Intelligence and Education* 107–121.
- Mizoguchi, R., Ikeda, M., 1996. Technical Report AI-TR-96-1, I.S.I.R., Osaka Univ Towards Ontology Engineering.
- Mondary, T., 2011. Construction d'ontologies à partir de textes. L'apport de l'analyse de concepts formels. (Phd).
- Morel, M., Claisse, S., 2010a. Système d'Information et de Surveillance MARtime pour l'identification des comportements Suspects, in: *Les 8èmes JST Du CETMEF*. Brest.
- Morel, M., Claisse, S., 2010b. Integrated, System for Interoperable sensors & Information sources for Common abnormal vessel behaviour detection & Collaborative identification of threat (I2C), in: *Proc. of the Ocean and Coastal Observation: Sensors and Observing Systems, Numerical Models and Information Systems*. Brest, France.
- Morris, B.T., Trivedi, M.M., 2008. A Survey of Vision-Based Trajectory Learning and Analysis for Surveillance. *IEEE Transactions on Circuits and Systems for Video Technology* 18, 1114–1127.
- Mouza, C. du, Rigaux, P., 2005. Mobility Patterns. *Geoinformatica* 9, 297–319.
- Napoli, A., 1997. Une introduction aux logiques de descriptions [WWW Document]. URL <http://hal.inria.fr/inria-00073375>
- Nguyen, V., 2011. *Ontologies and Information Systems: A Literature Survey* (No. DSTO-TN-1002). Defence Science and Technology Organisation.
- Niles, I., Pease, A., 2001. Towards a standard upper ontology, in: *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001, FOIS '01*. ACM, New York, NY, USA, pp. 2–9.
- Nilsson, M., van Laere, J., Ziemke, T., Edlund, J., 2008. Extracting rules from expert operators to support situation awareness in maritime surveillance, in: *2008 11th International Conference*

- on Information Fusion. Presented at the 2008 11th International Conference on Information Fusion, IEEE, pp. 1–8.
- Noucher, M., 2009. La donnée géographique aux frontières des organisations: approche socio-cognitive et systémique de son appropriation.
- Noyon, V., 2007. Modèle de vue relative et spatio-temporelle de trajectoires géographiques d'objets mobiles : application au contexte maritime.
- Noyon, V., Claramunt, C., Devogele, T., 2007. A Relative Representation of Trajectories in Geographical Spaces. *GeoInformatica* 11, 479–496.
- Noyon, V., Devogele, T., 2004. Système distribué de gestion d'objets mobiles, in: Actes Des Journées Des 7ième Journées Cassini. MSH Alpes, Grenoble, pp. 7–12.
- Parent, C., Spaccapietra, S., Zimanyi, E., 2006. Conceptual modeling for traditional and spatio-temporal applications: the MADS approach. Springer-Verlag.
- Patcha, A., Park, J.-M., 2007. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks* 51, 3448–3470.
- Patel-Schneider, P., Horrocks, I., 2006. OWL 1.1 Web Ontology Language Overview.
- Pease, A., Niles, I., Li, J., 2002. The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications, in: In Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web. p. 2002.
- Pelekis, N., Kopanakis, I., Kotsifakos, E.E., Frenzos, E., Theodoridis, Y., 2009. Clustering Trajectories of Moving Objects in an Uncertain World, in: *ICDM'09*. pp. 417–427.
- Pelekis, N., Theodoridis, Y., Vosinakis, S., Panayiotopoulos, T., 2006. Hermes - a framework for location-based data management, in: *Proceedings of the 10th International Conference on Advances in Database Technology*. Presented at the *EDBT'06*, Springer-Verlag, Berlin, Heidelberg, pp. 1130–1134.
- Piciarelli, C., Foresti, G.L., 2006. On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters* 27, 1835–1842.
- Prasad Sistla, A., Wolfson, O., Chamberlain, S., Dao, S., 1997. Modeling and querying moving objects, in: *Data Engineering, 1997. Proceedings. 13th International Conference On*. pp. 422–432.
- Psyché, V., 2007. Rôle des ontologies en ingénierie des EIAH : cas d'un système d'assistance au design pédagogique.
- Quillian, R., 1966. Semantic Memory.
- Rada, R., Mili, H., Bicknell, E., Blettner, M., 1989. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics* 19, 17–30.
- Randell, D.A., Cohn, A.G., 1989. Modelling Topological and Metrical Properties in Physical Processes., in: Eds), *Proceedings 1st International Conference on the Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, pp. 55–66.

- Randell, D.A., Cui, Z., Cohn, A.G., 1992. A Spatial Logic based on Regions and Connection, in: 3RD International Conference on Knowledge Representation and Reasoning. Morgan Kaufmann, San Mateo, pp. 165–176.
- Recio-García, J.A., Díaz-Agudo, B., González-Calero, P.A., Sanchez-Ruiz, A.A., Informáticos, D.S., 2006. Ontology based CBR with jCOLIBRI, in: Applications and Innovations in Intelligent Systems XIV. Springer-Verlag London, pp. 149–162.
- Recio-García, J.A., Díaz-Agudo, B., Sanchez-Ruiz, A.A., González-Calero, P.A., 2006. Lessons Learnt in the Development of a CBR Framework, in: Proceedings of the 11th UK Workshop on Case Based Reasoning. CMS Press, University of Greenwich, UK, pp. 60–71.
- Redmond, T., 2010. An Open Source Database Backend for the OWL API and Protege 4, in: OWL: Experiences and Directions. Presented at the OWLED 2010, Evren Surin, editor, California, USA.
- Renz, J., 2002. Qualitative Spatial Reasoning With Topological Information. Springer.
- Resnik, P., 1995. Using information content to evaluate semantic similarity in a taxonomy, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'95. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 448–453.
- Ressler, J., Dean, M., Benson, E., Dorner, E., Morris, C., 2007. Application of ontology translation, in: Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference, ISWC'07/ASWC'07. Springer-Verlag, Berlin, Heidelberg, pp. 830–842.
- Ressler, J., Dean, M., Kolas, D., 2007. Geospatial Ontology Trade Study (No. HM1582-05-C-0014). National Geospatial-Intelligence Agency.
- Rhodes, B.J., Bomberger, N.A., Zandipour, M., 2007. Probabilistic associative learning of vessel motion patterns at multiple spatial scales for maritime situation awareness, in: Information Fusion, 2007 10th International Conference On. pp. 1 –8.
- Ristic, B., La Scala, B., Morelande, M., Gordon, N., 2008. Statistical Analysis of Motion Patterns in AIS Data : Anomaly Detection and Motion Prediction. The 11th International Conference on Information Fusion 40–46.
- Riveiro, M., Falkman, G., Ziemke, T., 2008. Improving maritime anomaly detection and situation awareness through interactive visualization, in: Information Fusion, 2008 11th International Conference On. pp. 1 –8.
- Rodríguez, M., Egenhofer, M., 1999. Putting Similarity Assessments into Context: Matching Functions with the User's Intended Operations, in: Modeling and Using Context. Springer, pp. 310–323.
- Rodríguez, M.A., Egenhofer, M.J., 2003. Determining Semantic Similarity among Entity Classes from Different Ontologies. IEEE Transactions on Knowledge and Data Engineering 15, 442–456.
- Roussey, C., Laurini, R., Beaulieu, C., Tardy, Y., Zimmerman, M., 2004. Le projet Towntology : un retour d'expérience pour la construction d'une ontologie urbaine. Revue Internationale de Géomatique 14, 217–237.

- Roy, J., 2008. Anomaly detection in the maritime domain, in: Proceedings of SPIE. Presented at the Optics and Photonics in Global Homeland Security IV, Orlando, FL, USA, p. 69450W–69450W–14.
- Roy, J., Auger, A., DRDC, V., 2008. Knowledge and ontological engineering techniques for use in developing knowledge-based situation analysis support systems.
- Roy, J., Davenport, M., 2010. Exploitation of maritime domain ontologies for anomaly detection and threat analysis, in: Waterside Security Conference (WSS), 2010 International. Presented at the Waterside Security Conference (WSS), 2010 International, pp. 1–8.
- Salehi, M., 2009. Developing a Model and a Language to Identify and Specify the Integrity Constraints in Spatial Datacubes (Phd).
- Salvador, S., Chan, P., Brodie, J., 2004. Learning States and Rules for Time Series Anomaly Detection. *Artificial Intelligence* 23, 241–255.
- Sandu Popa, I., Kharrat, A., Zeitouni, K., Saint Pierre, G., 2009. Base de données de capteurs à localisation mobile Modèle et langage. *Ingénierie des systèmes d'information* 14, 35–58.
- Sboui, T., Bedard, Y., 2010. A systematic approach for managing the risk related to semantic interoperability between geospatial datacubes. *International Journal of Agricultural and Environmental Information Systems (IJAEIS)* 1, 20–41.
- Sboui, T., Bedard, Y., Rivest, S., 2009. Une approche basée sur la qualité pour faciliter l'intégration de modèles de cubes de données spatiales. *Revue internationale des nouvelles technologies de l'information (RNTI) - Fouille de Données Complexes RTNI-E-18*, 51–87.
- Schank, R.C., 1983. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge University Press, New York, NY, USA.
- Schank, R.C., Abelson, R.P., 1977. *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures*. Psychology Press.
- Schreiber, G., Wielinga, B., Breuker, J. (Eds.), 1993. *KADS: A Principled Approach to Knowledge-Based System Development*, 1st ed. Academic Press.
- Seco, N., Veale, T., Hayes, J., 2004. An Intrinsic Information Content Metric for Semantic Similarity in WordNet, in: ECAI'2004, the 16th European Conference on Artificial Intelligence.
- Shekhar, S., Lu, C.-T., Zhang, P., 2001. Detecting graph-based spatial outliers: algorithms and applications (a summary of results) 371–376.
- Simon, H.A., 1977. *The New Science of Management Decision*. Prentice Hall PTR.
- Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y., 2007. Pellet: A practical OWL-DL reasoner. *Web Semant.* 5, 51–53.
- Smith, B., 1996. Mereotopology: A Theory of Parts and Boundaries. *Data Knowl. Eng.* 20, 287–303.
- Smith, B., Mark, D.M., 2003. Do mountains exist? Towards an ontology of landforms. *Environment and Planning B: Planning and Design* 30, 411 – 427.

- Song, X., Wu, M., Jermaine, C., Ranka, S., 2007. Conditional Anomaly Detection. *IEEE Transactions on Knowledge and Data Engineering* 19, 631–645.
- Sowa, J.F., 1983. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley.
- Sowa, J.F., 1999. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Course Technology Inc.
- Spaccapietra, S., Cullot, N., Parent, C., Vangenot, C., 2004. On spatial Ontologies. Presented at the 6th Brazilian Symposium On Geoinformatics, Campos do Jordao, Brazil.
- Spaccapietra, S., Parent, C., Damiani, M.L., de Macedo, J.A., Porto, F., Vangenot, C., 2008. A conceptual view on trajectories. *Data & Knowledge Engineering* 65, 126–146.
- Staab, S., Studer, R. (Eds.), 2009. *Handbook on Ontologies*, 2nd ed. ed. Springer.
- Staab, S., Studer, R., Schnurr, H.-P., Sure, Y., 2001. Knowledge processes and ontologies. *Intelligent Systems, IEEE* 16, 26 – 34.
- Stocker, M., Sirin, E., 2009. PelletSpatial: A Hybrid RCC-8 and RDF / OWL Reasoning and Query Engine. *Language* 529.
- Strobl, C., 2008. Dimensionally Extended Nine-Intersection Model (DE-9IM). Shekhar, Shashi, Xiong, Hui.
- Studer, R., Benjamins, V.R., Fensel, D., 1998. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering* 25, 161–197.
- Temal, L., 2008. *Ontologie de partage de données et d’outils de traitement dans le domaine de la neuroimagerie*.
- Tsarkov, D., Horrocks, I., 2006. FaCT++ Description Logic Reasoner: System Description, in: *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR~2006)*. Springer, pp. 292–297.
- UNCD, 2011. *Review of Maritime Transport 2011* (No. UNCTAD/RMT/2011). United Nations Conference on Trade and Development.
- UNODC, 2011. *World Drug Report 2011* (No. E.11.XI.10). United Nations Office on Drugs and Crime.
- Uschold, M., Gruninger, M., 1996. *Ontologies: Principles, methods and applications*. *KNOWLEDGE ENGINEERING REVIEW* 11, 93–136.
- Uschold, M., King, M., 1995. *Towards a Methodology for Building Ontologies*, in: *In Workshop on Basic Ontological Issues in Knowledge Sharing*. Presented at the IJCAI-95.
- van Hage, W.R., Malaisé, V., de Vries, G., Schreiber, G., van Someren, M., 2009. Combining ship trajectories and semantics with the simple event model (SEM). *ACM Press*, p. 73.
- van Hage, W.R., Malaisé, V., De Vries, G.K.D., Schreiber, G., Van Someren, M., 2011. Abstracting and reasoning over ship trajectories and web data with the Simple Event Model (SEM). *Multimedia Tools and Applications* 57, 1–23.

- Van Heijst, G., Schreiber, A., Wielinga, B., 1997. Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies* 46, 183–292.
- Van Laere, J., Nilsson, M., 2009. Evaluation of a workshop to capture knowledge from subject matter experts in maritime surveillance, in: *12th International Conference on Information Fusion, 2009. FUSION '09*. Presented at the 12th International Conference on Information Fusion, 2009. FUSION '09, IEEE, pp. 171–178.
- Vandecasteele, A., Napoli, A., 2012a. Using Spatial Ontologies for Detecting Abnormal Maritime Behaviour, in: *OCEANS 2012 Proceedings*. Presented at the OCEANS 2012, Yeosu, Korea, p. 7.
- Vandecasteele, A., Napoli, A., 2012b. Enhancement of Ontology with Spatial Reasoning Capabilities to Support Maritime Anomaly Detection, in: *SOSE 2012*. Presented at the SOSE 2012, Genova, Italie.
- Varet-Pietri, M.-M., 2000. *L'ingénierie de la connaissance: la nouvelle épistémologie appliquée*. Presses Univ. Franche-Comté.
- Varzi, A.C., 2007. Spatial Reasoning and Ontology: Parts, Wholes, and Locations. *Handbook of Spatial Logics* 27, 945–1038.
- Wessel, M., Haarslev, V., Moller, R., 2000. Visual Spatial Query Languages: A Semantics Using Description Logic, in: *Diagrammatic Representation and Reasoning*. Springer-Verlag.
- Willems, C.M., 2011. Visualization of vessel traffic.
- Wolfson, O., Sistla, P., Xu, B., Zhou, J., Chamberlain, S., 1999. DOMINO: Databases fOr MovINg Objects tracking, in: *In ACM SIGMOD*. pp. 547–549.
- Wolfson, O., Xu, B., Chamberlain, S., Jiang, L., 1998. *Moving Objects Databases: Issues and Solutions*.
- Wu, Z., Palmer, M., 1994. Verbs semantics and lexical selection, in: *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, ACL '94*. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 133–138.
- Yan, Z., 2009. Towards Semantic Trajectory Data Analysis: A Conceptual and Computational Approach, in: *PhD Workshop*. Presented at the VLDB2009, Ecole Polytechnique Fédérale de Lausanne, Lyon, France, p. 3.
- Yan, Z., 2011. *Semantic Trajectories: Computing and Understanding Mobility Data (Phd)*.
- Yan, Z., Macedo, J., Parent, C., Spaccapietra, S., 2008. Trajectory Ontologies and Queries. *Transactions in GIS* 12, 75–91.
- Zheng, Y., Zhou, X., 2011. *Computing with Spatial Trajectories*. Springer.

Modélisation ontologique des connaissances expertes pour l'analyse de comportements à risque - Application à la surveillance maritime -

RÉSUMÉ : Dans nos sociétés où l'information est devenue omniprésente, la capacité à pouvoir capter et exploiter celle-ci est un enjeu majeur pour toute entité amenée à prendre une décision. En effet, de cette capacité dépendent ensuite les actions et les moyens d'interventions qui seront engagés. Mais, face à l'augmentation des données disponibles, au nombre croissant d'acteurs et à la complexification des menaces, l'environnement dans lequel évolue le décideur est marqué par une grande incertitude. Ce constat général se retrouve également au sein de la surveillance maritime qui se caractérise par des menaces hétérogènes, par une zone d'action très vaste et par un réseau important d'acteurs.

De ce fait, cette thèse se propose d'étudier les potentialités des ontologies spatiales à la fois comme objet de modélisation, de partage et d'inférence. L'objectif est de fournir aux experts du domaine un environnement adapté permettant la modélisation des connaissances relatives aux comportements anormaux de navires. Néanmoins, ces connaissances sont par nature spatio-temporelles. Il a donc été nécessaire d'étendre les fonctionnalités initiales du langage SWRL afin de prendre en compte ces caractéristiques.

Enfin, l'approche adoptée a ensuite été mise en application au sein du prototype *OntoMap*. De l'extraction des données, à l'analyse cartographique celui-ci offre les éléments nécessaires à la compréhension d'une situation anormale.

Mots clés : Ontologie Spatiale, Raisonnement Spatio-Temporel, Modélisation de comportements anormaux, Détection d'anomalie, Trajectoire sémantique

Ontological modeling of maritime expert knowledge for maritime behavior analysis - An application to the maritime domain -

ABSTRACT : In our societies, where information is everywhere the capacity to capture and analyze this information is a major issue for every entity that must take a decision. From this capacity depends the efficiency of the actions and measures that will be undertaken. But face with the large increase of available data, the increasing number of actors and the increasing complexity of threats, the strategic environment in which decision-makers operate is characterized by a significant uncertainty. This general observation is also present in the maritime domain characterised by heterogeneous threats, a large effective area and a large number of stakeholders.

Therefore, this proposed research explores the potentiality of spatial ontologies for modelling, sharing and inference. The purpose is to provide a suitable environment for the experts where knowledge about abnormal maritime behaviour can be modelled. However, this knowledge is spatio-temporal. It has therefore been necessary to extend the initial functionality of the SWRL language to take into account these characteristics.

Finally, the adopted approach had been implemented inside the *OntoMap* prototype. From retrieving the data to the cartographic analysis, this prototype offers all the necessary elements to understand an abnormal situation.

Keywords : Spatial Ontology, Spatio-Temporal reasoning, Modeling of Abnormal Behavior, Anomaly Detection, Semantic Trajectory