



HAL
open science

Contribution à l'extraction et à la représentation des connaissances de l'environnement maritime : proposition d'une architecture dédiée aux applications de navigation

Dieudonné Tsatcha

► To cite this version:

Dieudonné Tsatcha. Contribution à l'extraction et à la représentation des connaissances de l'environnement maritime : proposition d'une architecture dédiée aux applications de navigation. Modélisation et simulation. Université de Bretagne occidentale - Brest, 2014. Français. NNT : 2014BRES0118 . tel-02148222

HAL Id: tel-02148222

<https://theses.hal.science/tel-02148222>

Submitted on 5 Jun 2019

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.

THESE / UNIVERSITE DE BRETAGNE OCCIDENTALE

sous le sceau de l'Université européenne de Bretagne

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITE DE BRETAGNE OCCIDENTALE

Spécialité : Informatique

présentée par

Dieudonné TSATCHA

préparée dans l'Unité Mixte de recherche (EA n° 3634)
Institut de Recherche de l'École Navale

Ecole Doctorale des Sciences de la Mer

Contribution à l'extraction et à la
représentation des connaissances de
l'environnement maritime.
Proposition d'une architecture
dédiée aux applications de
navigation.

Thèse soutenue le 11 décembre 2014

devant le jury composé de :

Mme Bénédicte BUCHER

Docteur ès sciences (Hdr) – COGIT, Institut National de l'Information
Géographique et Forestière, Saint-Mandé / rapporteur

M. Mauro GAIO

Professeur – LIUPPA, Université de Pau et des Pays de l'Adour / rapporteur

M. Jean-Phillipe BABAU

Professeur – Lab-STICC, Université de Bretagne Occidentale / Président du
Jury

M. Alain BOUJU

Maître de conférences (Hdr) – L3i, Université de la Rochelle / examinateur

M. Christophe CLARAMUNT

Professeur – IRENAV, Ecole Navale, Brest / Directeur de thèse

M. Éric SAUX

Maître de conférences (Hdr) – IRENAV, Ecole Navale, Brest / examinateur

N° d'ordre : \NumeroOrdre

THÈSE

présentée à

L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE

Mention *Informatique - Géomatique*

par

Dieudonné TSATCHA

**Contribution à l'extraction et à la représentation
des connaissances de l'environnement maritime.
Proposition d'une architecture dédiée
aux applications de navigation**

soutenue le 11 Décembre 2014 devant la commission d'examen :

Composition du Jury

Rapporteurs : Dr. (Hdr) Bénédicte BUCHER - COGIT, IGN, Saint-Mandé
Pr. Mauro GAIO - LIUPPA, Université de Pau et des Pays de l'Adour

Examineurs : Pr. Jean-Phillipe BABAU - Lab-STICC, Président du jury, UBO, Brest
Mcf. (Hdr) Alain BOUJU - L3i, Université de la Rochelle

Directeur : Mcf. (Hdr) Éric SAUX - École Navale, Brest

Co-Encadrant : Pr. Christophe CLARAMUNT - IRENav, École Navale, Brest

Je dédie cette thèse à ma femme et mon fils

Remerciements

Les beaux moments de la vie ne durent jamais, pendant ces moments on découvre des personnes exceptionnelles avec qui on souhaite passer le reste de sa vie. Malheureusement, cela n'est pas possible car nous sommes toujours appelés à nous séparer à cause de nos ambitions personnelles. Cependant, il existe le mot « merci » qui permet de dire ou de rappeler haut et fort à toutes ces personnes l'importance qu'elles ont eu dans notre vie pendant les moments difficiles. En évitant toutefois de remercier toutes les personnes que j'ai rencontrées en un endroit précis de l'itinéraire emprunté au cours de ma vie, sinon même un livre entier ne pourra pas suffire pour leur présenter mes remerciements. Surtout que l'itinéraire de ma vie se compose d'un grand nombre de points d'intérêts et d'obstacles dans lequel j'ai rencontré des personnes formidables. Je vais juste m'intéresser aux points les plus saillants de ma vie ou j'ai été assisté par des personnes qui m'ont permis de continuer sans trébucher. C'est ainsi que pour celles qui ne seront pas citées dans ces remerciements, il n'en est de rien un oubli car je vous garde jalousement et précieusement dans mon cœur.

Tout commence un matin du 11 décembre 2014 à 12 heures 30 min, à l'école navale où j'apprends que j'étais devenu finalement docteur en géomatique avec la mention très honorable. Cette nouvelle m'a été annoncée par le président du jury de la soutenance M. Jean-Phillipe BABAU à qui je transmets mes remerciements les plus sincères de même qu'à tous les membres du jury : Mme Bénédicte BUCHER, M. Mauro GAIO et M. Alain BOUJU.

L'aboutissement d'une thèse ne pourrait se faire sans une équipe très compacte que je nomme le trio de combat (directeur, encadrant, thésard) où chaque entité joue un rôle important et irremplaçable. C'est dans ce contexte, que je salue l'aide précieuse que m'ont apportée mon directeur Mcf (Hdr) Eric Saux et mon co-encadrant le professeur Christophe Claramunt qui sont des personnes formidables et investies dans leur mission de recherche. C'est un héritage inestimable qu'ils m'ont transmis.

Je tiens aussi à remercier l'école navale et le ministère de la défense d'avoir mis à ma disposition des infrastructures adéquates pour la réalisation de cette thèse.

Cette thèse a été aussi possible par un cadre de vie remarquable. C'est dans cette initiative que je tiens à remercier tous les membres de ma famille qui m'ont toujours soutenu : ma femme Rosine Motouom Tamto, mon fils Eric Fomen Tsatcha, mes parents (Jean Fomen et Pauline Njemen et

que vos âmes reposent en paix et vos enfants ne vous oublieront jamais), ma grande sœur Sylvie Petgang, mes frères Bienvenu Tamen, Calvin Ngaha, Charles Noudja, Jules Njiké et mes sœurs Julienne Njamen, Doréthée Ngangué, Serge Irène Ngando et Nadine Sandjong, sans oublier les familles Tamen (à Douala), Tchounzou (à Yaoundé), Tamto (à Yaoundé), Fondjo (à Paris), Fongwo (à Douala), Tchouta (à Dusseldorf), Mbogla (à Yaoundé).

Je remercie aussi tous mes amis avec qui, j'ai fait de la collocation pendant mes études au Vietnam (Landry, Richard, Hervé) au Cameroun (Guy, Ghislain, Jean marie), en Ukraine (Serge, Ngomseu, etc.), mes amis d'enfance avec qui nous sommes toujours restés très proches (Blaise Kamdem, Hippolyte Kamdem, Nyonyo chimène, Josiane Kengné, Willy Chieukam, Michel Tchatchueng, Serge Poudeu), mes amis de l'Ecole navale (Georges (mon frère camer), Pierre-Luc, Lamia, Ines, Joseph, Nicolas, Clement, Ousmane, Wissame, Pierre, Marie-Coz (relectrice et la mère de l'IRENAV), Loic, Arnaud, Emery, Géraldine, Alexandra, Julien, Laurent Etienne, Fouté, Grégory, Ballack, Dossou, Brisco, Talla, etc.).

C'est également avec beaucoup de plaisir que je transmets mes remerciements à ceux qui m'ont permis de réaliser tous ces projets : Eric Afonsi, Souopgui (relecteur, USA), Chacode (relecteur, France) et ceux du Vietnam (Ly lu, Tuoi, Tri, Dinh, Zoan) et leurs familles pour leur hospitalité inestimable.

Afin d'aborder cette thèse avec beaucoup de sérénité, j'ai rencontré des personnes qui m'ont appris des choses précieuses. C'est dans ce contexte que je remercie le professeur Pham Thi Ha Duoug qui m'a initié dans la résolution des problèmes complexes, tout l'ensemble corps enseignant de l'Institut Française de l'Informatique au Vietnam (M. Moraru, M. Alain boucher, M. Quang, M. Vinh, etc.) et tous mes camarades de la promotion 14 à l'IFI.

Je ne saurai terminer ces remerciements sans vous dire que je vous aime tous et ne changer jamais.

Résumé

De nos jours, les applications informatiques autonomes sont au centre de grandes préoccupations de la recherche scientifique. Ces dernières sont destinées initialement à des systèmes d'aide à la décision dans des environnements contraints et dynamiques, communément appelés environnements complexes. Elles peuvent dès à présent, à l'aide des avancées de la recherche, permettre de construire et déduire leurs connaissances propres afin d'interagir en temps réel avec leur environnement. Cependant, elles sont confrontées à la difficulté d'avoir une modélisation fidèle du monde réel et des entités qui le composent. L'un des principaux objectifs de nos recherches est de capturer et modéliser la sémantique associée aux entités spatio-temporelles afin d'enrichir leur expressivité dans les SIG ou les systèmes d'aide à la décision. Un service de routage maritime dynamique a été déployé en exploitant cette modélisation. Cet algorithme a été démontré comme optimal en termes d'espace mémoire et de temps de calcul.

La sémantique capturée se compose de l'affordance et de la saillance visuelle de l'entité spatiale. Les connaissances associées à cette sémantique sont par la suite représentées par une ontologie computationnelle qui intègre des approches spatio-temporelles. Ces connaissances sont soit déduites du savoir de l'expert du domaine, soit extraites de gros volumes de données textuelles en utilisant des techniques de traitement automatique du langage.

L'ontologie computationnelle proposée nous a permis de définir un algorithme de routage maritime dynamique (fonction des événements ou objets présents dans l'environnement) fondé sur une heuristique itérative monocritère de plus courte distance et bidirectionnelle. L'algorithme *BIDA** proposé s'applique sur un graphe itératif qui est une conceptualisation d'une grille hexagonale itérative recouvrant la zone de navigation. Cet algorithme permet aussi la gestion de différents niveaux de résolution.

Toujours dans l'initiative de produire un modèle aussi proche que possible du monde réel, l'algorithme *BIDA** a été enrichi des stratégies multicritères afin de prendre en compte les différentes contraintes de la navigation maritime. Les contraintes globales et locales auxquelles nous nous sommes intéressés sont la profondeur des eaux, la distance de navigation et la direction de navigation.

Le modèle proposé permet ainsi d'enrichir les capacités cognitives des utilisateurs évoluant dans les environnements maritimes et peut aussi être utilisé pour construire des systèmes complètement

autonomes explorant ces environnements. Un prototype expérimental de navigation intelligente mettant en œuvre cette modélisation et proposant un service de routage maritime a été développé dans le cadre de cette thèse.

Mots clés : Intelligence artificielle, sciences de l'information géographique, ontologie, traitement automatique du langage, intelligence computationnelle, routage maritime, géométrie combinatoire, algorithmique combinatoire, complexité algorithmique, systèmes de gestion de bases de données (relationnelles et ontologiques).

Table des matières

Introduction	1
Plan du mémoire et guide de lecture	6
1 État de l'art	9
1.1 Représentation des entités spatio-temporelles dans les SIG	9
1.1.1 Dimension spatiale d'une entité	10
1.1.2 Dimension temporelle d'une entité	12
1.2 Intégration des approches intelligentes aux entités spatiales	17
1.2.1 Intelligence artificielle	18
1.2.1.1 Algorithmique et heuristique	19
1.2.2 Intelligence computationnelle	21
1.2.2.1 Apprentissage des connaissances	21
1.2.2.2 Systèmes intelligents dans l'acquisition des connaissances	23
1.2.2.3 Systèmes intelligents dans l'exploration des connaissances	24
1.3 Fusion des SIG et des approches intelligentes	24
1.3.1 Ontologie	25
1.3.2 Les formalismes de représentation	28
1.4 Conclusion	31
2 Ontologie computationnelle pour la représentation de la sémantique des entités spatio-temporelles	33

2.1	Enrichissement des entités spatio-temporelles par la sémantique issue du savoir d'un expert	34
2.1.1	Dimension sémantique de l'entité	34
2.2	Ontologies computationnelles pour l'environnement maritime	37
2.2.1	Modélisation d'une entité en «DL»	40
2.2.2	La norme «OGC» en logique de description	42
2.3	Formalisation de l'ontologie proposée	43
2.3.1	Formalisation de la dimension temporelle	46
2.3.2	Formalisation de la dimension spatiale	52
2.3.3	Formalisation de la dimension sémantique	53
2.3.4	Formalisation des changements d'une entité ou de sa perception	53
2.4	Implémentation de l'ontologie proposée	55
2.4.1	Les différents systèmes	55
2.4.2	Les règles en SWRL	58
2.5	Cas d'étude : la représentation du système de balisage	59
2.6	Conclusion	63
3	Enrichissement des entités spatio-temporelles à partir de techniques de TAL (Traitement automatique du langage)	65
3.1	Techniques de recherche d'informations	67
3.2	Extraction automatique de la sémantique associée à un concept	69
3.2.1	Vecteur conceptuel d'un mot	69
3.2.2	Vecteur conceptuel d'une phrase	72
3.3	Espace de décision pour l'extraction de la sémantique	74
3.3.1	Projection d'une phrase dans un espace de décision	74
3.3.2	Direction sémantique principale d'un concept	76
3.3.3	Cas d'étude	76
3.4	Utilisation des gloses dans la recherche de la sémantique	78
3.5	Conclusion	82
4	Service de routage dynamique et formalisation d'une trajectoire de navigation	85
4.1	Intérêt d'un routage approprié pour l'environnement maritime	86

4.2	Concepts fondamentaux des algorithmes basés sur des heuristiques	89
4.3	Choix du pavage ou maillage	93
4.3.1	Facteur de branchement	93
4.3.2	Nombre de nœuds contenus dans la solution	94
4.3.3	Nombre de nœuds dans une direction	94
4.3.4	Maillage hexagonal	97
4.4	Structure de données	103
4.5	Algorithme <i>BIDA*</i> sur un maillage hexagonal	109
4.6	Conclusion	115
5	Architecture du modèle proposé : application à la navigation sous contraintes	119
5.1	Les différents modules de l'architecture	121
5.1.1	Module de traitement automatique du langage	121
5.1.2	Module d'édition de l'ontologie	121
5.1.2.1	Activités maritimes	123
5.1.2.2	Carte marine	124
5.1.3	Module de stockage et d'utilisation des données	124
5.1.3.1	Serveurs de base de données	125
5.2	Prise en compte des contraintes de l'environnement	129
5.2.1	Contraintes globales	130
5.2.2	Contraintes locales	131
5.3	<i>BIDA*</i> pour des problèmes multi-constraints	133
5.3.1	Formalisation des différents critères	135
5.4	Conclusion	138
	Conclusion et perspectives	141
	Contributions	141
	Perspectives	144
	Bibliographie	145

Liste des figures

1	Exemple de sémantique des informations qui peuvent influencer les routes de navigation le jour (à gauche) et la nuit (à droite). Le jour, les entités sont identifiées par leur forme et couleurs, et la nuit par les signaux lumineux qu'elles émettent [Néméta, 2008].	3
2	Les trois domaines scientifiques intervenant dans cette thèse.	5
1.1	Le triangle du sens décrit les étapes de modélisation d'un concept et d'une entité.	10
1.2	Étapes d'évolution de l'entité dans sa dimension spatiale	11
1.3	Relations topologiques hiérarchisées du modèle <i>RCC8</i>	12
1.4	(a) Étapes d'évolution de l'entité dans sa dimension temporelle; (b) Ontologie de la dimension temporelle.	14
1.5	Relations temporelles de l'algèbre de [Allen et Hayes, 1990]	14
1.6	Illustration de la typologie des processus spatio-temporels	16
1.7	Illustration d'une entité spatio-temporelle	17
1.8	Différents domaines travaillant sur des systèmes intelligents. Extrait de [Russel et Norvig, 1995].	18
1.9	Illustration de l'organisation des problèmes <i>NP</i>	21
1.10	Structure d'un réseau neuronal artificiel	22
1.11	Illustration d'une conceptualisation de l'environnement maritime	26
1.12	Expressivité de la logique terminologique et de la logique fonctionnelle	29
1.13	Différentes catégories de formalisme	29
1.14	Expressivité de la logique de description programmable (DLP) et de la logique de <i>Horn</i>	31
2.1	Illustration des différentes catégories de la norme S57	39

2.2	Ontologie proposée pour une entité atomique ou élémentaire	45
2.3	Illustration des différentes bouées cardinales (Nord, Sud, Est, Ouest) vues de jour et de nuit	62
3.1	(a) Illustration des « synsets » du mot « port » générés à partir de l'application « visual thesaurus » ² et (b) Illustration des synsets extraits dans notre modèle, les cercles en vert représentent les « synsets » candidats de type nom, les cercles en jaunes représentent les « synsets » candidats de type adjectif.	71
3.2	Illustration de la structure d'un réseau lexical défini dans WordNet.	71
3.3	(Visualisation de l'espace de décision associé au concept (a) « anchorage area » (zone de mouillage) et (b) « cardinal buoy » (bouée cardinale).	78
4.1	Exemple de route proposée par le logiciel MaxSea où la trajectoire traverse la zone de terre compromettant ainsi la sécurité	88
4.2	(a) Graphe initial parcouru par l'algorithme (BSA^*) et (b) les deux partitions de l'algorithme de recherche se rencontrant au nœud n_2	91
4.3	(a) Partition avant après le « nipping » du nœud n_1 et (b) la partition arrière après « pruning » au nœud n_3	92
4.4	Facteur de branchement pour (a) une grille hexagonale, (b) une grille carrée, (c) une cellule non situé sur la diagonale d'une grille octile, (d) une cellule sur la diagonale d'une grille octile.	94
4.5	Exemple des déplacements optimaux sur les différents types de grille.	95
4.6	Les angles de déviation correspondant aux déplacements présentés à la figure 4.5.	95
4.7	Sélection des mailles optimales sur différentes grilles en fonction de la direction de déplacement.	97
4.8	Exemple de maillages parallèles M et M' sur des faces planaires. Ces maillages sont combinatoirement équivalents et les arêtes sont parallèles.	98
4.9	Schéma (a) d'un maillage hexagonal dans le système de coordonnées 4.1 et (b) d'un hexagone dans le système de coordonnées locales 4.2	99
4.10	Schéma d'un maillage hexagonal dans le système de coordonnées $\Gamma'(s, i', j')$	101
4.11	Différents faisceaux de propagation associés à leur faisceau symétrique correspondant.	102
4.12	Propagation utilisant les faisceaux symétriques.	103
4.13	Modélisation des cellules voisines dans un système de référence local (à gauche) ou global (à droite).	104

4.14	Illustration de la cellule racine dans le processus de modélisation.	105
4.15	Illustration de la modélisation du premier anneau.	106
4.16	Graphe correspondant à la structure de la figure 4.15.	106
4.17	Illustration des trois premières étapes dans le processus de modélisation du second anneau.	107
4.18	Illustration des trois dernières étapes dans le processus de modélisation du second anneau.	107
4.19	Graphe correspondant à la structure de la figure 4.17a.	108
4.20	Table de hachage illustrant les sommets déjà calculés par la structure de données au cours du processus de création de la cellule racine (i, j)	108
4.21	(a) Graphe initial avec les valeurs $h(n_i)$ associées à chaque nœud et (b) chemin solution.	110
4.22	Les deux premières étapes et le nœud sélectionné (rouge) utilisant l'algorithme IDA^*	110
4.23	Les deux dernières étapes et le nœud sélectionné (rouge) utilisant l'algorithme IDA^*	110
4.24	Différents types de multi-résolution (a) ouverture 3, (b) ouverture 4 et (c) ouverture 7 [Sahr, 2011].	112
4.25	Exemple d'un routage maritime itératif utilisant l'algorithme $BIDA^*$ sur une structure de données dynamique. Cas 1 : multi-résolution sur un maillage hexagonal en regroupement (ouverture 4).	115
4.26	Exemple d'un routage maritime itératif utilisant l'algorithme $BIDA^*$ sur une structure de données dynamique. Case 2 : résolution simple sur un maillage hexagonal.	116
5.1	Architecture globale du modèle.	122
5.2	Stratégie de recouvrement du R-tree et son arbre associé	127
5.3	Illustration de l'ensemble des points d'intérêt globaux formé uniquement des points s (en rouge) et t (en noir)	132
5.4	Illustration des points d'intérêt globaux issus des points de l'enveloppe convexe de la zone de non navigation.	132
5.5	(a) Illustration de la stratégie de routage suite à la rencontre de la bouée cardinale ouest de couleur jaune « Fillettes » et (b) la route finale solution.	137

- 5.6 (a) Illustration d'une route de navigation respectant l'ensemble des contraintes locales et globales de l'environnement. En bleu foncé, les zones d'accès restreint. . 138

Liste des tableaux

2.1	Premier pas vers la détermination d'un ensemble de facteurs génériques de la saillance [Landragin, 2004].	38
2.2	Syntaxe et sémantique pour la logique de description [Diego, 1996a]	44
2.3	Illustration du vecteur temporel et du domaine de définition de chacune de ses composantes	47
2.4	Comparaison des systèmes de raisonnement sur différentes caractéristiques <i>ALCQI</i>	56
2.5	Base de règles intégrée à notre base de connaissances	61
2.6	Illustration de l'agrégation d'une bouée cardinale Est vue de jour	62
2.7	Illustration de l'état courant d'une bouée cardinale Est vue de jour	62
2.8	Illustration de l'agrégation d'une bouée cardinale Est vue de nuit	63
2.9	Illustration de l'état courant d'une bouée cardinale Est vue de nuit	63
3.1	Désambiguïsation de la phrase « Le bateau est dans le port »	68
3.2	Désambiguïsation de la phrase « The ship is in the port »	73
3.3	Valeurs de projection du concept « low water » dans l'espace de décision (« back, stop, maneuver et continue »).	76
3.4	Valeurs de projection des phrases associées au concept de « zone de mouillage » dans un espace de décision ($V(\text{maneuver}), V(\text{stop}), V(\text{continue}), V(\text{back})$).	77
3.5	Valeurs de projection des phrases associées au concept de « bouée cardinale » dans l'espace de décision ($V(\text{maneuver}), V(\text{stop}), V(\text{continue}), V(\text{back})$).	78
3.6	Valeurs de projection des phrases associées au concept de « brouillard dense » dans l'espace de décision ($V(\text{maneuver}), V(\text{stop}), V(\text{continue}), V(\text{back})$) utilisant le concept de gloses.	79

4.1	Principales différences entre le routage maritime et le routage basé sur un réseau structuré	88
4.2	Complexité d' <i>IDA*</i> sur les différentes grilles. $D = l(u)$ et H correspond à l'effet de la stratégie heuristique sur l'algorithme.	96
4.3	Nombre de nœuds adjacents et valeur du facteur de branchement pour les différentes grilles.	97
4.4	Coefficients de variation entre les nœuds parent et fils.	104
4.5	Évaluation de la stratégie proposée par comparaison avec une stratégie de recherche fondée sur un maillage complet de l'environnement.	116
5.1	Différents types d'engins mobiles associés à leur environnement de navigation dédié.	120
5.2	Illustration des différents groupes de critères	134
3	Contribution des différents chapitres sur les modules de l'architecture du modèle proposé. 1) Module de traitement du langage, 2) Module d'édition de l'ontologie 3) Module de stockage et d'utilisation des données.	141

Introduction

L'exploration et la navigation dans des environnements contraints et complexes par des véhicules ou des engins autonomes, tels que l'environnement maritime, connaissent une évolution considérable. Un environnement est en particulier considéré complexe lorsqu'il comporte des éléments imprévisibles et dynamiques. Dans de tels environnements, les concepteurs de Systèmes d'Information Géographique (SIG) font face à des éléments dynamiques caractérisés par de nombreuses interactions et rétroactions. Les SIG devraient idéalement proposer des représentations intelligentes de l'environnement en intégrant des modèles spatio-temporels performant qui intègrent l'ensemble des connaissances de cet environnement. L'environnement qui nous intéresse dans le cadre de cette thèse est l'environnement maritime reconnu comme contraint et complexe de par ses règles de navigation, la densité du trafic, la diversité des engins y navigant (bateaux à moteur, voiliers, sous-marins, AUV³, etc.) et en raison des éléments dynamiques (courant, marée, vent, brouillard). Cette complexité de l'environnement maritime a de l'influence sur la sécurité, la visibilité et par conséquent les décisions à prendre au cours d'une navigation.

La sécurité de la navigation maritime a été un défi important pour la recherche et le développement au cours de ces dernières décennies même si beaucoup reste encore à faire. Les premiers challenges ont été initiés par les systèmes de transports intelligents «ITS»⁴ en proposant certaines solutions pour la visualisation (p. ex., navigation maritime en 3D⁵ [Goralski et Gold, 2008], l'analyse des comportements des navires à travers des flux de données AIS⁶), mais l'identification d'une approche de modélisation qui prenne en compte toutes les composantes de l'environnement maritime reste encore une tâche difficile à accomplir. Les principales difficultés proviennent de la complexité de l'environnement maritime, de ses entités et composantes dynamiques et de leurs interactions qui ont un impact direct sur la navigation.

En outre, les conditions de visibilité (brouillard, jour, nuit) peuvent influencer la perception des objets et par conséquent les actions de navigation à effectuer. La plupart des « ITS » essaient de réduire les risques de collision par l'amélioration de la représentation visuelle de l'environnement

3. Autonomous Underwater Vehicle

4. Intelligent Transportation Systems

5. <http://www.geovs.com/>

6. Automatic Identification System

[Claramunt *et al.*, 2005] ou par l'analyse des informations provenant de capteurs (GPS⁷, AIS, RADAR⁸, caméras infrarouges, LRIT⁹). Toutefois à notre connaissance, aucune de ces méthodes ne permet de faciliter la planification de la navigation. Dans ce travail, l'un des objectifs est de réduire la complexité de la représentation en interprétant les informations sémantiques générées par un ou plusieurs objets, et ceci à un moment et à un endroit précis dans un environnement donné. L'un des principaux résultats attendus de cette recherche est **une aide à la décision pour la navigation maritime fondée sur la sémantique des entités qui composent l'environnement, et ceci afin de proposer aux marins des routes de navigation qui soient sécurisées.**

L'intérêt de l'apport d'un processus d'aide à la décision est d'améliorer de façon significative les capacités cognitives du marin pendant des situations de stress élevé et de surcharge de travail. Dans ce contexte, nous supposons qu'une route satisfait la contrainte de sécurité de navigation si elle prend en compte la profondeur du relief sous-marin mais aussi la sémantique ou l'affordance des entités autour du navire pendant son déplacement. Une affordance est une qualité d'un objet ou d'un environnement qui permet à un individu d'améliorer l'action à effectuer [Gibson, 1977]. Les actions telles que continuer, attendre, manœuvrer, revenir en arrière sont les différentes actions que le marin est amené à prendre tout au long de la navigation. Cela implique de prendre en compte les zones réglementées, l'effet des courants, les signaux radio, le vent, etc. Compte tenu des connaissances qui proviennent des événements extérieurs, des comportements des objets situés à proximité du navire et de celles détectées par les capteurs, un marin doit rapidement obtenir des informations directionnelles quant à la route à suivre pour prendre la décision la plus appropriée. L'une des difficultés pour l'identification des objets par les marins est de tenir compte de leur saillance [Landragin, 2004]. La saillance est considérée comme la propriété permettant à une entité d'être différenciée de celles qui l'entourent. Par exemple, une bouée peut être perçue différemment par bonne ou mauvaise visibilité (la nuit ou lors de la présence de brume) (cf. figure 1). Ainsi, les systèmes d'aide à la navigation ont besoin de nouvelles techniques de stockage et de modélisation afin de tenir compte de l'ensemble des connaissances sur les entités, leurs propriétés et leurs relations.

La présence des ECDIS¹⁰ et des ENC¹¹ fournissent aux marins une aide précieuse. Les ECDIS sont des systèmes électroniques capables de visualiser la position géographique et les informations relatives aux entités d'une carte marine. Ils permettent de remplacer les cartes papier traditionnelles afin d'assurer la sécurité des usagers en mer. A ce jour, pour améliorer la sécurité en mer, les ECDIS intègrent et peuvent être couplés à diverses sources de données (ENC, GPS, AIS, RADAR, SONAR¹², gyroscope¹³, etc.) mais souvent aux formats différents. Ces diverses sources d'informa-

7. Global Positioning System

8. Automatic Identification System, RAdio Detection And Ranging

9. Long Range Identification and Tracking

10. Electronic Chart Display and Information System

11. Electronic Navigational Chart

12. SOund Navigational and RAnging

13. Un gyroscope est un instrument de navigation indiquant le nord géographique, en se basant sur l'effet gyroscopique. L'axe de rotation du gyroscope s'aligne ainsi sur l'axe des pôles.

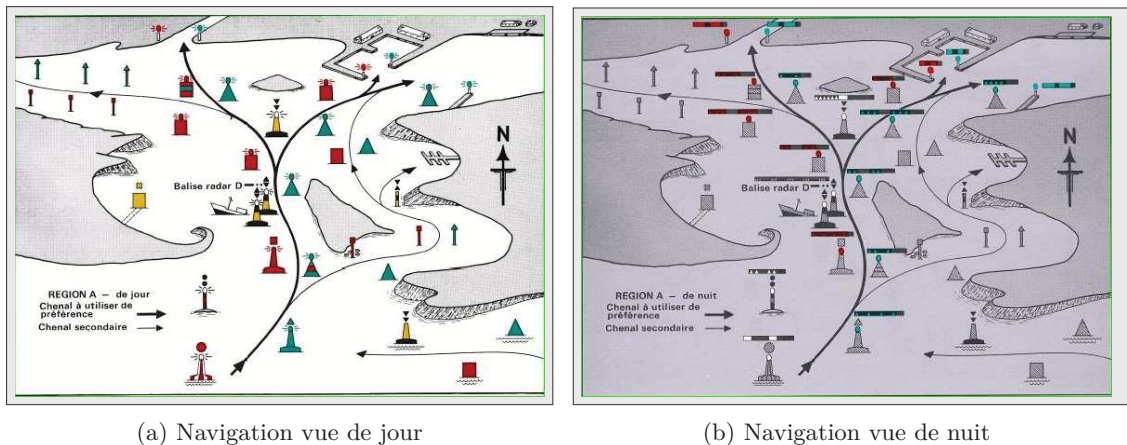


Figure 1 – Exemple de sémantique des informations qui peuvent influencer les routes de navigation le jour (à gauche) et la nuit (à droite). Le jour, les entités sont identifiées par leur forme et couleurs, et la nuit par les signaux lumineux qu’elles émettent [Néméta, 2008].

tion doivent être interprétées et manipulées par le marin en temps réel pour accomplir sa mission de navigation, une tâche qui reste jusqu’à ce jour difficile d’autant plus que certaines connaissances sont transcrites dans de gros volumes de données textuelles. C’est le cas notamment des règles de navigation définies dans le Règlement International pour Prévenir les Abordages en Mer (RIPAM) et des aides à la navigation contenues dans les instructions nautiques et les pilotes côtiers. Ces dernières constituent des aides précieuses puisque qu’elles décrivent un ensemble d’information et de procédures nautiques utiles (mais le plus souvent non obligatoires) en navigation côtière. Voici un exemple d’instructions :

« *DANGERS.* - Au milieu du Goulet de Brest et parallèlement à son axe, une chaîne de dangers divise le goulet en une Passe Nord et une Passe Sud draguées respectivement à 11 m et 12 m (drague hydrographique).

Le danger le plus au SW est le Plateau des Fillettes, marqué à l’Ouest par la bouée «Fillettes», cardinale Ouest lumineuse, et au Sud par la bouée «Kerviniou», latérale bâbord lumineuse. Au centre de la chaîne, la Basse Goudron est également balisée par une bouée latérale bâbord lumineuse. A l’extrémité NE de la chaîne, la Roche Mengam porte une tourelle de danger isolé, lumineuse, surmontée d’une éolienne. Le Plateau des Fillettes est couvert par le secteur rouge (260° - 307°) du feu du Petit Minou et celui (034° - 054°) du feu de la tourelle «Mengam».

INSTRUCTIONS. - Pour pénétrer dans la Passe Nord du Goulet, de jour, suivre l’alignement du château d’eau de Plougastel par la tourelle «Mengam» à 074,5°. Suivre ensuite l’axe de la passe.

La direction de la Passe Sud est donnée par le relèvement à 047,5° du phare du Portzic (axe du secteur blanc [045° - 050°] du feu auxiliaire directionnel du phare du Portzic). »

La dispersion des règles ou des connaissances dans de tels documents rend alors difficile la prise de

décision pendant les moments critiques.

L'environnement maritime est aussi de plus en plus sollicité pour diverses manifestations ou activités nautiques (jeux, sports nautiques, pêche, surveillance maritime, etc.). On y compte alors un nombre important d'utilisateurs et d'installations ou d'infrastructures artificielles. Cette croissance des entités ou d'objets qui y sont installés a un impact fort sur la sécurité car elle modifie considérablement le trafic et le déplacement des biens et des personnes.

Le fait de proposer ou de permettre l'affichage des informations pertinentes à un utilisateur, pouvant éventuellement tenir compte de son profil et de son activité, tout en respectant la privation des zones personnalisées ou restreintes (p. ex., zones d'essai militaire ou zones d'attente pour les cargos) est inexistant dans les systèmes de navigation actuels. De la même façon, les informations ou connaissances qui sont contenues sur une carte numérique ne sont généralement pas suffisantes afin d'avoir une représentation de l'environnement et une exploitation des informations aussi riches que possible et les plus réelles possibles. La saillance de l'entité est une des caractéristiques qui n'est pas prise en compte, jusqu'à ce jour, dans les applications maritimes. La saillance est avant tout visuelle dans le contexte de la navigation, mais elle est également structurelle et sémantique [Sorrows et Hirtle, 1999]. Elle permet d'améliorer l'action de navigation à réaliser face à celle-ci.

Suite à toutes ces insuffisances ou imperfections dans les systèmes existants, cette thèse a pour objectif de contribuer à l'extraction et à la représentation des connaissances de l'environnement maritime afin de proposer une architecture dédiée aux applications de navigation. Notre approche est à l'intersection de trois grands domaines de recherche, à savoir :

- les sciences de l'information géographique,
- l'intelligence artificielle,
- l'intelligence computationnelle.

La figure 2 est une illustration de la répartition de mes travaux dans ces trois domaines de recherche, où la gestion de la connaissance apparaît comme l'activité principale à modéliser, composante souvent absente dans les systèmes précédemment proposés. Dans le cadre de la thèse, une gestion efficace de la connaissance met en évidence les différentes thématiques décrites ci-après.

L'extraction de la connaissance consiste à rechercher de façon ciblée un ensemble de connaissances appropriées à un domaine à partir de gros volumes de données textuelles. Une telle extraction peut se déployer de façon automatique ou manuelle. Un déploiement automatique s'opère en particulier par des programmes basés sur le traitement automatique du langage (TAL) [Tsatcha *et al.*, 2013b]. Pour l'extraction manuelle, il appartient à un ensemble d'experts d'identifier les concepts d'un domaine et d'en établir des relations.

La structuration ou modélisation de la connaissance devra garantir la cohérence et la consistance des données ou des instances. Plusieurs modèles de structuration existent de nos jours :

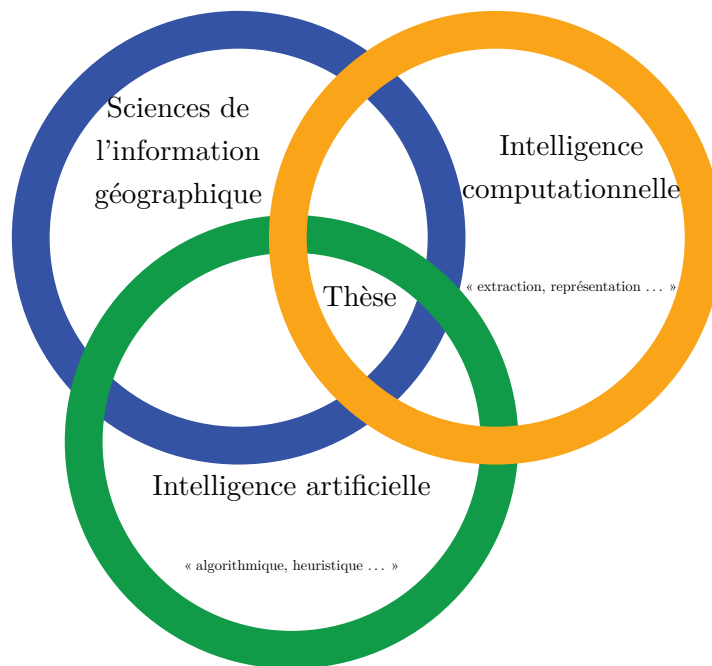


Figure 2 – Les trois domaines scientifiques intervenant dans cette thèse

- les structures entité-association,
- les structures orientées objet,
- les structures à base d'ontologie.

Nous avons utilisé une structure fondée sur des ontologies car elles offrent des formalismes et des approches permettant de raisonner, d'inférer des connaissances. Le raisonnement et l'inférence sur la connaissance permettent de produire de nouveaux concepts et d'intégrer de la sémantique. Elle propose également une structuration qui permet de mettre ensemble, dans une même application, les trois domaines illustrés par la figure 2.

La formalisation de la connaissance vient à la suite d'une structuration. Une question essentielle est le choix d'un formalisme dans lequel la connaissance doit être représentée. Plusieurs formalismes exploitant des ontologies sont possibles tels que les réseaux sémantiques, les graphes conceptuels, les *frames* [Minsky, 1974], la logique de description et la logique de description programmable [Schmidt-Schauß, 1989]. Nous utiliserons dans ce travail un formalisme fondé sur la logique de description programmable (DLP¹⁴). Il permet d'intégrer au formalisme défini des règles pour déterminer et produire de nouveaux concepts.

Le stockage et l'exploitation de la connaissance doit permettre d'utiliser de façon efficace la connaissance. En considérant que la connaissance nous aide à prendre des décisions sur des données qui sont considérées comme des informations que nous collectons et conservons, nous avons choisi un système de gestion de base de données fondé sur la technologie « triple store »¹⁵

14. Description Logic Programming

15. est une technologie permettant de représenter une proposition par des triplets (sujet, prédicat et objet)

[Sintek *et al.*, 2002]. Ce système permet de réaliser des requêtes avec un temps de réponse réduit et d'intégrer des moteurs et des raisonneurs sur les connaissances.

Les thématiques décrites précédemment nous ont permis de réaliser un prototype de routage maritime en temps réel. Il permet de trouver une route de navigation en temps réel tout en préservant la sécurité du marin et celle des usagers de l'environnement maritime. Ce routage s'applique sur une structure de données spatiales définie à partir d'un maillage bidirectionnel et itératif, basé sur des hexagones. Il permet de trouver une route optimale entre deux positions géographiques en tenant compte de la complexité de l'environnement maritime. L'optimalité de la route est fondée sur la prise en compte de plusieurs contraintes de navigation. Dans l'application proposée, nous considérons les contraintes globales de profondeur des eaux et de longueur de trajectoire ainsi que la contrainte locale de direction de navigation induite par la sémantique des entités proches.

Plan du mémoire et guide de lecture

Ce mémoire de thèse est organisée en cinq chapitres définis comme suit.

Chapitre 1 : État de l'art. Ce chapitre détaille le contexte de notre travail en mettant en évidence les derniers résultats proposés dans les domaines des sciences de l'information géographique, de l'intelligence artificielle et de l'intelligence computationnelle.

Chapitre 2 : Ontologie computationnelle pour la représentation de la sémantique. Ce chapitre propose une ontologie computationnelle utilisée pour formaliser et structurer les connaissances de l'environnement maritime. Dans ce chapitre, les connaissances structurées proviennent de l'expert et sont représentées par des triplets (sujet, prédicat, objets) sous un format OWL.

Chapitre 3 : Enrichissement des entités spatio-temporelles à partir des techniques de TAL (Traitement automatique du langage). Ce chapitre présente une stratégie d'extraction automatique de la sémantique associée aux entités spatio-temporelles dédiées à la navigation maritime à partir de documents textuels. La stratégie est fondée sur la notion de vecteur conceptuel. Dans ce chapitre, nous proposons différents cas d'étude afin d'illustrer cette stratégie.

Chapitre 4 : Service de routage dynamique et formalisation d'une trajectoire de navigation. Un des services couramment utilisé dans ces environnements complexes est le routage. Dans l'objectif d'exploiter la connaissance précédemment structurée et l'architecture fonctionnelle décrite dans le chapitre ci-après, nous avons proposé un algorithme de routage maritime bidirectionnel, dynamique (en fonction des évènements proches) et basé sur des hexagones. Ce chapitre décrit une structure de données modélisant la trajectoire couplée à une heuristique qui permet de rechercher des chemins optimaux en temps réel.

Chapitre 5 : Architecture du modèle dédiée aux applications de navigation. Ce chapitre propose un modèle de représentation et d'exploitation des entités dans les trois dimensions spatiale, temporelle et sémantique. Son architecture permet d'associer un module de traitement automatique du langage, un module de stockage de l'ontologie computationnelle et un module de stockage des données. Cette architecture offre un couplage simplifié et une interopérabilité. Nous utilisons des serveurs de données fondés sur une technologie « triple store ». Ces serveurs offrent un système de stockage ayant une architecture d'indexation de type Rtree [Beckmann *et al.*, 1990]. Les serveurs de données que nous avons utilisés sont Virtuoso¹⁶ et Parliament¹⁷. A l'aide du système « Jena » que ces serveurs intègrent, nous pouvons les utiliser pour lier les données « linked data » et les héberger sur différents serveurs de bases de données en appliquant des requêtes en local et à distance. L'architecture ainsi proposée permet de réduire de façon considérable le temps de réponse des requêtes sur des données spatio-temporelles par des langages de requête appropriés. La réduction de ce temps de réponse, nous a ainsi permis d'améliorer l'algorithme de routage en temps réel en considérant différentes contraintes globales et locales associées à l'environnement maritime.

16. <http://virtuoso.openlinksw.com/>

17. <http://parliament.semwebcentral.org/>

Chapitre 1

État de l'art

Sommaire

1.1 Représentation des entités spatio-temporelles dans les SIG	9
1.1.1 Dimension spatiale d'une entité	10
1.1.2 Dimension temporelle d'une entité	12
1.2 Intégration des approches intelligentes aux entités spatiales	17
1.2.1 Intelligence artificielle	18
1.2.2 Intelligence computationnelle	21
1.3 Fusion des SIG et des approches intelligentes	24
1.3.1 Ontologie	25
1.3.2 Les formalismes de représentation	28
1.4 Conclusion	31

LA réalisation d'applications « intelligentes » dédiées à la navigation dans des environnements larges et fortement contraints reste jusqu'à ce jour un challenge important de la recherche. Ce challenge est essentiellement dû à la dynamique des entités qui y résident et aux interactions entre ces entités. Dans ce contexte relativement complexe, il paraît logique d'aborder les domaines de l'intelligence artificielle et de l'intelligence computationnelle afin d'établir et de développer une approche intégrée avec celles issues des SIG. La section 1.1 décrit les différentes approches abordées dans les SIG pour représenter des entités dans leurs dimensions spatiale et temporelle. La section 1.2 analyse les approches issues de l'intelligence artificielle et de *l'intelligence computationnelle*. La section 1.3 étudie un modèle de représentation de connaissances basé sur un formalisme intégrant des concepts intelligents et les SIG.

1.1 Représentation des entités spatio-temporelles dans les SIG

De nos jours, la représentation et l'exploitation des entités dynamiques sont encore des problèmes impliquant plusieurs domaines scientifiques qui s'intéressent à la représentation des phénomènes.

Dans le cas des SIG, au cours de ces deux décennies écoulées, les chercheurs se sont particulièrement intéressés à l'intégration de la connaissance temporelle des entités dynamiques dans les applications [Egenhofer, 1989, Freksa, 1992, Langran, 1992, Claramunt *et al.*, 1997a]. Cette section explore les différentes stratégies développées dans ce cadre pour représenter les entités spatiales à l'aide de deux composantes essentielles, l'espace et le temps.

1.1.1 Dimension spatiale d'une entité

La modélisation de la spatialité est depuis longtemps un challenge important dans le domaine des SIG. Initialement, les travaux représentaient une entité spatiale à l'aide de son empreinte spatiale [Goodchild, 1990]. Dans cette thèse, une entité est considérée comme une instance d'un concept. Un concept représente une abstraction d'un élément du monde réel comme illustré sur la figure 1.1, pouvant être modélisé à partir de la règle du triangle du sens [Ogden et Richard, 1923]. Le triangle du sens associe à toute entité ou objet du monde réel, une conceptualisation ou un langage pouvant l'identifier de façon unique dans un modèle.

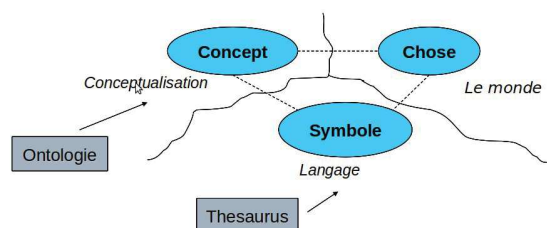


Figure 1.1 – Le triangle du sens décrit les étapes de modélisation d'un concept et d'une entité

L'empreinte spatiale d'une entité est considérée comme l'espace directement associé à cette entité. Elle a été longtemps utilisée dans la représentation des parcelles de terrain, rivières et bâtiments [Spéry *et al.*, 2001][Geraldine *et al.*, 2010]. Les empreintes spatiales pouvaient être assimilées à des surfaces, des points et des lignes pour leur représentation effective [Geraldine *et al.*, 2010]. Dans le domaine des SIG, deux techniques de représentation sont déployées pour modéliser une entité spatiale. La première technique est la représentation en vision raster où l'on structure l'espace en un certain nombre de pixels (c.-à-d. l'espace est vu comme une grille¹ et l'empreinte spatiale d'une entité est délimitée par un ensemble de pixels de cette grille) [Couclelis, 1992]. La deuxième technique utilise la vision vecteur où les données spatiales sont représentées par des primitives géométriques où l'empreinte spatiale peut être associée à des points, des lignes et des polygones. Dans cette thèse, ces deux techniques de représentation sont utilisées et il existe un ensemble de transformations qui permettent de passer d'une représentation à une autre. La vision raster nous permet de représenter par exemple les champs de vecteurs (courant, vent, etc.) tandis que

1. Une séquence régulière (pixels) ou irrégulière

la deuxième permet de représenter par exemple la géométrie des entités. Cette présentation en vectoriel de la géométrie de l’empreinte spatiale est faite de façon précise et les relations spatiales sont mieux évaluées.

Nous remarquons que ces seules informations ne permettent pas de représenter les interactions spatiales qui peuvent exister entre deux entités. Pour mieux représenter ces interactions, les relations spatiales ont été introduites (voir figure 1.2).

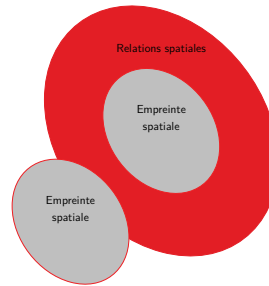
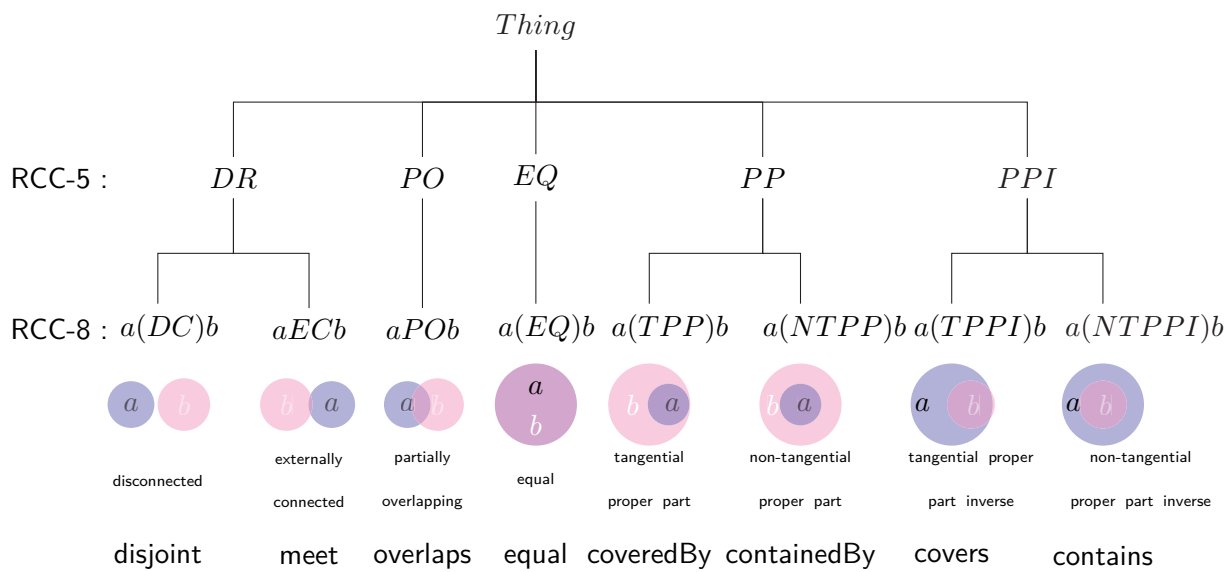


Figure 1.2 – Étapes d’évolution de l’entité dans sa dimension spatiale

Une relation spatiale permet de décrire la disposition entre un objet (simple ou complexe) et un autre en utilisant des éléments géométriques. Il existe quatre types de relations spatiales : la relation topologique, la relation méréologique, la relation d’orientation et la relation de proximité. Elles peuvent être décrites par des approches qualitatives et quantitatives. L’approche quantitative permet d’exprimer les relations spatiales à partir de critères géométriques. La distance euclidienne est l’un des exemples qui permet de quantifier la relation de proximité. Les critères non géométriques éventuellement fondés sur la perception humaine sont utilisés pour exprimer les approches qualitatives. Les relations dites non géométriques sont utilisées pour déterminer les propriétés qualitatives de l’espace. Les termes « près de », « loin de », « à une distance maximale » introduites par [Borrmann *et al.*, 2009] sont également utilisés pour exprimer une relation de proximité entre les entités. Les relations d’orientation réfèrent à la position d’un objet par rapport à l’observateur ou à un autre objet (p. ex. derrière, devant, au dessus, au nord, au sud, etc.). Selon le référentiel, elles peuvent être allocentriques ou extéro-centrées² ou égocentrique ou auto-centré³. Les relations topologiques ont été initialement proposées par [Egenhofer, 1989] et également décrites dans le modèle *RCC8* (*Region connection calculus*) par [Randell *et al.*, 1992] sous forme de 8 relations topologiques (Figure 1.3). La représentation de *RCC8* sous forme hiérarchique (Figure 1.3) permet, en les ordonnant, de caractériser les liens topologiques qui existent entre les relations. Les relations méréologiques sont des relations de parties ou de sous-ensembles d’un objet « partOf » [Motschnig-Pitrik et Kaasbøll, 1999].

2. Le référentiel n’est pas l’axe du corps

3. Le référentiel est l’axe du corps ou le méridien visuel vertical

Figure 1.3 – Relations topologiques hiérarchisées du modèle *RCC8*

Pour que cette représentation du monde réel soit la plus fidèle possible, il est important de s'intéresser à la dimension temporelle de l'entité. L'observation de l'entité dans son environnement exige de l'étudier aussi dans sa dimension temporelle.

1.1.2 Dimension temporelle d'une entité

Les transformations et les déplacements subis par des entités spatiales au cours de leur cycle de vie, de la naissance à la disparition, implique de considérer le temps comme une dimension pertinente. [Claramunt et Thériault, 1995] considèrent le temps comme un paramètre très important dans les SIG permettant entre autres, une analyse des phénomènes dynamiques du monde réel. Depuis les années 1990, l'un des challenges communément rencontrés dans les SIG est l'intégration de l'espace et du temps dans les applications.

Bien que les notions d'espace et de temps soient très familières aux humains depuis leur enfance, trouver une représentation satisfaisante reste toujours difficile. Il existe cependant des analogies entre le temps et l'espace qui sont décrites ci-dessous [Raza, 2001] :

- un objet est à un endroit unique à un moment donné,
- un objet peut être dans plusieurs endroits (apparition et disparition) à différents moments.

Par ailleurs, l'espace et le temps présentent aussi des différences. Quelques unes de leurs différences sont énoncées ci-dessous :

- le retour en arrière n'est pas possible dans le temps, un espace peut être traversé plusieurs fois dans le temps,

- le temps est dynamique et son mouvement est constant, un espace est statique car une place reste inchangée,
- le temps paraît souvent un ouvert fermé (non borné) et l'espace est au moins fermé ou borné [Easterfield *et al.*, 1991],
- Le temps a un ordre naturel ordonné dans le passé, le présent et le futur.

L'espace et le temps sont tous deux dotés d'une métrique permettant de les quantifier. Malgré ces différences, plusieurs travaux ont montré que la dimension temporelle est topologiquement équivalente à la dimension spatiale [Langran et Chrisman, 1988]. [Parkes et Thrift, 1980] en déduisent que les notions de temps et d'espace sont fortement corrélées et sont mêmes indissociables dans les SIG. Deux interprétations majeures sont utilisées pour décrire le temps. La première interprétation couramment appelée *le temps valide* est celle qui est propre au monde réel. Elle est différente de la deuxième interprétation dénommée *le temps transactionnel*. Le temps transactionnel indique la période de temps à laquelle un fait a été inséré dans une base de données. Cette période de temps est différente de la période au cours de laquelle le fait s'est produit dans le monde réel [Ahn et Snodgrass, 1986]. Les deux interprétations sont utilisées conjointement dans les SIG (modèle bi-temporel) afin de permettre une représentation harmonieuse [Worboys, 1994]. La modélisation du temps est matérialisée à l'aide d'une droite orientée sans début, ni de fin. Sur cette droite, les états temporels sont matérialisés par des points (c.-à-d., états sans durée comme les instants) ou par des intervalles (p. ex., les périodes). Ces deux états temporels (instant et période) constituent la signature temporelle d'une entité (voir figure 1.4(a)).

[Raza, 2001] montre que les opérations temporelles constituent un isomorphisme aux relations spatiales. Ces opérations peuvent être utilisées à l'aide d'une métrique ou d'un concept topologique. Les différentes opérations sont définies par [Allen et Hayes, 1990], où les auteurs montrent qu'une algèbre sur les instants peut tout à fait se dériver de celles des intervalles comme illustré sur la figure 1.5.

Pan [Pan, 2005] a proposé des relations d'agrégation temporelles 1.4(b). Elles permettent de représenter les séquences ou répétitions d'événements d'une entité (chaque jour, tous les n jours, secondes, mois, etc.). Un événement peut se produire de façon instantanée ou périodique (voir la figure 1.4(b)). L'intégration de l'agrégation temporelle dans notre modèle nous permet de représenter les phénomènes qui se produisent en cycle (les marées, la signalisation des bouées la nuit, etc.). La représentation de l'agrégation temporelle décrite par [Pan, 2005] est basée sur la description de l'horloge et du calendrier définis dans l'ontologie « DAML »⁴. Bien que le temps et l'espace aient

4. Darpa Agent Markup Language, est le nom du programme de recherche sur des langages par balises « xml, rdf et owl » lancé par l'agence de la défense américaine pour les recherches sur des projets avancés

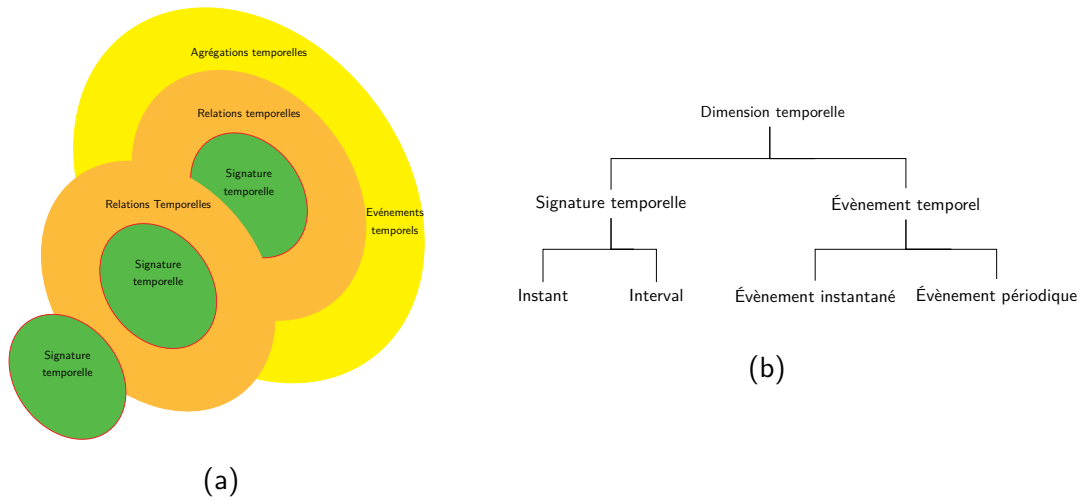


Figure 1.4 – (a) Étapes d'évolution de l'entité dans sa dimension temporelle; (b) Ontologie de la dimension temporelle

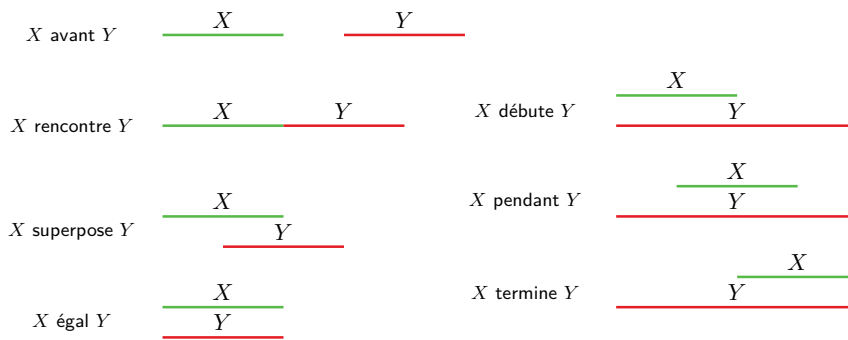


Figure 1.5 – Relations temporelles de l'algèbre de [Allen et Hayes, 1990]

des différences dans la représentation, il est judicieux de toujours les manipuler de façon transparente dans une même application. Certains modèles de représentation permettent la fusion des deux et sont présentés comme modèles spatio-temporels.

Modèles spatio-temporels d'une entité

Un modèle spatio-temporel est utilisé pour la représentation des phénomènes dynamiques des entités. Les modèles les plus répandus sont :

- Les modèles basés sur une représentation implicite des phénomènes spatio-temporels,
- les modèles basés sur le changement.

Les modèles basés sur une représentation implicite des phénomènes spatio-temporels ont été influencés par quatre approches :

- 1 les approches proposées dans les bases de données temporelles,

- 2** les approches de simulation,
- 3** les approches typiquement SIG [Langran, 1992],
- 4** les approches issues du raisonnement spatial qualitatif.

Les approches proposées dans les bases de données temporelles font parties des premiers travaux étudiés sur la modélisation de la composante temporelle pouvant avoir un impact pour les SIG [Tansel *et al.*, 1993]. Ces premiers travaux permettaient d'introduire de l'information temporelle dans une base de données relationnelle [Snodgrass et Ahn, 1986] [Gadia et Vaishnav, 1985]. Les approches de simulation permettent d'utiliser les techniques de simulation pour modéliser les phénomènes spatio-temporels et mettre en évidence l'aspect dynamique dans les SIG. Les approches typiquement SIG mettent en œuvre des requêtes pour des bases de données spatio-temporelles. Les approches basées sur un raisonnement qualitatif sont des travaux issus d'un raisonnement spatial qualitatif dans lesquels les changements ne sont pas explicitement représentés. Les changements qu'elles modélisent sont ceux qui influencent les relations spatiales mentionnées précédemment (relation topologique, relation de proximité, relation d'orientation, relation méréologique) [Cohn et Hazarika, 2001]. Cette approche se base sur le fait que les changements sont continus et utilise les modèles formels pour les représenter.

Les modèles basés sur le changement sont ceux qui tendent à se rapprocher de la réalité. Dans leur modélisation, l'objet est la primitive de base qui est une entité définie dans un espace [Bonnet, 1989]⁵. En considérant que les modèles de cette famille sont proposés par la communauté de recherche en *SIG* et en base de données spatio-temporelles, le terme objet fait référence à un objet géographique. Les changements proposés dans ces modèles sont ceux qui affectent l'objet géographique et non les séquences temporelles décrivant les objets du monde. Cette approche propose une stratégie de modélisation qui s'intéresse aux changements des relations et de leurs attributs. Les modèles les plus connus sont ceux proposés par [Hornsby et Egenhofer, 2000] qui se basent sur les changements de l'identité. A partir de ces approches, une identité est associée à une seule entité spatiale. Plusieurs contributions ont été apportées à ces approches sur les changements au cours des années 90's en proposant de nouveaux modèles comme ceux proposés par [Claramunt et Thériault, 1995] [Claramunt *et al.*, 1997a] [Claramunt *et al.*, 1997b]. Dans ces contributions, les auteurs présentent les changements comme un processus qui décrit l'évolution d'une entité spatiale vers un autre état qui lui est postérieur. Ces derniers mettent aussi en évidence une typologie de processus spatio-temporels (« STP »⁶) présenté à la figure 1.6. Ces processus sont classés selon trois familles :

- 1** les STP qui modifient une seule entité (comme par exemple la déformation, le changement de forme, l'expansion et le changement de taille),
- 2** les *STP* qui modifient les relations fonctionnelles entre plusieurs entités (comme par exemple

5. À ne pas confondre avec le terme « objet » issu des techniques de programmation orientée-objet. L'objet fait ici référence à la représentation d'un phénomène réel dans un système d'information.

6. Spatio Temporal Processes

la succession et la transmission) ,

3 les *STP* qui modifient les structures spatiales de plusieurs entités (comme par exemple l'union).

Toujours en restant dans le contexte de la navigation maritime, c'est cette approche basée sur des changements de l'objet géographique que nous avons modélisée dans cette thèse. Nous nous basons sur des *ontologies computationnelles* présentées au chapitre 2.1. L'intérêt de cette modélisation est d'extraire la sémantique pour garantir la sécurité des usagers de l'environnement maritime au cours de la navigation. La figure 1.7 est l'illustration d'une entité spatio-temporelle avec toutes ses différentes propriétés. C'est ainsi que nous l'avons modélisée.

L'intégration du temps et de l'espace dans un système exige de les traiter simultanément afin de produire un système aussi proche que possible du monde réel. Cette intégration a permis de s'intéresser à la fois aux dimensions spatiale et temporelle en proposant deux grandes catégories de modèles. La première catégorie s'intéresse à la représentation implicite des phénomènes spatio-temporels et la deuxième s'intéresse aux changements subis par des entités. Cependant, les modèles spatio-temporels ont besoin d'intégrer d'autres approches dites intelligentes pour représenter l'entité dans son environnement et d'enrichir les dimensions spatiale et temporelle par la prise en compte de la dimension sémantique. La section suivante analyse les différentes approches utilisées pour intégrer et rendre les systèmes intelligents.

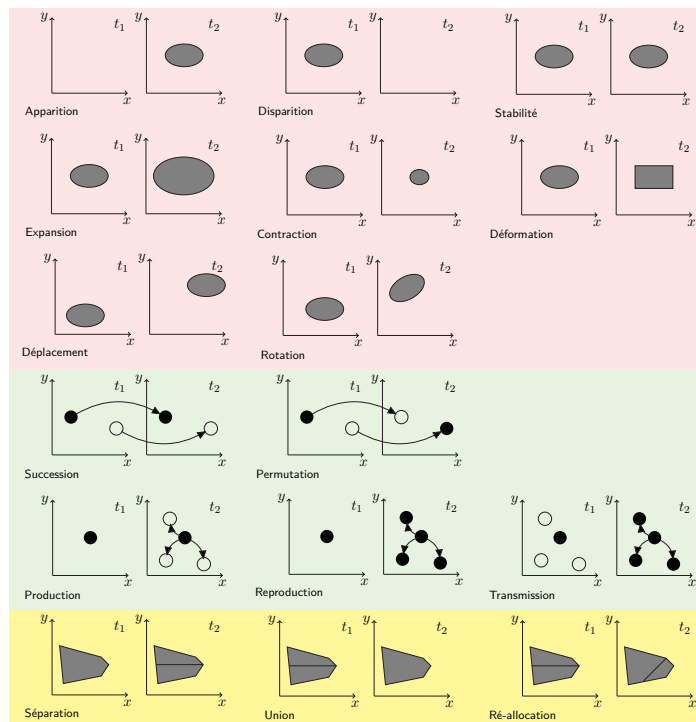


Figure 1.6 – Illustration de la typologie des processus spatio-temporels

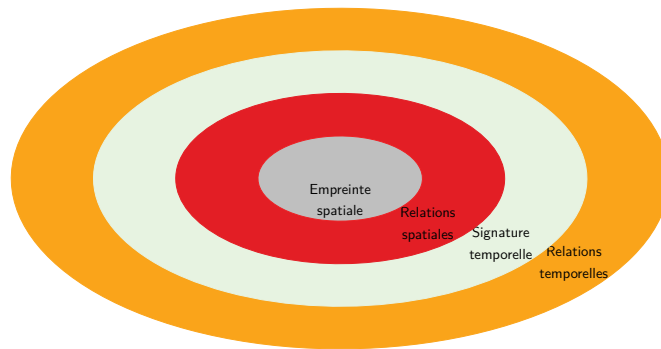


Figure 1.7 – Illustration d'une entité spatio-temporelle

1.2 Intégration des approches intelligentes aux entités spatiales

Afin de proposer une représentation de l'entité aussi proche que possible de la réalité et de mieux étudier les interactions qu'elles produisent dans leur milieu, il est primordial d'associer les approches d'intelligence artificielle et computationnelle aux modèles spatio-temporels appliqués sur des entités. Dans cette section, nous mettons en évidence la notion d'*intelligence* et les différents concepts proposés en intelligence artificielle et computationnelle. Certains de ces concepts ont déjà été intégrés dans plusieurs modélisations existantes des entités spatio-temporelles.

Notion d'intelligence en informatique

La notion d'intelligence est assez controversée dans son utilisation. Pour mieux décrire les notions qui se cachent dans ce concept d'intelligence, il est important de s'intéresser à sa définition dans un premier temps. Dans le dictionnaire [Sinclair, 1990], l'intelligence se définit comme suit :

- une personne est dite intelligente, si elle est capable de comprendre et d'entendre des choses,
- l'intelligence est la faculté de penser et de comprendre des choses à l'opposé de faire des choses de façon instinctive ou automatique.

Dans la première définition il apparaît que l'intelligence est un processus humain. La deuxième est plus large et laisse croire que des choses pourraient aussi être intelligentes [Negnevitsky, 2005]. Si les choses ont la capacité d'être intelligentes, qu'en sera-t-il de leur faculté à penser ? Le même dictionnaire, [Negnevitsky, 2005] trouve que la pensée est définie comme suit :

- la pensée est une activité du cerveau à considérer un problème ou à créer une idée d'après [Sinclair, 1990].

Toute chose ou toute personne devrait avoir un composant (p. ex. le cerveau chez les humains) lui permettant de comprendre des problèmes, de les résoudre et de prendre des décisions. Ainsi, l'intelligence est définie comme : *une capacité d'apprendre et de comprendre, de résoudre des problèmes et de prendre des décisions*. Cette notion d'intelligence a aussi particulièrement intéressé [McCarthy, 2007]. Pour lui, l'intelligence est à la fois un domaine de la science et de l'ingénierie spécialisé dans la construction de machines intelligentes par des programmes informatiques. Il a aussi pour objectif d'utiliser l'ordinateur pour comprendre l'intelligence humaine. Plus tard, il identifie l'intelligence à une activité *computationnelle* ayant la capacité d'atteindre des objectifs dans le monde. De toutes ces définitions, nous en déduisons que l'intelligence n'est pas seulement une qualité issue de la faculté humaine. Ainsi, toute chose ayant la capacité à résoudre des problèmes du monde qui l'entoure est dite intelligente. Deux grandes thématiques ont considérablement évolué depuis les origines de ce dilemme sur des machines intelligentes. La première communément appelée l'intelligence artificielle est considérée comme un domaine de l'informatique qui s'occupe du perfectionnement des machines pouvant réaliser des activités comme les humains. La deuxième est l'intelligence computationnelle qui s'oriente dans la représentation et l'acquisition des connaissances. L'intelligence computationnelle se focalise sur tout ce qui permet l'évolution du logiciel. La figure 1.8 est une représentation des domaines contribuant à l'évolution des systèmes intelligents. Cette évolution a créé plusieurs communautés de recherche afin de proposer de nouvelles stratégies permettant de résoudre des problèmes complexes du monde réel en un temps « raisonnable ».

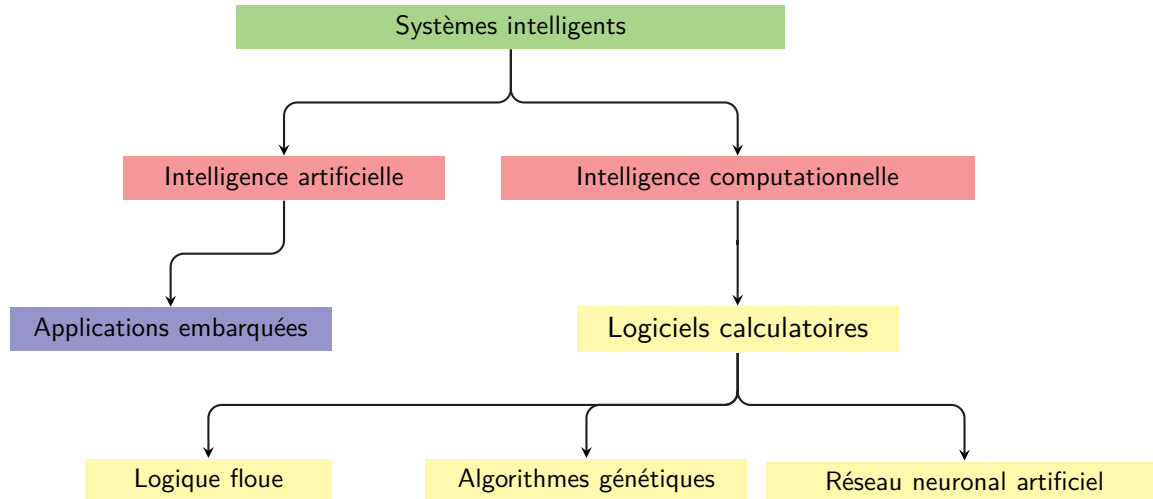


Figure 1.8 – Différents domaines travaillant sur des systèmes intelligents. Extrait de [Russel et Norvig, 1995]

1.2.1 Intelligence artificielle

L'intelligence artificielle (IA) est la fusion de deux grandes thématiques de recherche : l'intelligence artificielle humaniste (*IAH*) qui étudie les machines qui pensent et réagissent comme des humains

et l'intelligence artificielle rationaliste (*IAR*) qui examine les machines pouvant être construites sur la compréhension du comportement de l'intelligence humaine [Russel et Norvig, 1995]. Les premières recherches sur l'intelligence artificielle ont débuté dans les années 1940 avec pour principales motivations de prouver que les ordinateurs sont intelligents. A cette époque l'IA a aussi pour mission de reproduire des comportements qui exigent une intelligence humaine. Le principal challenge à cette époque était de savoir si les machines pouvaient penser. Question restée vague et floue jusqu'à ce que [Turing, 1950] propose l'article sur la machine de Turing, qui va fortement influencer les recherches sur les machines intelligentes. Ce travail a proposé des résultats qui sont restés universels et sont toujours appliqués à ce jour. Ce n'est que plus tard que John McCarthy fonde le terme intelligence artificielle. Il a été lui aussi le créateur du premier langage de haut niveau **lisp** (*lisp* est la plus ancienne famille de langages impératifs⁷ et fonctionnels⁸) et du premier langage de programmation *Fortran*. Le langage *lisp* a été initialement développé pour exécuter des programmes.

Les principaux travaux de John McCarthy étaient orientés sur la logique formelle. Faisant suite à ces différents travaux, il devenait ainsi le principal contributeur dans l'ingénierie des connaissances. Dans ces travaux, il apparaît peu de concepts et de méthodologies orientés vers la résolution des problèmes complexes. La principale difficulté de ces travaux était liée aux faibles capacités de stockage et de calcul des ordinateurs existant à cette époque. Pour la plupart des méthodes qu'il propose pour la résolution de problèmes, il déploie toutes les combinaisons possibles afin de trouver la solution. Ceci est dû au fait qu'aucune méthode proposée n'utilise des connaissances extraites du domaine du problème [Russel et Norvig, 1995]. Cette difficulté lance alors les recherches sur la complexité algorithmique et les heuristiques.

1.2.1.1 Algorithmique et heuristique

Dans l'intérêt de proposer des méthodes qui prennent en compte les connaissances issues d'un domaine du problème, les chercheurs ont proposé des méthodes de résolution de problèmes fondées sur des stratégies afin d'améliorer les techniques dites aveugles ou sans stratégies. Les méthodes aveugles fonctionnent sur des problèmes triviaux tout en sollicitant de grands volumes de stockage mémoire et un temps de calcul important. De plus, le passage à l'échelle (c'est-à-dire pour des grand échantillons) n'est pas toujours possible [Russel et Norvig, 1995]. Pour adapter les méthodes à des stratégies de passage à l'échelle des problèmes, les chercheurs vont étudier de façon précise le domaine du problème afin de déterminer des stratégies optimales. Une stratégie sera dite optimale quand elle minimise l'ensemble des ressources sollicitées pour résoudre un problème, le temps de calcul et l'espace de stockage mémoire étant les deux ressources le plus souvent considérées. Le

7. Un langage est dit impératif quand il décrit les opérations d'un programme en terme de séquences d'instructions exécutées par un ordinateur pour modifier l'état du programme

8. Le Langage fonctionnel considère plutôt une évaluation des fonctions mathématiques tout en écartant ainsi le concept de changement d'état et de mutation

coût des ressources sollicitées en temps de calcul a permis d'identifier deux grandes catégories de problèmes : la catégorie des problèmes dits P (Polynomiaux) et la catégorie des problèmes non déterministes NP . La première catégorie dite classe des problèmes P est la classe de l'ensemble de problèmes de décision⁹ dont la solution se trouve en temps polynomial. Plus précisément, si l'entier n est la taille du problème et p le temps ou le nombre d'étapes nécessaires pour trouver une solution, p peut s'exprimer par une fonction polynomiale d'ordre k en n définie par l'équation 1.1. Dans ce cas, la complexité du problème dans le pire des cas est donnée par l'expression $O(n^k)$.

$$p = f(k) = \sum_{i=1}^k n^i + cte \quad \text{et } n > k \quad \text{où } cte \text{ est une constante entière} \quad (1.1)$$

La deuxième catégorie des problèmes dits NP est l'ensemble des problèmes de décision tels que, pour une solution donnée, il est possible de vérifier cette solution en un temps polynomial. Les classes NP et P ne sont pas complémentaires (NP ne signifie pas non polynomial) et il existe aussi des problèmes qui ne sont ni P ni NP . Plus tard [Karp, 1972] en se basant sur les travaux de [Cook, 1971] proposait la classe NP complet (NPC) qui est une sous classe des problèmes polynomiaux non-déterministes NP . Un problème de la classe NP complet vérifie les deux propriétés ci-dessous :

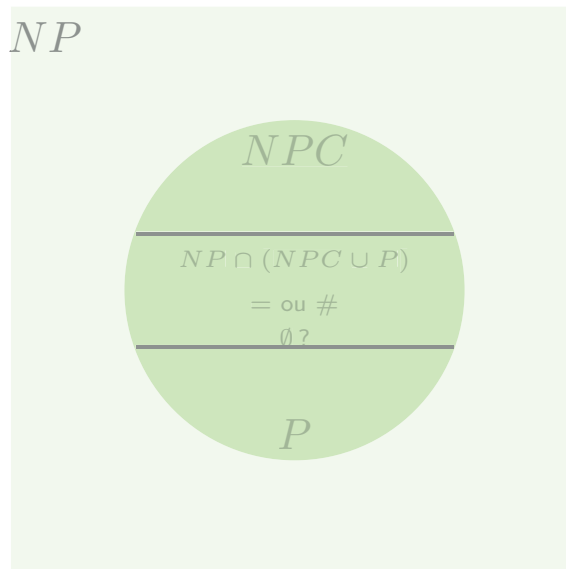
- 1** pour une solution donnée, on peut la vérifier en un temps polynômial ($P \subset NP$),
- 2** tous les problèmes de la classe NP se ramènent à celui-ci via une réduction polynomiale [Karp, 1972]. C'est-à-dire que ce problème est au moins aussi difficile que tous les autres problèmes de la classe NP .

Les problèmes qui vérifient uniquement la deuxième propriété seront dits NP -difficiles. Par ailleurs, il n'est pas possible de montrer que l'intersection entre les classes NP et $(NPC \cup P)$ est $= \emptyset$ ou $\neq \emptyset$? (c.-à-d, les problèmes NP ne sont formés que des problèmes P et NPC). Cette intersection est représentée par l'ovoïde sur la figure 1.9.

Confrontés au fait que les problèmes quotidiens sont généralement NPC (p. ex., le problème du voyageur de commerce, la recherche d'une clique maximale dans un graphe¹⁰, etc.) , même les chercheurs qui possèdent des ordinateurs avec de grandes capacités de stockage mémoire et avec une puissance de calcul très performante ne parviennent pas à trouver toujours les solutions optimales à ces problèmes avec un temps de calcul raisonnable. Pour faire face à ces contraintes en temps et en espace de stockage ainsi qu'à l'absence de pré-traitement et formalisation des données utilisées à l'entrée des problèmes, plusieurs nouveaux domaines ont été explorés : la représentation des connaissances, l'apprentissage algorithmique, le calcul neuronal. Ils font l'objet de la section suivante. Ces domaines sont aussi les précurseurs de l'intelligence computationnelle avec l'aide des langages de programmation, de la logique formelle, de l'anti-logique et des circuits intégrés.

9. Un problème de décision se résume à une question dont la réponse est oui/non sur une entrée bien définie.

10. la recherche du clique maximale est un problème qui consiste à trouver dans un graphe un sous-graphe total qui n'est pas inclus dans un autre sous-graphe total

Figure 1.9 – Illustration de l’organisation des problèmes NP

1.2.2 Intelligence computationnelle

Les principaux champs de recherche de l’intelligence computationnelle s’articulent autour de la logique floue, le réseau artificiel neuronal et les algorithmes génétiques. En regroupant ces trois disciplines, on observe leurs contributions immédiates dans trois thématiques complémentaires : l’apprentissage des connaissances, l’acquisition des connaissances et l’exploration des connaissances. Par ailleurs, la connaissance est la capacité à comprendre la théorie et la pratique d’un sujet ou d’un domaine. La connaissance est aussi la somme de tout ce qui est connu.

1.2.2.1 Apprentissage des connaissances

L’intégration des méthodes d’apprentissage aux systèmes intelligents visant à réagir comme l’homme ou d’autres êtres vivants demande un certain nombre de prérequis. Ces prérequis exigent une structuration préalable de la connaissance sollicitée. Au cours de l’apprentissage, le système intelligent est solitaire et il cherche à développer et à améliorer ses connaissances. L’apprentissage par des machines présente deux problèmes fondamentaux à savoir : l’épistémologie¹¹ et l’heuristique¹². L’apprentissage permet aussi de fournir en temps polynomial des solutions qui ne sont pas forcément optimales pour des problèmes NP difficile¹³. Deux grandes thématiques sont explorées dans l’apprentissage des connaissances. Nous pouvons citer entre autre les réseaux formels et la

11. L’épistémologie est le domaine de la philosophie des sciences qui étudie les sciences particulières, comme la théorie de la connaissance en général

12. L’heuristique se définit comme l’art d’inventer et faire des découvertes dans les domaines de l’optimisation combinatoire, de la recherche opérationnelle et de la théorie des graphes

13. Un problème NP difficile est un problème auquel tous les problèmes de la classe NP peuvent se réduire à lui

logique floue.

Apprentissage par un réseau formel

L'apprentissage par un réseau est décrit comme étant un modèle de raisonnement basé sur le cerveau humain. Il se base sur le fait que l'apprentissage est la tâche fondamentale et essentielle de la structure biologique du réseau neuronal. A l'aide de sa faible densité structurale (ou le couplage entre les composants), la structure neuronale a été facilement représentée et a permis de modéliser l'apprentissage. Motivés par cette structuration, les chercheurs ont essayé de transposer l'architecture du réseau neuronal sur des ordinateurs [Negnevitsky, 2005]. Cette transposition a abouti à un réseau neuronal dit artificiel. La figure 1.10 illustre la structure d'un réseau neuronal. Le réseau formel a été le principal précurseur dans l'apprentissage des connaissances. Il offre de nouvelles bases aux systèmes experts flexibles et indépendamment d'une quelconque application. Son principe de fonctionnement est de construire de nouvelles règles à partir de nouveaux faits obtenus soit par des interactions avec des humains soit à partir de sa propre base de connaissances. La difficulté du réseau formel est qu'il opère toujours sur des langages bien formulés. Cependant, les langages que nous utilisons sont souvent considérés comme vagues et flous [Negnevitsky, 2005].

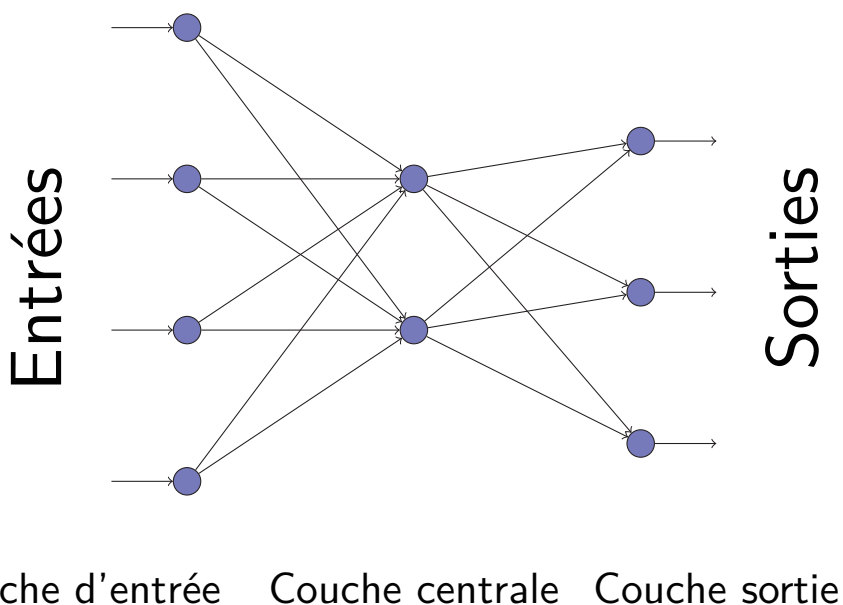


Figure 1.10 – Structure d'un réseau neuronal artificiel

Apprentissage par des systèmes experts flous

Basée sur les principes de la logique floue proposée par [Zadeh, 1965], la logique floue est définie comme un ensemble de principes mathématiques de la représentation des connaissances fondée sur des mesures de classes membres plus que sur des mesures de classes souches¹⁴. L'objectif de la logique floue est de refléter la pensée de l'homme. Elle a permis d'éviter le problème de goulot d'étranglement computationnel¹⁵ rencontré par les systèmes experts traditionnels. La logique floue a été utilisée dans des technologies vagues où les données d'entrée n'appartiennent pas forcément aux classes souches [Negnevitsky, 2005]. Elle écarte la possibilité de ne considérer que des résultats de probabilités pour déclencher de nouvelles règles. Dans son mode de fonctionnement, elle tient compte des mots clés du système et non à des valeurs numériques. Les termes « dense » et « absent » utilisés dans les expressions ci-dessous sont des exemples de terme définis dans la logique floue :

- si le brouillard est dense alors la `_distance_de_visibilité` est réduite,
- si le brouillard est absent alors la `_distance_de_visibilité` est correcte.

La recherche sur les réseaux formels et la logique floue ont permis de proposer de nouvelles méthodes utilisées dans l'acquisition et l'exploration des connaissances [Negnevitsky, 2005].

1.2.2.2 Systèmes intelligents dans l'acquisition des connaissances

Pour que la représentation de la connaissance soit effective dans un modèle à base de connaissances, la connaissance doit être initialement acquise. Plusieurs approches sont utilisées pour acquérir de la connaissance, d'autant plus qu'elle peut provenir de diverses sources d'information. Nous pouvons procéder à des observations ou à des interviews d'experts du domaine afin de produire un système à base de règles. Cependant cette tâche est difficile et coûteuse. D'autres approches consistent à acquérir de la connaissance de façon automatique par des techniques d'extraction, dans les gros volumes de documents structurés ou non structurés.

Les techniques ou méthodes utilisées dans ces approches d'acquisition automatique s'appuient généralement sur des algorithmes génétiques utilisés par les êtres vivants [Cleuziou *et al.*, 2011]. Les algorithmes qui ont fortement marqué ces méthodes sont les algorithmes inspirés par les colonies de fourmis et les colonies d'abeilles. Les comportements les plus marquants dans ces colonies sont identifiés par leur grande organisation. C'est à partir de ces comportements que sont établis des algorithmes génétiques. Ces algorithmes sont fondés sur une évolution dans le modèle computationnel. Le principe fondamental consiste à réaliser une simulation d'une population d'individus tout en évaluant leur performance. Le processus de simulation devra ainsi être répété un certain nombre de fois et la population obtenue à la i^{eme} simulation devra être utilisée pour la $(i + 1)^{eme}$ simulation. [Holland, 1975] fut le premier chercheur à établir le concept d'algorithme génétique. Trois grands événements ont marqué ensuite l'évolution des algorithmes : les algorithmes génétiques, les

14. Les classes souches sont les classes apparues généralement de façon binaire dans un processus de décision binaire

15. Est un état du système au cours duquel ses performances globales sont limitées

stratégies révolutionnaires et les programmes génétiques par représentation.

1.2.2.3 Systèmes intelligents dans l'exploration des connaissances

L'exploration est le fait de chercher avec l'intention de découvrir quelque chose d'inconnu. L'exploration de connaissances à partir de données est appelée la fouille de données [Groth, 1998]. La fouille de données peut être comparée à l'exploration de l'or dans des minerais. De grandes quantités de minerais doivent être traités avant que l'or puisse être extrait. Dans le même contexte, l'exploration de données peut nous aider à trouver *la connaissance cachée* à partir d'un ensemble de données. La fouille de données est aujourd'hui très utilisée par les entreprises. Par exemple, elle est utilisée pour identifier les clients susceptibles d'acheter un produit ou de demander un service. Dans l'analyse de tendance, elle est utilisée pour déterminer les tendances sur un marché, par exemple, le marché boursier. Dans la détection des fraudes, la fouille de données est utilisée pour identifier les fausses réclamations d'assurance, les achats frauduleux par carte de crédit [Groth, 1998].

Les techniques de la fouille de données sont inspirées de l'intelligence artificielle ou de la statistique. Les outils statistiques utilisés peuvent être élémentaires ou multidimensionnels, utilisant des techniques de généralisation de modèles linéaires (régression multiple, logistique) et d'analyse discriminante. Ces outils sont alors en concurrence avec d'autres approches plus algorithmiques visant à la construction d'arbres de décision, de règles d'induction et de réseaux neurones. Les différentes stratégies de modélisation de la connaissance présentées dans cette section nécessitent d'être structurées dans un formalisme de représentation afin d'être exploitées par d'autres domaines. La fusion de cette modélisation avec les approches en SIG, étudiées dans la section 1.1, a été réalisée sur le formalisme de la logique de description programmable « DLP ». Le choix de cette fusion est discuté dans la section 1.3.

1.3 Fusion des SIG et des approches intelligentes

Il apparaît clairement que le domaine des SIG a besoin de nouvelles approches intelligentes pour représenter au mieux les entités spatio-temporelles. Il est donc important de définir ou de choisir un formalisme qui devra fusionner les recherches réalisées dans les SIG et les résultats issus de l'intelligence artificielle. De nos jours, l'un des formalismes les plus prometteurs est fondé sur les ontologies. Les premiers travaux ne mettaient pas en valeur les dimensions temporelles et sémantiques dans les ontologies. Avec les récentes avancées sur la représentation des dimensions temporelles et spatiales, exploitant des moteurs d'inférences comme SWRL et des raisonneurs puissants comme PELLET tous deux basés sur le système JENA¹⁶, il est dorénavant possible d'intégrer les résultats d'intelligence artificielle dans les SIG afin de produire des applications robustes et intelligentes.

16. <https://jena.apache.org/>

1.3.1 Ontologie

Les premières années de création des ontologies remontent aux années 1980 [McCarthy, 2000]. D'après [McCarthy, 2000], l'ontologie est considérée comme une application de l'intelligence artificielle. Il décrit également l'ontologie comme une branche de la philosophie qui étudie des choses existantes. Plus tard, il a montré que l'ontologie et la réification sont fondamentalement les mêmes concepts. La réification consiste à transformer ou à transposer une abstraction en un objet concret ou à appréhender un concept comme une chose concrète.

L'ontologie fait partie de la description des concepts dans des systèmes à base de connaissances. Un système à base de connaissances n'est rien d'autre qu'une fusion entre une base de connaissances et un moteur d'inférence. Cette définition lui permet de se distinguer des programmes conventionnels structurés. Le schéma classique du processus d'inférence entre les règles, les causes et les faits classés selon les notions de déduction, d'abduction et d'induction est décrit ci-dessous :

- déduction : cause + règle = effet,
- abduction : effet + règle = cause,
- induction : cause + effet = règle.

Sa principale évolution est due aux recherches avancées sur le web sémantique [Guarino *et al.*, 2009]. Le but fondamental du web sémantique est de proposer des architectures bien structurées afin de faciliter la recherche et le suivi des informations ou des contenus sur le web. Plus tard, apparaît le thème de l'ontologie computationnelle qui n'est rien d'autre qu'un moyen de modéliser la structure d'un système. Pour [Guarino *et al.*, 2009], l'ontologie computationnelle consiste entre autres à mettre en évidence les entités et les relations qui peuvent émerger à partir de leur observation du monde réel et à les faire coïncider avec les objectifs de l'application. Les relations de l'ontologie sont unaire ou binaire et les concepts quant à eux sont organisés de façon hiérarchique et produisent une taxonomie.

Les ontologies dégagent en définitive deux grandes idéologies fondamentales à savoir : la généralisation et la spécialisation des concepts. D'après [Gruber, 1993], l'ontologie est vue comme une clarification de la spécification des concepts. Pour [Borst *et al.*, 1997], l'ontologie est une spécification¹⁷ convenable d'une conceptualisation¹⁸ partagée. Plus tard, au cours des années 1998, [Studer *et al.*, 1998] combinent ces deux définitions en déclarant que : « l'ontologie est une mise en œuvre d'une spécification explicite d'une conceptualisation partagée ». Cette définition de [Studer *et al.*, 1998] met en évidence trois notions majeures qui sont : 1) la notion de conceptualisation, 2) la notion de mise en œuvre personnelle et enfin, 3) la notion de partage. Bien qu'issue d'un

17. La spécification est un ensemble explicite d'exigences à satisfaire par un matériau, produit ou service

18. Représentation mentale générale et abstraite d'un objet ou idée que l'on se fait de quelque chose

processus mental, la conceptualisation a pu être formalisée à l'aide d'une extension de la structure relationnelle de [Genesereth et Nilsson, 1987].

La conceptualisation est décrite par [Gruber, 1993] comme un squelette de la représentation officielle des objets, concepts et d'autres entités qui revendiquent leur existence dans un domaine d'intérêt lié par des relations ou propriétés. Ces définitions permettent de classer la conceptualisation comme un processus mental et abstrait. Malgré cette complexité mentale et abstraite, [Genesereth et Nilsson, 1987] ont établi une représentation mathématique simple de la conceptualisation comme : « une extension et intention relationnelle structurée » et formalisée par un couple (D, R) où D est l'ensemble universel et R l'ensemble des relations sur D .

Extension relationnelle structurée

Considérons un ensemble D régissant un univers d'entités de l'environnement maritime utilisé au cours de la navigation. Nous aurons les concepts suivants qui émergeront de notre conceptualisation : *RouteMaritime*, *Balises*, *ZonesDeDanger*, etc. Dans ce domaine, les relations binaires qui pourraient apparaître entre ces trois concepts sont : *prendEnCompte*, *estEntouréPar*, *respecteDes*. Nous pouvons imaginer des relations unaires telles que toute entité a un identifiant (*aUnIdentifiant*). Nous en déduisons ainsi $D = \{P200009, P200010, P200011\}$ et les relations $R = \{RouteMaritime, Balises, ZonesDeDanger, prendEnCompte, estEntouréPar, respecteDes\}$. Cette modélisation est illustrée par la figure 1.11 et l'extension de ces relations sont définies comme suit :

- *RouteMaritime* = $P200009, \dots$
- *Balises* = $P200010, \dots$
- *ZonesDeDanger* = $P200011, P200012, \dots$
- *prendEnCompte* = $(P200009, P200011), \dots$
- *estEntouréPar* = $(P200010, P200011), \dots$
- *respecteDes* = $(P200009, P200010), (P200009, P200011), \dots$

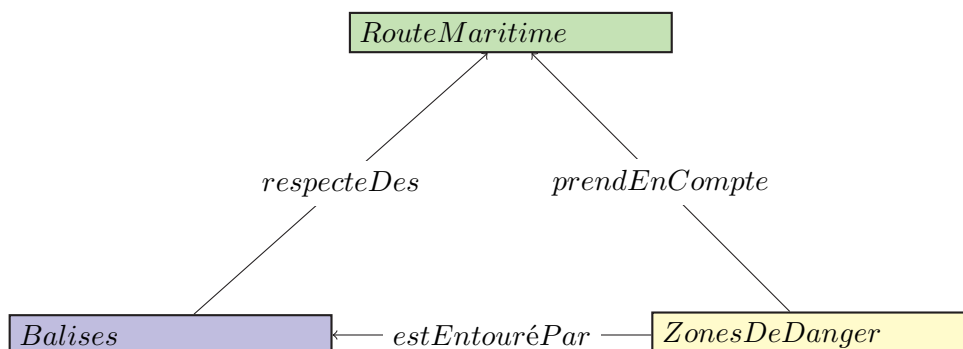


Figure 1.11 – Illustration d'une conceptualisation de l'environnement maritime

L'extension relationnelle structurée est confrontée à deux problèmes majeurs : elle n'est pas conforme aux changements du monde et elle est très spécifique au monde en termes de croyances. D'autre part, l'absence de conformité aux changements du monde se justifie par le fait que l'arrivée d'une nouvelle instance dans le modèle proposé ci-avant modifie la conceptualisation. Par exemple, si nous avons une nouvelle zone qui est identifiée, la relation *prendEnCompte* va directement changer de cardinalité. D'après [Genesereth et Nilsson, 1987], la conceptualisation est indépendante du changement du monde, ce qui engendre deux nouvelles conceptualisations car $(D, R) \# (D', R')$ où D' l'ensemble des entités et R' toutes les relations établies suite à un nouveau changement du monde. Le deuxième problème de la conceptualisation par extension est qu'elle est très spécifique au monde réel. Pourtant une conceptualisation vise à comprendre le sens d'une relation indépendamment d'un seul état donné du monde réel, puisque le sens de la relation peut varier dans un contexte précis. Dans ce cas, il est alors nécessaire de procéder à une intentionnalité¹⁹ des relations.

Intention relationnelle structurée

L'intentionnalité d'un concept ou d'une relation se définit comme l'ensemble des états maximum d'un monde (des croyances globales dans un contexte). Dans une modélisation, il faut initialement clarifier le « monde » et « l'état du monde ». Supposons par exemple deux entités liées par la relation *estEntouréPar*. Cette relation *estEntouréPar* peut avoir plusieurs interprétations dans la conceptualisation. Pour une première interprétation, une balise peut s'utiliser pour délimiter une zone de danger et dans une seconde interprétation, elle peut s'utiliser dans une zone de danger pour orienter la navigation. Ces deux interprétations engendrent de nouveaux changements du monde et par conséquent deux nouvelles conceptualisations d'après [Genesereth et Nilsson, 1987]. Pour pallier ces problèmes, il est nécessaire de s'intéresser à la notion d'intention relationnelle ou relation conceptuelle. [Guarino *et al.*, 2009] la définit comme suit. Considérons S un système arbitraire, D un ensemble arbitraire d'éléments distincts de S et W un ensemble d'états ou les possibles changements du monde de S . Le couple $\langle D, W \rangle$ est appelé domaine espace de S . Intuitivement, $\langle D, W \rangle$ fixe l'espace de variabilité de l'univers d'échanges D en respectant les états possibles W de S . Une relation intentionnelle (ou relation conceptuelle) $\phi(n)$ d'arité²⁰ n de $\langle D, W \rangle$ est une fonction totale $\phi(n)$ définie par l'équation 1.2.

$$\phi(n) : W \rightarrow (2^{D^n}) = (2^{(\|S\|)^n}) \quad (1.2)$$

L'intention relationnelle structurée ou la conceptualisation d'après [Guarino *et al.*, 2009] est définie par le triplet $C = (D, W, \mathcal{R})$ où \mathcal{R} est l'ensemble des relations conceptuelles du domaine espace $\langle D, W \rangle$. Dans le cas de l'exemple proposé ci-dessous, la conceptualisation selon [Guarino *et al.*, 2009] est décrite comme suit :

19. L'intentionnalité est le fait de donner de l'importance à comment l'on décrit les choses, sans s'intéresser au langage

20. l'arité d'une fonction, ou opération, est le nombre d'arguments ou d'opérandes qu'elle requiert.

$D = \{P200009, P200010, P200011, P200012\}$, $\mathcal{R} = \{RouteMaritime^1, Balises^1, ZonesDeDanger^1, prendEnCompte^2, estEntouréPar^2, respecteDes^2\}$ est l'ensemble des relations conceptuelles ou l'exposant sur la relation désigne son arité et $W = \{w1, w2, \dots\}$ l'ensemble des possibles changements du monde.

- pour tout changement du monde $w \in W$, $RouteMaritime^1(w) = P200009, \dots$
- pour tout changement du monde $w \in W$, $Balises^1(w) = P200010, \dots$
- pour tout changement du monde $w \in W$, $ZonesDeDanger^1(w) = P200011, P200012, \dots$
- $prendEnCompte^2(w1) = (P200009, P200011), \dots$
- $prendEnCompte^2(w2) = (P200009, P200011), (P200009, P200012), \dots$
- $prendEnCompte^2(w3) = \dots$
- $estEntouréPar^2(w1) = (P2000010, P200011), \dots$
- $estEntouréPar^2(w2) = \dots$

1.3.2 Les formalismes de représentation

Au vu des éléments détaillés au paragraphe précédent, l'ontologie apparaît comme le modèle de représentation par excellence pour représenter l'entité. Grâce à sa capacité à intégrer les fonctionnalités issues de l'intelligence, détaillées dans les sections 1.2.2 et 1.2.1, elle permet de représenter facilement la sémantique (la saillance, l'affordance, voir chapitre 2.1) associée à une entité dans un environnement donné. Par ailleurs, il est judicieux de proposer un formalisme adéquat à l'ontologie. Le formalisme de représentation d'une ontologie est fondé sur la logique terminologique.

La recherche en représentation des connaissances a conduit au développement de la logique terminologique [Schmidt-Schauß, 1989]. La logique terminologique est la représentation conceptuelle des termes d'un domaine. Elle est indépendante de la logique fonctionnelle et a été introduite dans le but de représenter la sémantique et l'information syntaxique du langage naturel. Elle est basée sur un formalisme de description des attributs. De nos jours, plusieurs résultats ont été apportés dans la description des attributs grâce aux succès des domaines de recherche suivants : la théorie des langages, la théorie des automates et la logique modale²¹. La logique terminologique est basée sur la description des attributs et a été démontrée comme étant très utile dans la compréhension expressive des formalismes [Schmidt-Schauß, 1989]. Il faut cependant noter que la logique terminologique et la logique fonctionnelle sont des sous-ensembles de la logique traditionnelle (la logique du premier ordre : « FOL²² ») voir la figure 1.12.

Bien que la logique terminologique ait été reconnue indécidable avant les années 1988, ce n'est qu'au

21. La logique modale est à l'origine de la logique aléthique qui prend en compte la croyance du locuteur dans la phrase

22. logique comportant un ensemble de symboles désignant des variables ou des prédicats liés par des connecteurs (et, ou) associant des quantificateurs existentiels et universels

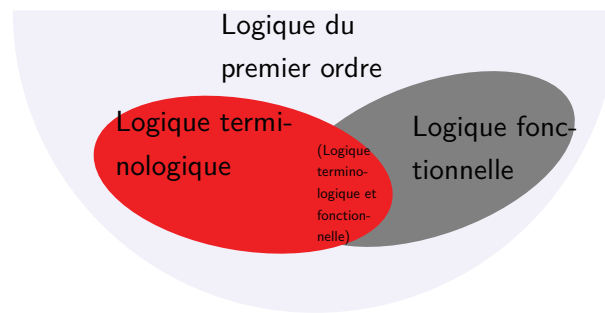


Figure 1.12 – Expressivité de la logique terminologique et de la logique fonctionnelle

cours de l'année 1989 qu'on a pu montrer qu'elle est un sous-concept du langage de représentation *KL-one* [Schmidt-Schauß, 1989][Schmolze *et al.*, 1985] qui est un langage décidable. Le langage de représentation *KL-one* est défini comme étant un système de représentation de la connaissance par des programmes en intelligence artificielle [Schmolze *et al.*, 1985].

A l'aide de la logique terminologique, plusieurs formalismes de représentation de la connaissance ont été développés et fondés sur des approches logiques et non logiques (voir la figure 1.13). Concernant les approches non logiques, les plus anciens sont les réseaux sémantiques introduits par [Quillan, 1967] et [Lehmann, 1992]. Ils représentent la connaissance sous forme d'un graphe orienté et étiqueté où les nœuds identifient les concepts et les arcs identifient les relations entre les concepts. La difficulté rencontrée dans les réseaux sémantiques apparaît dans le fait qu'ils ne représentent pas exactement la sémantique. Ceci est dû au fait que les arcs sont utilisés pour spécifier différentes relations.

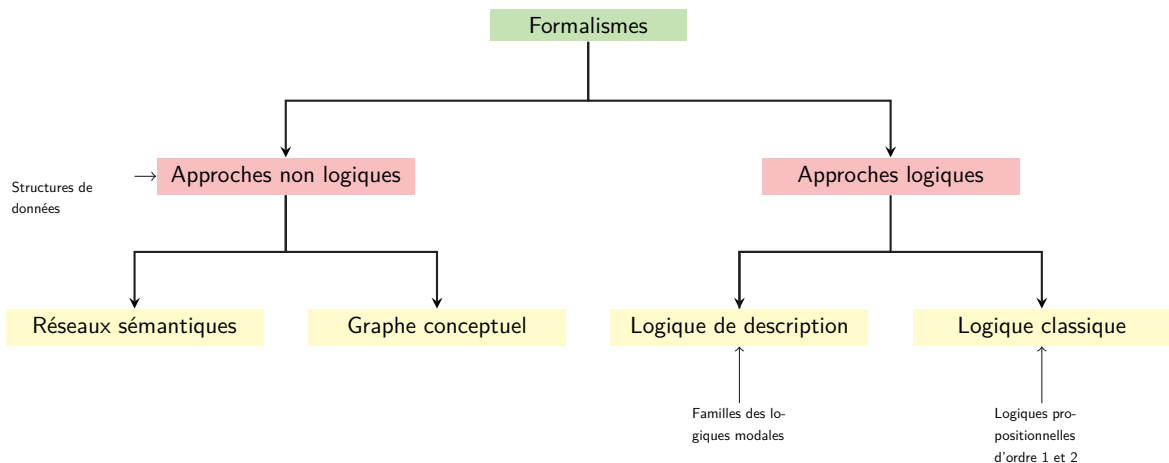


Figure 1.13 – Différentes catégories de formalisme

La « frame » proposée par [Minsky, 1974] est une amélioration du graphe conceptuel. Elle est modélisée par un ensemble de nœuds et relations présentés de façon hiérarchisée. Le niveau le plus

haut représente les choses qui sont toujours vraies (concepts ou classes). Le niveau le plus bas (nœud terminal) appelé aussi « slot » est rempli d'instances ou de données. Chaque nœud terminal est une instance membre d'une classe. Certaines informations sont associées aux attributs telles que : les valeurs, les restrictions des valeurs. Celles-ci sont attachées par des méthodes et procédures permettant de calculer d'autres valeurs selon le besoin. Les valeurs des nœuds terminaux sont assimilées à un domaine concret (p. ex. les entiers, les strings) ou identifiant d'autres « frames ». Cependant la sémantique définie dans la « frame » n'est pas claire en ce sens que la sémantique des différents types d'informations embarquées dans la « frame » est erronée [Domini *et al.*, 1997].

Finalement, pour pallier ce problème, [Schmidt-Schauß, 1989] a montré que le système qui permettait à la « frame » d'intégrer différents types d'informations ou de s'adapter à un réseau sémantique était la logique de description basée sur des approches logiques. Par la suite, plusieurs travaux sur les formalismes de représentation se sont basés sur le modèle de *Kl-one*. Issu du système *Kl-one*, la logique de description issue de la logique terminologique ou des concepts du langage communément appelé « DL » a été conçue pour la reconstruction de la logique et la spécification des connaissances représentant des systèmes. Elle est basée sur les concepts et les rôles. Les concepts décrivent un ensemble d'objets et les rôles ou propriétés sont interprétés comme des relations binaires entre les objets. La logique de description est utilisée pour représenter les concepts et les rôles complexes tout en utilisant simplement les opérations (négation, conjonction, disjonction) et les quantificateurs (universels et existentiels). Elle est composée d'une base de connaissances *Tbox* « Terminologic box », une base d'instances *Abox* « Assertion box » et des prédicats *Rbox* « Relation box ». La grammaire présentée par l'équation 1.3, est celle associée à la logique de description. L'équation 1.3 présente les différents constructeurs qui permettent de définir et déterminer les nouveaux concepts du modèle.

Dans cette formule C et T sont des concepts, A et D sont des concepts atomiques et R est un rôle atomique.

$$C, D \rightarrow A \mid C \cap T \mid \forall R.C \mid \exists R \quad (1.3)$$

Cette grammaire a connu une extension dénommée le modèle conceptuel de description de données *ALCQI* « Attribute Language with Complement and Query Information » [Diego, 1996a, Diego, 1996b, Woods et Schmolze, 1992] représentée par l'équation 1.4.

$$\begin{aligned} C, D \rightarrow & A \mid C \cap D \mid C \cup D \\ & \forall R.C \mid \exists R.C \mid \exists^{\geq n} R.C \mid R.C \mid \exists^{\leq n} R.C \\ R \rightarrow & P \mid P^- \quad P \text{ est un rôle atomique} \end{aligned} \quad (1.4)$$

La puissance du formalisme de la logique de description réside dans ses moteurs d'inférence et ses raisonneurs. Ses règles sont définies suivant la logique de *Horn* capable de représenter une logique conditionnelle (si \rightarrow alors). La logique de *Horn* est une disjonction des formules atomiques avec un littéral positif et précédé par un quantificateur universel (\forall) (voir équation 1.5). Elle est

issue de la logique fonctionnelle.

$$\begin{aligned} (\forall) \neg B_1 \vee \neg B_2 \vee \dots \vee \neg B_n \vee H & \text{ est équivalent à} \\ (\forall) B_1 \wedge B_2 \wedge \dots \wedge B_n \rightarrow H & \end{aligned} \quad (1.5)$$

Ces deux règles se résument comme suit : si $B_1, B_2 \dots B_n$ sont vraies alors H est aussi vraie ou bien si les conditions $B_1, B_2 \dots B_n$ sont satisfaites alors exécutez H . Le formalisme le plus récent en logique de description est la logique de description programmable (voir figure 1.14) qui a été proposée par [Grosf *et al.*, 2003]. Elle permet d'intégrer directement des règles dans le formalisme. Ce formalisme peut s'utiliser sous plusieurs formats (*owl, rdf, owl - rdf, rdf - ml ...*). Le format que nous avons utilisé dans notre thèse est le format *owl* intégré dans une base de données « triple store ».

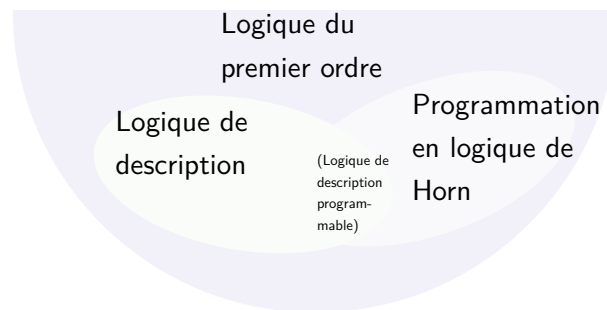


Figure 1.14 – Expressivité de la logique de description programmable (DLP) et de la logique de *Horn*

Dans la suite de ces travaux de thèse, nous nous sommes intéressés à la logique de description. A l'aide de ces bases solides dans son mécanisme de raisonnement, elle a été prouvée décidable et est actuellement est très utilisée non seulement dans le domaine de l'intelligence artificielle mais aussi dans d'autres domaines.

1.4 Conclusion

Ce chapitre met en évidence les différentes approches qui ont été utilisées dans les SIG et l'IA pour résoudre des problèmes du monde réel. Dans le cas des SIG, ce chapitre analyse les étapes de modélisation d'une entité dans deux dimensions : l'espace et le temps. Chacune de ces dimensions exige des méthodes ou des approches qui permettent de les représenter. La dimension spatiale définit des concepts tels que l'empreinte et les relations spatiales (de proximité, d'orientation, topologique et mérologique). La dimension temporelle présente quant à elle des concepts d'instant, d'intervalle, d'événement et d'agréations temporelles ainsi qu'un grand nombre de relations temporelles [Allen, 1984].

L'intégration de ces techniques dans un même modèle a finalement abouti au développement de modèles spatio-temporels. Cependant, les modèles spatio-temporels doivent intégrer de l'intelligence afin de représenter un environnement de manière aussi proche que possible de la réalité. Plusieurs approches issues de l'intelligence computationnelle et de l'intelligence artificielle ont été investiguées. L'intelligence artificielle s'intéresse à la réalisation de machines ou de modèles reproduisant les comportements humains tandis que l'intelligence computationnelle s'active dans la représentation et la gestion des connaissances afin de rendre des applications plus intelligentes. La fusion des approches issues des domaines SIG et celles issues de l'intelligence artificielle dans un même modèle doit être la plus transparente possible vis-à-vis des utilisateurs.

Cette représentation est rendue possible par l'ontologie. Pour certains chercheurs, l'ontologie est considérée comme une généralisation, une clarification et une spécification de concepts, et doit dégager des notions de conceptualisation, de mise en œuvre et de partage. La conceptualisation a pu être modélisée mathématiquement malgré sa complexité mentale et abstraite par [Guarino *et al.*, 2009]. La modélisation proposée par [Guarino *et al.*, 2009] met en évidence les notions d'extension et d'intention relationnelle. Par ailleurs, la fiabilité d'une ontologie dépend de ses formalismes qui sont issus des approches logiques et non logiques. Dans ce contexte, plusieurs formalismes ont été proposés (réseaux sémantiques, graphe conceptuel, logique de description, logique classique, etc.). De tous ces formalismes, la logique de description, dérivant de la logique terminologique, a été prouvée plus fiable pour la représentation de la sémantique dans divers domaines d'ingénierie des connaissances. Ainsi, les différentes stratégies de modélisation des connaissances présentées dans ce chapitre nécessitent d'être structurées en logique de description afin d'être exploitées par d'autres domaines. Cette étude fait l'objet du chapitre 2.

Chapitre 2

Ontologie computationnelle pour la représentation de la sémantique des entités spatio-temporelles

Sommaire

2.1	Enrichissement des entités spatio-temporelles par la sémantique issue du savoir d'un expert	34
2.1.1	Dimension sémantique de l'entité	34
2.2	Ontologies computationnelles pour l'environnement maritime	37
2.2.1	Modélisation d'une entité en «DL»	40
2.2.2	La norme «OGC» en logique de description	42
2.3	Formalisation de l'ontologie proposée	43
2.3.1	Formalisation de la dimension temporelle	46
2.3.2	Formalisation de la dimension spatiale	52
2.3.3	Formalisation de la dimension sémantique	53
2.3.4	Formalisation des changements d'une entité ou de sa perception	53
2.4	Implémentation de l'ontologie proposée	55
2.4.1	Les différents systèmes	55
2.4.2	Les règles en SWRL	58
2.5	Cas d'étude : la représentation du système de balisage	59
2.6	Conclusion	63

L^A proposition d'un modèle de représentation des entités spatio-temporelles aussi proche que possible de la réalité exige d'enrichir les modèles spatio-temporels actuels de nouvelles composantes sémantiques. La sémantique d'une entité est la troisième dimension permettant de représenter celle-ci.

Dans les modèles dédiés à la navigation maritime, la saillance et l'affordance sont les notions visées. La saillance d'une entité correspond généralement à la mise en relief d'un concept ou d'un ensemble de concepts qui se distinguent du contexte dans lequel ils sont immergés.

La saillance peut être visuelle, linguistique, sémantique, cognitive ou temporelle [Landragin, 2004]. Elle apparaît ainsi comme un facteur prépondérant dès que l'on s'intéresse à la modélisation du sens, aussi bien en interprétation qu'en génération dans la réalisation des systèmes automatiques [Landragin, 2004]. Par ailleurs, l'affordance est définie comme la capacité d'une entité ou d'un système à suggérer sa propre utilisation. Elle est utilisée dans différents domaines, notamment dans la psychologie cognitive, la psychologie de perception, le design, l'interaction homme-machine et l'intelligence artificielle [Landragin, 2004]. On doit à la psychologie sa définition originale : elle désigne *toutes les possibilités d'actions sur un objet*. Les actions appliquées sur un objet dépendent de la perception ou de la saillance visuelle que nous renvoie cet objet. Les propriétés de saillance et d'affordance sont complémentaires dans la représentation de la sémantique d'une entité. Ainsi, dans ce chapitre, la section 2.1 s'intéresse à la définition de la dimension sémantique d'une entité spatiale. Cette sémantique est par la suite enrichie par des connaissances provenant de l'expert du domaine à l'aide d'une ontologie computationnelle présentée dans la section 2.2. L'ontologie computationnelle proposée est développée par une étape de formalisation décrite à la section 2.3 avant d'être implémentée dans la section 2.4. La section 2.5 propose un cas d'étude de représentation d'un système de balisage.

2.1 Enrichissement des entités spatio-temporelles par la sémantique issue du savoir d'un expert

La dimension sémantique est importante dans ce travail de thèse. Les composantes de cette dimension devront être initialement identifiées par l'expert du domaine afin d'être utilisées pour enrichir les modèles spatio-temporels. Pour atteindre cet objectif, nous avons utilisé des ontologies computationnelles qui intègrent des techniques d'intelligence computationnelle et d'intelligence artificielle (voir chapitre 1.3). Cette ontologie nous permet d'intégrer les deux composantes principales de la navigation maritime que sont la saillance et l'affordance dans notre application d'aide à la navigation. Avant de présenter l'ontologie computationnelle proposée, nous avons préalablement étudié ces deux composantes tout en analysant leurs inter-relations dans la section suivante.

2.1.1 Dimension sémantique de l'entité

La sémantique est définie comme une branche de la linguistique qui étudie les signifiés, soit, ce dont parle un énoncé [Landragin, 2004]. On parle généralement de la syntaxe qui concerne le signifiant, soit ce qu'est l'énoncé. Dans le contexte de cette thèse principalement orientée vers la modélisation de l'environnement maritime à caractère dynamique, les composantes de la dimension

sémantique devront s'intégrer aux modèles spatio-temporels. Compte-tenu du fait que les modèles spatio-temporels qui nous intéressent sont ceux basés sur les changements spatio-temporels de l'entité, l'objet de cette section consiste à associer les composantes sémantiques et dynamiques essentielles à la navigation. Les composantes sémantiques auxquelles nous nous sommes intéressés sont la saillance et l'affordance.

Notion de la saillance

La saillance peut être visuelle, linguistique, sémantique, cognitive ou temporelle [Landragin, 2004]. La saillance visuelle est liée à la nature et à la disposition d'objets dans une scène, sans être clairement définie par rapport à des propriétés visuelles particulières. Quant à la saillance linguistique, elle est liée à la façon dont on présente ou traite un message et recouvre divers aspects de la structure informationnelle (ou structure communicative), sans être clairement définie par rapport aux composants de cette structure que sont les topiques, thème et *focus*¹ [Landragin, 2004]. Du point de vue cognitif, être saillant c'est ressortir en premier de la perception d'une scène compte-tenu :

- des représentations mentales de l'utilisateur,
- de ses intentions de perception [Edmonds, 1993],
- de son attention visuelle,
- de sa mémoire à court terme (qui se rapproche d'une scène vécue) [Rousselet et Fabre-Thorpe, 2003] ou de sa mémoire à long terme pouvant se décomposer en deux aspects :
 - la familiarité visuelle individuelle,
 - la familiarité visuelle culturelle,
- de ses expériences personnelles et de ses émotions.

Dans le cadre de ce chapitre, nous considérons uniquement la saillance visuelle, car elle joue un rôle important dans la navigation maritime. Elle permet de mettre en valeur une entité afin d'encadrer, d'orienter et de garantir la sécurité des biens et des personnes au cours de la navigation.

La saillance visuelle est liée aux particularités de la perception visuelle ; les facteurs physiques qui permettent de la déterminer peuvent être intrinsèques à la nature de l'objet en terme de luminosité (p. ex. la saillance du soleil ou d'une ampoule électrique du fait de leurs luminosités inhérentes) ou liés aux variations de perception des couleurs recouvrant deux aspects : un premier relatif aux temps de latence, et un second relatif à l'effet spatial des couleurs. Le temps de latence, c.-à-d., le décalage entre le début de l'excitation et celui de la sensation, varie selon les couleurs. En définitive, une entité est dite saillante d'un ensemble d'entités si elle se distingue par une propriété que les autres entités de l'ensemble n'ont pas. Les principaux critères qui permettent de mettre en évidence la saillance visuelle d'un objet sont décrits dans la table 2.1. En se référant à la table 2.1, nous en déduisons que la modification observée sur un critère de saillance d'une entité engendre une nouvelle interprétation et par conséquent une nouvelle sémantique. Il en résulte que le modèle spatio-temporel *STP* doit

1. mot latin qui veut dire foyer, c'est le lieu où plusieurs choses se concentrent.

associer une nouvelle saillance à chaque changement de l'entité. Une nouvelle saillance implique une nouvelle perception visuelle de l'entité et par conséquent une nouvelle affordance. Les notions de saillance et d'affordance sont toujours utilisées de façon transparente et complémentaire dans les modèles de représentation cognitive.

Notion d'affordance

Dans ses travaux, le psychologue Heinz Werner introduit le concept de monde primitif composé des objets appelés «objets d'actions» ou « objets-signaux ». Ces «objets d'actions» nous feraient agir ou du moins nous y pousseraient ; en considérant qu'en psychologie, une action se compose de trois composantes : la composante sensationnelle (facteur cognitif²), la composante sentimentale ou émotionnelle (facteur effectif) et la composante conative³ (activité motrice). Les « objets-signaux » dépendent d'une certaine relation ou mise en relation entre certaines qualités propres du monde et certaines actions ou dispositions à agir [Niveleau, 2006]. Sans cette relation, l'objet perdrait sa qualité de signal. En mettant en évidence son analyse de la perception sur les rapports entre le sujet⁴ et l'environnement, Gibson explique que cette affirmation doit se comprendre dans la cadre de la théorie écologique⁵ et non dans le cadre de la théorie physique⁶. Plus tard, Gibson concentre ses recherches exclusivement sur la dimension motrice de la perception visuelle, ouvrant ainsi la voie aux approches modernes sensorimotrices⁷ [Niveleau, 2006]. Il déclare également que l'environnement est formé de médium (milieu, intermédiaire) et de substances qui forment les objets et les sujets et leur permettent « différentes interactions ». Il définit le médium comme la matière dans laquelle se déplace le sujet et la matière qui n'appartient pas au médium représente les substances qui forment les objets. Pour Gibson, le terme affordance est considéré comme toute possibilité d'interaction offerte au sujet par son environnement. Dans ce contexte, l'air accorde la respiration, le sol accorde la marche, une pierre accorde la prise et ainsi de suite selon une relation qui dépend des caractéristiques physiques des entités mises en relation avec les caractéristiques physiques propres au sujet percevant. L'homme est le plus clair exemple ou sujet puisqu'il change ou produit de nouvelles affordances pour rendre son environnement plus facile à vivre [Simone, 2011]. Dans tous ses travaux, Gibson montre que le lien entre la perception visuelle et la notion d'affordance sont constitutives. La perception visuelle et l'affordance sont capables de guider et d'encadrer le sujet et doivent en conséquence être perçues continuellement. Ce qui met en évidence une fois de plus la relation entre la saillance visuelle et l'affordance au cours de la navigation.

La sécurité de la navigation exige de tenir compte de la saillance visuelle des entités comme indiqué

2. Il désigne l'ensemble des processus psychiques liés à l'esprit

3. Elle désigne, un effort, une tendance, une volonté, une impulsion dirigée vers un passage à l'action

4. le sujet est considéré comme un animal qui perçoit et agit [Niveleau, 2006]

5. Elle s'intéresse au monde en tant qu'environnement d'un sujet

6. Elle consiste à décrire le monde tel qu'il est, en allant de l'infiniment petit (les atomes, les particules, etc.) à l'infiniment grand (les galaxies, l'univers, etc.)

7. Concerne à la fois les phénomènes sensoriels et l'activité motrice, ou leurs interactions

dans la table 2.1. La prise en compte de la saillance exige de proposer une catégorisation des différents changements des entités spatiales pouvant modifier cette saillance. Étant donné que la saillance visuelle dépend des variations de couleurs, de formes et des temps de latence associés aux couleurs comme énoncé dans le 2.1.1, nous avons ainsi identifié quatre types de variations pouvant influencer ou modifier la saillance visuelle d'une entité et par conséquent son affordance.

Différents types de changement ou de perception d'une entité spatio-temporelle

La première catégorie, nommée par l'expression *variation de l'empreinte spatiale*, représente les entités spatio-temporelles qui subissent un changement de forme ou d'empreinte spatiale. Dans le modèle spatio-temporel *STP*, elle correspond aux fonctions *d'apparition, de disparition, d'expansion, de contraction, de déformation, d'union, de réallocation, de séparation* [Claramunt et Thériault, 1995]. La deuxième catégorie, nommée par l'expression *variation de l'emplacement spatial*, identifie les entités spatio-temporelles qui se déplacent dans l'espace en fonction du temps sans changer de forme. Elle correspond à la fonction *de déplacement* dans le modèle spatio-temporel. La troisième catégorie, nommée par l'expression *variation de perception* représente les entités spatio-temporelles dont la perception visuelle change. Enfin la quatrième catégorie, nommée par l'expression *variation de la durée de vie* identifie les entités qui vieillissent et qui par conséquent sont moins saillantes. Avant de procéder à une définition succincte de ces quatre catégories permettant d'exploiter la saillance sur des entités spatio-temporelles de l'environnement maritime, une étude préalable sur les ontologies computationnelles pour le domaine maritime a été faite dans les parties ci-après. Ces dernières sont utilisées pour modéliser l'impact des différentes variations sur la dimension sémantique.

2.2 Ontologies computationnelles pour l'environnement maritime

[Malyankar, 1999] fut l'un des premiers précurseurs ayant contribué à la représentation de l'environnement maritime par des ontologies. Dans ses travaux, il souligne la complexité d'intégrer l'information maritime dans des applications à cause de la diversité des sources d'acquisition de données. Il s'est beaucoup intéressé au contexte de la navigation côtière. Il a proposé une description d'une ontologie qui a permis la construction de l'ontologie computationnelle pour l'information maritime. Dans cette description, il parvient à considérer la symbologie des objets des cartes maritimes. Pour mettre en œuvre son ontologie, l'auteur s'est basé sur la norme *S57*⁹ qui définit les standards des cartes nautiques électroniques ou numériques. Cette normalisation a été définie par l'organisation hydrographique internationale OHI. Dans cette normalisation, nous trouvons deux grandes catégories d'informations. La première est constituée de l'ensemble des classes d'objets regroupant tous les concepts de l'environnement maritime [IHO, 2000]. Deux concepts appartiennent à une

9. <http://www.s-57.com/>

Facteurs génériques de saillance	Exemples visuels	Exemples linguistiques
Critère intrinsèque aux unités	Luminosité inhérente	Nom propre
Simplicité	Bonne forme	Interjection
Singleton	Singleton de couleur	Seul antécédent forme
Isolement	Isolement spatial	Apposition
Mise en évidence explicite	Éclairage spot	Accentuation
Construction dédiée	Valorisation du sujet	Représentatif
Placement à un endroit stratégique	Tiers du cadre	Début de l'énoncé
Répétition	Régularité	Répétition
Symétrie	Symétrie	Chiasme ⁸
Rupture dans la continuité	Disposition linéaire	Rythme d'élocution
Infraction à une règle implicite	Disposition incongrue	Raté, erreur
Exploitation d'une norme	Composition classique	Construction usuelle
Rupture dans la continuité	Symétrie	Symétrie
Structuration du message	Ligne force	Construction à topique

Tableau 2.1 – Premier pas vers la détermination d'un ensemble de facteurs génériques de la saillance [Landragin, 2004].

même classe d'objets si elles ont une description similaire dans leur mode d'usage. La deuxième catégorie comprend l'ensemble de classes d'attributs qui peuvent être utilisées pour décrire une classe d'objets de la première catégorie. Une classe d'objets ou une classe d'attributs est identifiée par un acronyme. Un acronyme est une séquence de 6 caractères majuscules. La figure 2.1 ci-dessous illustre les deux grandes catégories et présente une représentation hiérarchique de la norme S57. Cette normalisation dispose de 173 classes d'objets utilisées pour l'extraction des concepts de l'environnement maritime.

Avec ce grand nombre de classes d'objets, la tâche d'extraction et de mise à jour des contenus devient très laborieuse et est généralement faite de façon manuelle par des experts [Maedche *et al.*, 2001]. A l'aide des ontologies et des techniques d'extraction, la complexité de cette tâche peut être réduite et même un non-expert peut participer au peuplement du modèle. Il lui suffira alors de comparer les résultats issus de l'extraction avec ceux extraits des sources originelles. Les premiers résultats

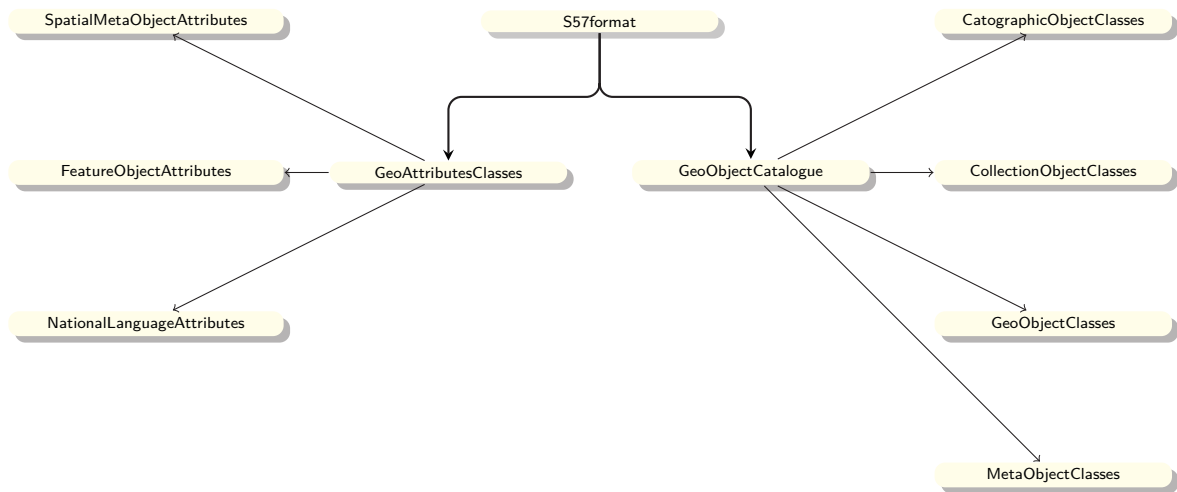


Figure 2.1 – Illustration des différentes catégories de la norme S57

des travaux de [Malyanka, 2002] ont été décrits sous un format *xml* appelé « MIML »¹⁰. Cette représentation était utilisée pour la recherche d'information et par conséquent nécessitait encore un processus de standardisation avant d'être acceptée ou utilisée dans le domaine maritime. L'un des objectifs de ces travaux était de proposer à partir du langage « MIML » des techniques de raisonnement intelligent. L'auteur a cependant été confronté à la complexité du milieu maritime, aux fortes interactions entre les entités et aux diverses règles de navigation à respecter.

Toutes ces règles ou descriptions permettent d'affirmer que la première étape d'une construction de l'ontologie maritime doit provenir d'une ontologie computationnelle [McCarthy, 2000], [Guarino *et al.*, 2009]. De plus, ces règles de navigation (incluant les collections de règles de privilèges et de régulations de trafic, les spécifications environnementales des zones protégées et les zones interdites) sont définies dans des documents [NOAA, 2011b]. Ces documents sont normalisés et fortement détaillés. Leur structuration permet de réaliser une extraction de connaissances fiable et efficace par des experts ou par des algorithmes d'extraction automatique fondés sur le « TAL ». [Malyankar, 2004] déclare plus tard que le second domaine de recherche après la structuration des informations de l'environnement maritime est de permettre aux différents partenaires académiques de travailler sur des problèmes du monde réel, sachant que s'intéresser au monde réel exige de tenir compte des notions de passage à l'échelle, variation dans le temps, de données textuelles ou non textuelles (non structurées et multi-formes) et de données temps réel ou statiques. [Malyankar, 2004] affirme également que le monde dynamique n'est pas seulement modélisé par des relations simples mais aussi pas des paramètres tels que le temps. Pour atteindre ces objectifs, il est nécessaire de bien modéliser le temps et l'espace et de proposer des systèmes fondés sur les bases du web sémantique afin de permettre la portabilité, le partage et le transfert des données. Malheureusement, les modèles existants pour représenter et faire des requêtes sur le temps ne sont pas appropriés à cause de leur incapacité à représenter les relations temporelles d'Allen et de représenter exhaustivement

10. est un langage de balisage pour l'information maritime

la richesse des informations spatiales (voir chapitre 1.1).

Les premiers travaux sur les ontologies maritimes ont permis de comprendre les grandes difficultés de la représentation de l'environnement maritime et de lancer des recherches dans plusieurs domaines afin d'améliorer les formalismes et les techniques de représentation des environnements dynamiques. Les principaux axes de cette recherche se sont orientés dans les formalismes de représentation du temps et de l'espace en logique de description programmable «DLP» sous un format «OWL» et «RDF»¹¹ [Kyzirakos *et al.*, 2012] [Kolas *et al.*, 2009].

2.2.1 Modélisation d'une entité en «DL»

Les récents travaux dans la gestion de l'information comme dans le web de données («linked data») ont permis de générer de nouveaux champs d'intérêt sur les requêtes et les raisonneurs spatio-temporels. Actuellement, la recherche sur la manipulation du web de données est devenue un sujet très important dans le domaine des bases données et en intelligence artificielle. Ces nouveaux travaux s'orientent vers l'intégration du web sémantique dans les sciences de l'information géographique. [Koubarakis *et al.*, 2012] se sont intéressés à la modélisation et aux techniques de requêtes. L'un des principaux défis est de trouver les moyens et méthodes pour rendre les données «RDF» disponibles sur le Web tout en les interconnectant avec les autres sources de données avec pour objectif d'accroître leurs valeurs [Bizer *et al.*, 2009]. Dans cette vision, des recherches ont commencé dans les SIG pour prendre en compte des résultats de la recherche initiée par [Egenhofer, 2002]. [Egenhofer, 2002] exhorte la communauté des SIG à porter une attention particulière à l'évolution des représentations des données spatiales pour le Web. Cette attention doit se concentrer sur cinq grandes thématiques détaillées ci-après :

1 Choix des modèles de données et des langages de requêtes

Le choix du modèle vise à trouver un modèle convenable pour rendre l'information spatiale et temporelle disponible sur le web. En d'autres termes, cela revient à choisir un modèle fiable fondé sur une structuration adéquate. De plus, ce choix doit permettre une extension appropriée des formats «RDF» ou «OWL» en logique de description «DL» et une nouvelle expressivité des requêtes «SPARQL»¹² appropriée au SIG.

2 Architecture et implémentation

L'architecture et l'implémentation doivent identifier les techniques de stockage et les types d'indexation des contenus stockés. Plusieurs techniques de stockage et d'indexation

11. modèle de graphe destiné à décrire de façon formelle les ressources Web et leurs métadonnées, de façon à permettre le traitement automatique

12. SPARQL est un langage de requête et un protocole qui permet de rechercher, d'ajouter, de modifier ou de supprimer des données «RDF» disponibles pat Internet

ont été proposées et les plus utilisées sont les *Rtree* [Guttman, 1984a] et les *Quadtree* [Samet et Webber, 1985]. Ces techniques de stockage sont décrites dans le chapitre 5. Elles seront couplées aux requêtes SPARQL pour une extraction efficace des données spatiales.

3 Interface utilisateur

Cette thématique s'intéresse à étudier comment présenter et développer des interfaces utilisateurs afin de transformer les besoins utilisateurs en des requêtes compréhensibles par le système. Par ailleurs, elle doit aussi proposer une API¹³ facile à déployer pour mettre en place une telle interface utilisateur.

4 Liaison entre les métadonnées existantes

La troisième thématique de recherches s'occupe de la transformation des métadonnées géographiques sous un format « RDF ». Cette transformation permettrait aux métadonnées d'être disponibles dans le web des données facilitant ainsi leur portabilité et leur interopérabilité.

5 Les applications

Pour cette dernière thématique, les chercheurs identifient les différentes applications qui pourront utiliser le web de données sur internet. Les secteurs tels que le transport, l'environnement, l'observation terrestre et la sécurité publique ont été identifiés comme les secteurs pouvant utiliser les résultats de ces travaux. Plusieurs applications de type web services ont déjà mis en œuvre certains résultats de ces recherches. C'est le cas du projet *Geoname*¹⁴ dont l'objectif est de produire « linked data » dans différents domaines. Les données de *Geoname* sont liées aux données de *DBpedia*¹⁵. Les noms des lieux sont proposés en plusieurs langues incluant la latitude, la longitude, l'élévation, les subdivisions administratives et les codes postaux. Toutes ces coordonnées sont définies suivant le système géodésique «WGS84». Le projet *Yago2* [Hoffart *et al.*, 2013] est une proposition d'ontologie où les entités sont associées aux informations spatio-temporelles importantes. Dans *Yago2* la description de la composante temporelle est considérée simplement en instant temporel. Parmi ces deux dernières applications et même parmi les plus récentes, aucune d'entre elles ne prend en compte la complexité de la représentation spatio-temporelle des entités en logique de description telle que proposée par [Allen, 1984, Claramunt et Jiang, 2000, Egenhofer, 2002, Egenhofer, 1989]. Cependant, certaines applications parviennent à intégrer des représentations spatio-temporelles mais utilisent généralement des approches fondées sur des bases de données relationnelles [Kolas *et al.*, 2009] à l'égard des systèmes de représentation de la connaissance fondée sur des

13. interface de programmation ou logiciel fournissant un ensemble normalisé de classes, de méthodes ou de fonctions qui serviront de façade pour le développement d'autres logiciels

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

15. <http://wiki.dbpedia.org/>

ontologies (voir la partie 2.4). Pour résoudre ce problème, le consortium «OGC»¹⁶ a proposé une nouvelle standardisation permettant de représenter l'entité dans ses dimensions spatiales et temporelles en logique de description.

2.2.2 La norme «OGC» en logique de description

Le consortium international des compagnies, agences gouvernementales et universités OGC a pour objectif d'harmoniser le développement des normes d'interopérabilité accessibles au public et de faciliter les recherches sur les thématiques énoncées dans le paragraphe ci-avant. OGC a récemment publié dans l'un de ses documents de spécification sur la représentation des entités spatiales une nouvelle normalisation [Perry et Herring, 2011]. Celle-ci décrit un ensemble de concepts liés aux données géo-spatiales et introduit de nouveaux termes pour l'empreinte spatiale. L'entité géographique est alors considérée comme une abstraction d'un phénomène du monde réel et peut avoir une variété d'attributs [Perry et Herring, 2011] [Perry et Herring, 2012]. Si nous considérons par exemple « le port maritime » comme entité spatiale, ces informations thématiques doivent être : son nom, sa ville d'attache et sa caractéristique ou son empreinte spatiale (c.-à-d. sa localisation sur le globe terrestre). La caractéristique spatiale peut être représentée en utilisant des géométries telles que le point, la ligne et le polygone. Chaque géométrie est définie suivant un système de coordonnées de référence (*WGS84*, *Mercator*, *etc.*) et les coordonnées sont stockées comme une séquence de caractères décrite en « Well Known Text » *WKT* ou « Geography Markup Language » *GML*.

Structuration d'une entité spatiale en DL

La structuration d'une entité spatiale décrite par l'OGC est basée sur des standards de représentation *WKT* et *GML* comme définie dans le document [Perry et Herring, 2012]. Le résultat de cette structuration est codifié sur le format *OWL* et *RDF*. Cependant, ces deux standards de stockage de données spatiales ont besoin de nouveaux langages de requêtes. En d'autres termes, ils devront intégrer toutes les notions liées à l'empreinte spatiale, les relations spatiales (topologique, méréologique, proximité, orientation), les notions liées à la signature temporelle et les relations temporelles proposées par [Allen, 1984]. Cette intégration a été inspirée des modèles basés sur les bases de données relationnelles. Ces derniers ont longtemps travaillé sur ces problématiques. Les approches proposées dans ces modèles ont ainsi facilité la structuration de l'entité spatiale en *RDF* ou *OWL*.

La fiabilité d'une structuration dépend aussi de ses techniques d'extraction de données ou des langages de requête. Dans ce contexte, les langages de requêtes de données ont été définies comme des extensions du langage *SPARQL*. Les deux langages de requêtes les plus répandus sont « *geosparql* » et « *stSparql* ». Ils sont détaillés dans le chapitre 5.

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

Les recherches sur la structuration d'entités spatiales pour le web de données dans le domaine des SIG et de l'intelligence artificielle ont connu une évolution importante ces dernières années. Elles poussent les limites rencontrées dans la modélisation des environnements fortement contraints et dynamiques en offrant une structuration de l'entité spatiale et des langages de requêtes dédiés permettant de capturer les changements d'une entité.

Une fois la structuration établie, il est question de trouver des instances ou des données appropriées au modèle permettant de peupler cette structuration. Cependant, les données de l'environnement proviennent de plusieurs infrastructures ayant des normes ou formats différents (AIS, RADAR, GPS, données cartographiques au format s57).

L'entité vue dans le contexte des SIG est toujours considérée sur trois dimensions : spatiale, temporelle et sémantique [Yuan, 1999, Raza, 2001]. Cependant, les modèles spatio-temporels du domaine maritime s'intéressent moins à la dimension sémantique car les notions d'affordance et de saillance ne sont pas prises en compte. L'absence de ces deux notions dans des applications de navigation peuvent compromettre la sécurité des usagers et des biens de l'environnement maritime. Ceci est dû au fait que ces deux notions ou composantes sont très importantes pour la représentation des environnements fortement dynamiques et particulièrement pour la navigation. Dans le cadre de la navigation maritime, la saillance de l'entité est très utilisée car celle-ci est souvent utilisée pour identifier des points d'intérêts ou remarquables ou « amers ». Il en est pour preuve les instructions nautiques du SHOM qui sont un ouvrage d'aide à la navigation maritime et dont les routes de navigations sont essentiellement fondées sur des points remarquables. Un point d'intérêt ou remarquable ne doit laisser aucune confusion possible pouvant nuire à la sécurité des usagers et doit préserver les critères décrits dans la table 2.1. Ainsi, la modélisation de la saillance d'une entité doit être aussi proche que possible de la réalité (en tenant compte de l'effet du jour et de la nuit ainsi que des conditions de visibilité).

Pour d'atteindre ces objectifs, nous avons proposé une nouvelle structuration de l'entité spatial dans laquelle nous mettons en évidence la saillance visuelle et l'affordance. Cette structuration a été réalisée par une ontologie représentée en logique de description. Elle propose un enrichissement de la structuration d'une entité spatiale proposée par le consortium OGC. La structuration proposée se compose de deux étapes, l'étape de formalisation et l'étape d'implémentation.

2.3 Formalisation de l'ontologie proposée

La formalisation effective de cette ontologie a nécessité de définir les concepts de bases de la grammaire utilisée pour représenter les axiomes en logique de description (DL) et les règles en logique du premier ordre (FOL). La table 2.2 illustre la grammaire associée à la logique de description et propose onze constructeurs. Les constructeurs qui sont fournis permettent de construire de nouvelles expressions ou concepts. Les concepts sont interprétés comme des sous ensembles d'un domaine et les rôles comme des relations binaires sur ce domaine. Une interprétation est définie par l'expression

$I = (\Delta^I, \cdot^I)$ formée d'un ensemble fini non vide Δ^I (le domaine de I) et d'une fonction \cdot^I définie par l'équation 2.1 qui associe chaque concept atomique A de \mathbf{A} à un sous ensemble \mathbf{A}^I de Δ^I , et associe également chaque concept rôle atomique P de \mathbf{P} à un sous ensemble \mathbf{P}^I de $\Delta^I \times \Delta^I$. Dans la table 2.2, les expressions \mathcal{C}, \mathcal{D} sont des concepts déduits des concepts atomiques \mathbf{A} et \mathcal{R}, \mathcal{S} sont des rôles déduits des rôles atomiques \mathbf{P} . Un concept peut être déduit d'un concept atomique soit par un axiome d'inclusion générale de concept ($C \subset A$) ou soit par un axiome de définition de concept ($C \equiv A$).

$$\begin{array}{l}
 \cdot^I: A \in \mathbf{A} \rightarrow \mathbf{A}^I \subset \Delta^I \\
 P \in \mathbf{P} \mapsto \mathbf{P}^I \subset \Delta^I \times \Delta^I
 \end{array}
 \left\{ \begin{array}{l}
 A \text{ est un concept atomique} \\
 \Delta^I \text{ l'ensemble des instances de } A \\
 P \text{ est un rôle atomique} \\
 \Delta^I \times \Delta^I \text{ l'ensemble des instances de } P
 \end{array} \right. \quad (2.1)$$

Expression en DL	Définition	Interprétation
$\mathcal{C}, \mathcal{D} \rightarrow \mathcal{A}$	Concept atomique	$A^I \subseteq \Delta^I$
\top	Haut	$\top^I = \Delta^I$
\perp	Bas	$\perp^I = \emptyset$
$\neg \mathcal{C}$	Complémentaire	$(\neg \mathcal{C})^I = \Delta^I \setminus \mathcal{C}^I$
$\mathcal{C} \sqcap \mathcal{D}$	Intersection	$(\mathcal{C} \sqcap \mathcal{D})^I = \mathcal{C}^I \sqcap \mathcal{D}^I$
$\mathcal{C} \sqcup \mathcal{D}$	Union	$(\mathcal{C} \sqcup \mathcal{D})^I = \mathcal{C}^I \sqcup \mathcal{D}^I$
$\forall \mathcal{R}. \mathcal{C}$	Quantificateur universel	$(\forall \mathcal{R}. \mathcal{C})^I = \{a \in \Delta^I \mid \forall b. \mathcal{R}^I(a, b) \Rightarrow \mathcal{C}^I(b)\}$
$\exists \mathcal{R}. \mathcal{C}$	Quantificateur existentiel	$(\exists \mathcal{R}. \mathcal{C})^I = \{a \in \Delta^I \mid \exists b. \mathcal{R}^I(a, b) \wedge \mathcal{C}^I(b)\}$
$\mathcal{R}, \mathcal{S} \rightarrow \mathcal{P}$	Rôle atomique	$P^I \subseteq \Delta^I \times \Delta^I$
$\exists^{n \geq} \mathcal{R}$	Rôle à cardinalité maximale	$(\exists^{n \geq} \mathcal{R})^I = \{a \in \Delta^I \mid \{(a, b) \in \mathcal{R}^I\} \geq n\}$
$\exists^{n \leq} \mathcal{R}$	Rôle à cardinalité minimale	$(\exists^{n \leq} \mathcal{R})^I = \{a \in \Delta^I \mid \{(a, b) \in \mathcal{R}^I\} \leq n\}$
$\exists \mathcal{R}^-$	Rôle réciproque	$(\exists \mathcal{R}^-)^I = \{(a, b) \in \Delta^I \times \Delta^I \mid (b, a) \in \mathcal{R}^I\}$

Tableau 2.2 – Syntaxe et sémantique pour la logique de description [Diego, 1996a]

A partir de la syntaxe décrite par les constructeurs ci-avant, la base de connaissance en logique de description est définie comme un triplet $\Sigma = (Tbox, Rbox, Abox)$. Dans cette base de connaissances Σ , $Tbox$ désigne l'ensemble des axiomes terminologiques. La $(Rbox)$ désigne l'ensemble de rôles et

(*ABox*) désigne les instances. Cette formalisation est utilisée pour définir ou proposer des ontologies opérant dans différents domaines.

L'utilisation de l'ontologie pour représenter les entités de l'espace maritime nécessite de considérer chaque entité dans sa structure *atomique*. Une entité est dite *atomique* lorsque sa composante spatiale est associée à au plus une empreinte spatiale. Une fois la représentation atomique établie, nous utiliserons les fonctions, les constructeurs et les règles pour capturer certaines informations relatives à la sémantique telles que la variation spatiale, la variation de l'emplacement spatial, la variation de perception et la variation de la durée de vie. Pour le faire, nous représentons l'entité en logique de description et les règles dans la logique du premier ordre (FOL). La figure 2.2 est une illustration de l'ontologie proposée. Dans cette représentation, chaque entité est considérée comme une entité atomique nommée « AtomicEntity ». Celle-ci peut avoir exactement une dimension temporelle, une dimension sémantique et exactement une dimension spatiale. Cet axiome est indiqué par l'équation 2.2. Afin de mieux décrire l'entité spatiale, il est important de réaliser une formalisation de chaque dimension de celle-ci.

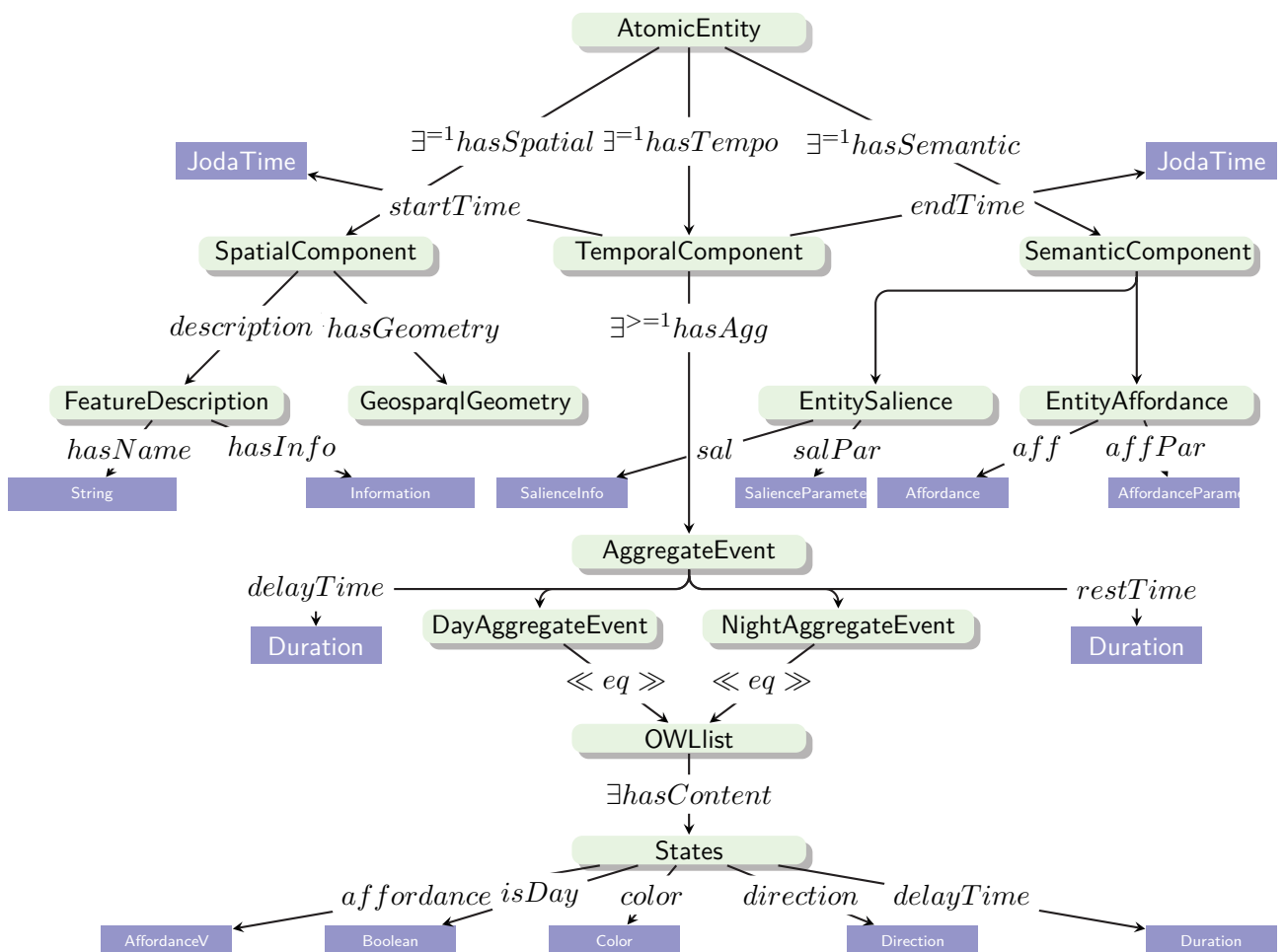


Figure 2.2 – Ontologie proposée pour une entité atomique ou élémentaire

$$\begin{aligned}
 \text{BasicEntity} \quad &\subseteq \exists^=1 \text{hasTempo.TemporalComponent} \wedge \\
 &\exists^=1 \text{hasSemantic.SemanticComponent} \wedge \\
 &\exists^=1 \text{hasSpatial.SpatialComponent}
 \end{aligned} \tag{2.2}$$

2.3.1 Formalisation de la dimension temporelle

Les modifications de la saillance (perception) et la durée de vie d'une entité exigent de proposer une dimension temporelle afin de capturer ses changements. [Yuan, 1999] propose un modèle relationnel pour représenter ces différents changements et montre l'intérêt de représenter ces changements pour des applications réelles. Nous avons proposé un axiome (voir équation 2.3), permettant de représenter ces changements.

$$\begin{aligned}
 \text{TemporalComponent} \quad &\subseteq \exists^{\geq 1} \text{hasAgg.AggregaEvent} \\
 &\exists \text{starTime.jodaTime} \wedge \\
 &\exists \text{endTime.jodaTime} \\
 \text{AggregaEvent} \quad &\subseteq \text{DayAggregaEvent} \vee \text{NightAggregaEvent} \wedge \\
 &\text{restTime}\{Duration\} \wedge \text{delayTime}\{Duration\}
 \end{aligned} \tag{2.3}$$

La dimension temporelle présente deux type de propriétés :

- les propriétés *starTime* et *endTime* sont les « dataproperty »¹⁷, elles désignent respectivement la naissance et la fin de vie de l'entité,
 - la propriété *hasAgg* est une relation binaire ou « objectproperty »¹⁸ qui permet de représenter les événements répétitifs des entités grâce au concept d'agrégations d'événements *AggregaEvent*.
- . Les propriétés *starTime* et *endTime* sont représentées par le type de données *jodaTime*¹⁹. Ce dernier est un type de données temporelles utilisé pour représenter une expression temporelle de l'ordre de la milliseconde. Il est matérialisé par le septuple (année, mois, jour, heure, minute, seconde, milliseconde) ou chaque composante du septuple contient la valeur de l'information temporelle que l'on souhaite matérialiser. Par exemple on peut souhaiter marquer le début d'une marée vive en pleine mer. On dira : la marée vive a commencé le 12 juin 2014 à 22 heures 0 min 0 seconde 0 milliseconde et elle est alors représentée en *jodaTime* par le septuple (2014, 6, 12 ,22, 0, 0, 0).

A l'aide de *jodaTime*, nous avons représenté les 11 relations d'Allen [Allen, 1984] et proposé les fonctions qualitatives (x heures plus tard ou plus tôt, où x est un entier) suivant la logique du premier ordre (FOL) utilisant un vecteur temporel dans le paragraphe suivant. Celle-ci peut se

17. propriété ou une relation dont sa portée ou son image est un type de données prédéfini dans le système (P. ex. chaine de caractère, entier, booléen, etc.)

18. propriété ou une relation dont sa portée ou son image est un concept du modèle

19. //www.joda.org/

faire de façon absolue ou de façon relative par rapport au temps courant. Le type *jodaTime* est modélisé par un vecteur temporel (voir la table 2.3).

La composante temporelle	Borne inférieure	Borne supérieure
<i>Année</i>	0	Ouverte
<i>Mois</i>	1	12
<i>Jour</i>	1	variant entre : {28,29,30,31} selon le mois
<i>Heure</i>	0	24
<i>Minute</i>	0	60
<i>Seconde</i>	0	60
<i>Milliseconde</i>	0	1000

Tableau 2.3 – Illustration du vecteur temporel et du domaine de définition de chacune de ses composantes

Vecteur temporel

Un vecteur temporel permet de représenter un instant temporel matérialisé par le septuple (année, mois, jour, heure, minute, seconde, milliseconde). Ce vecteur est formé d'une séquence de sept éléments représentant de la gauche vers la droite les différentes divisions du temps. Chaque composante est constituée d'une borne inférieure et d'une borne supérieure (seule l'année ne possède pas de borne supérieure). La table 2.3 illustre un vecteur temporel et les différentes bornes associées à chacune de ses composantes. Toute opération sur ce vecteur temporel doit garantir que les bornes sont respectées, à défaut il faut procéder à une harmonisation du vecteur. Nous avons intégré deux fonctions d'harmonisation, permettant de vérifier que les bornes supérieure et inférieure sont respectées.

Considérons que nous souhaitons représenter l'expression suivante sur un axe temporel : la marée vive sera signalée dans deux heures de temps et nous sommes actuellement le 16 juin 2014 à 23 heures 50 min 15 secondes 40 millicondes. Dans cette situation, l'heure de signalisation de la marée sera le 17 juin 01 heure 50 minutes 15 secondes 40 millisecondes car nous avons procédé à une harmonisation de la borne supérieure de la composante heure.

Une expression peut être référencée sur l'axe des temps en utilisant deux référentiels différents [Raza, 2001] à savoir :

- *le référentiel absolu*, dans l'exemple précédent, l'expression le 17 juin 01 heure 50 minutes 15 secondes 40 millisecondes est référencée de façon absolue dans l'axe des temps.
- *le référentiel relatif* : il existe deux approches de référencement relatif d'une expression temporelle dans l'axe des temps.

- 1 la première approche consiste à choisir l'instant courant comme référentiel. Dans l'exemple présenté ci-avant nous avons utilisé un référentiel relatif.
- 2 la deuxième approche utilise la composante du septuple temporel qui est citée dans l'expression comme référentiel. Supposons par exemple que nous ayons plutôt déclaré : « la marée vive a été signalée à la deuxième heure de la journée » et que nous soyons toujours le 16 juin 2014 à 23 heures 50 minutes 15 secondes 40 microsecondes. La composante citée dans notre modélisation est *l'heure*. Les composantes du septuple qui apparaissent avant le terme cité restent identiques au temps courant : l'année, le mois, le jour. Celles qui apparaissent après sont mises à zéro (minute, seconde, microseconde) et la composante citée prend la valeur indiquée dans l'expression. Ainsi, dans notre modélisation, l'expression ci-avant est représentée par le 16 juin 2014 à 2 heures 0 minute 0 seconde 0 milliseconde et correspond au septuple (2014, 06, 16, 2, 0, 0, 0).

Les fonctions que nous avons proposées pour représenter une expression temporelle jt dans un axe temporel sont $absolute(jt)$ et $relative(jt)$ où jt est de type *jodaTime*. La fonction $relative(jt)$ utilise un référentiel relatif et la fonction $absolute(jt)$ utilise un référentiel absolu.

Le vecteur temporel défini nous a permis de représenter différentes expressions temporelles que sont : un intervalle temporel et une durée de temps. Un intervalle temporel est la réunion de deux instants temporels. Soient $jt1$ et $jt2$ deux instants temporels avec ($jt1 < jt2$) et *TemporalInterval* un intervalle temporel, l'axiome associé à *TemporalInterval* est indiqué par l'équation 2.4. Dans le même contexte, pour déterminer une durée de temps entre deux instants temporels $jt1 = (a_6, a_5, a_4, a_3, a_2, a_1, a_0)$ et $jt2 = (b_6, b_5, b_4, b_3, b_2, b_1, b_0)$, nous devons initialement appliquer un opérateur différence \div entre leurs deux vecteurs temporels.

Considérons $jt1$ et $jt2$ deux vecteurs temporels dont souhaiterait calculer la différence \div définie par $jt \div jt2$. Le calcul de cette différence revient donc à déterminer les coefficients du vecteur temporel $res = (res_6, res_5, res_4, res_3, res_2, res_1, res_0)$ où $res = jt \div jt2$. Une fois que les coefficients ou les coordonnées du vecteur temporel res sont déterminés, la durée entre les deux instants est exprimée en milliseconde et calculée par le polynôme d'ordre au plus égal à 5 défini par l'équation 2.5. L'intérêt de calculer la durée de temps jusqu'à la milliseconde permet de représenter les changements décrits ci-avant avec de bonnes précisions.

$$TemporalInterval \subseteq jt1 \vee jt2 \quad (2.4)$$

$$durée = res_0 + 1000(res_1 + 60(res_2 + 60(res_3 + 24(res_4 + res_5 + res_5)))) \quad (2.5)$$

La détermination des coefficients $res_6, res_5, res_4, res_3, res_2, res_1$ et res_0 du vecteur temporel résultant res exige plusieurs étapes afin de réaliser l'opération \div définie entre deux vecteurs temporels. Ces étapes sont les suivantes :

- le calcul du nombre de jours écoulés res_6 entre les années a_6 et b_6 est déterminé par l'expression $res_6 = \sum_{i=a_6+1}^{b_6-1} Q_i$ où la valeur Q_i est égale à 366 si l'année est bissextile et 365 au cas contraire,
- le calcul du nombre de jours écoulés res_5 entre les mois d'une même année est défini par le nombre de jours ($nbajt1$) écoulés partant du mois qui suit a_5 jusqu'au douzième mois de l'année a_6 . Ceci est ajouté au nombre de jours écoulés ($nbajt2$) partant du début de l'année jusqu'au mois qui le précède b_5 dans la même année b_6 . Dans la formule $(nbajt1) = \overline{A_5} \times \overline{Mois}$ définie comme le produit vectoriel $\overline{A_5}$ et de \overline{Mois} où $\overline{A_5} = (0, 0, 0, \dots, 1, 1, 1, 1, 1)$ est un vecteur de valeurs binaires de taille 12 avec les $a_1 + 1$ dernières composantes égales à 1 et le reste à 0, dans la même formule $\overline{Mois} = (31, 28, 29, 30, \dots, 31)$ désigne un vecteur de 12 composantes dont les valeurs correspondent aux nombres de jours de chaque mois d'une année dans l'ordre croissant (*Janvier ... Décembre*).

Dans le même contexte, la formule $(nbajt2) = \overline{B_5} \times \overline{Mois}$, où $\overline{B_5} = (1, 1, 1, \dots, 0, 0, 0, 0)$ est un vecteur de valeurs binaires de taille 12 avec les $B_5 - 1$ premières composantes à 1 et le reste à 0. La valeur de res_5 est ainsi donnée par l'équation 2.6.

$$res_5 = nbajt1 + nbajt2 \quad (2.6)$$

- le calcul du nombre de jours écoulés res_4 au cours des mois a_5 et b_5 est défini comme le nombre de jours $nbmjt1$ écoulés, partant du jour qui suit a_4 jusqu'au dernier jour du mois a_5 . Ceci est ajouté au nombre de jours $nbmjt2$ écoulés partant du premier jour du mois jusqu'à la fin du jour $(b_6, b_5, b_4 - 1, 0, 0, 0, 0)$ qui précède b_4 . Nous avons ainsi $(nbmjt1) = \overline{A_4} \times \overline{Mois} - (a_4 + 1)$ où $\overline{A_4} = (0, 0, 0, \dots, 1, 0, 0, 0, 0, 0)$ est un vecteur de valeurs binaires de taille 12 avec uniquement $a_5^{ième}$ composante à 1 et $nbmjt2 = b_4$. La valeur de res_4 est ainsi donnée par l'équation 2.7.

$$res_4 = nbmjt1 + nbmjt2 \quad (2.7)$$

- les nombres d'heures res_3 , de minutes res_2 , de secondes res_1 et de millisecondes res_0 sont calculés en leur affectant respectivement en fonction des unités temporelles (heure, minutes, secondes et millisecondes) les composantes de l'instant temporel déterminées par la différence des instants temporels ($jt1s$) du jour qui suit a_4 marqué sur l'axe temporel par la fonction $relatif(jt1s)$ et de l'instant temporel $jt1$ ayant pour formule $(relatif(jt1s) - relatif(jt1))$, ceci ajouté à la différence des instants temporels $jt2$ et de $(jt2p)$ du jour qui précède b_4 marqué sur l'axe temporel par la fonction ayant pour formule $relatif(jt2p)$.

Les changements observés dans des modèles spatio-temporels sont considérés comme une agrégation d'événements [Pan, 2005]. Une agrégation d'événements est un ensemble d'événements qui se produisent de façon répétitive. Par exemple, chaque marée (morte ou vive) indique toujours les six niveaux de la mer. L'agrégation d'événements a été initialement proposée par [Pan, 2005] et celle qu'il propose ne prend pas en compte la durée de la période pendant laquelle l'événement se produit avant sa disparition. Par la suite, les événements observés sont généralement perçus (la saillance visuelle) différemment de jour et de nuit. Dans le cas des bouées, en journée leur saillance

est indiquée par leur couleur ou leur forme et pendant la nuit leur saillance est indiquée par leur feu de signalisation. Ainsi, une agrégation d'événements se décompose en deux sous-agrégations, celle de la nuit et celle du jour, présentée par l'équation 2.8. Chaque agrégation possède une propriété *delayTime* qui désigne la durée de temps pendant laquelle elle est en activité et la propriété *restTime* qui indique la durée de temps pendant laquelle elle reste inactive.

$$\begin{aligned} \text{AggregaEvent} &\subseteq \text{DayAggregaEvent} \vee \text{NightAggregaEvent} \\ &\text{restTime.}\{Duration\} \wedge \text{delayTime.}\{Duration\} \end{aligned} \quad (2.8)$$

L'absence ou la présence de la luminosité ambiante détermine le jour et la nuit utilise l'axiome proposé pour la saison (voir l'axiome 2.9).

$$\begin{aligned} \text{Saison} &\equiv \{\text{Printemps}, \text{Eté}, \text{Automne}, \text{Hiver}\} \\ \text{Printemps} &\subseteq \text{saisonP.}\{\Delta T^1\} \wedge \\ &\text{dayP.}\{\delta t^1 \text{unitPeriode}\} \wedge \\ \text{Eté} &\subseteq \text{saisonP.}\{\Delta T^2\} \wedge \\ &\text{dayP.}\{\delta t^2\} \wedge \\ \text{Automne} &\subseteq \text{saisonP.}\{\Delta T^3\} \wedge \\ &\text{dayP.}\{\delta t^3\} \wedge \\ \text{Hiver} &\subseteq \text{saisonP.}\{\Delta T^4\} \wedge \\ &\text{dayP.}\{\delta t^4\} \end{aligned} \quad (2.9)$$

A partir des différentes instances d'une saison et leurs paramètres durée de la saison, durée du jour pendant cette saison, nous allons utiliser la règle proposée par l'équation 2.10 en logique du premier ordre pour déterminer s'il y a la luminosité ambiante ou non. La règle $isDay(t, x, z, y, s^k)$ a pour paramètre t l'instant courant et $s^k = (\Delta T^k, \delta t^k)$ où ΔT^k indique la période de la saison, δt^k la période de nuit au cours de cette saison et s^k désignant une instance de la liste énumérée S c.-à-d. $s^k \in S = \{\text{Printemps}, \text{Eté}, \text{Automne}, \text{Hiver}\}$, où chaque instance est identifiée par l'entier $k=1..4$. Dans cette règle, la valeur booléenne x est à vraie si nous sommes le jour et prend la valeur fausse si nous sommes la nuit. Les périodes de nuit et d'une saison sont représentées par la fonction $relative(jt)$ (voir le paragraphe 2.3.1) et varient chaque jour.

$$\begin{aligned} isDay(t, x, z, y, s^k) &\leftarrow x \\ &[\exists y = faux \wedge t \in \Delta T^k \rightarrow y = vrai] \wedge \\ &[\exists z = faux \wedge t \in \delta T^k \rightarrow z = vrai] \wedge \\ &[x = z \wedge y] \end{aligned} \quad (2.10)$$

Cette présence de la luminosité joue un rôle dans le choix de l'agrégation. Nous distinguons deux sortes d'agrégations, l'agrégation de nuit $aggN$ et l'agrégation de jour $aggD$. La règle définie par l'équation 2.11 modélise le choix de l'agrégation selon l'instant courant. Considérons le prédicat $currentAgg$ indiquant l'agrégation courante représentée par la variable A .

$$\begin{aligned}
currentAgg(t, x, z, y, s^k, A) &\leftarrow A \\
&\left[(isDay(t, x, z, y, s^k) == vrai) \rightarrow A = aggD \right] \wedge \\
&\left[(isDay(t, x, z, y, s^k) == faux) \rightarrow A = aggN \right]
\end{aligned} \tag{2.11}$$

Au courant d'une agrégation d'événements, l'entité peut subir plusieurs états différents modifiant ainsi sa saillance et son affordance. Une agrégation est alors associée à une liste d'états. La représentation d'une liste ou d'autres conteneurs en «OWL» doit respecter la structuration proposée par [Drummond *et al.*, 2006] afin de conserver la décidabilité du langage «DL». Soit $State_i^d$ un état de jour situé à la $i^{ième}$ position de la liste $OWLlist$ de taille M . L'agrégation des événements de jour est définie par l'équation 2.12. De la même manière, nous pouvons aussi définir l'agrégation des événements de nuit à partir du $j^{ième}$ état de nuit $State_j^n$ d'une liste de taille N (voir équation 2.13).

$$\begin{aligned}
DayAggregaEvent &\equiv OWLlist \wedge \exists hasContents.\{State_0^d\} \quad hasNext \quad \exists(\\
&OWLlist \wedge \exists hasContents.\{State_1^d\} \quad hasNext \quad \exists(\\
&OWLlist \wedge \exists hasContents.\{State_2^d\} \quad hasNext \quad \exists(\\
&OWLlist \wedge \exists hasContents.\{State_3^d\} \quad hasNext \quad \exists(\\
&\dots \\
&OWLlist \wedge \exists hasContents.\{State_{M-2}^d\} \quad hasNext \quad \exists(\\
&OWLlist \wedge \exists hasContents.\{State_{M-1}^d\} \quad hasNext \quad EmptyList))))))
\end{aligned} \tag{2.12}$$

$$\begin{aligned}
NightAggregaEvent &\equiv OWLlist \wedge \exists hasContents.\{State_0^n\} \quad hasNext \quad \exists(\\
&OWLlist \wedge \exists hasContents.\{State_1^n\} \quad hasNext \quad \exists(\\
&OWLlist \wedge \exists hasContents.\{State_2^n\} \quad hasNext \quad \exists(\\
&OWLlist \wedge \exists hasContents.\{State_3^n\} \quad hasNext \quad \exists(\\
&\dots \\
&OWLlist \wedge \exists hasContents.\{State_{N-2}^n\} \quad hasNext \quad \exists(\\
&OWLlist \wedge \exists hasContents.\{State_{N-1}^n\} \quad hasNext \quad EmptyList))))))
\end{aligned} \tag{2.13}$$

Soit un état courant $State_i^c$ de l'agrégation de nuit ou de jour avec $c \in \{n, d\}$ et $l = 1 \dots n$ où $n \in \{M, N\}$ est égal au nombre d'états. Cet état correspond à la situation de l'entité au moment où un observateur ou sujet la perçoit. Il est défini par l'équation 2.14. Dans l'axiome d'un état, l'ensemble des valeurs de l'affordance est défini par « AffordanceV » et ses instances ont été restreintes aux actions ou affordances (continu, stop, back, maneuver) qui sont les décisions possibles proposées par l'expert (voir chapitre 3). A ces termes, nous avons ajouté le terme « inconnue » qui indiquera que l'action n'a pas été définie par l'expert. Dans ce cas, notre modèle ira rechercher dans les sources de données l'action correspondante. Cette technique de recherche de la sémantique fera

l'objet du chapitre 3.

$$\begin{aligned}
 State & \subseteq \text{affordance}.\{AffordanceV\} \wedge \\
 & \text{isDay}.\{Boolean\} \wedge \text{color}.\{Color\} \wedge \\
 & \text{direction}.\{Direction\} \wedge \text{delayTime}.\{Duration\} \\
 AffordanceV & \equiv \{\text{arrêt, manoeuvre, continue, retour, inconnue}\} \\
 Boolean & \equiv \{\text{vrai, faux}\} \\
 Color & \equiv \{\text{vert, bleu, blanc, noir, rouge, orange, \dots}\} \\
 Duration & \equiv \{\text{unityear, unitmonth, unitweek, unitday} \\
 & \text{unithour, unitmin, unitsec, unitsec}\} \\
 Direction & \equiv \{\text{North, South, East, West} \\
 & \text{North - West, North - East, South - West, South - East}\}
 \end{aligned} \tag{2.14}$$

Pour déterminer l'état courant de l'entité nous utilisons la règle définie par l'équation 2.15. Dans cette règle, le paramètre t désigne l'instant courant, ta désigne la durée de l'agrégation, tb désigne l'instant de mise en œuvre de l'agrégation, L^w désigne l'état courant solution, L désigne la liste des états de jour et de nuit (équation 2.13 et 2.12) et la valeur R indique le temps restant obtenu par division euclidienne de la différence de l'instant courant et l'instant de naissance de l'entité ($|t - tb|$) sur la durée d'une agrégation ta . Concernant la fonction $extract(R, L, L^w)$, elle permet de déduire l'état courant de la liste L à partir de R . Soit $L(q)$ un élément de la liste L avec $q = 1 \dots n$ où n est la taille de la liste et $L^t(q)$ le temps d'activité de l'élément $L(q)$. L'état L^w déterminé à un instant t parmi la liste d'états dont le reste R est indiqué par l'équation 2.15, représente le premier état de la liste qui satisfait la relation (voir équation 2.16).

$$\begin{aligned}
 currentState(t, ta, tb, L, R) & \leftarrow L^w \\
 & [\exists R = (t - tb) \text{ modulo}(ta)] \wedge \\
 & [L \neq \text{vide} \rightarrow L^w = extract(R, L, L^w)]
 \end{aligned} \tag{2.15}$$

$$\begin{aligned}
 extract(R, L, l) & \leftarrow (\exists min = n + 1 \wedge \\
 & [\forall \sum_{q=1}^n L^t(q) - R > 0 \wedge q < min] \\
 & \rightarrow min = q) \wedge \\
 & \rightarrow l = L(min)
 \end{aligned} \tag{2.16}$$

2.3.2 Formalisation de la dimension spatiale

La composante spatiale est définie par l'empreinte spatiale représentée par une géométrie suivant la norme décrite par le consortium *OGC* [Perry et Herring, 2011] et d'une propriété *description* du concept *FeatureDescription* (voir les axiomes de la règle 2.17). Le concept *FeatureDescription* permet de renseigner le nom de l'entité par la propriété *hasName* et des informations liées à l'entité telles que (le type de projection, le type de géométrie, etc.).

$$\begin{aligned}
SpatialComponent &\subseteq description.FeatureDescription \wedge \\
&hasGeometry.GeosparqlGeometry \\
FeatureDescription &hasName.\{String\} \wedge hasInfo.\{Information\}
\end{aligned} \tag{2.17}$$

2.3.3 Formalisation de la dimension sémantique

La sémantique est une composante majeure dans cette thèse. Elle nous permet de déterminer l'affordance et la saillance associées à une entité. De plus, la définition de la dimension sémantique exige de considérer un ensemble de contraintes qui doivent être satisfaites à tout moment afin de déterminer la saillance et l'affordance associés à l'entité. Ces contraintes sont modélisées par les concepts (*saillanceParameter*) pour la saillance et *AffordanceParameter* pour l'affordance. L'équation 2.18, représente les axiomes associées à la dimension sémantique. Dans ces axiomes, la propriété *distanceVisibility* permet d'identifier la distance au dessous de laquelle l'entité est visible et la propriété *surfaceVisibility* désigne la surface minimale à laquelle l'entité est visible. Cette surface est le plus souvent déterminée par des algorithmes d'inter-visibilité dans les *SIG*.

$$\begin{aligned}
SemanticComponent &\subseteq EntitySalience \vee EntityAffordance \\
EntitySalience &\subseteq sal.\{SalienceInfo\} \wedge salPar.\{SalienceParameter\} \\
EntityAffordance &\subseteq aff.\{Affordance\} \wedge affPar.\{AffordanceParameter\} \\
SalienceParameter &\subseteq distanceVisibility.\{Double\} \wedge surfaceVisibility.\{Double\} \\
AffordanceParameter &\subseteq \{Fairly, Meanly, Huge, VeryHuge\}
\end{aligned} \tag{2.18}$$

2.3.4 Formalisation des changements d'une entité ou de sa perception

Dans cette partie, nous formalisons les changements de l'entité spatio-temporelle et de sa perception sur la sémantique (saillance et affordance). Ces derniers ont un impact sur la sémantique qu'elle produit dans son environnement (voir partie 2.1.1). Ces variations sont les suivantes : la variation de l'empreinte spatiale, la variation de l'emplacement spatial, la variation de perception et la variation de la durée de vie.

- *la variation de l'empreinte spatiale* s'intéresse aux entités spatiales qui subissent une ou plusieurs interactions avec d'autres entités dans l'environnement. Ces interactions ont été modélisées par des modèles spatio-temporelles fondés sur le changement [Claramunt *et al.*, 1997a]. Parmi ces changements, il y a les changements qui modifient l'empreinte spatiale de l'entité (élongation, apparition, disparition, déformation, etc.). Cette variation de l'empreinte spatiale peut modifier les critères de saillance énoncés dans la table 2.1 et ainsi compromettre la navigation. Par exemple, une entité qui subit une variation importante peut complètement disparaître de l'environnement détruisant ainsi la saillance ou l'affordance qui lui sont associées. La règle définie par l'équation 2.19 modélise la variation spatiale d'une entité (p. ex. une érosion importante

peut faire disparaître une dune maritime). Cette variation devient importante quand l'intersection $intersection(G, G', g)$ entre l'empreinte initiale et celle obtenue à la suite d'un changement de l'entité est égale à l'ensemble vide (« \emptyset »). Dans cette règle, $emprS(E, SP, G, G', g, x)$, E représente l'entité, SP est la dimension spatiale, G est l'empreinte spatiale, G' est l'empreinte spatiale de l'entité après avoir subi un changement, g est le résultat de l'intersection et x est le coefficient permettant d'évaluer l'impact de la variation spatiale. Si la valeur $x = 0$, la disparition dans l'environnement de l'entité est complète et si $x = 1$ l'entité n'a subi aucune variation de l'empreinte spatiale.

$$\begin{aligned}
 emprS(E, SP, G, G', x) \quad \leftarrow x = 1 \\
 & [AtomicEntity(E)] \wedge \\
 & [hasSpatial(E, SP) \wedge hasGeometry(SP, G)] \wedge \\
 & [intersection(G, G', g)] \wedge \\
 & \left[\rightarrow x = \frac{Surf(g)}{Surf(G)} \right]
 \end{aligned} \tag{2.19}$$

– la variation de l'emplacement spatial s'intéresse à l'emplacement de l'entité ou à son déplacement. Elle s'observe lorsque le centre de gravité de l'entité se déplace dans l'espace. Elle a un impact direct sur la distance de visibilité à laquelle l'observateur ou le sujet voit l'entité spatiale. Par exemple, plus une entité est loin de son observateur, plus sa visibilité est réduite et par conséquent sa saillance visuelle diminue. La saillance peut être inexistante si la distance qui sépare l'observateur et l'entité est supérieure à la distance de visibilité maximale définie dans le concept de *SaillanceParameter*. Cette fonction de la variation de l'emplacement spatial est décrite par la règle de l'équation 2.20. Dans la règle $implS(E, SP, O, G', distanceVisibility, x)$, E représente l'entité, SP est la composante spatiale, G l'empreinte spatiale, O est la position de l'observateur, $distanceVisibility$ est la distance de visibilité au-dessus de laquelle l'entité n'est plus visible, x est le rapport permettant d'évaluer le déplacement. Si la valeur de $x > 1$ alors l'entité n'est plus visible à partir du point d'observation O .

$$\begin{aligned}
 implS(E, SP, O, G', distanceVisibility, x) \quad \leftarrow x = 1 \\
 & [AtomicEntity(E)] \wedge \\
 & [hasGeometry(SP, G)] \wedge \\
 & [\exists Odistance(G, O, d)] \wedge \\
 & \left[\rightarrow x = \frac{distanceVisibility}{d} \right]
 \end{aligned} \tag{2.20}$$

– la variation de la durée de vie se définit en fonction du temps valide et permet de calculer la durée de vie de l'entité à un moment donné. Cette durée de vie peut être utilisée afin de vérifier le vieillissement de l'entité (voir la table 2.1). Une entité vieillissante devient moins saillante et doit être signalée aux marins afin de ne pas compromettre la saillance visuelle. La variation de la durée de vie $varD(E, tf, t, x)$ prend en entrée l'entité E , l'instant de fin de vie de l'entité tf , la valeur Booléenne x est vraie si la fin de vie de l'entité est atteint et fausse sinon.

$$\begin{aligned}
varD(E, tf, t, x) \leftarrow & x = faux \\
& [AtomicEntity(E)] \wedge \\
& [endTime(E, tf)] \wedge \\
& [t > tf] \wedge \\
& [\rightarrow x = vrai]
\end{aligned} \tag{2.21}$$

- la variation de la perception de couleurs ou ces signalisations permet de déterminer la couleur ou le signal généré par l'entité en un moment donné. La variation de perception de couleur est très utilisée dans le domaine maritime et la couleur des entités diffère le jour et la nuit. La nuit l'entité est identifiée par des signaux cadencés et la journée par sa forme et la couleur. Considérons un feu de signalisation en la navigation terrestre, elle comprend trois variations de couleurs au cours d'un cycle. Elle peut passer du vert, au rouge et au jaune avec des temps de latence différents pour chaque couleur. La variation de couleurs est déterminée par l'équation 2.22 qui permet d'extraire la couleur de l'état courant (voir la règle 2.14). La variation de perception de couleur de l'entité correspond à la couleur l'état courant d'une entité dans lequel se trouve une entité en un moment donné. L'état courant de l'entité se détermine par l'équation 2.15.

$$\begin{aligned}
varC(E, tf, t, x, ta, tb, L, R) \leftarrow & \exists c \in Color \wedge c = State.color \\
& [\exists t varD(E, tf, t, x) = vrai] \wedge \\
& [L \neq vide \rightarrow State = currentState(t, ta, tb, L, R)]
\end{aligned} \tag{2.22}$$

2.4 Implémentation de l'ontologie proposée

L'implémentation de notre ontologie a été réalisée en utilisant le format OWL afin de respecter la logique de description. Les règles ont été réalisées avec du langage SWRL dont la syntaxe est décrite par l'équation 2.24 basé sur le système JENA. Il existe plusieurs systèmes permettant de réaliser l'ontologie et ces règles. Parmi ces systèmes, le plus populaire est JENA et est couramment utilisé pour développer les requêtes sur des « linked data ». Bien que JENA soit le système que nous avons choisi dans cette thèse plusieurs autres systèmes existent. Le tableau 2.4 résume les différents systèmes existant dans le domaine du web sémantique avec leurs caractéristiques. Les raisonneurs de ces systèmes utilisent généralement deux stratégies à appliquer sur des connaissances. Les stratégies en avant et en arrière sont les deux stratégies utilisées.

2.4.1 Les différents systèmes

Les systèmes les plus répandus sont décrits dans les paragraphes ci-dessous.

JENA

Le système JENA intègre un moteur d'inférence hybride (c.-à-d. un processus de raisonnement basé sur l'algorithme de tableau²⁰ [Baader et Sattler, 2001] s'appliquant de l'avant vers l'arrière). Il propose aussi plusieurs raisonneurs, à savoir un raisonneur transitif, un raisonneur sur des règles *rdfs*²¹, un raisonneur sur OWL et un raisonneur des règles génériques. Dans le cadre de cette thèse nous utilisons le moteur d'inférence SWRL qui est une extension de JENA²² approprié pour l'inférence en local (c.-à-d. directement sur le fichier OWL) et le raisonneur utilisé est *pellet* qui reconnaît le langage SWRL.

Système de raisonnement	Type	Langage supporté	Mécanisme d'inférence	RDF(S) supporté	Implémentation
<i>CWM</i>	homogène	N3, N-Triples, RDF/XML	Avant	Axiomes	Python
<i>Euler</i>	homogène	N3	Arrière	Axiomes	Java et C#
<i>JENA</i>	homogène	N3, N-Triples, RDF/XML	fusionné (avec un tableau)	Axiomes	Java
<i>TRIPLE</i>	homogène	logique floue	Arrière (avec un tableau)	Axiomes	XSB-Prolog et Java
<i>SEW</i>	homogène	RuleML, N-Triples, RDF/XML	Arrière (avec un tableau)	Règles	XSB-Prolog
<i>SWI</i>	hybride	RDF/XML	Arrière (avec un tableau)	Règles	Prolog

Tableau 2.4 – Comparaison des systèmes de raisonnement sur différentes caractéristiques *ALCQI*

CWM et Euler

CWM²³ est un système qui propose un moteur basé sur un algorithme de tableau en avant et développé en python spécialement pour le web sémantique. Les raisonneurs utilisent généralement deux stratégies à s'appliquer sur des connaissances. Les stratégies en avant et en arrière sont deux stratégies utilisées. Le système CWM supporte des formats *N – TRIPLES*, *RDF/XML* et la

20. algorithme utilisé pour le raisonnement sur des ontologies DL

21. rdf schéma

22. <https://jena.apache.org/>

23. <http://www.w3.org/2000/10/swap/doc/cwm.html>

notation 3 (format $N3$ ²⁴) et il est capable de passer d'un format à un autre. En plus, le système a une variété de «builtins»²⁵ fonctions généralement implémentées pour résoudre les problèmes en cryptographie utilisant les fonctions mathématiques qui manipulent le temps.

TRIPLE

Le « triple » est un système du web sémantique supportant les formats RDF et un sous ensemble des OWL Lite²⁶. Sa syntaxe est basée sur la logique floue et la logique du premier ordre. Ces triplets sont généralement représentés sur la forme $S[P \rightarrow O]$ où S signifie le sujet, P le prédicat et O l'objet [Sintek *et al.*, 2002]. Dans un triplet, l'ensemble des déclarations se partageant le même sujet peut être regroupé sous forme d'une séquence représentée par l'équation 2.23. Ce système de triplet est implémenté dans le serveur de bases de données *virtuoso* que nous présenterons au chapitre 3.

$$S[P1 \rightarrow O1; P2 \rightarrow O2; P3 \rightarrow O3; \dots] \quad (2.23)$$

SEW

SEW est le système natif et complet du *XSB – Prolog* [Sagonas *et al.*, 1994] qui implémente un moteur RDFs incluant le format *xml* et les types de données. Il est capable de lire un graphe sous format *rdf/xml* et N-TRIPLES. Il est aussi capable d'intégrer une logique du premier ordre en *RuleML*²⁷ qui a une possibilité de prendre en compte une notation 3 ($N3$). C'est ce système qui a permis de construire des raisonneurs comme SWRL.

SWI-Prolog

La distribution SWI-prolog inclut une librairie du web sémantique efficace et mise au point pour de grands volumes stockages et des triplets. Elle peut être inférée par des programmes *Prolog*. Elle possède une approche hybride qui permet d'implémenter séparément les raisonneurs et des triplets. Elle utilise des règles *Prolog*.

Metalog et IBL

Metalog et IBL sont deux systèmes permettant une spécification de bases de connaissances utilisant l'anglais comme un langage contrôlé. Ils résolvent le problème des règles superflues du RDFs

24. format de représentation des triples

25. technique qui permet d'ajouter de nouvelles fonctionnalités au système en se basant sur la logique du premier ordre

26. Version de «OWL» qui est décidable

27. norme de représentation des connaissances du web sémantique qui a servi de base pour le SWRL et d'autres langages

les rendant ainsi difficile à maîtriser. L'interface qu'ils proposent permet à l'utilisateur de spécifier facilement les règles qui seront ensuite traduites dans un programme logique. IBL utilise une structuration de la logique du second ordre et Metalog traduit les déclarations directement en *prolog* et les évaluant avec SWI-Prolog.

2.4.2 Les règles en SWRL

Une ontologie en DL est une abstraction syntaxique qui contient une séquence d'annotations, d'axiomes et de règles. Les annotations sont des propriétés qui permettent de renseigner certaines informations à l'ontologie telles que : les commentaires, la date de création, le propriétaire de l'ontologie, etc. Un axiome en DL est défini soit par des classes d'inclusion ou soit par des classes d'équivalence, présenté dans la partie 2.3.3. La définition d'une règle en DL est basée sur le langage SWRL²⁸ (Semantic Web Rule Language). Le langage SWRL est utilisé dans le web sémantique pour construire des règles. Il combine le langage OWL-DL et le langage RuleML « Rule Markup Language (Unary/Binary Datalog) ». Il se compose d'un mélange de règles et des constructeurs OWL définis à la table 2.2 incluant des annotations ontologiques, les axiomes de classes, des propriétés, des faits et des instances OWL aussi bien que les règles elles-mêmes. Un axiome de règle en SWRL respecte la logique de *Horn* (voir section 1.3) c.-à-d. il est composé d'un antécédent *Antecedent* (correspondant au corps de la logique *Horn*) et une conséquence *Consequent* (correspondant à l'entête de la logique *Horn*). La conséquence et l'antécédent sont formés d'un ensemble d'atomes ou du vide. Un atome peut être une règle de la forme :

- « $C(x)$ » (prédicat unaire),
- « $D(x)$ » (concept atomique),
- « $P(x, y)$ » (prédicats binaires),
- « $sameAs(x, y)$ » (propriété binaire qui permet de déterminer les éléments de x qui sont identiques de ceux de y)
- « $differentFrom(x, y)$ » (propriété binaire qui permet de déterminer les éléments de x qui sont différents de ceux de y)
- « $builtin(b, z_1, \dots, z_n)$ » (fonctions définies et qui peuvent être étendues par l'utilisateur respectant la relation 2.26).

Les atomes ($x, y, b, z, z_1, z_2, \dots, z_n$) peuvent aussi se référer aux variables (*i-variable*, *d-variable*), aux individus ou instances (*I-variable*), aux données littérales *D-variable* (des nombres, des chaînes de caractères, etc.).

$$axiome: : = règle \tag{2.24}$$

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

$$\begin{aligned}
rule: & : = 'Implies(['URIreference] \{annotation\} antecedent consequent)'' \\
Antecedent: & : = 'Antecedent('{atom}')'' \\
Consequent: & : = 'Consequent('{atom}')''
\end{aligned} \tag{2.25}$$

$$\begin{aligned}
atom: & : = \text{description}'('i - object)'' \\
& | \text{dataRange}'('d - object)'' \\
& | \text{IndividualvaluedProperty}'('i - object \ i - object)'' \\
& | \text{sameAs}'('i - object \ i - object)'' \\
& | \text{differentFrom}'('i - object \ i - object)'' \\
& | \text{builtIn}'('i - object \ i - object)''
\end{aligned} \tag{2.26}$$

$$\begin{aligned}
i - object: & : = \quad i - variable \ | \ individualID \\
d - object: & : = \quad d - variable \ | \ dataLiteral \\
i - variable: & : = \quad I - variable}'('URIreference)'' \\
d - variable: & : = \quad D - variable}'('URIreference)''
\end{aligned} \tag{2.27}$$

La formalisation des différentes dimensions de l'entité spatiale a permis de structurer et représenter les entités de l'environnement maritime. Pour illustrer cette formalisation, un cas d'étude a été proposé dans la section suivante.

2.5 Cas d'étude : la représentation du système de balisage

Le principal souci en navigation côtière est d'éviter les zones de danger représentées par la côte, les récifs et les faibles fonds. L'évitement de ces zones de danger s'appuie sur de nombreuses aides représentées par la signalisation maritime (surtout abondante sur les côtes des pays développés), le balisage cardinal en mer et le balisage latéral dans les chenaux. Les amers sur la côte permettent par ailleurs de tracer sur la carte des relèvements de garde tout en évitant des zones dangereuses. Deux amers alignés permettent une navigation par alignement, particulièrement pratique et sûre dans un chenal. Il est alors judicieux d'intégrer ces signalisations dans les modèles de représentation pour refléter la réalité. Dans le cadre de ce chapitre, nous présentons le cas d'étude d'un balisage que nous avons modélisé par l'ontologie proposée à la figure 2.2 et la base de règles illustrée par la table 2.5.

Cas d'étude d'une bouée cardinale *Est*

Il existe 4 sortes de bouées cardinales. Chaque bouée a une forme et des couleurs spécifiques lui permettant d'indiquer la route de navigation à suivre comme décrit ci-dessous :

- 1 les deux cônes pointant vers le haut désignent la *cardinale Nord* de couleur jaune et noir (en partant du bas vers le haut) et exigent de passer au *Nord* de la bouée.

- 2 les deux cônes pointant vers le bas désignent la *cardinale Sud* de couleur noir et jaune (en partant du bas vers le haut) et exigent de passer au *Sud* de la bouée.
- 3 un cône pointant vers le haut et l'autre vers les bas ayant tous deux une même base désignent la *cardinale Est* de couleur noir, jaune et noir (en partant du bas vers le haut) et exigent de passer à l'*Est* de la bouée.
- 4 un cône pointant vers le haut et l'autre pointant vers le bas se rencontrant tous deux sur un même sommet désignent la *cardinale Ouest* de couleur jaune, noir et jaune (en partant du bas vers le haut) et exigent d'orienter la navigation à l'*Ouest*.

Ces différentes cardinales sont illustrées par la figure 2.3. Dans cette illustration apparaît la forme et les couleurs des bouées cardinales et les différentes signalisations qui leur sont associées de nuit. Toutes ces connaissances sur les cardinales ont été intégrées dans notre modèle et présentées dans les paragraphes ci-après.

Dans ce cas d'étude, nous présentons uniquement l'état d'une bouée cardinale *Est* vue de jour et de nuit. La table de règle 2.5 nous a permis de déterminer l'agrégation courante de l'entité (agrégation de nuit ou de jour) afin de déterminer l'état correspondant et ses propriétés à un moment donné.

La détermination de l'état courant d'une entité passe d'abord par la détermination de l'agrégation des états de l'entité auquel elle appartient. La table 2.6 présente l'agrégation des états de jour d'une bouée cardinale *Est*. Dans cette illustration, *currentTime* indique le temps courant et la valeur 2014 – 06 – 17T13 : 08 : 51 : 666 exprimé en *jodaTime* que nous avons présenté à la section 2.3 désigne la date du 17 juin 2014 à 13 heures 8 minutes 51 secondes et 666 millisecondes. *currentAgg* identifie le nom de l'agrégation courante associée à la bouée, *hasAggregateBegin* et *hasAggregateEnd* désignent respectivement le début et la fin de l'agrégation, *durationAgg* marque la durée de vie de l'agrégation et *hasAggregateDuration* indique la durée d'une agrégation. S'il est à zero alors l'agrégation ne possède qu'un seul état (elle ne varie pas). Une fois l'agrégation de l'entité choisie, il faut déterminer l'état courant de l'entité. Cet état appartient à la liste des états présentés par l'axiome 2.12 et est extrait par la règle 2.16. Les propriétés de l'état courant de la bouée cardinale *Est* choisie comme cas d'étude et identifiée à la date 2014–06–17T13 : 08 : 51 : 666 sont décrites par la table 2.7.

La table 2.8 présente les propriétés de l'agrégation d'une bouée cardinale *Est* de notre modèle vue de nuit. Cette agrégation correspond aux signalisations de nuit figurant sur la figure 2.3. Nous observons un rythme de scintillements rapide et groupé (par bloc de trois éclats) de couleur blanche. Chaque bloc de couleur blanche possède une durée de scintillement d'une seconde. Par contre, l'espacement de deux groupes ou blocs est considéré comme un temps de repos de l'entité dans notre modèle et dure 5 secondes. Il est aussi considéré comme un état de couleur noire de notre modèle. Dans ce cas d'étude la bouée cardinale *Est* vue de nuit est représentée par l'agrégation décrite par la table 2.8 et l'état courant de cette bouée cardinale *Est* dans notre modèle à la date du 2014 – 06 – 17T22 : 39 : 29 : 374 est détaillée par la table 2.9.

Catégorie sélectionnée	Règles en SWRL	Interprétation
Entité	EntityInstance(?p), hasName(?p, ?q), identifyEntityDimensions(?q, ?cs, ?ct, ?cse) →component(?p, ?ct)	Retourne les informations sur la composante temporelle de l'entité <i>ct</i>
L'agrégation d'événement	EntityAggregateEvent(?p), hasAggregateBegin(?p, ?t0), hasAggregateDuration(?p, ?d0), hasAggregateName(?p, ?n0), CurrentEvent(?timeLife, ?n0, ?d0, ?state), TimeDuration(?t0, ?ct, ?timeLife), jtime(?ct) → component(?p, ?n0), durationAgg(?p, ?timeLife)	Retourne la durée de vie de l'agrégation
Saison	Saison(?p), hasSaisonBegin(?p, ?t1), hasSaisonEnd(?p, ?t2), buildJtimeInterval(?t1, ?t2, ?T) → hasInterval(?p, ?T)	Retourne l'intervalle de temps associé à chaque saison au cours d'une année
Saison	Saison(?p), hasDayBegin(?p, ?td1), hasDayEnd(?p, ?td2), hasSaisonEnd(?p, ?t2), hasSaisonbegin(?p, ?t1), buildJtimeInterval(?a1, ?a2, ?T), buildJtimeInterval(?ad1, ?ad2, ?Td), correspondFixTime(?t1, ?a1), correspondFixTime(?t2, ?a2), correspondFixTime(?td1, ?ad1), correspondFixTime(?td2, ?ad2), isBelongTo(?ct, ?T, ?r2), isBelongTo(?ct, ?Td, ?r1), isDay(?r1, ?r2, ?light), jtime(?ct) → hasDayInterval(?p, ?Td), hasSaisonInterval(?p, ?T), sunLight(?p, ?light)	Cherche la saison actuelle à partir du temps courant et la valeur <i>sunLight</i> est 1 si nous sommes le jour et 0 sinon (voir équation 2.10)

Tableau 2.5 – Base de règles intégrée à notre base de connaissances

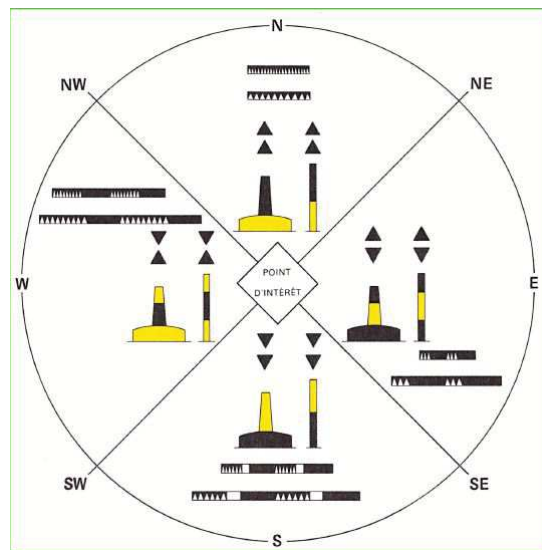


Figure 2.3 – Illustration des différentes bouées cardinales (Nord, Sud, Est, Ouest) vues de jour et de nuit

Les propriétés de l'agrégation	Les valeurs
currentTime	2014-06-17T13 :08 :51 :666
currentAgg	BoyCar1DayAggregateEvent
hasAggregateBegin	2014-05-01T03 :48 :59 :803
hasAggregateEnd	2019-05-01T03 :58 :59 :803
durationAgg	100551221,00 sec
hasAggregateDuration	0

Tableau 2.6 – Illustration de l'agrégation d'une bouée cardinale Est vue de jour

Les propriétés l'état courant	Les valeurs
hasName	BoyCarDayStateEast
hasStateDuration	0
hasColor	<i>Black_Yellow</i>
hasDirection	East
hasAffordance	Maneuver

Tableau 2.7 – Illustration de l'état courant d'une bouée cardinale Est vue de jour

Les propriétés de l'agrégation	Les valeurs
currentTime	2014-06-17T22 :39 :29 :374
currentAgg	BoyCar1NightAggregateEvent
hasAggregateBegin	2014-05-01T03 :48 :59 :803
hasAggregateEnd	2019-05-01T03 :58 :59 :803
durationAgg	100551221,00 sec
hasAggregateDuration	6

Tableau 2.8 – Illustration de l'agrégation d'une bouée cardinale Est vue de nuit

Les propriétés l'état courant	Les valeurs
hasStateName	BoyCarNightState2East
hasStateDuration	8=1+1+1+5 est la somme des temps de tous les états
hasColor	Black
hasDirection	East
hasAffordance	Maneuver

Tableau 2.9 – Illustration de l'état courant d'une bouée cardinale Est vue de nuit

2.6 Conclusion

Dans cette section, nous avons proposé un enrichissement des modèles spatio-temporels de l'environnement maritime par l'ajout d'une sémantique afin de garantir la sécurité des usagers au cours de la navigation. Les composantes de la dimension sémantique auxquelles nous nous sommes intéressées sont l'affordance et la saillance. Ces deux composantes sont complémentaires et très utilisées dans des systèmes de guidage ou de navigation. Ce besoin d'intégrer ces deux composantes de la dimension sémantique dans des modèles spatio-temporels nous a conduit à proposer une ontologie computationnelle. L'ontologie computationnelle utilisée modélise une entité dans trois dimensions (spatiale, temporelle, sémantique). La dimension spatiale associe l'empreinte spatiale à la description de la géométrie définie suivant la norme proposée par le consortium *OGC*. La composante temporelle représente une entité comme une agrégation d'événements se produisant de façon répétitive. La dimension sémantique s'intéresse à la modélisation de la saillance et de l'affordance et celles-ci peuvent être modifiées par la dynamique de l'environnement. Afin de permettre à l'entité de s'accorder à la dynamique de l'environnement, quatre types de variations ont été identifiées (la variation de l'empreinte spatiale, la variation de l'emplacement spatial, la variation de la perception, la variation de la durée de vie). Ces variations permettent de déterminer la saillance et

l'affordance en fonction des changements subis par l'entité dans son environnement. Le but de cette catégorisation est de retrouver la saillance ou l'affordance d'une entité à un moment donné. Dans le contexte de cette thèse, l'affordance et la saillance ont été déterminées par l'expert du domaine. Il peut cependant arriver que l'expert du domaine ne parvienne pas à proposer l'affordance à associer à une entité. Le système proposé utilisera des stratégies de recherche permettant de déterminer l'affordance correspondante. Cette stratégie utilise les approches proposées en traitement automatique du langage et est détaillée dans le chapitre suivant.

Chapitre 3

Enrichissement des entités spatio-temporelles à partir de techniques de TAL (Traitement automatique du langage)

Sommaire

3.1	Techniques de recherche d'informations	67
3.2	Extraction automatique de la sémantique associée à un concept	69
3.2.1	Vecteur conceptuel d'un mot	69
3.2.2	Vecteur conceptuel d'une phrase	72
3.3	Espace de décision pour l'extraction de la sémantique	74
3.3.1	Projection d'une phrase dans un espace de décision	74
3.3.2	Direction sémantique principale d'un concept	76
3.3.3	Cas d'étude	76
3.4	Utilisation des gloses dans la recherche de la sémantique	78
3.5	Conclusion	82

L'INTÉGRATION de données provenant de divers sites ou de différents serveurs crée des volumes de données considérables. La richesse de ces données attire de plus en plus des chercheurs qui proposent des algorithmes pour extraire de la connaissance [Bizer *et al.*, 2009]. L'un des domaines pionnier dans ces travaux de recherche est le traitement automatique du langage naturel. Il fournit des méthodes et des techniques modernes pour extraire des connaissances [Tsatcha *et al.*, 2013b]. Le résultat de ces travaux en extraction automatique de connaissances est intégré dans de nombreuses applications d'aide à la décision. C'est le cas des applications opérant dans l'environnement maritime qui sont aussi confrontées à de gros volumes de données provenant

de diverses infrastructures ou sources (AIS, RADAR, carte maritime, etc.) [Tsatcha *et al.*, 2013b]. Les applications existantes à ce jour ont des difficultés à exploiter ces données car celles-ci ne sont pas structurées, rendant difficile pour des experts de produire des modèles qui prennent toujours en compte toutes les connaissances produites par ces données.

Ainsi, les résultats de cette extraction permettent d'enrichir le modèle proposé au chapitre 2. Le modèle proposé a été réalisé à partir des connaissances fournies par des experts du domaine maritime. Finalement, le modèle issu de ces deux stratégies permet de produire des applications robustes qui devraient être capables de trouver la connaissance relative à une entité même si celle-ci n'a pas encore été identifiée par l'expert.

La méthode que nous proposons dans cette section est basée sur le «TAL» et est combinée à un principe de segmentation de gros volumes de documents. Elle a été appliquée à la problématique des systèmes de transports intelligents opérant dans le domaine de la navigation maritime. Elle utilise des documents de la régulation de la navigation maritime [NOAA, 2011a],[IHO, 2011], [IHO, 2011], [Pearson, 2008], [Néméta, 2008].

Pour mener à bien cette recherche, le traitement automatique du langage naturel est appliqué à deux niveaux. Le premier niveau concerne l'extraction des termes du domaine. Cette tâche a été rendue possible par l'utilisation du logiciel Yatea¹, proposé par [Aubin et Hamon, 2006], couplé avec TreeTagger [Schmid, 2012] afin d'extraire les termes ayant le plus grand nombre d'occurrences dans les documents. Cependant, les termes avec un petit nombre d'occurrences mais qui sont importants dans les décisions de navigation sont également considérés. Les termes extraits des documents [NOAA, 2011a],[IHO, 2011], [IHO, 2011], [Pearson, 2008], [Néméta, 2008] sont mis dans un corpus . Le corpus² utilisé contient 16 010 phrases définies avec 413 076 mots (175 578 noms, 269 22 verbes , 20 703 adjectifs , 9 106 adverbes). Le deuxième niveau détermine la sémantique des objets par rapport à un ensemble fini d'actions établies par un expert (voir chapitre 2.1).

Cette extraction est rendue possible grâce à la technique des vecteurs conceptuels appliqués aux phrases décrivant un objet. Ces vecteurs conceptuels sont projetés par la suite dans un espace de décision. En extrayant la sémantique des objets, il est possible de trouver la décision ou la future zone que le marin devrait suivre. Cette section se concentre principalement sur le deuxième niveau en supposant que le premier est une étape du prétraitement réalisée par le logiciel Yatea.

Dans l'intérêt de mieux étoffer cette section, elle est organisée comme suit. La section 3.1 présente les différentes techniques de recherche d'information dans de gros volumes de documents textuels et parfois non structurés (en termes de sémantique). La section 3.2 met en évidence l'intérêt des algorithmes d'extraction de connaissances et de sémantique dans l'environnement maritime et propose une technique d'extraction basée sur les vecteurs conceptuels des mots. La section 3.3 propose un espace de décision permettant de déterminer la contribution ou le lien sémantique d'un concept par

1. Yatea est un logiciel gratuit utilisé pour disambiguisation lexicale de documents.

2. Un corpus est un ensemble de documents, artistiques ou non (textes, images, vidéos, etc.), regroupés dans une optique précise.

rapport à une décision ou à une action ou à une affordance ou par rapport à un autre concept.

3.1 Techniques de recherche d'informations

L'extraction de l'information est un vaste domaine de la recherche scientifique. Ces premières activités ont commencé au cours des années 1960 à partir des expériences en indexation linguistique [Cleverdon, 1962]. C'est au cours des années 1990 que sont apparus les premiers moteurs de recherches basées sur la recherche des mots clés. La technique basée sur la recherche de mots clés consiste à retrouver un mot clé dans un document et sans toutefois procéder à une représentation de la requête et du document au préalable [Salton et MacGill, 1983]. Quelques moteurs de recherches sont apparus dans cette période :

- «Archie»³,
- «Wanderer»⁴,
- «Yahoo»⁵,
- «Lycos»⁶,
- «Excite»⁷.

Avec les avancées du « TAL », il est actuellement possible de fusionner les approches en «TAL» et l'extraction d'informations (voir chapitre 1.2.2.2) afin de rechercher de l'information sémantique associée à une requête [Strzalkowski *et al.*, 1994, Dumais *et al.*, 1997, Potthast *et al.*, 2008]. Trois étapes sont primordiales dans le processus de recherche d'information sémantique [Hiemstra, 2001] :

- 1** la représentation des documents,
- 2** la représentation de la requête,
- 3** une comparaison des documents et la requête tous deux préalablement structurés.

La structuration des documents et de la requête est rendue possible de façon automatique par des outils de désambiguïsation. L'objectif consiste à décomposer une phrase en mots auxquels sont associés pour chacun sa catégorie grammaticale, son lemme et sa fonction dans la phrase. Bien qu'il existe plusieurs outils de désambiguïsation, le plus stable que nous ayons trouvé (qui permet la désambiguïsation des termes français) est *Yatea*. Il peut également produire un flux «XML» que l'on peut directement stocker dans une base de données. Le tableau 3.1 présente un exemple de désambiguïsation de la phrase *Le bateau est dans le port* où « DT » signifie le déterminant, « NN » le nom, « VBZ » le verbe et « IN » la préposition.

Dans l'intérêt d'améliorer l'efficacité d'un processus de recherche d'information, les documents doivent être transformés dans un modèle approprié. Les modèles les plus utilisés sont les suivants :

3. <http://people.lis.illinois.edu/~chip/projects/timeline/1990archie.htm>
4. <http://history-computer.com/Internet/Conquering/Wanderer.html>
5. <http://fr.search.yahoo.com/>
6. <http://www.lycos.com/>
7. <http://www.excite.com/>
8. http://en.wikipedia.org/wiki/Brown_Corpus#Partofspeech_tags_used

Mot	Catégorie grammaticale ⁸	Lemme
Le	DT	le
bateau	NN	bateau
est	VBZ	être
dans	IN	dans
le	DT	le
port	NN	port

Tableau 3.1 – Désambiguïsation de la phrase « Le bateau est dans le port »

- le modèle ensembliste,
- le modèle probabiliste,
- le modèle algébrique.

Le modèle ensembliste représente les documents sous la forme de mots et de phrases. Le modèle probabiliste utilise des méthodes de probabilité comme la théorie bayésienne pour représenter les documents. Quant au modèle algébrique, il représente les documents et la requête sous forme de vecteurs ou de matrices.

Généralement, la plupart des systèmes affectent un score aux documents et les classent en fonction de leur score. Le score d'une requête dans un document est déterminé par le nombre de fois où la requête apparaît dans le document. Il ne prend pas en compte la disposition des termes de la requête dans le document (aucune considération de la relation sémantique, de la relation de parenté et de la correspondance des groupes grammaticaux entre les termes des mots à comparer). La relation sémantique s'observe lorsque la requête contient exactement les mêmes groupes grammaticaux classés dans l'ordre d'apparition après la désambiguïsation avec le supposé groupe de mots auquel il pourrait correspondre dans une phrase d'un document. Supposons que la requête soit « le beau bateau ». Dans ce cas, il n'existe aucune relation sémantique entre cette requête et la phrase « Le bateau est dans le port » car elles ne possèdent pas les mêmes groupes grammaticaux dans le même ordre. [Tsatsaronis et Panagiotopoulou, 2009] dans leur article mettent en évidence l'intérêt de capturer les relations sémantiques existant entre les termes dans l'extraction de l'information sémantique. Dans cette partie, nous proposons un algorithme basé sur les vecteurs conceptuels et la désambiguïsation où les résultats sont fortement liés à la relation de parenté entre les termes. Les bases de ces travaux sont inspirées de la notion du vecteur conceptuel initialement proposée par [Lafourcade *et al.*, 2002]. Les vecteurs conceptuels ont été principalement utilisés pour

l'extraction d'information et pour une représentation significative des «LSI» (Latent Semantic Indexing) modélisée à partir des «LSA» (Latent Semantic Analysis) étudiée en psycholinguistique [Salton et MacGill, 1983]. Notre approche est inspirée de [Chauché, 1990], qui propose un formalisme pour la projection des notions en sémantique linguistique dans un espace vectoriel. Le vecteur conceptuel (ou vecteur de concepts) d'un mot est un ensemble de mots dans lequel chaque mot détermine un concept où ce mot peut être utilisé. Le vecteur conceptuel d'une phrase prend en compte tous les concepts de la phrase. Cette dernière est basée sur la somme directe des vecteurs conceptuels des mots qui composent la phrase.

3.2 Extraction automatique de la sémantique associée à un concept

L'extraction automatique de la sémantique des entités est fondée sur la théorie des vecteurs conceptuels. La définition d'un vecteur conceptuel est basée sur les synsets (synonymes set) d'un mot. Dans la littérature, le synset d'un mot est un ensemble de mots interchangeables, dénotant un sens ou un usage particulier [Beckwith *et al.*, 1990]. Tous les mots composant le synset, excepté le premier mot, sont appelés les synsets candidats. Il en résulte que la définition du vecteur conceptuel associé à un mot est basé sur l'ensemble des synsets et une mesure évaluant la distance entre le mot et chaque synset candidat.

3.2.1 Vecteur conceptuel d'un mot

Le vecteur conceptuel d'un mot est ainsi organisé en fonction de la catégorie grammaticale (adjectif (a), adverbe (r), nom(n), verbe (v)) à laquelle le mot appartient et trié suivant un ordre décroissant de la valeur de distance dans chaque catégorie grammaticale.

Dans ce travail, nous avons utilisé la distance définie dans RiWordnet⁹ [Daniel, 2012] qui est une application développée pour produire des mots interactifs (littérature computationnelle) proposée par Daniel Howe. Plus formellement, soit w un mot, $S^c = (s_i^c)_{i=1}^{n^c}$ l'ensemble des n^c synsets de w et $C^c = (c_i^c)_{i=1}^{m^c}$ l'ensemble de m^c mots candidats de la catégorie grammaticale c alors le vecteur conceptuel $V(w)$ du mot w est défini comme une union pondérée de mots exprimés dans chaque catégorie grammaticale :

$$V(w) = \left(\bigcup_{l=1}^{m^a} c_l^a \delta(c_l^a, w) \right)_a \left(\bigcup_{l=1}^{m^r} c_l^r \delta(c_l^r, w) \right)_r \left(\bigcup_{l=1}^{m^n} c_l^n \delta(c_l^n, w) \right)_n \left(\bigcup_{l=1}^{m^v} c_l^v \delta(c_l^v, w) \right)_v \quad (3.1)$$

Dans l'intérêt de mieux illustrer nos propos, nous proposons un exemple pour trouver le vecteur conceptuel du mot « port ». Soit $S^a = (s_i^a)_{i=1}^1$, $S^r = \emptyset$, $S^n = (s_i^n)_{i=1}^5$ et $S^v = (s_i^v)_{i=1}^8$ et $S^v = (s_i^v)_{i=1}^8$ les différents synsets du mot port extraits de WordNet (voir Figure 3.1b) :

– Adjective (a)

9. RiWordnet est un interface de programmation sur WordNet qui est basé sur des données de la langue anglaise

- s_1^a : port, larboard (located on the left side of a ship or aircraft)
- Noun (n)
 - s_1^n : port (a place (seaport or airport) where people and merchandise can enter or leave a country)
 - s_2^n : port, port wine (sweet dark-red dessert wine originally from Portugal)
 - s_3^n : port, embrasure, porthole (an opening (in a wall or ship or armored vehicle) for firing through)
 - s_4^n : larboard, port (the left side of a ship or aircraft to someone who is aboard and facing the bow or nose)
 - s_5^n : interface, port ((computer science) computer circuit consisting of the hardware and associated circuitry that links one device with another (especially a computer and a hard disk drive or other peripherals))
- Verb (v)
 - s_1^v : port (put or turn on the left side, of a ship) "port the helm"
 - s_2^v : port (bring to port) "the captain ported the ship at night"
 - s_3^v : port (land at or reach a port) "The ship finally ported"
 - s_4^v : port (turn or go to the port or left side, of a ship) "The big ship was slowly porting"
 - s_5^v : port (carry, bear, convey, or bring) "The small canoe could be ported easily"
 - s_6^v : port (carry or hold with both hands diagonally across the body, especially of weapons) "port a rifle"
 - s_7^v : port (drink port) "We were porting all in the club after dinner"
 - s_8^v : port (modify (software) for use on a different machine or platform)

L'ensemble des mots candidats du mot « port » est défini par $C^a=(\text{bâbord})$, $C^r = \emptyset$, $C^n=(\text{meurtrière, hublot, bâbord, interface})$ et $C^v = \emptyset$, $C^r = \emptyset$ et $C^v = \emptyset$. L'expression $C^g = \emptyset$ désigne qu'il n'y a pas de synset candidat, c.-à-d qu'il n'y a pas d'autre sens dans la catégorie grammaticale g se rapportant au mot « port ». A la suite de cette description, nous en déduisons que le vecteur conceptuel du mot port est donné par :

$$V(\text{port})=(\text{larboard}[1.00])_a(\text{embrasure}[1.00]\text{porthole}[1.00]\text{larboard}[1.00]\text{interface}[1.00])_n$$

En définitive, la distance δ entre deux mots w_1 , w_2 , ou entre un mot et un mot candidat dans le contexte du vecteur conceptuel est calculée comme suit : Soient $G=\{\text{adjectif}(a), \text{adverbe}(r), \text{nom}(n), \text{verbe}(v)\}$, l'ensemble des catégories grammaticales dans le dictionnaire lexical WordNet $P=\{P^a, P^r, P^n, P^v\}$ l'ensemble des parents communs dans le réseau lexical WordNet. Les mots w_1 et w_2 ont un parent commun s'ils partagent des mêmes relations sémantiques (hypernymes, hyponymes, holonymes et toponymes etc.). La distance entre ces deux mots est définie ainsi :

10. <http://www.visualthesaurus.com/landing/>

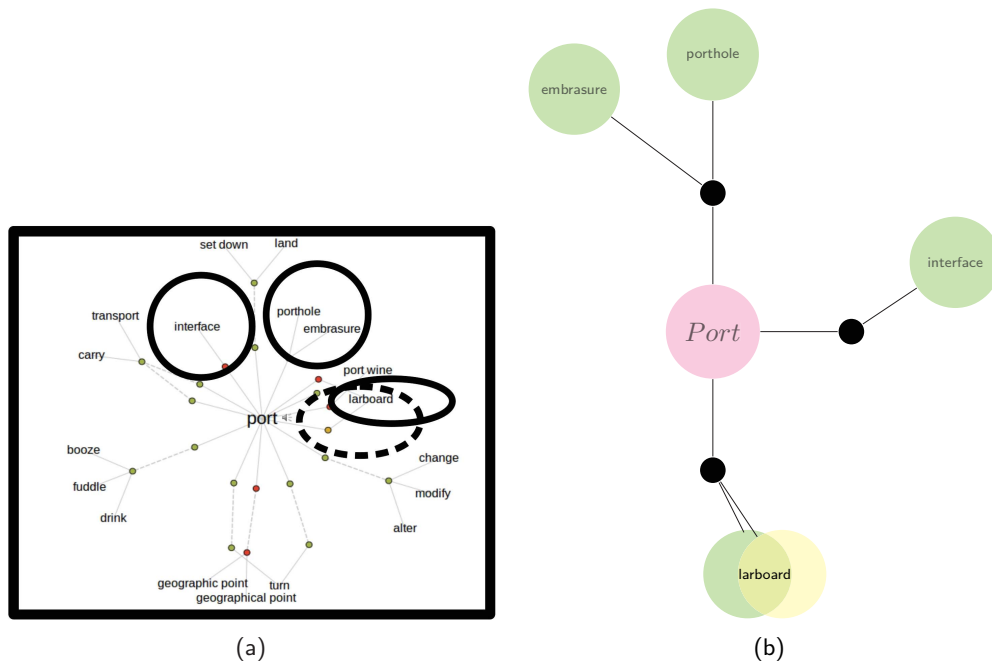


Figure 3.1 – (a) Illustration des « synsets » du mot « port » générés à partir de l'application « visual thesaurus »¹⁰ et (b) Illustration des synsets extraits dans notre modèle, les cercles en vert représentent les « synsets » candidats de type nom, les cercles en jaunes représentent les « synsets » candidats de type adjectif.

$$\delta(w_1, w_2)^{11} = \begin{cases} 1 & \text{si } w_1 \text{ et } w_2 \text{ n'ont pas de parent commun} \\ \left\| \left[\frac{\min(d(w_1, P^g), d(w_2, P^g))}{\min(d(w_1, P^g), d(w_2, P^g)) + d(P^g, R)} \right]_{g \in G} \right\| & \end{cases} \quad (3.2)$$

où $d(w_1, w_2)$ est égale au nombre d'arcs entre les nœuds w_1 et w_2 , R est le nœud racine dans le réseau lexical et $\| \cdot \|$ est la norme infinie. Plus la valeur de la distance entre deux mots est petite, plus leur proximité sémantique est grande.

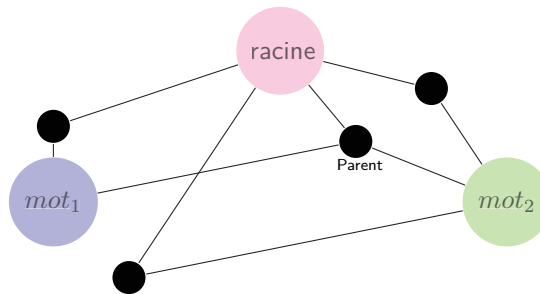


Figure 3.2 – Illustration de la structure d'un réseau lexical défini dans WordNet.

11. http://www.rednoise.org/rita/wordnet/documentation/riwordnet_method_getdistance.htm

La figure 3.2 représente un exemple d'organisation de mots dans le dictionnaire «WordNet» pour une catégorie grammaticale quelconque. En considérant un tel graphe, la distance $\delta(mot1, mot2)$ entre les deux mots est égale à $\|\frac{1}{1+1}\| = 0.5$.

3.2.2 Vecteur conceptuel d'une phrase

La description du vecteur conceptuel d'un mot décrit précédemment est un atout pour décrire celui d'une phrase. En supposant qu'une phrase est une collection de mots ayant plusieurs sens, nous pouvons définir le vecteur conceptuel d'une phrase comme la somme directe des vecteurs conceptuels de mots qu'elle contient. Pour chaque vecteur conceptuel de mot, on ne considère que la catégorie grammaticale qui correspond à celle du mot dans la phrase. Actuellement, dans «WordNet» les synsets des mots sont générés dans quatre catégories grammaticales (adjectif, adverbe, nom, verbe). Pour un mot ayant les « synsets » différents selon leur emploi dans la phrase ou dans la partie du discours, tous les vecteurs conceptuels seront considérés indépendamment de leur emploi ou partie « part of speech » dans laquelle ils sont utilisés. Si nous considérons la phrase précédente, la catégorie grammaticale du mot « bateau » dans la phrase «Le bateau est dans le port» est «Nom». Si nous souhaitons extraire le vecteur conceptuel de cette phrase, nous allons uniquement considérer le vecteur conceptuel du mot « bateau » dans la catégorie grammaticale «Nom». Un vecteur conceptuel de phrase rassemble un ensemble de concepts auxquels la phrase est associée. Dans la définition du vecteur conceptuel, on ne prend pas en compte le type (c.-à-d., déclaratif, exclamatif, interrogatif et impératif) et la forme (c.-à-d., affirmative/négative, active/passive, neutre/emphatique) de la phrase. Considérons s une phrase composée de n mots w_i , alors le vecteur conceptuel de la phrase est défini par :

$$V(s) = \sum_{i=1}^{n-1} V(w_i) \oplus V(w_{i+1}) \quad (3.3)$$

Dans cette expression, la somme directe de deux vecteurs conceptuels est définie comme l'union des mots candidats organisés en catégories grammaticales, où chaque mot est pondéré par la distance au concept qu'il représente (cf. équation 3.2). La somme directe du vecteur conceptuel est habituellement utilisée pour étendre les champs de concept de l'espace de concepts (c.-à-d., le contexte).

Soient w_i and w_j deux mots et $V(w_i)$ et $V(w_j)$ leur vecteur conceptuel correspondant, la somme directe de ces deux vecteurs conceptuels est exprimée par :

$$V(w_i) \oplus V(w_j) = \left(\bigcup_{k=i,j} \left(\bigcup_{l=1}^{m_{w_k}^a} c_{l,w_k}^a \delta(c_{l,w_k}^a, w_k) \right) \right)_a \left(\bigcup_{k=i,j} \left(\bigcup_{l=1}^{m_{w_k}^r} c_{l,w_k}^r \delta(c_{l,w_k}^r, w_k) \right) \right)_r \quad (3.4)$$

$$\left(\bigcup_{k=i,j} \left(\bigcup_{l=1}^{m_{w_k}^n} c_{l,w_k}^n \delta(c_{l,w_k}^n, w_k) \right) \right)_n \left(\bigcup_{k=i,j} \left(\bigcup_{l=1}^{m_{w_k}^v} c_{l,w_k}^v \delta(c_{l,w_k}^v, w_k) \right) \right)_v$$

Dans le cas particulier où $w_i = w_j$, $V(w_i) \oplus V(w_i) = V(w_i)$.

A titre d'exemple, la phrase suivante $s = \text{« The ship is in the port »}$ avec le tableau 3.2 comme résultat de la désambiguïstation. Le vecteur conceptuel des différents mots de la phrase s est décrit

Mot	Catégorie grammaticale	Lemme
The	DT	the
ship	NN	ship
is	VBZ	be
in	IN	in
the	DT	the
port	NN	port

Tableau 3.2 – Désambiguïstation de la phrase « The ship is in the port »

comme suit :

- $V(\text{the}) = \emptyset$
- $V(\text{ship}) = \emptyset$
- $V(\text{is}) = (\text{be}[0.00]\text{exist}[0.00]\text{equal}[0.00]\text{constitute}[0.00]\text{represent}[0.00]\text{comprise}[0.00]\text{follow}[0.00]\text{embody}[0.00]\text{personify}[0.00]\text{live}[0.00]\text{cost}[0.00])_v$
- $V(\text{in}) = (\text{inwards}[1.00]\text{inward}[1.00])_n(\text{inch}[0.00]\text{indium}[0.00])_r$
- $V(\text{the}) = \emptyset$
- $V(\text{port}) = (\text{embrasure}[1.00]\text{porthole}[1.00]\text{larboard}[1.00]\text{interface}[1.00])_n$

La somme directe entre $V(\text{ship})$ et $V(\text{port})$ est :

$$V(\text{ship}) \oplus V(\text{port}) = (\text{embrasure}[1.00]\text{porthole}[1.00]\text{larboard}[1.00]\text{interface}[1.00])_n$$

Le résultat normalisé (voir section 3.3.1) du vecteur conceptuel est :

$$\begin{aligned} V(\text{«The ship is the in the port»}) &= V(\text{the}) \oplus V(\text{ship}) \oplus V(\text{is}) \oplus V(\text{in}) \oplus V(\text{the}) \oplus V(\text{port}) \\ &= (\text{inch}[0.00]\text{indium}[0.00])_r \\ &\quad (\text{inwards}[0.38]\text{inward}[0.40]\text{embrasure}[0.47] \\ &\quad \text{porthole}[0.52]\text{larboard}[0.58]\text{interface}[0.66])_n \\ &\quad (\text{be}[0.00]\text{exist}[0.00]\text{equal}[0.00] \\ &\quad \text{constitute}[0.00]\text{represent}[0.00]\text{comprise}[0.00]\text{follow}[0.00] \\ &\quad \text{embody}[0.00]\text{personify}[0.00]\text{live}[0.00]\text{cost}[0.00])_v \end{aligned}$$

11. http://en.wikipedia.org/wiki/Brown_Corpus#Partofspeech_tags_used

3.3 Espace de décision pour l'extraction de la sémantique

L'intérêt de cette section est d'extraire l'information sémantique associée à un concept. Afin de répondre à cet objectif, nous avons introduit un espace de décision où les différents vecteurs conceptuels des phrases décrivant le concept sont projetés. L'espace de décision contient la liste des options possibles, obtenues généralement après une « macro-phase » d'une stratégie de décision. [Jankowski et Nyerges, 2003] résumant la macro-phase d'une stratégie de décision en trois étapes :

- 1 définition de l'intelligence des valeurs, objectifs et critères,
- 2 description de l'ensemble des options possibles,
- 3 les différents choix de recommandations.

Les options possibles devront être liées et validées par un expert. Les vecteurs conceptuels des mots, qui sont corrélés à la sémantique que l'on recherche, définissent alors les axes sémantiques (c.-à-d., la base) de l'espace de décision.

3.3.1 Projection d'une phrase dans un espace de décision

La projection d'une phrase s dans un espace de décision correspond à la projection des mots candidats du vecteur conceptuel de s (c.-à-d., $V(s)$). Cette projection nous permet d'évaluer leur contribution dans chaque direction sémantique de la base. Nous en déduisons ainsi la direction principale détaillée dans le prochain paragraphe.

Considérons que nous souhaitons calculer la contribution de x_i^c d'un mot candidat de $V(s)$ dans une direction sémantique d d'une catégorie c où n est le nombre de mots candidats du vecteur conceptuel $V(d)$ définissant l'axe sémantique de l'espace de décision. Si c_i^c est un mot candidat commun des deux vecteurs conceptuels $V(s)$ et $V(d)$ avec pour poids δ_s et δ_d respectivement alors :

$$x_i^c = \frac{(1 - \delta_s * \delta_d)}{n}. \quad (3.5)$$

x_i^c est égale à zéro si le mot candidat de $V(s)$ n'appartient pas à $V(d)$. Dans l'équation 3.5, la valeur est pondérée par le nombre de mots candidats n dans un axe sémantique. Un mot candidat de $V(d)$ ayant un poids égal de 1 (c.-à-d., une faible contribution sémantique) n'a pas de contribution dans un axe sémantique (c.-à-d., $x_i^c = 0$ si $\delta_s = 1$ et $\delta_d = 1$) et n'est pas pris en compte dans la décision finale. Pour résoudre ce problème, il a été judicieux de normaliser le vecteur conceptuel d'une phrase (section 3.2.2), ce qui rejette le cas où un poids est égal à 1.

La contribution d'une phrase dans la catégorie c est la somme des contributions m_c mots candidats de la catégorie, c.-à-d., $x^c = \sum_{i=1}^{m_c} x_i^c$. Avec le même raisonnement, nous parvenons à déduire que la contribution de la phrase est la somme des contributions dans chaque catégorie c.-à-d.,

$x = x^a + x^r + x^n + x^v$. A partir de cette équation, nous en déduisons que plus la contribution sémantique dans une direction est grande et plus la valeur de projection x est grande. Ce processus est répété dans toutes les directions sémantiques qui contribuent à l'espace de décision.

Nous illustrons notre principe avec l'exemple suivant dans lequel nous cherchons la contribution du mot « stay » dans la direction sémantique « stop ». Nous définissons initialement les vecteurs conceptuels de ces mots.

$$\begin{aligned}
V(\text{stop}) = & (\text{halt}[0.00]\text{block}[0.00]\text{check}[0.00] \\
& \text{arrest}[0.00]\text{blockade}[0.12] \text{bar}[0.14] \text{end}[0.14] \text{finish}[0.14]\text{barricade}[0.22] \\
& \text{break}[0.29]\text{cease}[0.67]\text{intercept}[0.73]\text{kibosh}[1.00] \\
& \text{terminate}[1.00]\text{contain}[1.00]\text{quit}[1.00]\text{discontinue}[1.00])_v \\
& (\text{halt}[0.00]\text{stoppage}[0.00]\text{stopover}[0.00] \\
& \text{layover}[0.00]\text{arrest}[0.00]\text{check}[0.00]\text{hitch}[0.00]\text{stay}[0.00] \\
& \text{occlusive}[0.00]\text{plosive}[0.00]\text{period}[0.00]\text{point}[0.00] \\
& \text{diaphragm}[0.00]\text{catch}[0.00]\text{blockage}[0.00]\text{block}[0.00] \\
& \text{closure}[0.00]\text{occlusion}[0.00])_n
\end{aligned}$$

$$\begin{aligned}
V(\text{stay}) = & (\text{remain}[0.00]\text{rest}[0.00]\text{stick}[0.00] \\
& \text{bide}[0.00]\text{abide}[0.00]\text{continue}[0.00]\text{detain}[0.00]\text{delay}[0.00] \\
& \text{persist}[0.00]\text{outride}[0.00]\text{quell}[0.00]\text{appease})_v \\
& (\text{arrest}[1.00]\text{check}[0.33]\text{halt}[0.33]\text{stop}[0.33]\text{hitch}[0.50] \\
& \text{stoppage}[1.00])_n
\end{aligned}$$

Nous déduisons ensuite les valeurs de projection des mots candidats du vecteur conceptuel $V(\text{stay})$ dans chaque catégorie. La valeur de projection de tous les mots candidats est nulle exceptée ceux qui sont communs à $c_1^n=\text{arrest}$, $c_2^n=\text{check}$, $c_3^n=\text{halt}$, $c_4^n=\text{hitch}$, $c_5^n=\text{stoppage}$. Les valeurs de projection sont déterminées ainsi :

- $x_1^n = \frac{1-0.00*1}{35} = 0.029$, avec $c_1^n=\text{arrest}$ et $n = 35$, $\delta_{stop}=0.00$, $\delta_{stay}=1$
- $x_2^n = \frac{1-0.00*0.33}{35} = 0.029$, avec $c_2^n=\text{check}$ et $n = 35$, $\delta_{stop}=0.00$, $\delta_{stay}=0.33$
- $x_3^n = \frac{1-0.00*0.33}{35} = 0.029$, avec $c_3^n=\text{halt}$ et $n = 35$, $\delta_{stop}=0.00$, $\delta_{stay}=0.33$
- $x_4^n = \frac{1-0.00*0.50}{35} = 0.029$, avec $c_4^n=\text{hitch}$ et $n = 35$, $\delta_{stop}=0.00$, $\delta_{stay}=0.50$
- $x_5^n = \frac{1-0.00*1}{35} = 0.029$, avec $c_5^n=\text{stoppage}$ et $n = 35$, $\delta_{stop}=0.00$, $\delta_{stay}=1$

Pour la catégorie « Nom », nous en déduisons que la contribution du mot « stay » dans la direction « stop » est : $x^n = x_1^n + x_2^n + x_3^n + x_4^n + x_5^n = 0.15$. Il en résulte que la contribution finale est :

$$x = x^a + x^r + x^n + x^v = 0.15.$$

3.3.2 Direction sémantique principale d'un concept

Dans cette section, nous nous intéressons à déterminer la direction sémantique associée à un concept. Pour aboutir à un tel résultat, plusieurs étapes sont nécessaires. La première étape de la méthode consiste à identifier les phrases dans le corpus qui décrivent ce concept et à projeter chacune d'elle dans l'espace de décision. La seconde étape permet de déterminer la contribution de chaque phrase dans l'espace de décision. La troisième étape consiste à sommer la contribution de toutes les phrases sur chaque axe de l'espace de décision et à déterminer la direction sémantique dans laquelle la somme de leurs contributions est la plus importante. Cette direction sémantique est appelée la direction principale de toutes les directions sémantiques de l'espace de décision.

Dans le cas où au moins deux directions ont le même score, la direction qui assure la sécurité de navigation est privilégiée. D'après les experts, les décisions suivantes sont classées selon l'ordre décroissant de sécurité : « back, stop, maneuver et continue ». Ainsi dans le tableau 3.3, la direction correspondant au concept « low water » concept est de « retourner ».

<i>Affordance</i> <i>Phrase</i>	maneuver	stop	continue	back
Total	0.12	0.12	0.02	0.12

Tableau 3.3 – Valeurs de projection du concept « low water » dans l'espace de décision (« back, stop, maneuver et continue »).

3.3.3 Cas d'étude

Afin de mieux illustrer cette approche basée sur l'analyse de la sémantique associée aux concepts, nous avons pris comme cas d'étude le concept de « anchorage area »(zone de mouillage) et celui de « cardinal buoy » (bouée cardinale). Les phrases qui décrivent le concept « anchorage area » sont les suivantes ;

- s_1 : “Any vessel anchored outside of the prescribed anchorage limits must move to a prescribed **anchorage area** when space becomes available”.
- s_2 : “Whenever, in the opinion of the captain of the port such action may be necessary, he may require any or all vessels in any designated **anchorage area** to moor with two or more anchors”.
- s_3 : “Reserved anchors shall be placed well within the anchorage areas, so that no portion of the hull or rigging will at any time extend outside of the **anchorage area**”.
- s_4 : “Except in cases where unforeseen circumstances create conditions of imminent peril, or with the permission of the captain of the port, no vessel shall be anchored in baltimore harbor and

patapsco river outside of the **anchorage areas** established in this section for more than 24 hours”.

s_5 : “Any vessel anchoring, under great emergency, within this area shall be placed as close to an **anchorage areas** as practicable, and shall move away immediately after the emergency ceases”.

Dans la seconde étape de ce processus, il est question d’extraire le comportement que le marin pourrait adopter devant le concept de «zone de mouillage». Nous avons restreint notre cas d’étude aux actions ou affordances « back, stop, maneuver et continue » qui est la liste des décisions possibles proposées par l’expert. Pour l’expert en navigation maritime, ces actions décrivent les différentes possibilités que peut prendre le marin face à un objet du monde réel. Chacune de ces actions a la définition suivante :

- 1 « maneuver » indique que le marin doit changer sa trajectoire de navigation ;
- 2 « stop » indique que le marin doit immobiliser son bateau ;
- 3 « back » signifie qu’il devra rentrer parce que l’environnement devient dangereux ou impraticable (par exemple, lors d’un brouillard dense ou d’un courant marin élevé) ;
- 4 « continu » est une proposition faite au marin pour qu’il suive la même trajectoire parce qu’aucun événement d’insécurité n’a été détecté.

L’affordance principale qui peut être associée au concept de «zone de mouillage» est *stop*, parce qu’il détient la direction sémantique la plus grande avec pour valeur : $0.06 = 0.01+0.00+0.00+0.01+0.04$. Les coordonnées de projection de chaque phrase dans un espace de décision sont résumées dans la table 4.3 et la figure 3.3a montre une visualisation de cet espace de décision.

<i>Affordance Phrase</i>	maneuver	stop	continue	back
s_1	0.00	0.01	0.00	0.00
s_2	0.01	0.00	0.00	0.01
s_3	0.01	0.00	0.00	0.01
s_4	0.00	0.01	0.00	0.00
s_5	0.01	0.04	0.01	0.00
Total	0.03	0.06	0.01	0.02

Tableau 3.4 – Valeurs de projection des phrases associées au concept de « zone de mouillage » dans un espace de décision ($V(\text{maneuver}), V(\text{stop}), V(\text{continue}), V(\text{back})$).

Intéressons nous maintenant à rechercher l’affordance associée à « bouée cardinale ». En supposant les mêmes documents que décrits précédemment, les phrases suivantes ont été extraites :

s_1 : “For example, a particular **cardinal buoy** represented through a symbol on a chart”.

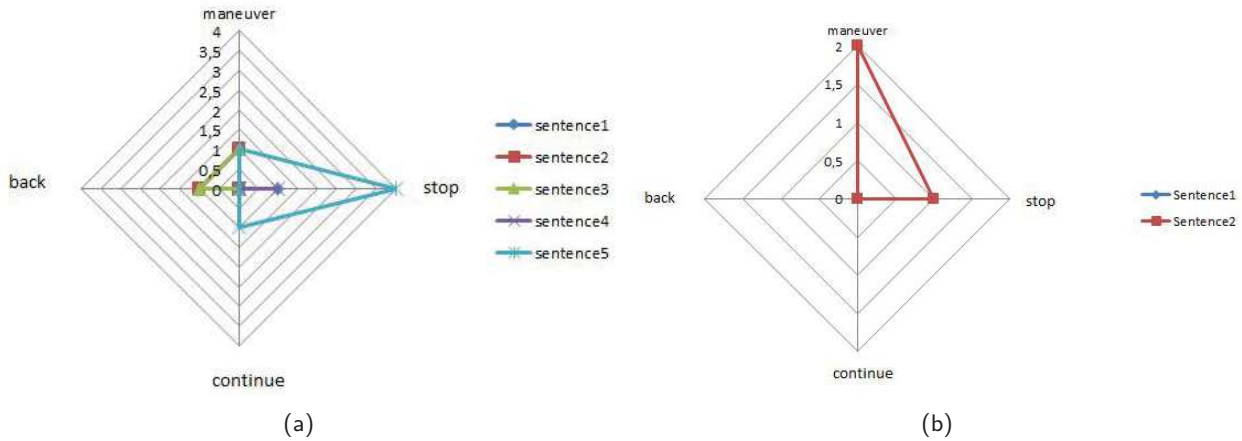


Figure 3.3 – (Visualisation de l’espace de décision associé au concept (a) «anchorage area» (zone de mouillage) et (b) «cardinal buoy» (bouée cardinale).

s_2 : “The top marks of **cardinal buoys** consist of the combination of two black cones mounted one above the other on the top of the buoy with the following, combinations :

- a. both cones pointing up = North cardinal,
- b. both pointing down = South cardinal,
- c. one pointing up and the other down with their bases together = East cardinal,
- d. one pointing up and the other pointing down with their points together = West cardinal.

La valeur de l’affordance principale associée au concept « bouée cardinale » est *maneuver*, avec pour valeur la plus élevée $0.04=0.02+0.02$. La table 3.5 résume les valeurs de projection de chaque phrase dans l’espace de décision et montre une visualisation de cet espace de décision sur la figure3.3b.

<i>Affordance Phrase</i>	maneuver	stop	continue	back
s_1	0.02	0.00	0.00	0.00
s_2	0.02	0.01	0.00	0.00
Total	0.04	0.01	0.00	0.01

Tableau 3.5 – Valeurs de projection des phrases associées au concept de «bouée cardinale» dans l’espace de décision ($V(\text{maneuver}),V(\text{stop}),V(\text{continue}),V(\text{back})$).

3.4 Utilisation des gloses dans la recherche de la sémantique

Dans certains cas, il peut arriver qu’un mot ou une phrase dans un espace de décision fournisse un vecteur conceptuel dont la norme euclidienne est égale à zéro (c.-à-d., chaque contribution sémantique est égale à zéro). En conséquence aucune décision ne peut émerger. Confrontés à cette

situation, nous avons choisi d’explorer les gloses afin d’améliorer nos résultats. Les gloses d’un mot sont les définitions de celui-ci. Par exemple, une glose du mot « port » peut être « une place (port ou aéroport) où les gens et les marchandises peuvent entrer ou quitter un pays » (identique à la définition du synset s_1^n de la section 3.2). Comme aucun résultat de décision n’a été retenue en utilisant l’approche proposée, le principe consiste alors à s’intéresser aux définitions du mot (c.-à-d. à ses gloses) pour extraire l’information sémantique qui y est associée. De nouvelles valeurs de contribution seront calculées pour les phrases ayant un vecteur dont la contribution est nulle. Celles-ci seront calculées à partir des gloses de chaque mot de la phrase. Pour chaque mot w , nous projetons chaque glose dans l’espace de décision (voir la section 3.3.1). La glose la plus pertinente en termes de contribution sémantique pour chaque mot w est celle qui sera utilisée pour la définition du mot dans le contexte du document. La norme infinie ($|||_{\infty}$) est la norme infinie utilisée pour trouver la glose la plus pertinente. Finalement, les coordonnées générées par chaque mot de la phrase considérée sont sommées pour trouver la nouvelle contribution de la phrase. Par exemple, soient deux phrases de notre corpus associées au concept de « brouillard dense ».

s_1 : “D’Avril à Septembre, il y a seulement quelques jours avec les “brouillards denses”.”

s_2 : “**Le brouillard dense** est une activité régulière en Mer et qu’on devra prévoir en cas de chaleur inhabituelle, d’hiver humide et en printemps”.

Ces deux phrases ne génèrent aucune décision dans notre espace de décision pour le concept de « brouillard dense » puisque la contribution sémantique sur tous les axes est nulle, c.-à-d :

s_1 : maneuver[0.00]stop[0.00]continue[0.00]back[0.00]

s_2 : maneuver[0.00]stop[0.00]continue[0.00]back[0.00]

Dans ce cas, les gloses des termes des phrases s_1 et s_2 sont explorées pour trouver une décision plus précise. Les projections des différentes gloses dans l’espace de décision sont indiquées par les résultats présentés dans la table 3.6 et conduisent à la décision de *retour*.

$\frac{Affordance}{Phrase}$	maneuver	stop	continue	back
s_1	0.00	0.01	0.01	0.12
s_2	0.00	0.05	0.02	0.35
Total	0.00	0.06	0.03	0.47

Tableau 3.6 – Valeurs de projection des phrases associées au concept de « brouillard dense » dans l’espace de décision ($V(\text{maneuver}), V(\text{stop}), V(\text{continue}), V(\text{back})$) utilisant le concept de gloses.

Description de l’algorithme

- 1: **procedure** *segmentationofDocument*(*AllDocument D, File corpus*) ▷ Cette fonction prend plusieurs type de documents (formats : pdf,txt,html) en entrée et les segmente en mots.

La sortie de cette segmentation est stockée dans un fichier corpus

Entrée AllDocument D

2:

Sortie File corpus

3:

4: **pour chaque** document d dans D **faire**

5: $corpus \leftarrow chunker(d)$ \triangleright segmente le document en mots et les sauvegarde dans le ou les fichiers du corpus

6: **fin pour**

7: **fin procedure**

8: **procedure** $generateConceptualVector(Concept\ c, ConceptualVector\ v_c)$ \triangleright Cette fonction permet de générer un vecteur conceptuel v du concept c . Le concept est extrait automatiquement dans l'ensemble des documents en utilisant un logiciel de désambiguïsation comme Yatea couplé avec un Treetagger

Entrée Concept c

Sortie Vecteur conceptuel v_c

9: réel d

10: $v_c \leftarrow \emptyset$

11: $pos \leftarrow getAllGrammaticalCategories(c)$

12: $pos \leftarrow C \{n, a, r, v\}$ \triangleright ou **n** indique le nom, **a** indique l'adjectif, **r** indique l'adverbe et **v** indique le verbe

13: **pour chaque** cat dans pos **faire**

14: $SynsetCandidate \leftarrow generateAllSynset(c, cat)$ \triangleright retourne l'ensemble des synsets du concept c dans la catégorie cat

15: **pour chaque** synset dans SynsetCandidate **faire** \triangleright Calcul de la distance sémantique $\delta(synset, c)$ en utilisant l'équation 3.2

16: $d \leftarrow computeDistance(synset, c)$

17: **fin pour**

18: $v_c \leftarrow v_c \cup \{synset[d]\}$

19: **fin pour**

20: **fin procedure**

21: **procedure** $generateConceptualVectorSentence(Sentence\ s, VecteurConceptuel\ v_s)$

Entrée Sentence s

Sortie Vecteur conceptuel v_s

22: concept c

23: $v_s \leftarrow \emptyset$

24: **pour chaque** concept c dans s **faire**

25: $generateConceptualVector(c, v_c)$

26: $v_s \leftarrow v_s \oplus v_c$ \triangleright Réalisation de la somme directe entre deux vecteurs

27: **fin pour**

28: **fin procedure**

29: **procedure** $generateSpace(Directions\ D, Espace\ E)$ \triangleright Cette fonction permet de générer les vecteurs conceptuels représentant la base dans l'espace de décision

Entrée Directions D

Sortie Espace E

30: $E \leftarrow \emptyset$

31: **pour chaque** $d \in D$ **faire**

32: $D \leftarrow \{maneuver, continu, stop, back\}$ ▷ Les directions sémantiques utilisées

33: $generateConceptualVector(d, v_d)$

34: $E \leftarrow E \cup v_d$

35: **fin pour**

36: **fin procedure**

37: **procedure** *selectedSentences*(*File corpus*, *Concept c*, *List S*) ▷ Cette fonction sélectionne les phrases S associées au concept c

Entrée File corpus, Concept c

Sortie List S

38: $S \leftarrow \emptyset$

39: **pour chaque** sentence s dans corpus **faire**

40: **si** $contain(s, c) = true$ **alors** ▷ Cette fonction est à vraie si c est dans s

41: $S \leftarrow S \cup s$

42: **fin si**

43: **fin pour**

44: **fin procedure**

45: **procedure** *computeCoordWithGlosesOfSentence*(*Sentence s*, *Vector Coord*, *Direction D*)

Entrée Sentence s, Direction D

Sortie Vecteur Coord

46: $VecteurCoordg, bestCoord \leftarrow (0, 0, 0, 0)$

47: **pour chaque** word w dans s **faire**

48: $generateGloses(w, G)$ ▷ Cette fonction est intégrée dans la librairie *WordNet*, il prend un mot w et génère ses gloses

49: **pour chaque** gloses g dans G **faire**

50: $projectionSentencesInSpace(g, D, Coordg)$

51: **si** $\|Coordg\|_{\infty} \geq \|bestCoord\|_{\infty}$ **alors**

52: $bestCoord \leftarrow Coordg$

53: **fin si**

54: **fin pour**

55: $Coord \leftarrow Coord + bestCoord$ ▷ Somme de deux vecteurs à 4 dimensions

56: **fin pour**

57: **fin procedure**

58: **procedure** *projectionSentencesInSpace*(*List S*, *Directions D*, *Vector Coord*)

Entrée List S, Directions D

Sortie Vector Coord

59: $Vecteurcoord \leftarrow (0, 0, 0, 0), coordg \leftarrow (0, 0, 0, 0)$ ▷ Les coordonnées de la phrase dans un espace de décision

60: $generateSpace(D, E)$

61: **pour chaque** direction v_d dans E **faire**

62: $genConceptualVector(d, v_d)$

63: **pour chaque** sentence s dans S **faire**

```

64:      generateConceptualVectorSentence( $s, v_s$ )
65:      pour chaque concept  $c_i$  dans  $(v_s \cap v_d)$  faire
66:           $x_i \leftarrow \text{computerCoordinate}(c_i, v_s, v_d)$       ▷ Calcule  $x_i$  avec la formule de l'équation 3.5
67:           $\text{coord}[d] \leftarrow \text{coord}[d] + x_i$ 
68:      fin pour
69:  fin pour      ▷ Nous prenons en compte les différentes gloses de la phrase
70:  si  $\| \text{coord} \|_2 = 0$  alors
71:       $\text{computeCoordWithGlosesOfSentence}(s, \text{Coordg}, D)$ 
72:  fin si
73:  si  $\| \text{Coordg} \|_2 > 0$  alors
74:       $\text{Coord} \leftarrow \text{Coord} + \text{coordg}$ 
75:  sinon
76:       $\text{Coord} \leftarrow \text{Coordg} + \text{coord}$ 
77:  fin si
78:  fin pour
79: fin procedure
80: procedure  $\text{computeHighCoordinate}(\text{Vecteur } \text{Coord}, \text{ Directions } D, \text{ direction } \text{sem})$       ▷ Cette
    fonction retourne la contribution principale dans l'espace de contribution
Entrée Vecteur Coord
Sortie float val, direction sem      ▷ val est la valeur de la contribution du concept sur la direction
    sémantique sem
81:   $\text{val} \leftarrow \text{maxIndice}(\text{Coord})$       ▷ La fonction max retourne l'indice de la valeur maximale du vecteur
82:   $\text{sem} \leftarrow \text{getDirection}(D, \text{val})$       ▷ La fonction getDirection retourne la direction correspondant à
    l'indice
83: fin procedure
84: procedure principal      ▷ Nous cherchons l'espace de décision associé au concept
85:   $\text{AllDocument} D$ 
86:   $\text{File } \text{corpus}$ 
87:   $\text{Concept } c = \text{'cardinal buoy'}$ 
88:   $S \leftarrow \emptyset$       ▷ Liste des phrases sélectionnées
89:   $\text{DirectionsDir} \leftarrow \{\text{maneuver}, \text{continu}, \text{stop}, \text{back}\}$       ▷ Conforme à notre description précédente
90:   $\text{segmentationofDocument}(D, \text{corpus})$ 
91:   $\text{selectedSentences}(\text{corpus}, c, S)$ 
92:   $\text{projectionSentencesInSpace}(S, \text{Dir}, \text{Coord})$ 
93:   $\text{computeHighCoordinate}(\text{Coord}, \text{Dir})$ 
94: fin procedure

```

3.5 Conclusion

Dans cette section on introduit de façon détaillée une stratégie d'extraction de la sémantique associée à un concept. Cette extraction utilise un ensemble de documents segmentés dans un corpus. Les documents utilisés ont été écrits par des experts du domaine ce qui garantit la richesse de ces documents. A l'aide du corpus produit, il est possible d'utiliser les techniques de «TAL» afin de

proposer un algorithme d'extraction de la sémantique. L'algorithme que nous avons proposé utilise le fait que chaque concept est structuré dans une base de donnée lexicale (p. ex. WordNet) comme un ensemble de synonymes cognitifs (synsets) en fonction de leur catégorie grammaticale : noms, verbe, adjectif et adverbe.

L'utilisation des synsets d'un mot, nous a permis d'intégrer la technique du vecteur conceptuel à l'algorithme proposé. Un vecteur conceptuel prend en compte les mots candidats du synset et leur distance au mot. L'intérêt de la sémantique extraite permet de réaliser l'aide à la décision pour la navigation. Le but de cette étape est de produire la décision associée à un concept. Pour déterminer cette décision, nous utilisons un espace de décision suivant lequel nous calculons les contributions sémantiques de chaque phrase décrivant le concept dans le corpus. La décision retenue sera celle qui produira la contribution la plus élevée parmi les différentes décisions de l'espace. La décision extraite a permis d'enrichir la propriété «affordance» du concept sémantique du modèle ontologique «AtomicEntity» proposé à la section 2.1. La valeur de cette propriété peut être indiquée par l'expert si elle est connue et si elle ne l'est pas, la stratégie du vecteur conceptuel proposée dans cette section pourrait l'extraire.

Il peut arriver qu'aucune décision ne soit retenue d'un ensemble de phrases si toutes ces phrases génèrent des contributions nulles sur toutes les directions de l'espace de décision. La stratégie a été améliorée pour résoudre ce problème en se basant sur les concepts de gloses. Cette stratégie a été appliquée dans le contexte maritime plus particulièrement dans l'aide à la navigation mais pourrait aussi facilement être appliquée dans d'autres domaines.

Chapitre 4

Service de routage dynamique et formalisation d'une trajectoire de navigation

Sommaire

4.1	Intérêt d'un routage approprié pour l'environnement maritime	86
4.2	Concepts fondamentaux des algorithmes basés sur des heuristiques . . .	89
4.3	Choix du pavage ou maillage	93
4.3.1	Facteur de branchement	93
4.3.2	Nombre de nœuds contenus dans la solution	94
4.3.3	Nombre de nœuds dans une direction	94
4.3.4	Maillage hexagonal	97
4.4	Structure de données	103
4.5	Algorithme <i>BIDA</i>* sur un maillage hexagonal	109
4.6	Conclusion	115

LA mobilité est une activité couramment rencontrée dans des environnements complexes. Le routage fait partie de l'un des services de la mobilité. Il est défini comme le mécanisme par lequel des chemins sont sélectionnés, le plus souvent dans un réseau, pour acheminer les données d'un expéditeur jusqu'à un ou plusieurs destinataires. Il se dénomme « pathfinding » dans le domaine des SIG ou dans la programmation des jeux. Dans l'objectif de présenter un cas d'étude permettant de justifier l'architecture fonctionnelle décrite au chapitre 2, nous avons proposé un algorithme de routage maritime.

Ce chapitre présente une structure de données couplée à une heuristique appropriée et appliquée en temps réel afin de trouver des chemins optimaux dans des environnements larges et non structurés par des réseaux. A l'aide des données maritimes de la rade de Brest décrites dans un format *S57*

présenté au chapitre 2 nous avons pu déployer un service très utilisé dans le domaine maritime. Le service proposé est le routage maritime appliqué en temps réel. Ce dernier permet de trouver une route de navigation en temps réel tout en préservant la sécurité des navigateurs et celle des usagers de l'environnement maritime. Il s'applique sur une structure de données spatiales permettant de trouver une route optimale entre deux positions géographiques en tenant en compte de la complexité de l'environnement maritime (éviter des zones interdites et des zones de danger). L'optimalité de la route est fondée sur une heuristique de minimisation de critères (la distance, etc.) à respecter lors de la navigation. Notre approche est basée sur un algorithme de parcours en profondeur pour la recherche de la route optimale et le critère à minimiser est la distance. Celle-ci est réalisée sur des cônes directionnels (des faisceaux symétriques) et des cellules hexagonales construites de façon complémentaires et itératives. Cette approche a été modélisée par un graphe dynamique dans l'intérêt de réduire le temps d'accès aux données et l'espace mémoire. Pour mieux aborder ce chapitre, la section 4.1 met en évidence la particularité et l'intérêt de proposer des algorithmes appropriés à l'environnement maritime. Elle est suivie par la section 4.2 qui présente les concepts fondamentaux des algorithmes basés sur des heuristiques. Ensuite la section 4.3 analyse les différents types de maillage appliqués pour le routage afin de choisir celui qui est le plus approprié pour la navigation maritime. La section 4.4 propose une structure de données afin de stocker le maillage déployé par le routage. Enfin, le chapitre se termine par la section 4.5 qui décrit les particularités de l'algorithme et présente un cas d'étude sur un environnement réel.

4.1 Intérêt d'un routage approprié pour l'environnement maritime

La planification de route entre deux positions géographiques est toujours restée un problème non trivial dans le domaine maritime. L'un des facteurs de complexité est la croissance élevée du nombre d'entités dans cet environnement et des règles de navigation (cf. chapitre 2) qui rend la navigation difficile à accomplir même par des professionnels de la navigation. Les difficultés rencontrées par ces professionnels de la navigation sont : le choix de la bonne règle à appliquer en temps réel, la distance de visibilité réduite par des phénomènes imprévisibles (p. ex. le brouillard), le tracé de route fastidieux sur des cartes papier.

Plusieurs algorithmes ont déjà été proposés par des systèmes de transports intelligents pour les environnements terrestres dans lesquels la recherche de chemin est contrainte par le réseau. C'est le cas des engins dans les systèmes « RGS »¹ et les systèmes temps réels « AVDS »² dont la stratégie de routage est fortement contrainte par le réseau routier.

L'environnement maritime a aussi besoin d'algorithmes qui lui sont propres (c.-à-d., qui tiennent compte de sa complexité) pour trouver une route optimale entre deux points géographiques (c.-à-d., une origine et une destination). Même si certaines routes de navigation sont prédéfinies pour de

1. Route Guidance System

2. Automated Vehicle Dispatching System

grands bateaux, l'environnement maritime reste un environnement ouvert dans lequel la navigation peut s'appliquer dans toutes les directions créant ainsi un problème qui rend la navigation maritime assez différente des autres environnements contraints par un réseau.

Le principal défi considéré dans ce chapitre vise à améliorer les techniques de recherche de chemin en temps réel, où le chemin proposé est en permanence contraint par des objets et des phénomènes qui modifient l'environnement et la navigation. Quand nous parlons de temps réel, notre temps de réponse est de l'ordre des secondes ce qui permet de prendre en compte les différents changements (p.ex., la modification de la visibilité (brouillard, nuit), la perception des bateaux et la disparition et la réincarnation des objets (voir chapitre 2)) et l'évitement des collisions. Notre approche considère les propriétés spécifiques à l'environnement maritime (cf. la table 4.1). La table 4.1 résume les différences majeures entre le routage maritime et celui basé sur un réseau structuré ou connu à l'avance. De plus, plusieurs propriétés et règles régissent le routage maritime et mettent en évidence sa particularité :

- la régulation du trafic,
- la régulation des collisions [U.S Coast Guard, 1999],
- les zones de danger (c.-à-d., les lignes ou zones de séparation de trafic),
- les zones interdites (p.ex., les zones d'essai militaires),
- les objets mobiles (p.ex., les bateaux),
- les changements dynamiques tels que les marées, les courants, le brouillard, le vent

Les objets et leurs saillances dépendent des conditions de la perception : la visibilité et la temporalité jour et nuit. Toutes ces contraintes ont un impact fort sur les ressources (le temps de calcul et l'espace mémoire) pénalisant ainsi le routage en temps réel.

A partir de ces constats, il est primordial de proposer des applications de routage appropriées à l'environnement maritime. Bien que, certains systèmes d'informations proposés pour la navigation maritime intègrent des algorithmes de navigation [Jianping Xu, 1994],[Yu *et al.*, 2003], aucun d'entre eux ne prend en compte toutes les spécificités de l'environnement maritime précédemment énoncées. Plusieurs algorithmes de routage maritime proposés à ce jour pour la course de voiliers ne prennent en compte que les effets des phénomènes météorologiques (vent, courant) sur des données rasters [Ahmadi *et al.*, 2008]. La figure 4.1 montre un exemple de route proposée par le logiciel MaxSea¹. Ce logiciel considère seulement deux facteurs pour la réalisation de routes maritimes :

- les prévisions de la météo,
- la vitesse du voilier.

Cependant, cette route ne préserve pas la sécurité et ne peut donc être considérée comme une solution de routage en temps réel.

Toujours dans l'objectif de contribuer à la réalisation de systèmes d'aide à la décision qui permettront d'augmenter la faculté cognitive du capitaine dans des situations de stress élevé en proposant des routes à suivre, ce travail s'est concentré sur les problèmes de routage appliqués à des envi-

1. <http://www.maxsea.com>

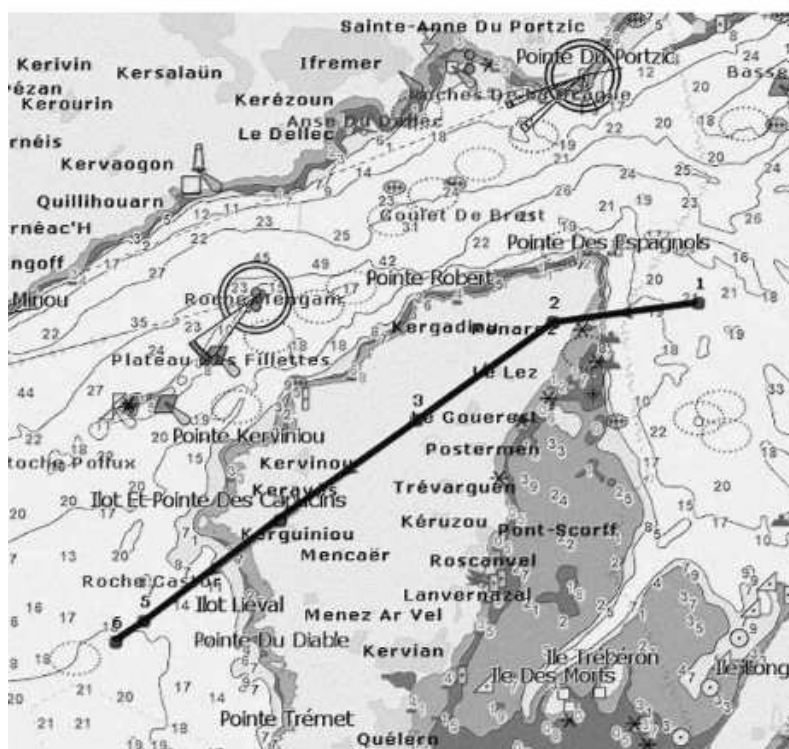


Figure 4.1 – Exemple de route proposée par le logiciel MaxSea où la trajectoire traverse la zone de terre compromettant ainsi la sécurité

	Espace	Entités	Temps de réponse	Planification de routes
Routage maritime	Ouvert	Dynamique	A une grande influence	Principalement basé sur des landmarks
Routage basé sur un réseau	Contraint	Dynamique	A une faible influence	Fortement contraint par le réseau

Tableau 4.1 – Principales différences entre le routage maritime et le routage basé sur un réseau structuré

ronnements maritimes en deux dimensions (c.-à-d., en cartographie 2D). La navigation sur laquelle nous avons travaillé est la navigation côtière par opposition à la navigation hauturière. La navigation hauturière s'effectue lorsque le navire navigue en haute mer (hors de vue de la terre) et la navigation côtière, lorsqu'il navigue non loin des côtes (en vue de la terre). La phase de transition, entre navigations hauturière et côtière est l'atterrissage. La navigation hauturière présente moins de contraintes (moins d'obstacles et de règles à respecter) que la navigation côtière. Le fait de travailler à proximité des côtes réduit la taille de l'échantillon des données. Cette réduction de la

taille de l'espace de travail par rapport au globe terrestre tout entier, nous conduit à utiliser une représentation plane et non une représentation ellipsoïdale³ qui tient compte de la rotondité de la terre. Dans tout ce travail, notre environnement a été modélisé par le format *S57* (cf. chapitres 2 et 3) enrichi par une ontologie spatio-temporelle. Les empreintes spatiales des entités du format *S57* sont représentées au format vectoriel (en projection WSG84). La compréhension des approches de routage demandent un certain nombre de pré-requis. La section suivante expose les différentes thématiques exploitées par les algorithmes de recherche de chemins optimaux.

4.2 Concepts fondamentaux des algorithmes basés sur des heuristiques

La réalisation d'un routage efficace en SIG nécessite le choix de deux concepts fondamentaux. Le premier est le choix de la grille et le second est le choix de l'algorithme qui sera appliqué sur cette grille dont l'abstraction se présente comme un graphe orienté. Dans la littérature certaines approches contredisent cette réalité en proposant des algorithmes approchés basés sur la minimisation d'équations linéaires ou non linéaires [Dolinskaya, 2012]. La difficulté de ces applications réside dans l'incapacité de raisonner en temps réel sur des composantes spatiales ou de coupler leurs approches aux bases de données spatio-temporelles. Par conséquent, elles ne peuvent pas être utilisées pour de l'aide à la décision en temps réel. Pourtant l'aide à la décision est une fonctionnalité majeure de notre travail de thèse. Ceci nous a donc amené à choisir une grille structurée d'une façon particulière et a sollicité une étude approfondie sur les différents types de grille et leur impact (en terme de complexité) sur la navigation. Plusieurs applications dédiées aux systèmes de transport, qui ont intégré des algorithmes de planification ou de routage, superposent une grille sur la région d'étude et un algorithme est utilisé afin de trouver le meilleur chemin sur cette grille. Généralement, la région est modélisée par un maillage carré, hexagonal ou *octile*⁴ « Regular Square Grid » [Yap, 2002]. [Cui et Shi, 2011] ont montré que le choix de la grille utilisée peut avoir un impact sur la performance des algorithmes de recherche de chemin. [Sahr *et al.*, 2003] et [Tong *et al.*, 2013] ont introduit des systèmes globaux de discrétisation de grille (DGGSs)⁵ qui sont considérés comme une structure prometteuse pour la représentation de l'information géospatiale du globe terrestre fondée sur des grilles hexagonales. Dans nos travaux, nous avons montré que la construction du nombre de cellules composant le chemin solution peut être réduit en termes d'espace mémoire et de temps de calcul en utilisant une approche itérative.

Concernant, le deuxième choix qui se concentre sur les algorithmes, deux grandes catégories d'algorithmes sont communément utilisés pour trouver un chemin optimal connectant deux points dénommés s (point de départ) et t (point d'arrivée) :

- les algorithmes unidirectionnels,
- les algorithmes bidirectionnels.

3. ellipsoïde est une surface du second degré de l'espace euclidien à trois dimensions

4. carré qui offre huit directions de déplacements

5. Discrete Global Grid Systems

Chacune de ces deux grandes catégories est subdivisée en plusieurs classes d'algorithmes. La classe d'algorithmes la plus utilisée est celle dite admissible. Un algorithme sera dit admissible s'il garantit de trouver une solution valide qui satisfait un chemin minimal parmi l'ensemble des solutions existantes. Les bases des premiers algorithmes admissibles ont été proposées par [Moore, 1957] et [Dijkstra, 1959]. Les algorithmes de recherche de chemin s'appliquent régulièrement sur des graphes qui sont des abstractions du maillage d'une région donnée.

Considérons un maillage (c.-à-d., une partition de la zone de recherche en cellules) associé à un graphe direct $G(X, U)$ où X est une collection des nœuds (c.-à-d., des cellules) et U est une collection d'arcs. Chaque arc de U peut être considéré comme une paire ordonnée de nœuds de X , commençant par un nœud n_0 incluant le point initial s . A chaque itération, l'algorithme explore le graphe² en suivant une recherche ou un parcours en profondeur ou en largeur et en évoluant pas à pas (partant du nœud vers le meilleur de ses successeurs) jusqu'à atteindre le nœud final n_N qui contient le point d'arrivée t . Dans la conceptualisation, un chemin sera alors défini comme un sous graphe de G où les nœuds de X sont ordonnés de façon croissante et où le nœud initial n_0 est le nœud racine. Un chemin peut aussi être identifié comme une séquence de nœuds commençant par le nœud racine n_0 et se terminant par le nœud final n_N décrit comme suit :

$$u = (e_{n_0 n_1}, e_{n_1 n_2}, e_{n_2 n_3}, e_{n_3 n_4}, e_{n_4 n_5}, \dots, e_{n_{N-1} n_N})$$

où $s \in n_0$, $t \in n_N$ et $e_{n_{k-1} n_k}$ est chemin formé par les nœuds n_{k-1} et n_k . De la même manière on peut l'écrire ainsi :

$$u = (n_0, n_1, n_2, n_3, n_4, \dots, n_N).$$

La longueur du chemin est calculée en utilisant la distance l , c.-à-d. :

$$l(u) = \sum_{i=0}^{N-1} \|e_{n_i n_{i+1}}\|, \text{ où } \|\cdot\| \text{ est la } L_2 \text{ norme.}$$

Pour trouver le meilleur successeur n_j du nœud courant n_i il faut minimiser la distance $l(n_i, n_j) + \lambda_{n_0, n_i}$ où $l(n_i, n_j)$ est le coût associé à l'arc entre les deux nœuds n_i et n_j , et λ_{n_0, n_i} est le coût minimal évalué pour atteindre le nœud n_i partant de la racine n_0 . Dans la littérature l'algorithme de *Dijkstra* a connu une extension plus efficace appelée A^* . A^* est un algorithme appartenant à la grande classe des algorithmes heuristiques utilisés pour résoudre des problèmes complexes. Dans son principe, il introduit une nouvelle fonction de coût ou fonction heuristique pour atteindre le nœud final n_N . Pour l'algorithme A^* , l'estimation du coût d'un nœud courant n_i est alors définie par la fonction $f(n_i) = g(n_i) + h(n_i)$, où $g(n_i)$ est le coût du chemin parcouru partant de n_0 à n_i et $h(n_i)$ est la fonction heuristique qui n'est qu'une estimation du coût restant pour atteindre le nœud cible ou le nœud final n_N à partir de n_i . Il a été prouvé que A^* est un algorithme admissible car il ne surestime jamais le coût de la fonction f et le calcule à chaque itération par opposition

2. Dans cette conceptualisation proposée, un graphe est associé à un maillage.

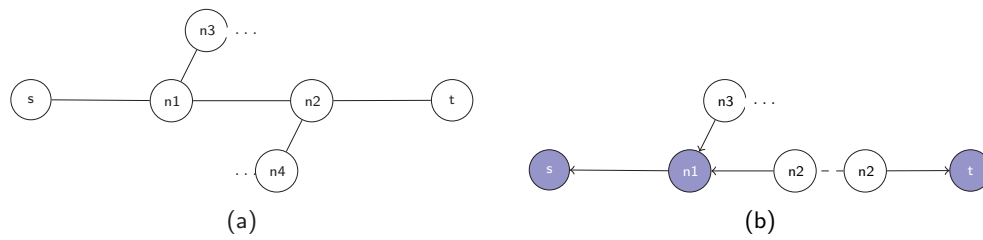


Figure 4.2 – (a) Graphe initial parcouru par l'algorithme (BSA^*) et (b) les deux partitions de l'algorithme de recherche se rencontrant au nœud $n2$

avec l'algorithme de *Dijkstra* qui lui estime le coût de cette fonction à la fin du processus. Une des avancées en *IA* pour les algorithmes de recherche de chemin a été de trouver la version itérative en profondeur de A^* , appelée (IDA^*)⁶ proposée par [Korf, 1985], où la recherche en profondeur est améliorée par une connaissance du graphe. L'intérêt de cette approche est de visiter seulement les nœuds pouvant appartenir à la solution optimale. L'algorithme (IDA^*) est une amélioration de la recherche par périmètres proposée par [Dillenburg et Nelson, 1994]. Par ailleurs tous les algorithmes de recherche de chemin optimal peuvent avoir une version bidirectionnelle.

Les algorithmes bidirectionnels appartiennent à la deuxième catégorie des algorithmes. Ils créent deux partitions de recherche (Avant et Arrière, voir figure 4.3b). L'objectif est de trouver le chemin optimal en utilisant une heuristique dans les deux directions « Front arrière »⁷ et « Front Avant »⁸ partant du nœud racine n_0 et du nœud final n_N . Une des premières heuristiques bidirectionnelles a été proposée par [Pohl, 1971] dénommée $BHPA^*$. Cette heuristique a été améliorée plus tard par [Kwa, 1989] qui ajoute plusieurs opérations spécifiques « nipping » et « pruning » « trimming (removing) » et « screening (not placing) » à son heuristique (BSA^*)⁹. Considérons un cas d'exemple où les opérations « nipping », « pruning » sont appliquées dans le graphe (voir figure 4.2a). L'un des objectifs primordiaux de ces opérations est d'éviter les explorations des nœuds déjà visités et de prévenir des expansions redondantes ou inutiles dans les deux partitions [Pulido *et al.*, 2012]. Soient deux points s et t respectivement les points de départ et d'arrivée de l'arbre de recherche 4.2a. L'arbre de recherche est parcouru par l'algorithme bidirectionnel (BSA^*). La partition avant va de s à t et a déjà sélectionné les nœuds s et $n1$. La partition arrière va de t à s et a seulement sélectionné le nœud t . Considérons que les deux partitions se rencontrent au nœud $n2$ (voir figure 4.2b). Si la partition arrière continue à visiter le nœud $n1$, il se pose alors un problème de discontinuité au nœud $n1$ de la partition arrière nommée « nipping » (voir figure 4.3a) puisque la solution optimale a déjà été trouvée de s à $n1$ par la partition avant. Pour les mêmes raisons (la solution optimale a déjà été trouvée de $n2$ à t par la partition arrière), l'algorithme présente une discontinuité de la partition avant nommée « pruning » si elle visite le nœud $n3$ (voir figure 4.3b)

6. Iterative Depth A^*

7. L'avant de la partition arrière

8. L'avant de la partition en avant

9. Bidirectional Search A^*

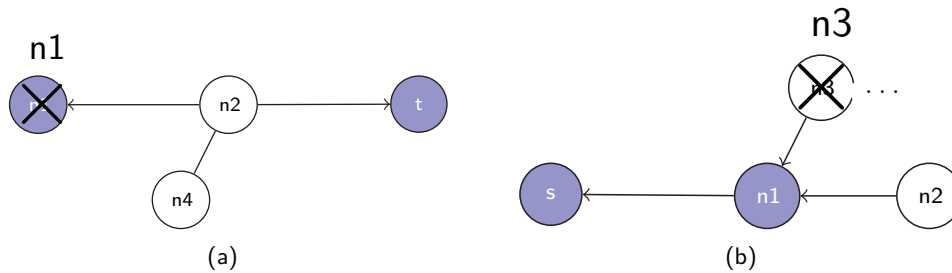


Figure 4.3 – (a) Partition avant après le « nipping » du nœud $n1$ et (b) la partition arrière après « pruning » au nœud $n3$

Le fait d'appliquer les deux heuristiques a un inconvénient crucial illustré par la métaphore des missiles (où les deux partitions peuvent se traverser sans se toucher). Cet inconvénient est dû au fait qu'ils appliquent une heuristique avec un principe d'évaluation *Front à fin*¹⁰. En comparant judicieusement *BHPA** et *BSA**, [Sint et de Champeaux, 1977] et [de Champeaux, 1983] ont proposé une évaluation *Front à front*¹¹. Quelques années plus tard, [Kaindl et Kainz, 1997] ont montré que l'évaluation « Front à front » est plus efficace que l'évaluation « Front à fin ». Dans les années 1990 [Dillenburger et Nelson, 1994] et [Manzini, 1995] ont introduit une approche de recherche en périmètre. Cette approche est basée sur une évaluation *Front à front* utilisant l'algorithme du parcours en largeur du premier d'abord¹² où les nœuds générés sont regroupés en zones de recherche délimitées par des frontières en fonction de leur coûts. Une approche plus évoluée a été proposée par [Kaindl et Kainz, 1997] qui met au point une *approche différentielle* combinatoirement moins coûteuse en appliquant des changements dynamiques de la fonction heuristique. Son principe se base sur une fonction d'estimation heuristique $h(n_i)$ qui est déduite d'autres fonctions heuristiques. Dans l'objectif de proposer une route optimale de navigation maritime en temps réel, nous discuterons dans les sections suivantes de la structure de données itératives à base de graphe que nous avons proposée et sur laquelle s'applique un algorithme de routage bidirectionnel en profondeur implémentant une évaluation « Front à front » dénommée (*BIDA**) [Tsatcha *et al.*, 2013a]. La mise en place d'un tel algorithme nécessite de discuter du choix du pavage ainsi que du processus d'évaluation itératif des nœuds du graphe.

10. qui estime le coût minimal $h(n_i)$ d'un nœud courant n_i au nœud final n_N

11. qui estime le coût minimal du nœud courant n_i d'un chemin d'une partition en utilisant une fonction heuristique $h(n_i, n_j)$ où n_j est le nœud le plus en avant de l'autre partition

12. Cet algorithme consiste à parcourir le graphe en largeur en évaluant chaque nœud du graphe suivant le principe premier entré et premier sorti (schéma d'un parcours en largeur)

4.3 Choix du pavage ou maillage

La recherche de chemin utilise le plus souvent une grille pour recouvrir une région qui sera ensuite évaluée par un algorithme de recherche. Le choix de la grille peut réduire de façon importante le coût de la route solution optimale. [Cui et Shi, 2011] montre comment les différentes représentations de cartes de planification peuvent avoir un impact sur la performance des algorithmes de recherche de chemin. Ainsi, le choix du pavage dans les algorithmes de routage est motivé ou contraint par 3 facteurs à respecter ou à considérer pour améliorer la complexité en temps de calcul et en espace mémoire. Ces trois facteurs sont les suivants :

- 1** la valeur du facteur de branchement¹³ définie par [Korf, 1985] doit être minimale. Le critère est illustré par les figures 4.4a, 4.4b, 4.4c, 4.4d et leurs valeurs sont résumées dans la table 4.3.,
- 2** le nombre de nœuds contenus dans le chemin solution doit être minimisé. D'après [Yap, 2002], le chemin solution minimise aussi le nombre de changements de direction.,
- 3** le nombre de cellules (ou maille) doit être minimal dans une direction [Yousefi et Donohue, 2004]. Ce critère doit être capable de déterminer la plus longue séquence dans une direction tout en minimisant le nombre de changements de direction et le nombre total de nœuds qui composent le chemin solution (Figure 4.7).

4.3.1 Facteur de branchement

Le facteur de branchement peut être défini comme étant le nombre de fils d'un nœud du graphe ou la moyenne des prochains états possibles de tous les états générés par l'algorithme de recherche [Korf, 1985]. La complexité des algorithmes de recherche de chemin est toujours donnée en fonction de deux paramètres : le facteur de branchement et la profondeur de la solution. Ainsi, le routage devra s'effectuer sur un graphe de recherche qui est une abstraction d'une grille. Les différents types de mailles qui composent la grille et qui permettent d'intégrer un algorithme de recherche de chemin sont : les hexagones, les carrés et les octiles. L'algorithme de recherche doit trouver un chemin optimal sans retour arrière et un tel objectif est atteint en ne visitant que les nœuds qui ont une grande probabilité de conduire à la solution.

Le facteur de branchement des différentes grilles est illustré par les figures 4.4a, 4.4b, 4.4c et 4.4d et résumé par la table 4.3. Ce dernier doit être optimisé pour réduire la complexité de l'algorithme de recherche, en supposant que toutes les directions sont permises quel que soit le maillage et dans un système de référence absolu¹⁴ défini par [Frank, 1998]. En considérant ainsi les situations illustrées les figures 4.4a, 4.4b, 4.4c et 4.4d, une grille carrée offre quatre directions de déplacement (c.-à-d., Nord, Est, Sud et Ouest) avec un facteur de branchement de 3 où le nœud précédemment

¹³. La valeur du facteur des branches est identifiée comme le nombre possible de nœuds adjacents au nœud courant utilisés pour éviter les retours en arrière

¹⁴. un système de référence absolu est composé de quatre directions principales Avant, Retour, Droit, Gauche et de quatre directions dérivées Avant-Gauche, Avant-Droit, Retour-Gauche, Retour-Droit.

évalué n'est pas pris en compte (voir la figure 4.4b). Une grille octile offre toutes les possibilités de déplacement généralement utilisées en navigation maritime (*Nord, Nord Est, Est, Sud Est, Sud, Sud Ouest, Ouest, Nord Ouest*) mais les directions *Sud Est, Sud, Sud Ouest* sont éliminées à cause des possibilités de *retour arrière* (voir la figure 4.4d). Il en résulte un facteur de branchement égal à 5. Cependant, son facteur de branchement est réduit à trois directions pour le cas des cellules qui ne sont pas sur la diagonale (voir Figure 4.4c). Une grille hexagonale propose six directions et un facteur de branchement de 3 « (Nord Est, Nord, Nord Ouest) ».

Le type de grille sera choisi afin de satisfaire les critères 1), 2) et 3).

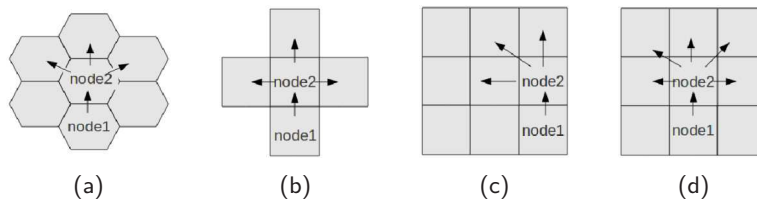


Figure 4.4 – Facteur de branchement pour (a) une grille hexagonale, (b) une grille carrée, (c) une cellule non situé sur la diagonale d'une grille octile, (d) une cellule sur la diagonale d'une grille octile.

4.3.2 Nombre de nœuds contenus dans la solution

Le nombre de nœuds contenus dans le chemin solution est aussi un critère pertinent pour la complexité des algorithmes de recherche. Le chemin solution est considéré à partir d'un point de départ s vers un point d'arrivée t . Quelle que soit la grille, le chemin optimal est donné par un graphe u qui minimise le nombre de nœuds où la distance $l(u)$ (voir section 4.2). [Yap, 2002] a prouvé qu'un chemin optimum minimise le nombre de changements de direction. Les figures 4.5 et 4.6 illustrent ce concept sur différents maillages. Par exemple pour les grilles octiles et hexagonales, il n'est pas nécessaire de calculer la plus grande profondeur sur la direction verticale. Leur plus petit chemin est donné sur la direction diagonale comme l'indique la figure 4.6. Un changement de direction peut être réalisé quand l'angle entre la ligne verticale et la ligne oblique liée à la cellule finale est respectivement de $\frac{\pi}{4}$ et $\frac{\pi}{3}$ pour les grilles octile et hexagonale (voir figure 4.6b et 4.6c). [Yap, 2002] résume le coût des algorithmes de recherche sur les différentes grilles dans la table 4.2. Le maillage hexagonal est meilleur que les maillages carrés et octile puisqu'il réduit la longueur du chemin.

4.3.3 Nombre de nœuds dans une direction

Le « clustering » des mouvements est la plus longue séquence de nœuds dans une direction qui permet de réduire le nombre de changements de direction et le nombre total de nœuds contenus dans le chemin solution. La figure 4.7 montre que la grille octile est la meilleure des grilles puisqu'elle calcule efficacement les chemins suivant les directions verticale, horizontale et diagonale. Cependant,

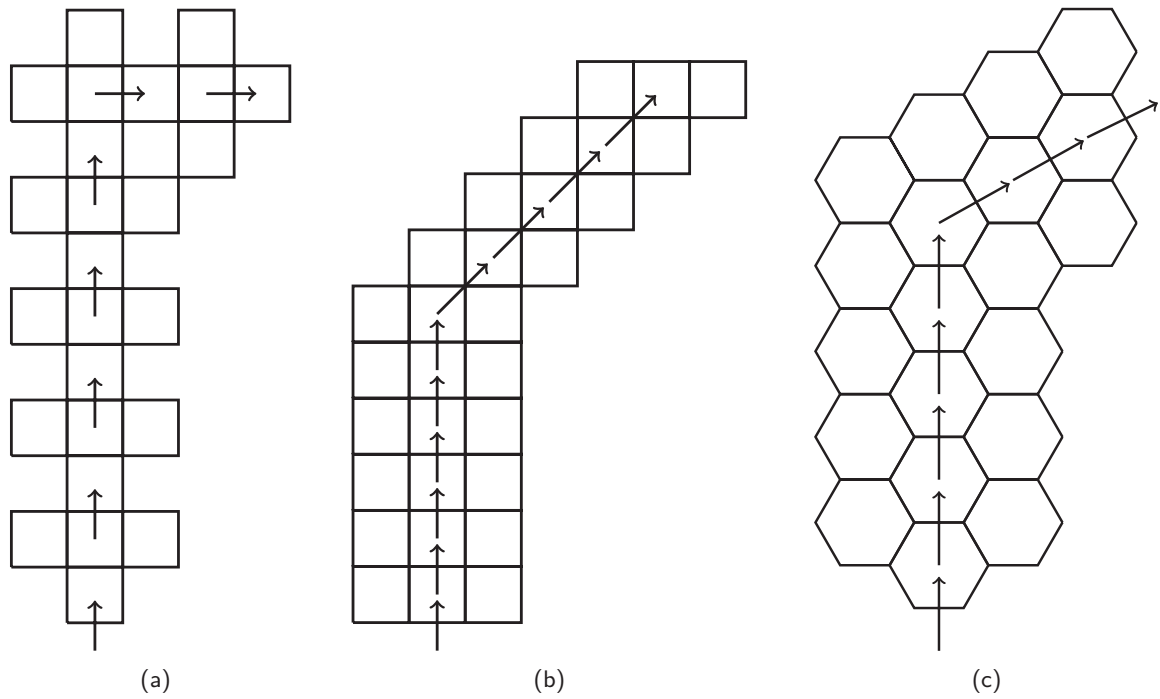


Figure 4.5 – Exemple des déplacements optimaux sur les différents types de grille.

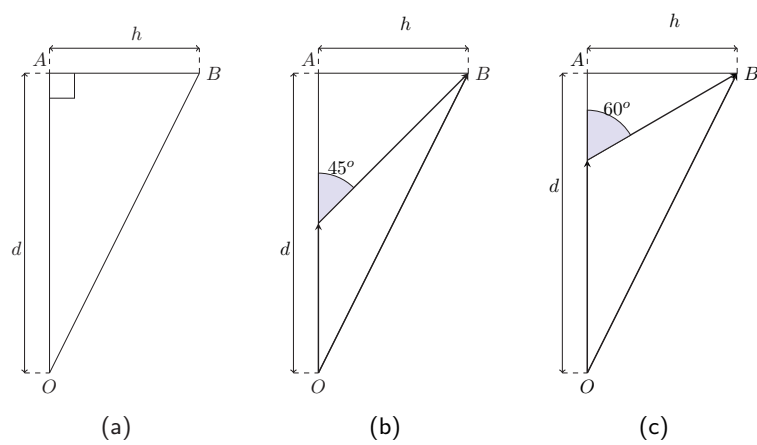


Figure 4.6 – Les angles de déviation correspondant aux déplacements présentés à la figure 4.5.

Grille	Carrée	Hexagonale	Octile
Profondeur moyenne	$1.00D$	$0.81D$	$0.71D$
Complexité d' IDA^*	$O(3.00^{D-H})$	$O(2.42^{D-H})$	$O(2.77^{D-H})$

Tableau 4.2 – Complexité d' IDA^* sur les différentes grilles. $D = l(u)$ et H correspond à l'effet de la stratégie heuristique sur l'algorithme.

le temps de calcul n'est pas optimal sur la direction diagonale car le facteur de branchement pour des cellules situées sur la diagonale est de 5. Similairement aux grilles carrées qui ont un facteur de branchement de 3, les déplacements sur la direction diagonale ne sont pas intéressants (contrainte liée à la longueur du chemin qui doit être minimale). Enfin, le maillage hexagonal ne satisfait pas le changement minimum de direction dans le cas horizontal, mais offre un temps de calcul minimal avec un facteur de branche égal à 2. C'est pour cette raison que [Yousefi et Donohue, 2004] utilisent une grille hexagonale pour modéliser un contrôleur de surcharge pour le trafic aérien.

Si nous considérons le cas de la navigation maritime qui est notre principal environnement de recherche, les bateaux préfèrent garder une même direction pendant un long moment (troisième facteur) avec pour objectif de réduire la distance pour atteindre la destination (second facteur). En plus, il devra aussi être capable de modifier sa trajectoire en temps réel (premier facteur) en vue de garantir les évitements de collision avec d'autres objets (bateaux, objets flottants, etc.) ou aux changements de l'environnement. Dans ce cas, la trajectoire construite sera modifiée si une entité du bateau génère une information susceptible de modifier cette dernière. Dans le chapitre 3, nous avons proposé une approche de modélisation pour capturer les informations générées par des entités maritimes à l'aide des informations stockées dans de gros volumes de données appelés corpus. Après étude des différents polygones généralement utilisés dans le maillage des environnements, il apparaît que l'hexagone est le meilleur d'entre eux car il offre un bon compromis entre les différents facteurs présentés précédemment. Déplus, deux cellules hexagonales consécutives se partagent un même côté sur toutes les directions (voir figure 4.7) ce qui implique que le chemin est toujours modélisé par une bande pouvant identifier un couloir de navigation. Ce couloir peut être associé à une bande de sécurité ou à l'intérieur de laquelle le bateau se déplace en toute sécurité. En ce qui concerne les grilles « octiles », les diagonales de deux cellules voisines se joignent en un sommet. Cette situation pourrait en un moment donné proposer une trajectoire ou un chemin se réduisant à un point. Une telle approche de modélisation n'est pas appropriée pour la navigation maritime en ce sens que les routes de navigation sont définies par des couloirs de navigation délimités par des zones de sécurité.

En conclusion, la maille hexagonale permet la génération de grille optimale qui offre le plus d'avantage pour trouver une solution optimale pour le problème de planification de navigation. La section

Grille	Carrée	Hexagonale	Octile
Noeuds adjacents	4	6	8 pour les cellules situées sur la diagonale ou 5 pour les cellules situées non sur la diagonale
Facteur de branchement	3	3	5 pour les cellules situées sur la diagonale et 3 pour les cellules non situées sur la diagonale

Tableau 4.3 – Nombre de nœuds adjacents et valeur du facteur de branchement pour les différentes grilles.

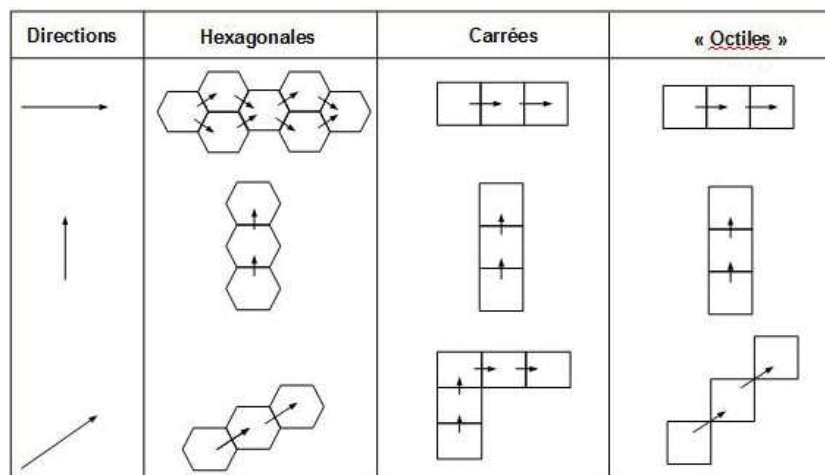


Figure 4.7 – Sélection des mailles optimales sur différentes grilles en fonction de la direction de déplacement.

suivante se concentre sur la structure de données utilisée pour mettre en œuvre l'algorithme *BIDA** sur une maille hexagonale. L'algorithme *BIDA** est ensuite détaillée dans la section 4.5.

4.3.4 Maillage hexagonal

Le but de cette section est de trouver les directions de propagation basées sur un maillage hexagonal utilisées dans la recherche bidirectionnelle tout en conservant la complémentarité du chemin solution. Pour atteindre un objectif de cette envergure, nous sommes amenés à suivre deux étapes importantes. La première étape présentée dans la section 4.3.4 développe le processus de modélisation déployé dans le maillage hexagonal. Cette modélisation n'admet pas de chevauchement des cellules hexagonales lors de la fusion des chemins solutions issus des deux fronts. La seconde étape, détaillée dans la section 4.3.4, sélectionne les directions les plus efficaces à partir d'un ensemble de directions de propagation possibles. La direction choisie sera celle qui minimise les contraintes exigées par l'application. Le développement de cette section sera suivi par la section 4.4 qui propose une structure de données pour sauvegarder les cellules hexagonales du chemin solution.

Construction d'un maillage hexagonal complémentaire

Le principe de ce processus a été inspiré du parallélisme appliqué sur des faces planaires. Les auteurs les plus actifs dans ce domaine sont [Pottmann *et al.*, 2007a] qui ont montré que le parallélisme peut être utilisé pour produire une vue uniforme ou un maillage complémentaire tout en gardant les nœuds optimaux et des propriétés géométriques. Si nous considérons deux maillages M et M' présentés dans la figure 4.8, ces auteurs ont pu prouver que les maillages sont combinatoirement équivalents si et seulement s'ils ont une correspondance 1-1 entre les sommets et les arêtes. Il en résulte alors que ces deux maillages ne sont considérés parallèles que s'ils ont des arêtes correspondantes [Pottmann *et al.*, 2007b].

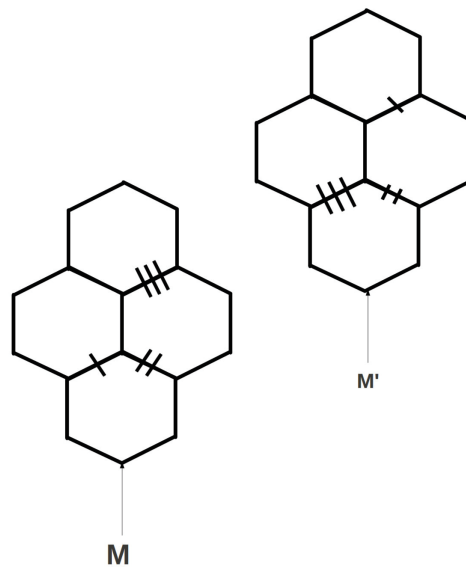


Figure 4.8 – Exemple de maillages parallèles M et M' sur des faces planaires. Ces maillages sont combinatoirement équivalents et les arêtes sont parallèles

En utilisant ces résultats de parallélisme, nous avons proposé un maillage complémentaire partant des points s et t . Ces deux points correspondent respectivement aux points de départ et d'arrivée définis précédemment dans un espace à deux dimensions. Il existe toujours une propagation générée par s qui recouvre le point final t avec un seul hexagone. Les propriétés du parallélisme entre deux maillages montrent qu'il existe un seul hexagone (nœud) d'origine de centre b qui contient le point t et dont le maillage généré par lui-même rencontre le maillage généré à partir du centre s . Le calcul du centre b est réalisé en considérant que le maillage hexagonal est représenté par une matrice à deux dimensions ayant pour indices i et j correspondant respectivement aux indices des colonnes et des lignes en utilisant des axes zigzag (voir figure 4.9a). Trouver la cellule hexagonale qui contient t équivaut à trouver les indices (i, j) de la cellule hexagonale qui recouvre t . Les coordonnées du centre sont obtenues à partir des formules détaillées ci-dessous, où nous considérons $\Gamma(O, \vec{i}, \vec{j})$ le système de coordonnées indiqué dans la figure 4.9.

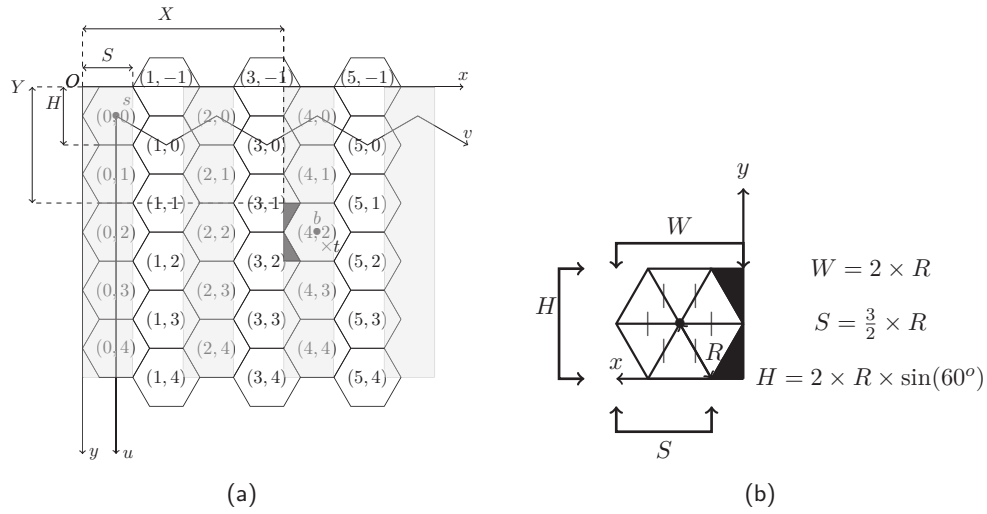


Figure 4.9 – Schéma (a) d'un maillage hexagonal dans le système de coordonnées 4.1 et (b) d'un hexagone dans le système de coordonnées locales 4.2

$$\Gamma(O, \vec{i}, \vec{j}) \quad (4.1)$$

$$\gamma(O', \vec{u}, \vec{v}) \quad (4.2)$$

La méthode pour calculer le centre $b = (b_x, b_y)$ de l'hexagone qui contient le nœud final $t = (t_x, t_y)$ est réalisée en deux étapes. La première subdivise l'espace avec des carrés formant ainsi une matrice à deux dimensions. Chaque carré de la grille construite sera indexé par les indices (i_{tile}, j_{tile}) et qui sont les coordonnées de son point supérieur gauche. Dans la figure 4.9, les largeurs des lignes et des colonnes sont respectivement égales à H et à S . La seconde étape utilise les indices du carré qui contient le nœud final afin de déterminer l'indice (i_{hex}, j_{hex}) de l'hexagone final et les coordonnées de son centre b . Les indices (i_{tile}, j_{tile}) du carré qui contient t sont calculés par l'équation 4.3 :

$$\begin{aligned}
 i_{tile} &= \lfloor \frac{t_x}{S} \rfloor \\
 j_{tile} &= \begin{cases} \lfloor \frac{t_y}{H} \rfloor & \text{lorsque } i_{tile} \text{ est paire} \\ \lfloor \frac{t_y - \frac{H}{2}}{H} \rfloor & \text{lorsque } i_{tile} \text{ est impaire} \end{cases} \\
 &= \lfloor \frac{y_{ts}}{H} \rfloor \text{ avec } y_{ts} = t_y - i_{tile} \bmod 2 \times \frac{H}{2}
 \end{aligned} \quad (4.3)$$

Par ailleurs, le point t peut être défini dans le système de coordonnées local $\gamma(O', \vec{u}, \vec{v})$ associé au carré qui contient t (voir figure 4.9b). Ses coordonnées locales (t_{x_tile}, t_{y_tile}) sont définies par l'équation 4.4 :

$$t_{x_tile} = t_x - i_{tile} \times S \quad \text{et} \quad t_{y_tile} = y_{ts} - j_{tile} \times H \quad (4.4)$$

L'étape suivante concerne l'identification des cellules hexagonales (trois cellules hexagonales qui rencontrent le carré (i_{tile}, j_{tile}) recouvrant le point t . Ce point t appartient soit à la zone gris clair ou à la zone gris foncé dans la figure 4.9a. La frontière de ces deux zones est définie par l'équation :

$$x_{tile} = R \times \left| \frac{1}{2} - \frac{y_{tile}}{H} \right| \quad (4.5)$$

Il en résulte que les indices (i_{hex}, j_{hex}) de l'hexagone qui contient le point $t(t_x, t_y)$ sont :

$$\begin{aligned} i_{hex} &= \begin{cases} i_{tile} & \text{lorsque } t_{x_tile} > R \times \left| \frac{1}{2} - \frac{t_y}{H} \right| \\ i_{tile-1} & \text{autrement} \end{cases} \\ j_{hex} &= \begin{cases} j_{tile} & \text{lorsque } t_{x_tile} > R \times \left| \frac{1}{2} - \frac{t_y}{H} \right| \\ j_{tile} - i_{tile} \bmod 2 + d & \text{autrement} \end{cases} \\ \text{avec } d &= \begin{cases} 1 & \text{lorsque } t_{y_tile} > \frac{H}{2} \\ 0 & \text{autrement} \end{cases} \end{aligned} \quad (4.6)$$

Le paramètre d change dans le cas où t appartient à la partie grise supérieure de l'hexagone ($d = 0$) ou à la partie grise inférieure de l'hexagone ($d = 1$). Finalement, les coordonnées du centre de l'hexagone qui contient le point t sont :

$$b_x = i_{hex} \times S + R \quad \text{et} \quad b_y = j_{hex} \times H + \frac{H}{2} \quad (4.7)$$

En centrant le système de référence Γ sur le point s par une translation de vecteur $(R, \frac{H}{2})$, nous obtenons de nouvelles coordonnées pour le centre b :

$$b_x = i_{hex} \times S \quad \text{et} \quad b_y = j_{hex} \times H \quad (4.8)$$

Généralement deux types de maillage hexagonal peuvent être appliqués. L'un peut avoir une orientation dite « plate » (Figure 4.9b) ou dite « pointue » (Figure 4.10b). Cependant, le choix de l'orientation ne modifie pas la structure de données proposée. La seconde orientation peut être définie à partir de la première par un changement de système de coordonnées. Dans les prochaines sections, nous travaillerons avec une maille hexagonale ayant une orientation pointue. Considérons $\Gamma'(s, \vec{i}', \vec{j}')$ le nouveau système de coordonnées présenté dans la figure 4.10. Γ dérive de Γ' en appliquant une rotation d'angle $\frac{\pi}{2}$ (voir l'équation 4.9). Dans ce nouveau système de coordonnées, le point $b = (b_x, b_y)$ est alors défini par l'équation 4.10.

$$\begin{pmatrix} \vec{i} \\ \vec{j} \end{pmatrix} = \begin{pmatrix} \cos(\frac{\pi}{2}) & -\sin(\frac{\pi}{2}) \\ \sin(\frac{\pi}{2}) & \cos(\frac{\pi}{2}) \end{pmatrix} \times \begin{pmatrix} \vec{i}' \\ \vec{j}' \end{pmatrix} = \begin{pmatrix} -\vec{j}' \\ \vec{i}' \end{pmatrix} \quad (4.9)$$

$$\begin{aligned} \begin{pmatrix} b_x \\ b_y \end{pmatrix}_{\Gamma} &= (b_x \quad b_y) \times \begin{pmatrix} \vec{i} \\ \vec{j} \end{pmatrix} = (b_x \quad b_y) \times \begin{pmatrix} -\vec{j}' \\ \vec{i}' \end{pmatrix} = -b_x \vec{j}' + b_y \vec{i}' \\ &= (b_y \quad -b_x) \times \begin{pmatrix} \vec{i}' \\ \vec{j}' \end{pmatrix} = \begin{pmatrix} b_y \\ -b_x \end{pmatrix}_{\Gamma'} \end{aligned} \quad (4.10)$$

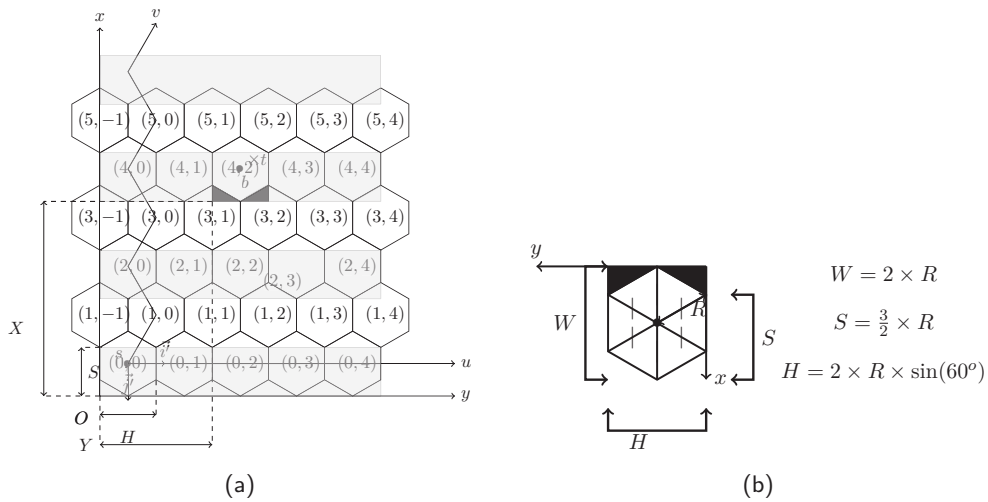


Figure 4.10 – Schéma d'un maillage hexagonal dans le système de coordonnées $\Gamma'(s, i', j')$.

A la fin de cette présentation, nous en déduisons qu'à partir du point de départ s , ce processus de modélisation est capable de calculer le centre b de cellule hexagonale contenant le point d'arrivée t . Elle permet au maillage généré à partir de s de rencontrer celui produit à partir de b de telle sorte que les deux forment un pavage complet de l'espace utilisé à la résolution des problèmes de routage. Dans la prochaine section, nous nous intéresserons aux directions de propagation opérées sur chaque partition « avant et arrière » dans le processus de recherche bidirectionnelle.

Directions de propagation dans la recherche bidirectionnelle

Dans cette section nous introduisons une méthode permettant de déterminer direction de propagation dynamique en temps réel tout en évitant les « retours en arrière ». Une direction de propagation peut être assimilée à un faisceau de propagation. En considérant les maillages hexagonaux, l'objectif est de trouver l'ensemble des hexagones contenus dans un faisceau correspondant à une direction quelconque (c.-à-d., la zone de recherche) sur lequel notre algorithme *BIDA** pourrait s'appliquer. Chaque zone de recherche est définie par trois hexagones adjacents. Le chiffre trois correspond à

la valeur du facteur de branchement décrit dans la section 4.3. La structure de l'hexagone et ses six côtés permettent de mettre en place six faisceaux où chaque faisceau est centré sur l'hexagone associé à chaque côté et lui correspond un faisceau symétrique (voir figure 4.11).

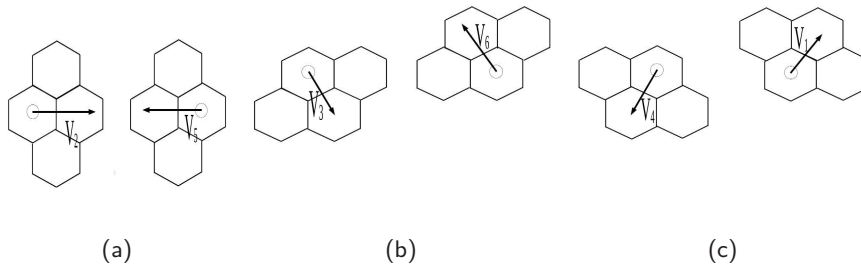


Figure 4.11 – Différents faisceaux de propagation associés à leur faisceau symétrique correspondant.

Pour un hexagone donné de centre c , l'espace est partitionné en section coniques $(Cone_i)_{i=1}^6$ centré sur cet hexagone. Chaque section conique est obtenue par une rotation d'angle $\frac{\pi}{3}$ autour de c . Les trois cellules hexagonales adjacentes à l'hexagone de centre c seront utilisées pour la recherche de la plus pertinente dans l'algorithme de recherche de chemin. Une section conique est identifiée par sa direction de propagation nommée V_i qui est sa bissectrice intérieure.

La sélection du faisceau de propagation est un problème de minimisation de contraintes. Il faut déterminer les faisceaux qui minimisent les critères requis par l'application. La navigation ou les déplacements impliquent de prendre en compte deux types fondamentaux de contraintes : les contraintes spatiales et temporelles. Chacune d'elles peut être une contrainte de cohérence ou d'exigence. D'une part, les contraintes de cohérence sont génériques et ne dépendent pas d'un projet ou d'une application. Elles sont toujours liées aux propriétés intrinsèques des entités qui se trouvent dans l'environnement. D'autre part, les contraintes d'exigence sont un exemple de construction de la réglementation. Ces contraintes d'exigence sont aussi considérées comme une excellente stratégie de construction des règles ou des exigences des clients, qui peuvent varier selon l'application [Borrmann *et al.*, 2009].

Par conséquent, si nous considérons $(c_s, C_T)_i$ la paire de contraintes spatiales et temporelles associée au faisceau i , le défi de l'algorithme est de trouver le faisceau qui satisfait l'ensemble des contraintes ou qui minimise les fonctions de modélisation de ces contraintes. [Schwalb et Vila, 1998] définissent le problème de satisfaction de contraintes temporelles comme une solution informatique permettant de représenter et d'améliorer des requêtes sur les empreintes et relations temporelles (voir chapitre 1).

Cette section explore les contraintes directionnelles et la prochaine section 4.5 dédiée à *BIDA** satisfait les contraintes topologiques et de proximité. La contrainte directionnelle est exprimée pour être aussi proche que possible de la direction $D = \overrightarrow{b_i b_j}$ définie à partir de la droite joignant

les centres b_i et b_j des cellules associées aux fronts avant et arrière (voir figure 4.12). Le processus bidirectionnel conduit à utiliser des faisceaux symétriques sur les deux parties de la propagation. Ainsi, la direction optimale V_i minimise le critère défini par l'équation 4.11 :

$$\widehat{D, V_i} = \text{Min}_{i=1..6} \widehat{D, V_i} \quad (4.11)$$

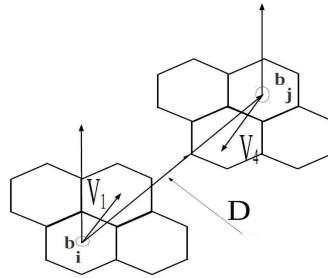


Figure 4.12 – Propagation utilisant les faisceaux symétriques.

4.4 Structure de données

Dans cette section, nous développons une structure de données pour implémenter l'algorithme *BIDA** tout en prenant en compte le problème de minimisation de l'espace mémoire et de temps de calcul. Cela implique que la structure de données devra être capable de stocker les deux faisceaux symétriques de la recherche bidirectionnelle. Cette structure devra ainsi être dynamique, incluant les relations qui lient les cellules parentes et leur séquence de cellules voisines. Le processus devra être optimisé avec comme objectifs que les coordonnées et les sommets des hexagones ne devront être construits qu'une seule fois. Puisque les hexagones adjacents ont des sommets et des côtés en commun, un sommet peut être partagé par au plus 3 hexagones et un côté peut être partagé par au plus 2 hexagones. La stratégie vise à ce qu'un sommet partagé par des hexagones ne soit construit qu'une et une seule fois. Pour les autres hexagones partageant le même sommet, il suffit de faire une mise à jour. Le même principe est appliqué pour les côtés partagés par deux hexagones. La structure de données est modélisée par deux graphes distincts orientés et complémentaires où les nœuds racines sont des hexagones contenant les points s et t dont les coordonnées des centres sont définies dans la section 4.3.4. Les nœuds sont reliés par un arc (parent-fils). Nous définissons ci-après les types de relations entre deux nœuds voisins :

$$\begin{aligned} R_{pc} &: \text{ relation parent-fils} \\ R_{cp} &: \text{ relation fils-parent} \\ R_{bb}^p &: \text{ relation frère-frère ayant pour parent p} \end{aligned} \quad (4.12)$$

La structure de données proposée permet de construire et de stocker les hexagones par un processus

dynamique. La figure 4.13 montre que le maillage hexagonal peut être modélisé avec une vision globale et locale. Concernant la représentation dans un système de référence absolue, un hexagone est identifié avec les indices de ligne et de colonne (i, j) comme expliqué dans la section 4.3.4. Si nous observons le système de référence local centré sur l'hexagone courant, nous avons six hexagones situés dans les directions suivantes « (Nord Est, Est, Sud Est, Sud Ouest, Ouest, Nord Ouest) » identifié par V_i dans la section précédente : « Nord Est » est marquée par 1, « Est » par 2, « Sud Est » par 3, « Sud Ouest » par 4, « Ouest » par 5 et « Nord Ouest » par 6. Dans le système de référence absolue, les relations d'indices (ligne,colonne) entre deux voisins peuvent être exprimées. Si nous supposons la variation $(\delta_i, \delta_j)_d$ suivant la direction d allant d'un nœud (i, j) à un nœud (i', j') , les indices peuvent être calculés en utilisant l'équation 4.13. La table 4.4 présente les coefficients de variations associés aux différentes directions.

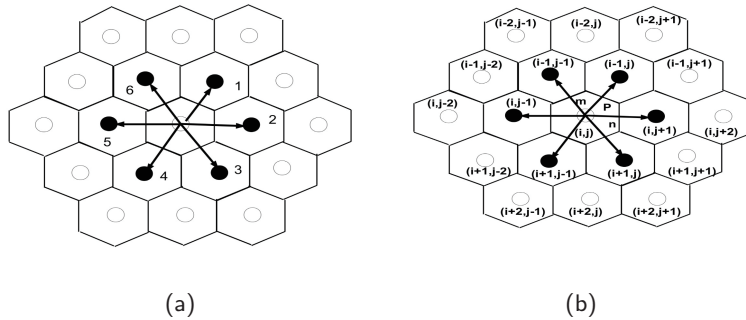


Figure 4.13 – Modélisation des cellules voisines dans un système de référence local (à gauche) ou global (à droite).

$$(i', j') = [-1]^i (\delta_i, \delta_j)_d + (i, j) \quad (4.13)$$

d	1	2	3	4	5	6
$(\delta_i, \delta_j)_d$	$(-1, 0)$	$(-1, 1)$	$(0, 1)$	$(1, 1)$	$(1, 0)$	$(0, -1)$

Tableau 4.4 – Coefficients de variation entre les nœuds parent et fils.

Pour mieux aborder les paragraphes ci-dessous, nous avons proposé la définition d'une notation d'une cellule hexagonale dans notre structure de données. Soit une cellule (i, j) définie dans le graphe, sa notation est la suivante :

$$\left. \begin{array}{l}
d_{pc} : \text{ direction parent-fils de la cellule courante} \\
\text{vue du parent} \\
(i, j) : \text{ les indices de l'hexagone courant (ligne, colonne)} \\
m, n : \text{ les directions des frontières respectivement à gauche et à droite de la direction } d_{pc} \\
S : \text{ l'ensemble des directions identifiant les fils}
\end{array} \right\} \begin{array}{l}
m [Hex(i, j)]_S^{d_{pc}}
\end{array} \quad (4.14)$$

L'explication de notre modélisation est détaillée dans le paragraphe suivant. D'après la figure 4.11b, le nœud du graphe associé à la cellule $(i - 1, j)$ est marqué par ${}^0_2[Hex(i - 1, j)]^1_{6,1,2}$. Ses directions sont $m = 6$, $d_{pc} = 1$ et $n = 2$ définies à partir des indices (i, j) de la cellule parent et correspondant au faisceau ayant pour direction V_1 illustré par la figure 4.11c (configuration droite). L'ensemble des directions $S = \{6, 1, 2\}$ sont définies à partir des indices $(i - 1, j)$ de la cellule courante et identifie les cellules voisines suivantes possibles $(i - 2, j)$, $(i - 2, j + 1)$ et $(i - 1, j + 1)$ permettant d'étendre la structure de données. Considérant la définition d'un maillage hexagonal suivant toutes les directions et à la même résolution, notre modélisation est réalisée par un processus de grossissement itératif. A la première étape de la modélisation, la cellule racine (i, j) est similaire à la figure 4.14. Au commencement la cellule racine (i, j) est assimilée à un point. Il peut être identique au point de départ s ou au centre de l'hexagone contenant le point t . A partir d'une rotation autour de ce centre, les cellules possibles sont mises à jour et stockées dans la structure S (cf. équation 4.14) A la fin du processus la structure de données est réduite à une cellule racine labellisée ${}^0_0[Hex(i, j)]^0_{1,2,3,4,5,6}$. Par convention une direction nulle ($m = 0$, $d_{pc} = 0$ or $n = 0$) implique que la cellule n'a aucun parent.

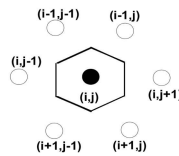


Figure 4.14 – Illustration de la cellule racine dans le processus de modélisation.

La seconde étape consiste en la création des six cellules filles de la cellule initiale ou cellule racine. Cette étape permet de réaliser le premier anneau autour de la cellule racine (Figure 4.15). Dans l'objectif de produire une structure de données optimisée, les relations R_{bb}^p doivent être mises à jour. Le graphe présenté à la figure 4.16 montre la structure de données associée au premier anneau de cellules où la cellule (i, j) est la cellule racine. Chaque arc est orienté du fils vers le parent et étiqueté avec la direction d_{cp} . La direction d_{cp} n'est pas simplement utilisée dans l'intérêt d'étiqueter les arcs du graphe des figures 4.16 et 4.19 mais permet aussi de mettre à jour les nœuds fils contenus dans S d'une cellule mère. On en déduit que si le nœud fils « voit » son parent dans la direction d_{cp} alors son parent le « voit » dans la direction d_{pc} (cf. équation 4.15) :

$$d_{pc} = (d_{cp} + 2) \bmod 6 + 1 \quad (4.15)$$

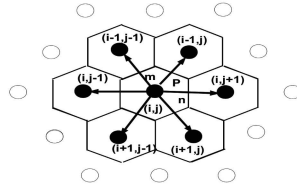


Figure 4.15 – Illustration de la modélisation du premier anneau.

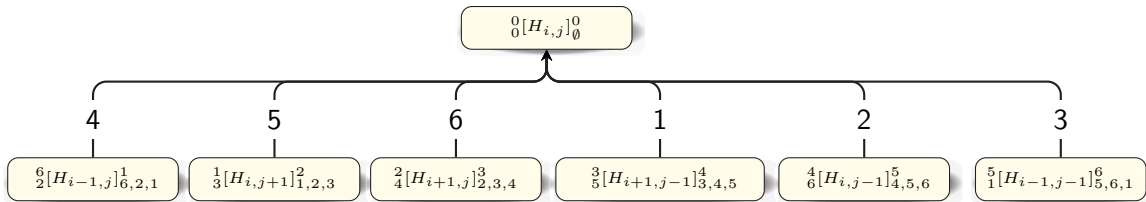


Figure 4.16 – Graphe correspondant à la structure de la figure 4.15.

Afin de produire une structure de données qui grossit progressivement et de façon dynamique, nous allons appliquer le processus précédent sur les cellules suivantes ce qui conduira à la construction du second anneau. Le résultat de ce grossissement est montré par la figure 4.17a et correspondant à la création du graphe présenté dans la figure 4.19. Commenant par la première cellule fille $\frac{6}{2}[H_{i-1,j}]_{6,2,1}^1$, trois cellules filles sont construites dans les directions 6, 1 et 2 (voir la figure 4.17a). Cela conduit à la définition des cellules $\frac{5}{1}[H(i-2,j)]_{5,6,1}^6$, $\frac{6}{2}[H(i-2,j+1)]_{6,1,2}^1$ et $\frac{1}{3}[H(i-1,j+1)]_{1,2,3}^2$ (Figure 4.19). Ces nouvelles cellules seront considérées comme des filles de la cellule courante $(i-1,j)$ ce qui conduit à la mise à jour des relations frère-frère R_{bb}^p entre les trois cellules sœurs et les plus anciennes (c.-à-d. mise à jour des relations entre les cellules sœurs $(i,j+1)$, $(i-1,j+1)$ et les cellules sœurs $(i-1,j-1)$, $(i-2,j)$). Les nouvelles cellules sœurs à mettre à jour sur la frontière sont indiquées par les m et n dans la notation de la cellule.

Ceci implique donc de créer un lien entre les sœurs identifiées par les directions m et $(m+4) \bmod 6 + 1$ (d'après la cellule courante $(i-1,j)$) de même que les sœurs identifiées par les directions n et $n \bmod 6 + 1$. Dans l'exemple précédent, l'ensemble S de la cellule $(i,j+1)$ est modifié en incluant une nouvelle sœur dans sa direction 1. Ainsi, le nombre 1 est retiré de S et la cellule $(i,j+1)$ devient $\frac{1}{3}[H(i,j+1)]_{2,3}^2$ (voir la figure 4.19). Le même principe est appliqué à la cellule $(i-1,j-1)$ où S est modifié en incluant sa nouvelle sœur dans la direction 1. Ainsi, le nombre 1 est retiré de S et la cellule $(i-1,j-1)$ devient $\frac{5}{1}[H(i-1,j-1)]_{5,6}^6$ (voir la figure 4.19). Ce processus est répété sur les cellules filles suivantes (voir les figures 4.17b, 4.17c, 4.18a, 4.18b, 4.18c) jusqu'à la fin du second

anneau. Les itérations suivantes sont réalisées selon le même principe. La structure de données est ainsi optimisée de sorte qu'aucun nœud du graphe n'a subi de duplication et n'est calculé qu'une seule fois au cours du grossissement de la structure de données.

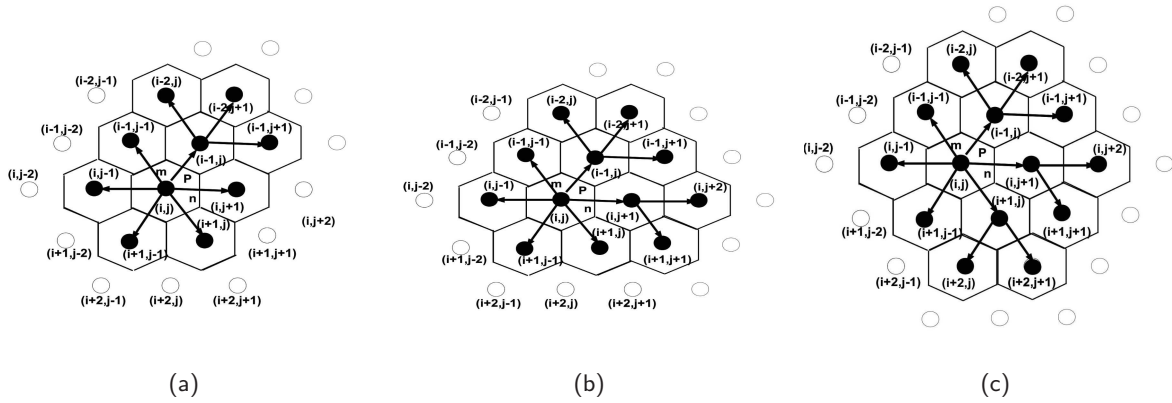


Figure 4.17 – Illustration des trois premières étapes dans le processus de modélisation du second anneau.

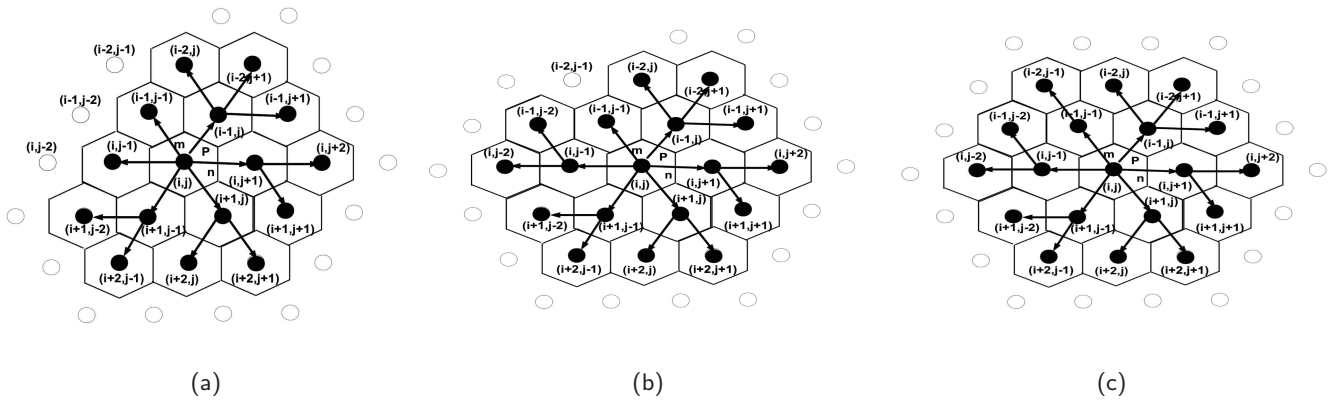


Figure 4.18 – Illustration des trois dernières étapes dans le processus de modélisation du second anneau.

La précédente structure de données a été présentée dans le contexte général d'un maillage complet. Si nous utilisons une direction de propagation, comme définie dans la figure 4.11, nous pouvons réduire le temps de calcul en proposant un sous-graphe associé au graphe complet. Ceci réduit le coût de calcul et fournit une structure de données optimale pour l'application d'un algorithme bidirectionnel itératif de recherche de chemin.

Du point de vue de la visualisation et des calculs, deux cellules voisines sont reliées par deux sommets communs. En supposant les sommets $(v_i)_{i=1}^6$ de l'hexagone (où v_1 est le sommet supérieur de la cellule de l'hexagone) et pour éviter les calculs répétés des sommets, chaque direction V_i est

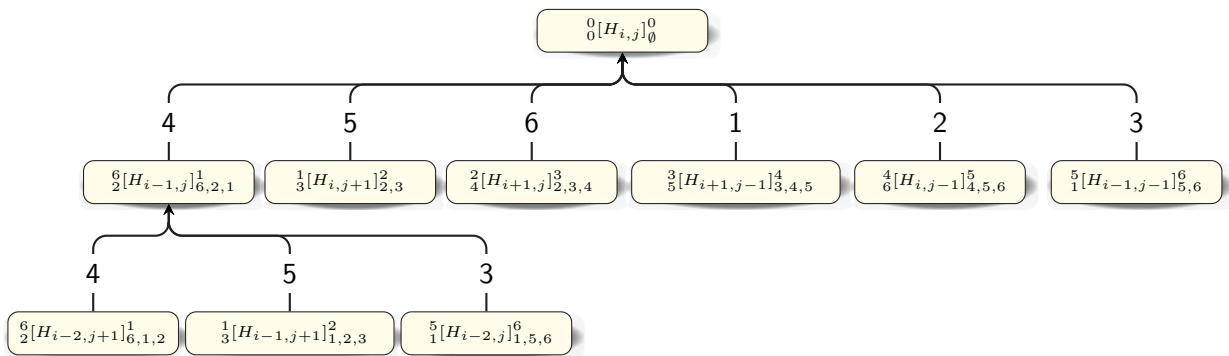


Figure 4.19 – Graphe correspondant à la structure de la figure 4.17a.

liée à un sommet v_i dans la cellule hexagonale. D’après la figure 4.20, les rotations autour du centre de l’hexagone (i, j) dans les directions $(V_i)_{i=1}^6$ conduisent à la création de six sommets v_i . Comme présenté dans la section 4.5, chacun de ces sommets est commun à la cellule parente (i, j) et à ses deux filles. Chaque cellule hexagonale est stockée dans une table de hachage permettant de sauvegarder tous les sommets précédemment décrits tout en évitant les redondances de calculs de sommets dans le maillage. L’optimisation de la structure améliore aussi le processus de visualisation des hexagones. Les mises à jour de la table de hachage sont réalisées en prenant en compte le processus de grossissement de la maille. Par exemple, après la création de la cellule racine (i, j) (figure 4.20), le sommet v_2 appartient à trois cellules hexagonales (i, j) , $(i - 1, j)$ et $(i, j + 1)$. La cellule (i, j) , a deux sommets en commun v_4 et v_5 déjà construits (en observant la table de hachage ligne par ligne) avec la cellule voisine $(i - 1, j)$. Si nous nous positionnons sur une cellule de la table de hachage, il est aisé d’en déduire des cellules qui n’ont pas encore été définies et leur direction de construction.

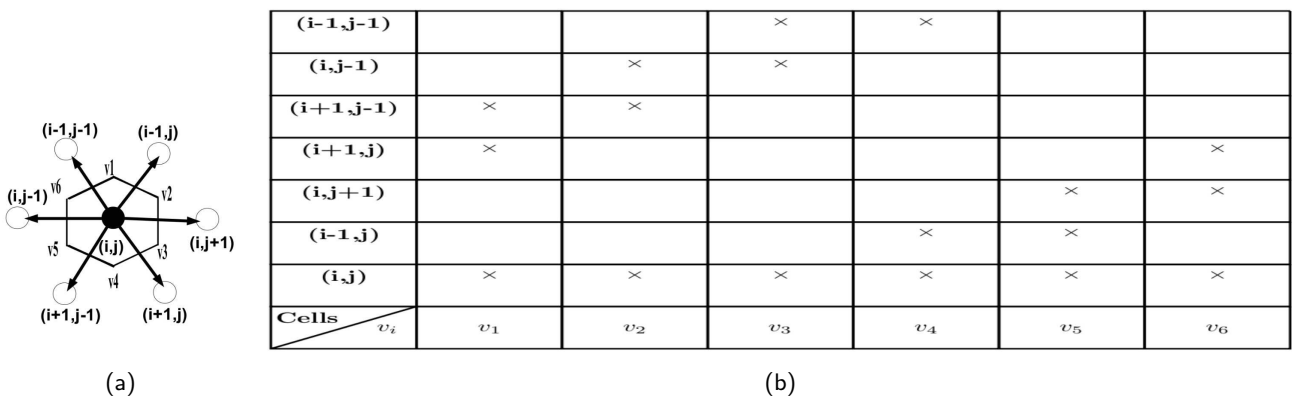


Figure 4.20 – Table de hachage illustrant les sommets déjà calculés par la structure de données au cours du processus de création de la cellule racine (i, j) .

4.5 Algorithme *BIDA** sur un maillage hexagonal

Généralement la complexité des algorithmes de recherche de chemins est toujours déterminée en fonction de trois paramètres : le temps de calcul, l'espace mémoire et le coût du chemin. L'espace mémoire correspond au nombre de blocs de la mémoire vive de la machine utilisée par l'algorithme tandis que le temps de calcul est la durée de temps nécessaire au *CPU* pour l'exécution de l'algorithme. En considérant ces trois paramètres, il s'avère que *IDA** est asymptotiquement c.-à-d. connu comme optimal parmi tous les autres algorithmes dans un arbre de recherche [Korf, 1985]. Sa particularité provient du fait qu'il intègre efficacement une stratégie de recherche itérative. En plus *IDA** est le seul algorithme capable de trouver une solution optimale quand les ressources disponibles sont limitées. L'algorithme *IDA** a une version bidirectionnelle. En se basant sur les interprétations suivantes : « *If the problem is to determine the shortest path from a given origin to a given destination, the number of comparisons can often be reduced in practice by fanning out from both the origin and the destination, as if they were two separate independent problems. . . . However, once the shortest path between a node and the origin or the destination is found in one problem, the path is conceptually replaced by a single arc in the other problem.* » [Dantzig, 1960] proposa le premier algorithme bidirectionnel de recherche de chemin.

Avec la même ambition [Nicholson, 1966] et [Dreyfus, 1969] ont proposé plusieurs approches pour améliorer la précédente description. A partir de cette interprétation [Korf, 1985] met au point une recherche bidirectionnelle qui optimise l'espace et le temps en cherchant simultanément vers l'avant du point de départ (état initial) et vers l'arrière du point d'arrivée (état final) jusqu'à ce qu'un état commun soit trouvé sur les deux frontières de recherche. La table 4.2, nous permet de déduire que le maillage hexagonal offre le plus grand compromis entre la longueur du chemin et la complexité de l'algorithme. En supposant que nous utilisons un schéma de hachage et un maillage hexagonal, il en résulte que la complexité d'un algorithme bidirectionnel en temps et en espace est en $O(2.42^{\frac{D-H}{2}})$.

Le principe de l'algorithme *IDA** est le suivant : pour chaque itération, une recherche en profondeur d'abord (*DFS*)¹⁵ est initialement réalisée pour trouver le prochain nœud dans le graphe. La *DFS* explore les trois nœuds possibles dans un faisceau initié par le facteur de branchement et met à jour leur fonction de coût $f(n_i) = g(n_i) + h(n_i)$. Le coût de la fonction $f(n_i)$ considère que $g(n_i)$ est le coût du chemin parcouru de s à n_i et $h(n_i)$ est une estimation heuristique du coût restant de n_i à t . La valeur du seuil utilisée pour une itération est le coût minimum parmi l'ensemble des coûts possibles à l'itération précédente. Le nœud sélectionné à l'itération suivante est celui dont la fonction de coût est égale au seuil. L'intérêt de cet algorithme est de ne pas minimiser la profondeur du chemin mais plutôt de minimiser le coût total du chemin (c.-à-d., la fonction f). Un exemple de nœud retenu est illustré par les figures 4.21, 4.22 et 4.23. Le fait de restreindre notre algorithme aux faisceaux de propagation ne change pas les propriétés d'admissibilité de l'algorithme *IDA**.

L'algorithme présenté ci-après fournit les concepts principaux utilisés dans l'implémentation de

15. Depth First Search

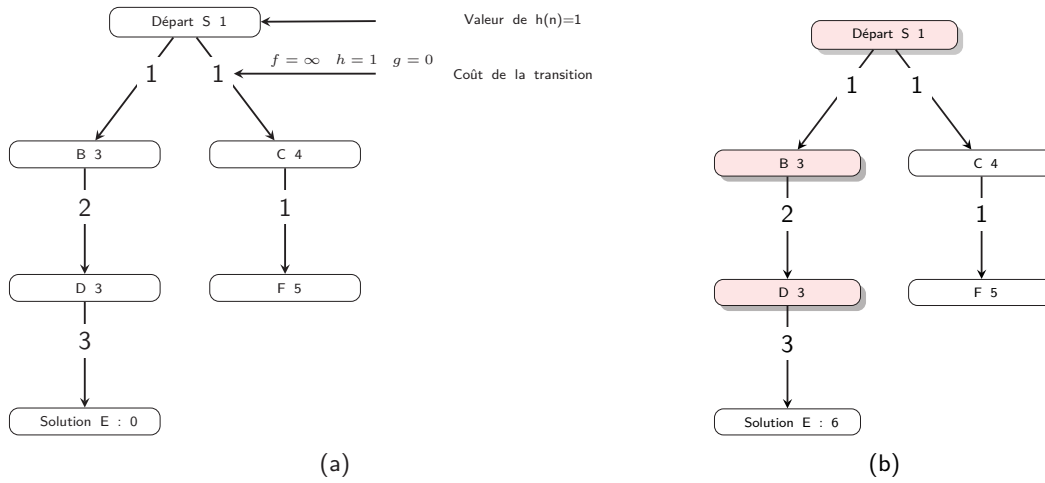


Figure 4.21 – (a) Graphe initial avec les valeurs $h(n_i)$ associées à chaque nœud et (b) chemin solution.

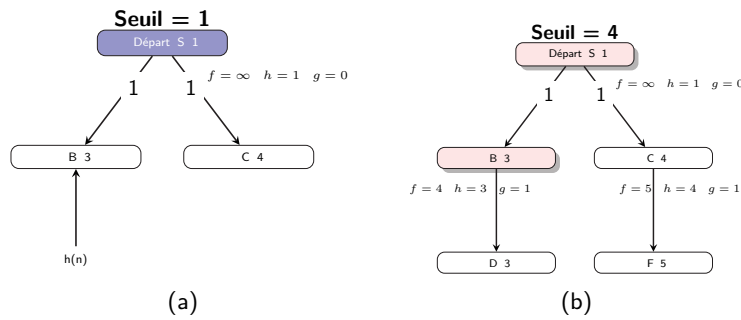


Figure 4.22 – Les deux premières étapes et le nœud sélectionné (rouge) utilisant l'algorithme IDA^* .

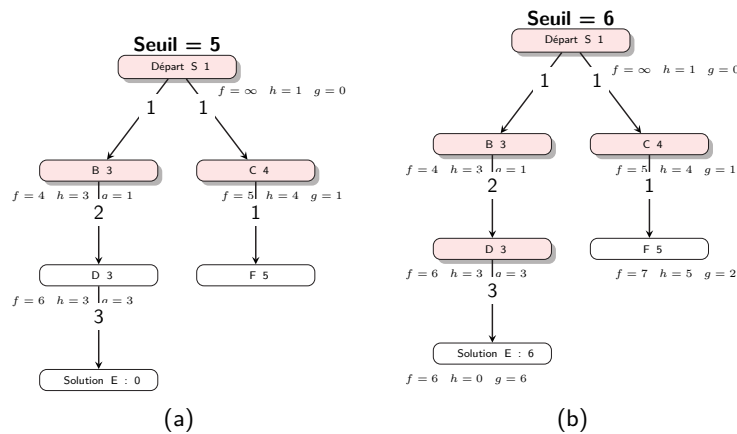


Figure 4.23 – Les deux dernières étapes et le nœud sélectionné (rouge) utilisant l'algorithme IDA^* .

l'algorithme *BIDA**. Le cas d'étude a été appliqué à un environnement réel qui est décrit à la section 4.5. *Map* représente l'ensemble des objets géographiques (empreintes spatiales et attributs) utilisés pour l'environnement maritime. La variable *tip* entraîne la suspension d'un processus en cours d'exécution durant une période donnée afin d'éviter les accès concurrents dans le processus. A chaque processus est liée une structure de données dynamique nommée $Mesh^i$ ($i = 1, 2$) associée à un graphe est composé d'un ensemble de cellules hexagonales définies par l'équation 4.14. Pour la création des cellules hexagonales, à chaque étape un nœud $N^{(i,j)}$ correspond à un sommet qui peut être assimilé à une case de la table de hachage (voir la figure 4.20). Toujours dans l'initiative d'éviter des opérations redondantes, chaque maillage est géré par deux piles de nœuds. La première pile de nœuds est l'ensemble des nœuds prometteurs qui devront être explorés et évalués en utilisant la fonction $f = g + h$. Elle est nommée pile ouverte ($Stack_1$). Cette pile $Stack_1$ a une architecture LIFO¹⁶. La seconde pile regroupe l'ensemble des nœuds visités qui n'ont pas sélectionnés comme ceux conduisant au but. Elle est nommée pile des nœuds fermés ($Stack_2$). Le rayon de l'hexagone est défini par le paramètre r . La taille du rayon joue un rôle sur la résolution du chemin solution. Deux approches peuvent être prises en compte. Dans le premier cas, r peut être considéré invariant pendant tout le processus de recherche de chemin et assure simplement un seul niveau de résolution comme présenté dans la figure 4.26. Une seconde approche par multi-résolution peut être réalisée sur des systèmes globaux de grilles discrètes représentées par un groupe (ouverture) de 3, 4 ou 7 hexagones [Sahr, 2011] (voir la figure 4.24). Dans la modélisation de ces différents regroupement (ouverture), le centre des cellules est préservé ; seul le rayon change de résolution dynamiquement. Trois paramètres permettent de changer la résolution des cellules : i) le centre c ii) le rayon r et iii) l'angle de déviation α calculé entre les axes symétriques de deux résolutions consécutives. Ainsi, il est possible de déterminer les propriétés (c.-à-d. r' , c' et α') des grilles hexagonales à faible résolution à partir des paramètres des grilles de résolutions plus grandes (c.-à-d., r , c and α). Dans le cas du regroupement à 4 (ouverture 4), les paramètres sont calculés comme suit : $r' = 2 * r$, $c' = c$ et $\alpha = 0$. La figure 4.25 est un exemple de l'algorithme *BIDA** où une multi-résolution de regroupement à 4 est appliquée. Le principal défaut rencontré en utilisant une telle stratégie de multi-résolution est qu'il met en péril les conditions de complémentarité des maillages avant et arrière (voir la section 4.3.4). Néanmoins, un des intérêts est de permettre l'optimisation de la recherche (temps de calcul et espace de stockage) en offrant la possibilité de travailler sur de grandes résolutions dans un premier temps avant de passer à des résolutions plus faibles afin d'affiner les résultats (en cas de conflits). Un cas applicatif a été réalisé pour montrer l'utilité de l'algorithme *BIDA** dans la résolution de problèmes du monde réel et notamment dans un environnement maritime. Les résultats sont présentés dans la section suivante.

Entrée Point de départ s et point d'arrivée t : deux positions géographiques.

1: r : le rayon de la cellule hexagonale.

2: *Map* : l'ensemble des couches thématiques représentant l'environnement

Sortie *Path* : l'ensemble des nœuds (c.-à-d. sommets et ses indices) définissant le chemin solution.

16. Last In and First Out

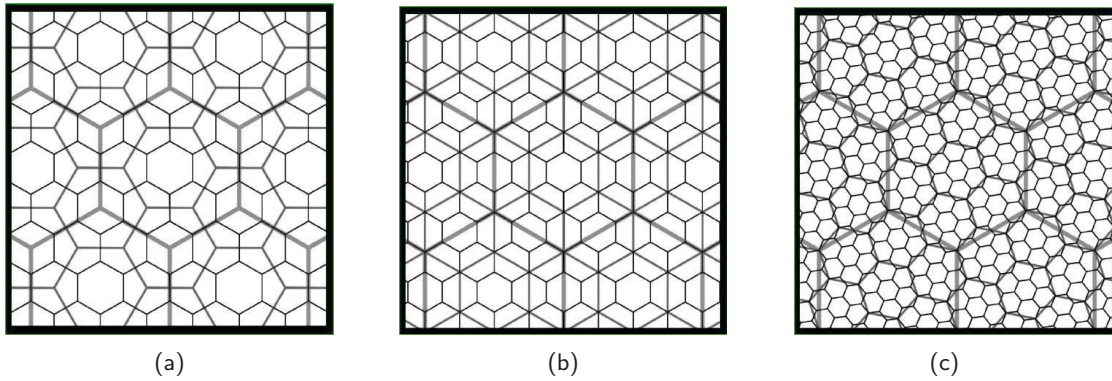


Figure 4.24 – Différents types de multi-résolution (a) ouverture 3, (b) ouverture 4 et (c) ouverture 7 [Sahr, 2011].

```

3: fonction BIDA*(s, t, r, Map)    ▷ La fonction bidirectionnelle principale basée sur deux « threads ».
4:   Mesh1, Mesh2, Stack11, Stack12, Stack21, Stack22, Path1, Path2 ← ∅
5:   Cell, Pnode ← ∅
6:   ObstacleArea ← selectFeaturesFromMap(Map)                                ▷ Cette fonction
   sélectionne l'ensemble des couches ou objets géographiques à éviter pendant la navigation (p.ex., zones
   interdites, zones de danger, bouées, etc.).
7:   tant que joinMeshes(Mesh1, Mesh2) ≠ true faire
8:     si currentThread() = T1 alors
9:       Cell ← updateMesh(T1, s, t, r, Mesh1, Stack11, Stack12, Mesh2)
10:      Pnode ← parentNode(Cell)
11:      Path1 ← Path1 ∪ Pnode
12:      wait(tip)
13:     sinon si currentThread() = T2 alors
14:       Cell ← updateMesh(T2, s, t, r, Mesh2, Stack21, Stack22, Mesh1)
15:       Pnode ← parentNode(Cell)                                ▷ Mise à jour des indices.
16:       Path2 ← Path2 ∪ Pnode
17:       wait(tip)
18:     fin si
19:   fin tant que
20:   Path ← mergingPath(Path1, Path2)
21:   retourner Path                                             ▷ Retourne le chemin optimal.
22: fin fonction

```

Entrée *T*_{*j*} : le « thread » de la procédure

23: *s* et *t* : deux positions géographiques.

24: *r* : le rayon de la cellule hexagonale.

Entrée et

Sortie *Mesh*^{*j*} : un maillage directionnel.

25: *Stack*₁ : la pile des nœuds sélectionnés pour le chemin de solution.

26: *Stack*₂ : la pile des nœuds visités et évalués en utilisant la fonction $f = g + h$. Le nœud le plus prometteur pour le chemin de solution est au dessus de la pile.

27: **fonction** *updateMesh*(*T*_{*j*}, *s, t, r, Mesh*^{*j*}, *Stack*₁, *Stack*₂, *Mesh*^{*i*}) ▷ La procédure retourne la meilleure

cellule dans la maille $Mesh^j$ de l'algorithme de recherche et met à jour les sorties.

```

28:    $Cell \leftarrow \emptyset$ 
29:   si  $Mesh^j = \emptyset$  alors
30:     si  $currentThread() = T_1$  alors
31:        $N^{(0,0)} \leftarrow buildFirstNode(s, r)$   $\triangleright s$  est le centre de la première cellule et  $N^{(0,0)}$  est le premier
       nœud du front avant (Figure 4.15).
32:     sinon
33:        $N^{(0,0)} \leftarrow buildLastNode(s, t, r)$   $\triangleright$  La fonction calcule le centre  $b$  du dernier nœud en
       utilisant de la formule présentée dans la section 4.3.4 et génère le premier nœud du front arrière.
34:     fin si
35:      $pushStack(N^{(0,0)}, Stack_2)$   $\triangleright$  Mettre le premier nœud au sommet de la pile des nœuds visités.
36:     sinon
37:     si  $Stack_2 \neq \emptyset$  alors
38:        $N^{(i,j)} \leftarrow popStack(Stack_2)$   $\triangleright$  Cette fonction sélectionne le meilleur nœud au sommet de la
       pile.
39:     si ( $positionSelected(N^{(i,j)}, Stack_1) = true$ ) alors  $\triangleright$  Cette fonction vérifie si le nœud
        $N^{(i,j)}$  correspond à une position ou à un espace géographique qui a été déjà sélectionné et stocké dans
       la pile  $Stack_1$ 
40:       retourner  $\emptyset$ 
41:     fin si
42:     si ( $meetObstacle(N^{(i,j)}, ObstacleArea) = true$ ) alors
43:       retourner  $\emptyset$ 
44:     sinon
45:        $PartialCell \leftarrow extractRelationsFromMesh(Mesh^j, N^{(i,j)})$   $\triangleright$  Centré sur le nœud  $N^{(i,j)}$ ,
       la fonction extrait l'ensemble des sommets (et leurs relations) déjà construites dans le maillage  $Mesh^j$ .
       Pour le premier nœud  $N^{(0,0)}$ , cette fonction calcule la première cellule hexagonale  ${}^0_0[Hex(0,0)]_{1,2,3,4,5,6}$ .
46:        $Cell \leftarrow buildCell(PartialCell, N^{(i,j)}, Mesh^j)$   $\triangleright$  Calcule la cellule associée au nœud
        $N^{(i,j)}$  à partir de la fonction  $PartialCell$  ainsi que son complémentaire (c.-à-d. les sommets de  $N^{(i,j)}$ 
       non encore définis) (Figure 4.20).
47:        $storeCell(Cell, Mesh^j)$ 
48:        $V_i \leftarrow optimalDirection(Mesh^j, Mesh^i)$ 
49:        $Beam \leftarrow generateThreeBestNodes(Cell, V_i)$ 
50:        $pushStack(sort(Beam), Stack_2)$ 
51:        $pushStack(N^{(i,j)}, Stack_1)$ 
52:       retourner  $Cell$ 
53:     fin si
54:   fin si
55: fin si
56: fin fonction

```

Déploiement de l'algorithme BIDA* en environnement maritime

L'algorithme BIDA* a été appliqué dans un environnement maritime afin de trouver un chemin optimal joignant deux points géographiques donnés (point de départ et point d'arrivée). Ce chemin

sera la trajectoire à suivre par le bateau.

Dans le processus *BIDA** à chaque itération les zones de danger sont évitées en utilisant les relations spatiales topologiques « equal, disjoint, intersect, touch, cross, within, contain, overlap » [Freeman, 1975]) entre les cellules (i, j) du maillage et les différents objets constituant l'environnement. Si une intersection existe, la recherche essaie de trouver d'autres chemins ou solutions parmi les cellules sœurs (voir la section 4.4). Dans le cas où nous ne trouvons aucune solution parmi les cellules « sœurs », la stratégie est de retourner à un niveau supérieur dans l'arbre de recherche. Une fois retourné à la cellule « parent », nous rechercherons une solution parmi ses sœurs. La stratégie peut être appliquée jusqu'au retour à la cellule racine. Dans ce cas, l'application n'a trouvé aucune solution. Dans notre cas d'étude, la fonction de coût g (respectivement h) évalue une distance de « Manhattan » ou de « taxi-distance » entre le point de départ s (respectivement le point d'arrivée¹⁷ t) et le point courant.

Notre approche a été appliquée sur des données maritimes extraites de cartes nautiques numériques (ENC). Ces cartes sont définies sur des données vectorielles utilisant les primitives géométriques telles que point, ligne, polygone exprimée dans le système de référence géodésique ED50¹⁸. Le format utilisé à cet effet est le format S57¹⁹ (voir chapitre 2). Notre exemple utilise la carte numéro 7066 fournie par le (²⁰ « Service Hydrographique et Océanographique de la Marine » SHOM, Brest, France) et représente une zone maritime autour de la ville de Brest en France.

Dans notre application, les utilisateurs doivent d'abord soumettre le point de départ s , le point d'arrivée t et le rayon r de la cellule hexagonale. Le rayon de la cellule hexagonale est arbitrairement choisi et ce choix a un impact sur le temps de calcul. Plus il est grand, plus le temps de calcul est réduit. Concrètement, il peut être utilisé pour indiquer la zone de sécurité à respecter autour du bateau associée à une distance minimale et définie en fonction de la vitesse et le taux de giration²¹ du bateau. Les figures 4.25 et 4.26 illustrent respectivement une utilisation de la multi-résolution et d'une résolution unique sur un maillage hexagonal. Dans notre cas d'étude les cellules pertinentes du chemin bidirectionnel sont affichées en respectant les fonctions heuristiques tout en évitant les zones de danger comme la zone de terre (zones gris clair), les bouées (cercles en gris clair) et les zones interdites (zones noires et sombres). La zone de mer est représentée par une couleur blanche. Le prototype a été implémenté en langage de programmation « Java » en utilisant les fonctionnalités de la librairie java topologie suite (JTS²²) pour appliquer les relations spatiales. La ligne noire illustre la route proposée et les cellules hexagonales correspondent au graphe de nœuds déployés pour la solution. Chaque nœud du graphe est construit avec exactement ses trois fils. Les trois nœuds en noir de la figure 4.25 sont les fils du nœud courant sur les fronts « arrière » et « avant ».

17. Dans le concept de recherche bidirectionnelle ce point est celui le plus avancé du faisceau symétrique déterminé dans la section 4.3.4

18. European Datum 1950

19. http://www.iho.int/iho_pubs/IHO_Download.htm

20. <http://www.shom.fr>

21. ROT est un attribut dans le système d'identification automatique du format AIS.

22. <http://www.vividsolutions.com/jts/jtshome.htm>

La table 4.5 propose certains résultats pour évaluer la performance de la stratégie développée. Les critères considérés ci-après sont le rapport γ indiquant le gain de nœuds par rapport aux problèmes de recherche de chemin ordinaires et le temps de calcul. Le rapport γ évalue le nombre de cellules sollicitées par l'algorithme proposé en comparaison avec le nombre de cellules nécessaires pour effectuer un pavage complet de la région considérée ($Nb_{hexagon} = \frac{\text{l'aire de la region}}{\text{l'aire d'un hexagone}}$). Par exemple, le rapport γ_1 dans le premier cas de la table 4.5 est égal à $\frac{142177-1014}{142177} = 0.99$. Ceci signifie que seulement 1% des cellules de la grille pavant la région a été nécessaire pour déterminer la solution optimale. Le temps de calcul indiqué dans la dernière colonne de la table 4.5 a été obtenu à partir d'un ordinateur comportant un processeur « Intel Celeron E3200 (2.40 GHz) ». Nous remarquons donc que notre stratégie minimise le temps processeur par un stockage optimal des nœuds. Ceci est d'autant plus vrai lorsque l'espace de recherche est vaste.

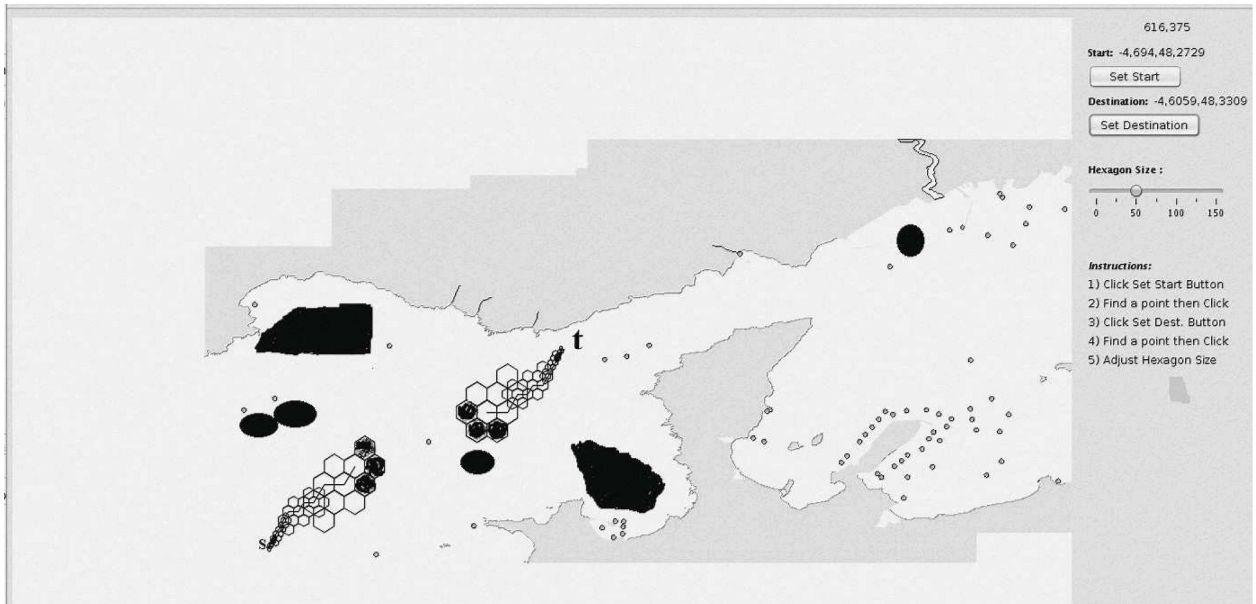


Figure 4.25 – Exemple d'un routage maritime itératif utilisant l'algorithme *BIDA** sur une structure de données dynamique. Cas 1 : multi-résolution sur un maillage hexagonal en regroupement (ouverture 4).

4.6 Conclusion

Ce chapitre introduit une structure de données spatiale utilisé dans la recherche de chemin en temps réel dans des environnements ouverts. Elle est basée sur une structure de données dynamique. Cette structure de données permet d'appliquer un algorithme *BIDA** dans l'intérêt d'extraire une route optimale en termes de distance et de complexité. La structure de données proposée présente plusieurs propriétés qui (i) détermine un maillage complémentaire d'hexagones produisant un maillage pour le chemin solution et (ii) optimise la direction de propagation en suivant l'une des trois directions proposée par facteur de branchement. Cette dernière a été implémentée à partir d'un graphe

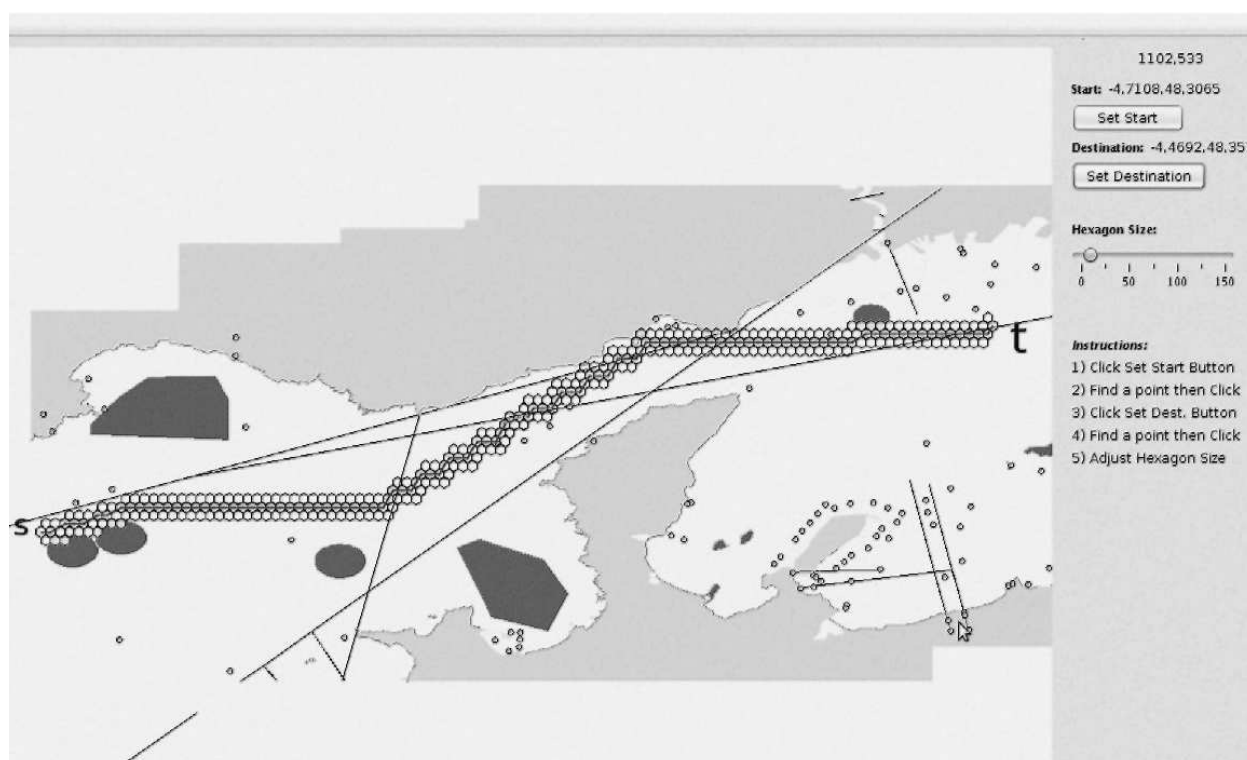


Figure 4.26 – Exemple d'un routage maritime itératif utilisant l'algorithme *BIDA** sur une structure de données dynamique. Case 2 : résolution simple sur un maillage hexagonal.

Point de départ s (km)	point d'arrivée t (km)	rayon de l'hexagone (m)	nombre de cellules dans le graphe	nombre de cellules dans le maillage de l'environnement	nombre de cellules dans le chemin solution	γ	temps de calcul (ms)
-524282.64 5379318.29	-497390.83 5382653.57	50	1014	142177	338	0.99	21350
-524282.64 5379318.29	-510401.87 5377558.01	50	576	142177	172	0.99	8207
-524313.63 5379102.12	-497019.12 5382653.57	96	567	38568	185	0.98	5309
-524313.63 5379102.12	-497019.12 5382653.57	122	432	23881	144	0.98	5332
-524313.63 5379102.12	-497297.90 5385000.62	150	558	15798	159	0.96	11477
-505228.18 5375396.25	-497297.90 5385000.62	103	240	15798	80	0.98	2969

Tableau 4.5 – Évaluation de la stratégie proposée par comparaison avec une stratégie de recherche fondée sur un maillage complet de l'environnement.

hiérarchique orienté en évitant la duplication des nœuds du graphe et les redondances de calcul des informations sur les nœuds. Par ailleurs, cette structure de données réduit considérablement le temps de calcul et le coût de l'espace mémoire disponible pour la recherche de chemins en temps

réel. Dans le cadre de la navigation maritime, l'utilisation d'une stratégie de recherche de route bidirectionnelle permet de réaliser une prévision sur la route à suivre en temps réel. En considérant que les applications en temps réel peuvent faciliter l'aide à décision du capitaine pendant les situations de stress ou de surcharges d'informations. ce travail a été appliqué dans les données statiques et à l'aide des résultats des travaux du (chapitre 3). La suite de ces travaux a été appliquée sur des données dynamiques représentées par des modèles spatio-temporelles (voir chapitre 5). Les travaux futurs devront prendre en compte deux contraintes principales. La première est l'évitement des zones de danger en prenant en compte la météorologie et les informations sur l'environnement telles que le vent et les courants. La seconde est la sémantique des informations générées par une région ou un objet (p.ex., les bouées, les alignements) en temps réel qui modifient la trajectoire du bateau.

Chapitre 5

Architecture du modèle proposé : application à la navigation sous contraintes

Sommaire

5.1	Les différents modules de l'architecture	121
5.1.1	Module de traitement automatique du langage	121
5.1.2	Module d'édition de l'ontologie	121
5.1.3	Module de stockage et d'utilisation des données	124
5.2	Prise en compte des contraintes de l'environnement	129
5.2.1	Contraintes globales	130
5.2.2	Contraintes locales	131
5.3	BIDA* pour des problèmes multi-constraints	133
5.3.1	Formalisation des différents critères	135
5.4	Conclusion	138

LA navigation maritime doit prendre en compte un ensemble d'informations ou de données utiles pour leur modélisation afin d'être exploitées par des applications intelligentes comme souligné au chapitre introductif. Ces connaissances spatiales, temporelles et sémantiques ont besoin d'une formalisation et d'une structuration présentées aux chapitres 2 et 3 avant d'être stockées dans des bases de données (voir le paragraphe 5.1.3.1). Une fois ces données ou informations stockées dans des bases de données, il est alors possible de raisonner sur celles-ci afin de produire des connaissances nécessaires au développement des services dédiés à la navigation maritime.

Considérant les connaissances spatiales, temporelles et sémantiques, nous proposons une architecture dédiée aux applications de navigation. Cette architecture repose sur trois modules indépendants permettant l'édition de l'ontologie computationnelle, l'extraction de connaissances à partir

d'un traitement automatique du langage et enfin le stockage et l'exploitation des données. L'architecture a été déployée, sans pour autant être restrictif, dans le cadre d'un service de routage maritime sous-contraintes. L'algorithme initialement proposé dans le chapitre 4 a été enrichi afin d'intégrer l'ensemble des connaissances acquises sur les entités.

La prise en compte de contraintes dans l'algorithme de routage nécessite de les regrouper en deux grandes catégories : les contraintes globales et les contraintes locales. Les contraintes de distance de la trajectoire et de profondeur sont les deux contraintes globales. La contrainte de distance permet de contraindre la longueur de la trajectoire. Cette dernière est intrinsèquement intégrée dans l'heuristique que nous avons proposé dans le chapitre 4. En complément, la contrainte globale de profondeur nous assure de trouver un chemin solution dans une zone ou région à l'intérieur de laquelle la navigation est possible par un engin mobile. La table 5.1 illustre les différents environnements dédiés à quelques engins de navigation (avions, bateaux, voitures, etc.). Dans le contexte de la navigation maritime, la zone navigable sera définie comme une zone de profondeur suffisante.

Environnements dédiés	Air	Mer	Réseau routier
Types de navigation	Navigation aérienne	Navigation maritime	Navigation terrestre
Engins	Avion	Bateau	Voiture

Tableau 5.1 – Différents types d'engins mobiles associés à leur environnement de navigation dédié.

Les contraintes locales sont liées aux contraintes proches de la position du navire. Nous nous sommes restreints dans notre étude à la contrainte locale de direction issue de la sémantiques des entités. Parmi les entités prises en compte, on retrouve :

- 1** les entités (généralement placées par l'homme) présentes afin d'éviter les zones où la navigation est impossible. Elles permettent de garantir la sécurité des usagers et des biens au cours de la navigation. A titre d'exemple, nous y trouvons l'ensemble des entités du système de balisage.
- 2** les entités présentes sur les cartes marines, représentées par l'homme et utilisées pour faciliter et améliorer la navigation. Nous avons par exemple, les alignements ou lignes de navigation. Contrairement, au premier groupe d'entités, les éléments de ce groupe ne sont pas physiquement présents dans l'environnement maritime.

Dans ce chapitre, la section 5.1 présente l'architecture globale du système dédié aux applications de navigation maritime. La section 5.2 présente plus en détails les contraintes globales et locales considérées dans cette étude. Ensuite, la section 5.3 définit un algorithme de routage multicritère afin de prendre en compte les précédentes contraintes. La stratégie de routage multicritère est appliquée sur des données maritimes extraites de cartes nautiques numériques de la rade de Brest. Les caractéristiques de cette carte ont été présentées dans la section 4.5 et la représentation des

connaissances sémantiques associées aux entités de l'environnement sont décrites dans la section 5.1. Enfin, la section 5.4 conclut ce chapitre.

5.1 Les différents modules de l'architecture

La mise en œuvre d'une telle architecture doit pouvoir gérer les informations ou données en utilisant les approches issues du traitement du langage, de l'ontologie computationnelle, des techniques de stockage des connaissances et des données (spatiales, temporelles et sémantiques). Les différents composants de l'architecture de notre modèle sont considérés comme trois modules indépendants (voir figure 5.1). Ces différents modules sont : 1) le module du traitement de langage et d'acquisition des données, 2) le module d'édition et de représentation des connaissances via des ontologies et 3) le module de stockage et de transfert des connaissances.

5.1.1 Module de traitement automatique du langage

Ce module (module 1 dans la figure 5.1) est une implémentation des méthodes proposées au chapitre 3 et permet l'enrichissement sémantique des entités spatio-temporelles. Il s'intéresse à l'extraction des concepts liés à la navigation maritime et à la désambiguïsation des phrases qui décrivent ces concepts. Les concepts et phrases extraits des corpus sont stockés dans une base de données pour une utilisation ultérieure. L'extraction et la désambiguïsation sont réalisées par le logiciel *Yatea* qui produit un flux de contenu en xml de concepts et de phrases. Le flux de données xml obtenu est ensuite transmis à une base de données (module 3 dans la figure 5.1) afin de permettre le calcul de l'affordance associée à chaque concept (cf. chapitre 3).

5.1.2 Module d'édition de l'ontologie

Ce module s'intéresse à l'édition de l'ontologie exploitant des connaissances fournies par l'expert et des connaissances déduites des documents relatifs à la navigation et à la sécurité maritime (module 1). Ces connaissances sont structurées par une ontologie computationnelle implémentée à partir du logiciel protégé¹. Ce module comporte deux composants : le modèle de données et le modèle de règles. Le modèle de données utilise des triplets « triple store » (sujet-predicat-objet) détaillés au chapitre 2 et représentés sous le format OWL, tandis que les règles sont définies en logique du premier ordre et implémentées dans le langage SWRL (voir chapitre 2). Les sources d'information utilisées pour le modèle de données proviennent de la rade de Brest et sont structurées au format S57 (voir chapitre 2).

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

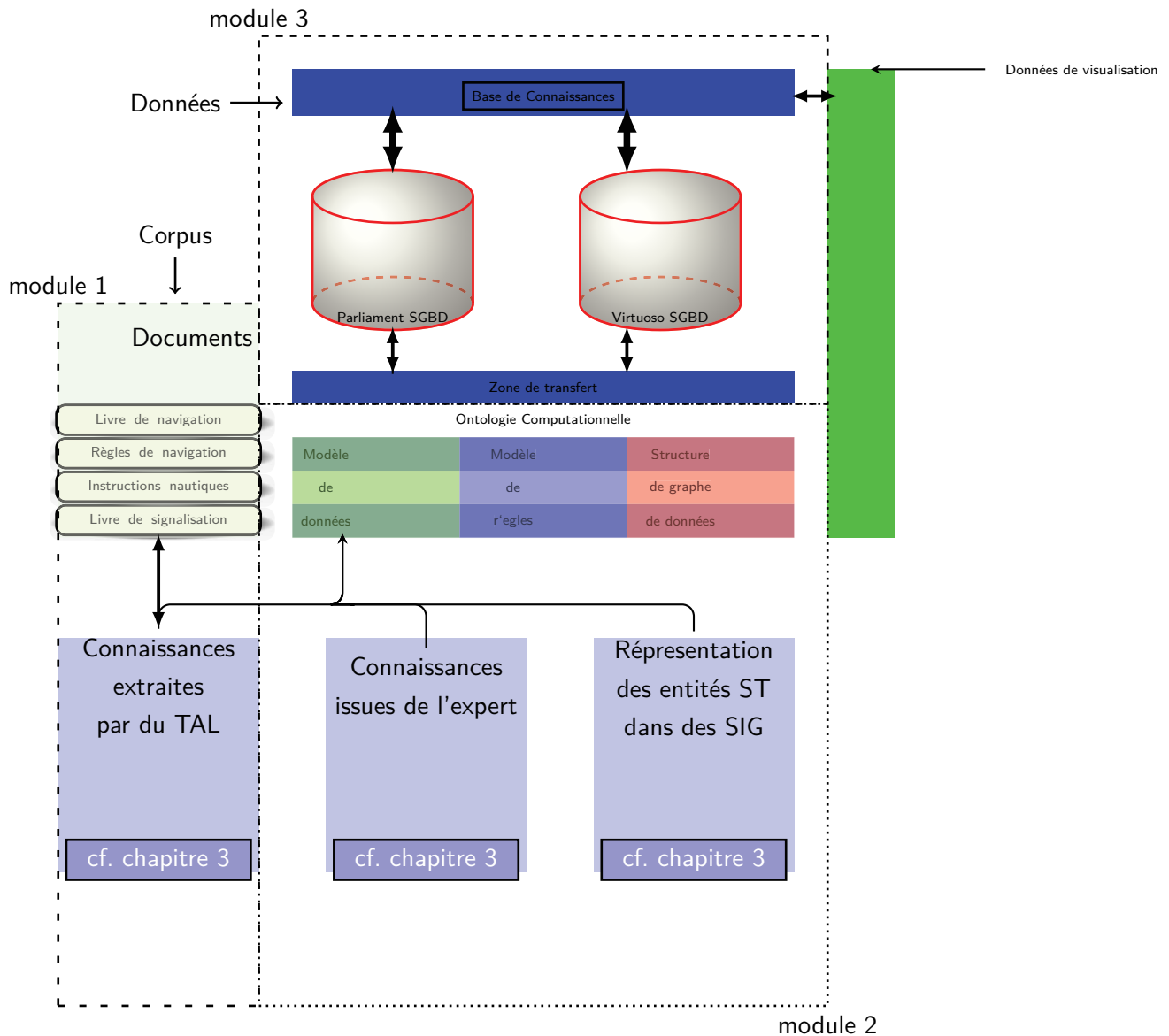


Figure 5.1 – Architecture globale du modèle

La proposition d'une ontologie computationnelle qui prend en compte le modèle de données et le modèle des règles du domaine maritime exige de réaliser une catégorisation des données maritimes. Les différentes catégories que nous avons identifiées sont les suivantes :

- une catégorie concernant les activités maritimes,
- une catégorie concernant les connaissances issues de la carte marine au format S57,
- une catégorie concernant les connaissances issues des capteurs,
- une catégorie concernant les connaissances issues des données textuelles.

Les deux principales catégories que nous avons structurées via une ontologie dans le cadre de cette thèse sont celles associées aux activités maritimes et à la carte marine.

5.1.2.1 Activités maritimes

Les activités maritimes prises en compte sont :

- l'aquaculture qui est l'élevage d'organismes aquatiques en mer ou en eau douce.
- les cultures marines qui s'intéressent à l'élevage de coquillages (conchyliculture) tels que les huîtres creuses, les huîtres plates, les moules, etc., de poissons (pisciculture) et crustacés (crevettes, etc.). La culture d'algues (algoculture) est aussi considérée.
- les activités de loisirs, qui se composent des activités liées au tourisme, aux sports nautiques, à la baignade et à la pratique de la voile ou bien du surf.
- les marais salants ou salines se qui localisent dans un ensemble de bassins de faible profondeur, appelés carreaux, dans lesquels sont récoltés le sel, obtenu par évaporation de l'eau de mer, sous l'action combinée du soleil et du vent.
- la navigation maritime qui est la principale activité que nous avons explorée dans le cadre de cette thèse. Elle concerne toutes les activités humaines de circulation sur les mers et océans. Elle désigne aussi l'ensemble des techniques et méthodes qui permettent au navire de déterminer sa position, de calculer la route à suivre pour se rendre à sa destination, en toute sécurité. Elle est rendue possible par la prise en compte d'un ensemble d'objets. Ces objets se décomposent en deux sous catégories, à savoir : les objets naturels et artificiels.

Les objets naturels se composent d'objets naturels statiques et d'objets naturels dynamiques. Nous avons par exemple comme objets naturels statiques : la végétation, les récifs, les rochers, etc. En ce qui concerne les objets naturels dynamiques, nous avons par exemple les spongiaires², les mollusques³, les poissons⁴ et les humains qui peuvent être les marins, les pêcheurs, les touristes, les militaires, etc.

Les objets artificiels sont aussi identifiés par deux catégories à savoir : les objets artificiels dynamiques et les objets artificiels statiques. Parmi les objets artificiels dynamiques, nous avons par exemple les bateaux qui peuvent être motorisés (bateaux de pêche, navires, cargos, bateaux de plaisance, navires marchands, bateaux militaires (porte-avions, frégates, etc.)) ou non motorisés (voiliers, pirogues, etc.).

Les objets artificiels statiques représentent généralement des récifs artificiels. Un récif artificiel est une structure immergée, construite ou placée délibérément sur le fond marin dans le but d'imiter certaines fonctions d'un récif naturel destiné à protéger, régénérer, concentrer et ou valoriser les peuplements de ressources marines vivantes. Son objectif est aussi de protéger, restaurer et régénérer des habitats aquatiques, ainsi que de promouvoir la recherche, les possibilités récréatives et l'utilisation de la zone à des fins éducatives. Nous avons par exemple les brise-lames ou les

2. qui forment les éponges. Ce sont des animaux très primitifs qui constituent des structures considérées comme les plus simples.

3. ce sont des animaux qui possèdent un corps mou avec ou sans coquille externe ou interne. Leur corps est composé d'une tête, d'une masse viscérale, et d'un pied.

4. ils forment un groupe de vertébrés très hétérogène, dont les caractéristiques communes sont la présence d'appareil branchial, d'une peau écailleuse et de nageoires.

corps-morts. En plus des récifs, nous avons les objets permettant de faciliter la navigation maritime comme les bouées, les amers (voir chapitre 2) ou les objets permettant la communication ou le transport (câbles, pipelines) ainsi que les objets relatifs à l'extraction des ressources naturelles (plates-formes). Certaines classes d'objets provenant de carte marine numérique au format S57 permettent d'enrichir les catégories présentées précédemment.

5.1.2.2 Carte marine

La carte marine propose une représentation de l'environnement maritime. Elle peut être décrite suivant le modèle proposé par la NOAA⁵ « National Oceanic and Atmospheric Administration » ou les instances des entités sont des objets contenus dans la représentation S57 (voir section 2.2). La représentation de la carte maritime proposée par NOAA se compose de sept catégories qui sont :

- 1 les informations générales sur la carte,
- 2 les zones interdites et les zones de danger,
- 3 les informations hydrographiques,
- 4 les informations sur la neige,
- 5 les zones d'essai militaire,
- 6 les marées et courants,
- 7 les informations topologiques et les installations offshore.

Dans le format S57, les informations hydrographiques sont relatives aux données concernant le bord des côtes, les profondeurs données par les sondes⁶, les zones réglementées, la nature des fonds, les courants et les marées, les roches et les épaves ou encore les routes. Les informations topographiques sont réparties en 3 sous-catégories : les caractéristiques culturelles, les caractéristiques naturelles et les ports et services.

Les connaissances de l'environnement maritime ainsi structurées sont transférées dans des bases de données dédiées permettant leur stockage afin d'être utilisées de façon efficace et en temps réel. La mise en œuvre de ce stockage a été réalisée par le module 3 décrit dans la section suivante.

5.1.3 Module de stockage et d'utilisation des données

Le stockage et l'utilisation des données sont rendues possible par des techniques développées dans le web sémantique. Les données sont exploitées au sein de serveurs de données (voir chapitre 2). Les données utilisées sont relatives aux données spatio-temporelles liées à l'environnement maritime (cartographie marine et activités maritimes). Elles sont généralement exprimées aux formats svg,

5. <http://encdirect.noaa.gov>

6. indique la profondeur minimale en un lieu donné. Des isobathes (ou courbes isobathymétriques) permettent de représenter sur les cartes la topographie du fond marin.

gml, xml et csv. Nous avons par exemple la représentation de l'ensemble des bateaux et du système de balisage qui est réalisée sous le format svg. Nous avons aussi l'ensemble des connaissances cartographiques en un lieu donné (format S57) qui est représenté sous le format csv ainsi que les différents documents nautiques définis dans un corpus représentés sous un format xml.

Ces données sont stockées dans le serveur de la base de données Virtuoso qui sera détaillé ci-après. Elles sont extraites du serveur de base de données par des requêtes utilisant le protocole « geosparql » qui est une extension du sparql. L'équation 5.1 est un exemple de requête « geosparql » permettant d'extraire les entités spatiales à l'intérieur du polygone Polygon((7 50, 7 53, 9 53, 9 50, 7 50)). Afin d'avoir des temps de réponse raisonnables sur le stockage et l'extraction des ontologies, il est intéressant de choisir les serveurs de bases de données les plus performants et qui intègrent le langage sparql.

```

PREFIX geo : < http : // www . opengis . net / ont / geosparql >
PREFIX geof : < http : // www . opengis . net / def / function / geosparql >
PREFIX sf : < http : // www . opengis . net / ont / sf >
SELECT DISTINCT
*
WHERE
?bGeomgeo : asWKT ?bWKT .
FILTER ( geof : sf Within ( ?bWKT , " Polygon ( ( 7 50 , 7 53 , 9 53 , 9 50 , 7 50 ) ) " 3f : wktLiteral ) )

```

(5.1)

5.1.3.1 Serveurs de base de données

Les bases de données sont formées des entités composant l'environnement maritime. En relation avec l'ontologie définie, nous nous sommes restreints aux entités associées à la carte marine et aux activités maritimes. Une entité est définie à partir de ses composantes spatiale, temporelle et sémantique comme précisé au chapitre 2.

Les bases de données performantes permettent d'accroître l'efficacité des applications destinées à la surveillance des environnements maritime, aérien et urbain [Beckmann *et al.*, 1990]. Ainsi, ces bases de données possèdent des techniques d'indexation qui optimisent et rendent transparente l'extraction des contenus spatiaux et non spatiaux. Plusieurs techniques d'indexation ont été proposées [Finkel et Bendley, 1976], [Guttman, 1984b] [Sellis et Faloutsos, 1987]) et les plus utilisées sont les Quadtree [Finkel et Bendley, 1976] et les R-tree [Beckmann *et al.*, 1990]. La structure de données R-tree utilise un B-arbre pour la représentation des contenus et la structure de données Quadtree [Finkel et Bendley, 1976] utilise un graphe hiérarchique issue d'une division récursive de la région considérée.

Les serveurs de base de données ontologique les plus utilisés, développant des techniques d'indexa-

tion basées sur des R-tree, sont :

- Parliament⁷,
- AllegroGraph⁸,
- OWLIM⁹,
- Strabon¹⁰,
- Virtuoso¹¹.

Parmi tous ces serveurs, Parliament est présenté comme le plus performant et est régulièrement mis à jour. Il est présenté dans le paragraphe ci-après.

Structure de données R-tree

Cette structure de données proposée par [Guttman, 1984b] est une variante des techniques d'indexation conçues pour organiser les collections d'objets spatiaux. Elle représente la donnée spatiale par le plus petit rectangle à deux dimensions qui la recouvre. Sa technique de construction est fondée sur un recouvrement progressif des données spatiales qui produit à la fin un B-arbre dans lequel chaque nœud de l'arbre correspond au plus petit rectangle contenant d -rectangles (d est le nombre de fils du nœud). Chaque nœud (excepté les nœuds feuilles) stocke un couple d'informations (R,P) où R est le plus petit rectangle et P un pointeur sur les nœuds fils. Les nœuds feuilles ou nœuds terminaux, contiennent des pointeurs qui leur permettent de référencer les données réelles dans la base de données. Chaque nœud feuille se compose également de deux informations (R,O), où R est le plus petit rectangle qui contient l'objet spatial O. Dans la structure de données R-tree, les frontières des rectangles formés par des nœuds différents peuvent se chevaucher. Cependant, un objet est associé à un seul nœud. La figure 5.2a est une illustration de la représentation d'un ensemble d'objets spatiaux construits par un R-tree avec son B-arbre associé (voir figure 5.2b). Les données spatiales réelles sont représentées par des rectangles en traits continus et les nœuds terminaux par des rectangles en trait mixte.

La structure de données R-tree respecte un ensemble de propriétés qui permettent d'éviter un déséquilibre de l'arbre lors des opérations de suppression, d'insertion et de mis à jour de l'arbre. Elle garantit une recherche d'informations dans l'arbre en ordre logarithmique. Les principales propriétés sont les suivantes :

- 1** Soient M et m respectivement le nombre maximum et minimum de fils que peut avoir un nœud (excepté les nœuds feuilles), on a toujours $m \geq M/2$.
- 2** Tout nœud (excepté les nœuds feuilles) a au moins deux nœuds fils.
- 3** Toutes les feuilles sont à la même profondeur dans l'arbre.

7. <http://parliament.semwebcentral.org/>
 8. <http://franz.com/agraph/allegrograph/>
 9. <http://owlim.ontotext.com/display/OWLIMv54/Home>
 10. <http://www.earthobservatory.eu/Strabon>
 11. <http://virtuoso.openlinksw.com/>

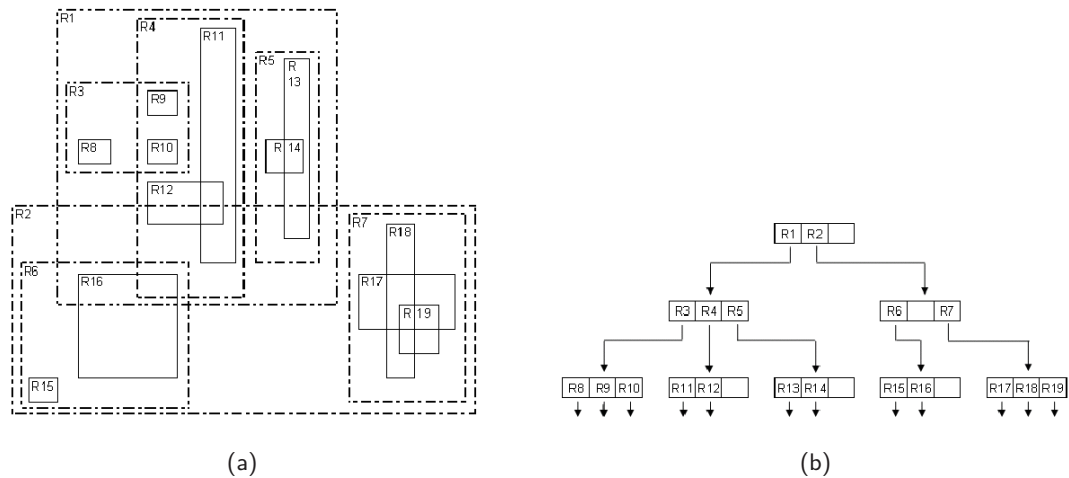


Figure 5.2 – Stratégie de recouvrement du R-tree et son arbre associé

Dans le cadre de cette thèse, nous avons utilisé un serveur de base de données fédérateur, nommé Virtuoso, qui joue le rôle d'un méta système de données (DBMS) « Data Base Management Systems » et qui propose une manipulation transparente de plusieurs systèmes de base de données autonomes. Les bases de données ainsi constituées peuvent être géographiquement décentralisées et inter-connectées par un réseau. La tâche de centralisation du système fédéré revient à Virtuoso qui permet ainsi une harmonisation des requêtes malgré le fait que le système soit constitué de serveurs bases de données totalement autonomes. Virtuoso est libre et permet de stocker divers formats de données (json, xml, svg, etc.) et permet d'utiliser des techniques de requêtes basées sur du sql ou du sparql. De nos jours, il peut facilement s'adapter au contexte spatial et il est très performant en association avec d'autres serveurs spécialement dédiés à la gestion et le stockage d'informations spatiales. Dans l'architecture que nous proposons, le serveur de base de données Virtuoso est couplé au serveur de base de données Parliament.

Parliament¹² est un serveur qui associe deux techniques intéressantes :

- une technique d'indexation sur des contenus,
- une capacité à raisonner sur des informations spatio-temporelles.

Parliament a été proposé par l'équipe de recherche et de développement de « BBN technologie »¹³.

Il associe :

- une indexation temporelle, qui permet de répondre efficacement à des questions comme "trouver tous les événements qui ont eu lieu entre les temps X et Y".
- un protocole GeoSPARQL, qui est le standard récemment adopté par la norme OGC pour représenter et stocker les données géospatiales. En utilisant son indexation spatiale, le serveur de base de données Parliament peut efficacement répondre à des questions comme "trouver tous les éléments situés dans la région X".

12. <http://parliament.semwebcentral.org/>

13. <http://bbn.com/>

- un format rdf, lui permettant de réaliser la réification et aussi le stockage des autres données.
- un moteur d’inférence performant, lui permettant de déterminer de nouveaux faits à partir d’une base de règles organisée en triple store. Il permet aussi de mettre en œuvre l’inférence proposée par le modèle RDFS (voir chapitre 2) ainsi que certaines propriétés d’OWL (par exemple, les classes d’équivalentes et les propriétés inverse, symétrique, fonctionnelle et transitive)

Langages de requêtes

Dans l’objectif d’extraire des serveurs de bases de données des informations structurées en logique de description décrites par la norme OGC, plusieurs langages de requêtes ont été proposés. Ils sont tous issus du langage de requête *sparql*. Parmi ces langages de requêtes, les plus répandus sont les langages Geosparql et stSparql. Le langage de requête Geosparql a été implémenté dans le serveur de base de données Parliament [Kolas *et al.*, 2009]. Le langage stSparql est une extension de Geosparql, développé pour le serveur de base de données Strabon [Kyzirakos *et al.*, 2012] utilisé pour le projet européen Telios¹⁴. Ces deux langages de requêtes représentent la ressource spatiale suivant les standards WKT, GML (cf, chapitre 2) et la ressource temporelle utilise la structuration proposée par [Pan, 2005]. Dans le cadre de cette thèse, le langage que nous avons utilisé est le Geosparql implémenté par le serveur de base de données Parliament.

Le langage Geosparql est basé sur le protocole sparql appliqué sur des données spatiales et représentées sous un format RDF. Dans sa structure, le langage Geosparql présente des exigences pour produire des vocabulaires d’un langage de requête telles que les classes, les propriétés et les fonctions [Perry et Herring, 2011]. Il a pour principaux composants.

- Le Noyau : c’est le composant qui définit les classes de hauts niveaux, produites par des utilisateurs avec des vocabulaires de modélisation de l’information spatiale. Les classes proposées par ce composant sont : geo:SpatialObject et geo:Feature. La classe geo:SpatialObject a pour instance toute entité ayant une représentation spatiale. La classe geo:Feature représente toutes les caractéristiques spatiales d’une entité.
- L’extension des géométries : c’est le composant qui produit un vocabulaire utilisé pour stocker et extraire une information à partir de sa géométrie. La principale erreur de conception du geosparql primitif était d’utiliser les littéraux pour encoder les valeurs des géométries comme une simple unité. Une extension a été introduite en considérant deux types de données geo:wktLiteral et geo:gmlLiteral. Cette extension a été suivie par une définition d’autres normes du standards OGC utilisé pour encoder pour encoder des géométries (P. ex. WKT et GML). Le type geo:wktLiteral est composé d’une URI identifiant le système de coordonnées géodésiques suivi d’un WKT qui encode la géométrie. Dans le même contexte, les littéraux de types geo:gmlLiteral se composent d’un élément GML valide qui permet d’implémenter un objet GML. Les stratégies fondées sur des données sérialisées ont été implémentées par les deux langages de requêtes (stSparql et Geosparql).

14. <http://www.earthobservatory.eu/Strabon>

Les deux langages de requêtes stSparql et Geosparql utilisent des littéraux identiques pour représenter des informations relatives à la géométrie de l'entité. Ces deux langages exploitent la possibilité de sérialisation des géométries suivant les standards WKT et GML. Ils expriment aussi les relations spatiales *RCC8*. Ils intègrent également tous deux le raisonneur spatio-temporel nommé *pelletSpatial* [Koubarakis *et al.*, 2012]. Mais, il existe cependant de petites disparités entre les deux langages de requêtes. La principale différence entre ces deux langages est identifiée dans leurs implémentations. Le langage stSparql est actuellement plus évolué que le langage Geosparql en ce sens qu'il intègre de nos jours toutes les propriétés de transitivité sur les relations spatiales *RCC8*. Ces propriétés ne sont pas encore toutes implémentées dans le langage Geosparql. Le langage Geosparql est régulièrement mis à jour par rapport au langage stSparql qui n'est plus mis à jour depuis l'année 2013. Ainsi, dans le cadre de cette thèse, nous avons utilisé le langage Geosparql et le serveur de base de données Parliament présenté au paragraphe ci-avant.

Le découpage de notre architecture en trois modules a permis de mieux structurer et stocker les connaissances spatiale, temporelle et sémantique afin d'être explorées par des langages de requêtes appropriés (Geosparql, stSparql, etc.). L'optimisation du temps de réponse des requêtes nous a alors permis d'intégrer cette connaissance de l'environnement maritime à l'algorithme de routage dynamique *BIDA** présenté au chapitre 4. Il en résulte la mise en place d'une stratégie de routage sous contraintes qui est détaillée dans les sections ci-après.

5.2 Prise en compte des contraintes de l'environnement

La navigation fondée sur des points remarquables ou des points d'intérêts est très importante en milieu ouvert et complexe. L'environnement maritime est un exemple où la navigation avec des points remarquables est appliquée. L'initiative de proposer une trajectoire construite en temps réel à l'aide de points remarquables est un défi permanent dans le domaine du « pathfinding ». En considérant qu'une trajectoire de navigation doit prendre en compte les phénomènes dynamiques et les contraintes générées par le milieu, les algorithmes de recherche de trajectoires doivent considérer l'ensemble de ces complexités dans leur modélisation. Deux grandes catégories de contraintes sont généralement mises en évidence : les contraintes globales et les contraintes locales qui ont été présentées dans la partie introductive de ce chapitre. L'ontologie computationnelle développée dans le chapitre 2 permet d'associer les contraintes à la solution proposée au chapitre 4 et de répondre ainsi aux besoins de la navigation maritime principalement basée sur des points d'intérêts. La nouvelle solution obtenue intègre à la fois les dimensions spatiale, temporelle et sémantique modélisées et présentées aux chapitres 2 et 3. La sémantique générée par chaque entité de l'environnement est liée à la saillance et l'affordance. La section 5.2.1 présente la stratégie utilisée pour satisfaire les contraintes globales. La présentation des contraintes locales sera alors réalisée dans la section 5.2.2.

5.2.1 Contraintes globales

Dans le cadre de cette thèse, nous nous sommes intéressés à deux contraintes globales : *la contrainte de profondeur* et *la contrainte de distance*. La contrainte de distance a initialement été prise en compte dans le chapitre 4 et a permis de proposer une heuristique permettant d'optimiser le processus et de minimiser « la longueur » du chemin solution. La distance à laquelle nous nous sommes intéressés est la distance euclidienne. Le routage proposé doit aussi satisfaire la contrainte globale de profondeur, c'est-à-dire que le chemin proposé devra se situer dans une zone de navigation de profondeur suffisante (généralement définie à partir du tirant d'eau du navire auquel on ajoute une marge de sécurité appelée pied de pilote). Il faut dans ce cas contraindre la trajectoire de navigation à cette zone de navigation en créant des points de passage obligés appelés points d'intérêt.

Les points d'intérêt globaux définissent l'ensemble des sites utilisés pour satisfaire une ou plusieurs contraintes globales liées à la navigation maritime. La séquence ou la succession des points d'intérêts globaux définit des points de passage ou des points de changement de direction dans la stratégie de routage. Une navigation sera effective si les points de départ et d'arrivée satisfont initialement la contrainte globale de profondeur et si les segments qui composent la trajectoire de navigation sont tous contenus dans une zone de navigation (de profondeur suffisante). Dans ce paragraphe, nous proposons une stratégie permettant de déterminer les points d'intérêts dits « globaux » nécessaires pour orienter la navigation dans cette zone de navigation.

La première étape consiste à déterminer si il existe un espace valide pour la recherche de trajectoire. Nous considérons que si deux points d'intérêt (initialement les points de départ s et d'arrivée t) sont contenus dans une **même** zone de navigation NA_i alors il existe un chemin continu reliant ses deux points. En conséquence, il faut vérifier si il existe une même zone de navigation contenant simultanément les points s et t . Considérons les points s et t , respectivement le point de départ et le point d'arrivée, contenus simultanément dans une même zone de navigation NA_i . La direction de navigation est donnée par la droite D définie par l'équation 4.11 (page 103). Si D est contenue entièrement dans la zone de navigation NA_i alors il n'est pas nécessaire d'introduire des points de réorientation ou de changement de direction. En revanche, si cette dernière intersecte une zone de non navigation NNA_j , des points d'intérêt globaux sont nécessaires afin de contraindre la navigation par ces points. La droite (st) divise alors l'espace (ou le plan) de recherche en deux demi-espaces (ou deux demi-plans). Si les deux demi-espaces vérifient le critère qu'il existe une **même** zone de navigation contenant simultanément les points s et t , soit le choix de la zone de recherche de la séquence des points d'intérêts globaux est arbitraire, soit la recherche est effectuée dans les deux zones de navigation. Ce choix ne présente aucun impact sur la complexité de l'algorithme. Par ailleurs, si un seul demi-espace vérifie le critère alors la zone de navigation contenue dans ce demi-espace est utilisée pour rechercher la séquence de points d'intérêts globaux. Enfin, si aucun des demi-espaces ne vérifie le critère, la stratégie de recherche ne peut pas être appliquée et il n'existe alors aucun chemin maritime permettant de relier s et t . Dans les paragraphes qui suivent, une stratégie a été développée afin de trouver des points d'intérêt globaux dans une zone de recherche

précédemment identifiée.

Dans l'objectif de déterminer les points d'intérêts globaux les plus pertinents pour la navigation, il est important d'appliquer une stratégie. Le point d'intérêt est dit pertinent s'il permet d'améliorer la navigation. Plusieurs hypothèses ont été adoptées afin de mieux mettre en évidence cette stratégie de recherche des points d'intérêt globaux. Considérons POI , l'ensemble de points d'intérêt pouvant permettre de trouver un chemin possible de navigation. Ces points d'intérêts sont déterminés en utilisant la relation d'intersection entre la direction de navigation $D = (st)$ et les zones de navigation NA_i (« Navigation Area ») et les zones de non navigation NNA_j (« Non Navigation Area »). Nous en déduisons deux cas possibles :

- 1** si le segment $[s, t]$ déduit de la direction de navigation $D = (st)$ est entièrement contenue dans une même zone de navigation, c.-à-d. $[s, t] \cap NA_i = [s, t]$, alors les points d'intérêt globaux sont respectivement les points initial et final, c.-à-d. $POI = \{s, t\}$ (voir figure 5.3) et l'algorithme de routage est appliqué à partir de ces deux points ;
- 2** si la direction de navigation (st) rencontre une zone de non navigation, c.-à-d. $\exists j$ tel que $[s, t] \cap NNA_j \neq \emptyset$, alors la droite (st) divise l'espace de recherche en deux demi-espaces $E1$ et $E2$ qui sont utilisées comme des zones de recherche des séquences de points de repères globaux. Considérons alors que le recherche des points d'intérêt s'effectue dans le demi-espace E_1 . Ce demi-espace inclut une zone de navigation NA_i susceptible d'apporter une solution de routage et une zone de non navigation NNA_j en intersection avec la direction de navigation D . L'objectif est alors de trouver un chemin solution dans la zone de navigation NA_i tout en évitant les zones de non navigation potentiellement incluse dans NA_i et contournant la zone de non navigation NNA_j . Cela revient à rechercher une zone continue allant de s à t contenue dans la zone de navigation NA_i (et évitant les zones de non navigation NNA_j).

La stratégie de recherche des points d'intérêt consiste à définir la courbe polygonale minimale (en terme de nombre de points) contenue dans la zone de navigation NA_i et approchant la frontière de la zone de non navigation NNA_j . Les points composant cette courbe polygonale approchée sont alors définis comme les points d'intérêt globaux. La courbe approchée est issue de l'utilisation d'un algorithme de simplification de type Douglas-Peucker [Douglas et Peucker, 1973] appliqué récursivement sur la courbe modélisant l'enveloppe convexe des points formant la zone de non navigation NNA_j et dont l'approximation est contrainte par la zone de navigation. Finalement, l'algorithme de routage est alors appliqué sur chaque paire de points d'intérêt tout en gérant la continuité entre les différentes portions de routage successives (Figure 5.4).

5.2.2 Contraintes locales

Le routage appliqué dans un environnement maritime exige aussi de prendre en compte les propriétés sémantiques associées aux entités qui composent l'environnement ou sa représentation cartographique. Une contrainte locale est souvent associée à la sémantique d'une entité généralement



Figure 5.3 – Illustration de l'ensemble des points d'intérêt globaux formé uniquement des points s (en rouge) et t (en noir)

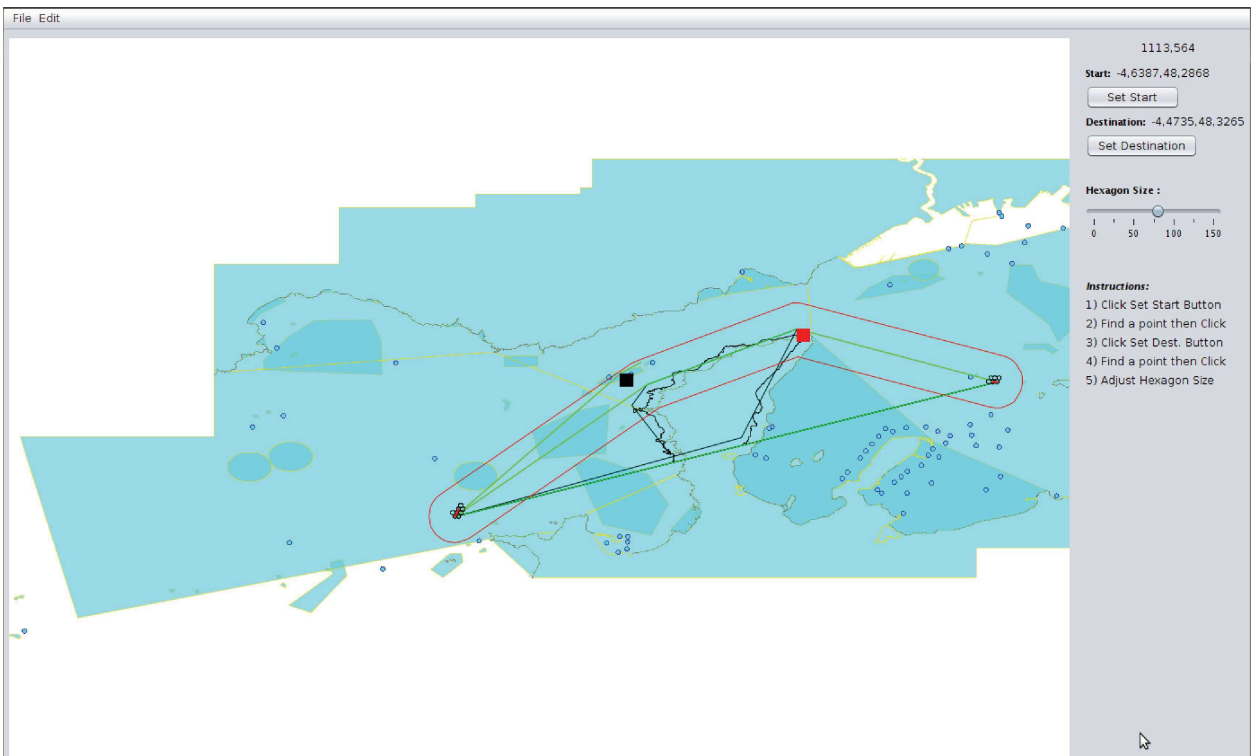


Figure 5.4 – Illustration des points d'intérêt globaux issus des points de l'enveloppe convexe de la zone de non navigation.

introduite par l'homme et destinée à orienter et encadrer la navigation. A titre d'exemple, les

bouées ou les lignes de navigation sont des entités destinées à contraindre localement la navigation. On pourra distinguer les contraintes locales dites « fortes » associées (aux bouées cardinales ou latérales) des contraintes locales dite « faibles » associées (aux lignes de navigation) qui sont des aides à la navigation. Il existe différentes catégories de contraintes locales représentées à partir de l'ontologie proposée au chapitre 2. La contrainte locale que nous avons considérée est *la contrainte de direction de navigation*. Par exemple, dans le cas d'une entité de type « Bouée cardinale nord », la direction de navigation sera orientée vers le nord de la bouée cardinale. De la même façon, un alignement d'amers impliquera tout d'abord un changement de cap du navire afin de se positionner sur cet alignement, puis un autre changement de direction de navigation pour suivre cet alignement. Ainsi la stratégie proposée dans la section suivante consiste à prendre en compte simultanément les différentes contraintes locales dont les valeurs, associées aux propriétés sémantiques des entités, peuvent être directement intégrées ou inférées à partir de la base de connaissances définie dans l'ontologie computationnelle (voir chapitre 2). Les différentes contraintes locales considérées seront celles comprises dans la zone de navigation (NA_i).

Les contraintes globales et locales illustrées dans les paragraphes 5.2.1 et 5.2.2 ont été prises en compte par la stratégie de routage dynamique fondée sur l'algorithme *BIDA**. Cette stratégie est présentée dans la section 5.3. La stratégie proposée a ainsi permis d'améliorer le routage initial basé sur la seule contrainte globale de distance (chapitre 4) et de répondre ainsi aux besoins des marins qui prennent en compte plusieurs contraintes au cours de leur navigation.

5.3 *BIDA** pour des problèmes multi-constraints

Avant de présenter l'algorithme *BIDA** appliqué sur plusieurs contraintes, il est important de procéder à une définition des algorithmes multi-constraints ou multi-critères. Un chemin soumis à plusieurs contraintes nommé P se définit comme un chemin partant d'une source s vers une destination t d'un graphe orienté $G(V, E)$ où V est l'ensemble des sommets et E est l'ensemble des arcs, tel que $w_i(P) < c_i$, où $w_{i=1\dots n}$ est une fonction associée à la i^e contrainte définie de E vers R_0^+ avec $c_i \in R_0^+$ une valeur constante également associée à la i^e contrainte.

Suite à cette définition, nous observons que les contraintes peuvent être de plusieurs types ou de natures différentes. En plus, certaines contraintes peuvent influencer d'autres contraintes car elles sont appliquées simultanément sur le même chemin P . Il est alors important de catégoriser les différents critères d'évaluation des contraintes afin de proposer une stratégie qui devra les prendre en compte. Les différentes catégories de critères que nous avons identifiées sont les critères cumulatifs ou additifs [Fargier *et al.*, 2013] auxquels nous avons ajouté les critères décisionnels et catalyseurs :

- Les critères catalyseurs sont utilisés pour accélérer la stratégie sans être évalués. Dans le cadre de la navigation, ils accélèrent la stratégie en réduisant l'espace de recherche de la solution optimale. L'espace de recherche désigne l'ensemble des directions ou orientations pouvant conduire à la solution. Le critère catalyseur utilisé dans cette stratégie est la direction de navigation. Le

stockage des critères catalyseurs est inutile dans une structure de données. Ces critères réduisent également de façon considérable la taille de la structure de données.

- Les critères cumulatifs ou additifs désignent les critères dont la valeur du critère à une étape donnée est importante pour l'étape suivante. Un critère cumulatif ou additif est défini par la relation $\sum_{i=1}^N C_{ij} < \beta_i$, où C_{ij} est la valeur associée au i^e critère, j est l'étape à laquelle le critère a été évalué et β_i est la borne supérieure. Le critère cumulatif dans cette stratégie est le critère de distance. Ce dernier est évalué par des fonctions ou des heuristiques. Il est important de stocker les valeurs de ce critère dans la structure de données.
- Les critères décisionnels permettent de vérifier à tout moment si la valeur courante du critère est dans une plage de valeurs prédéfinies. Cette plage de valeurs correspond à un intervalle de valeurs selon laquelle la navigation est rendue possible. Tout nœud du graphe (voir chapitre 4) pondéré par une valeur n'appartenant pas à cet intervalle est rejeté comme un nœud ne pouvant conduire à la solution. Le critère décisionnel dans cette stratégie est le critère de profondeur. La navigation est rendue possible lorsque les valeurs de profondeur sont comprises entre les valeurs α_{inf} et α_{sup} , ainsi toute valeur de profondeur qui n'est pas dans cet intervalle n'est pas prise en compte pas la stratégie.

Contraintes	Critères cumulatifs	Critères catalyseurs	Critères décisionnels
Direction de navigation	Non	Oui	Non
Profondeur de navigation	Non	Non	Oui
Distance	Oui	Non	Non

Tableau 5.2 – Illustration des différents groupes de critères

Appliquer une stratégie qui prend en compte ces différents groupes de critère exige d'établir un ordre de priorité sur ceux-ci. La table 5.2 illustre les groupes auxquels appartiennent les différentes contraintes que nous avons pris en compte. Une fois défini un ordre de priorité, il revient à évaluer les différentes contraintes de chaque groupe de critère. L'ordre de priorité est défini en fonction de l'exigence du groupe de critère dans la préservation de la sécurité. En partant des plus importantes aux moins importantes, nous identifions les critères décisionnels, les critères catalyseurs et les critères cumulatifs. De plus, au sein d'un même groupe de critères, il se peut que plusieurs entités exigent de prendre en compte plusieurs contraintes. Dans cette situation, les contraintes dites fortes seront considérées avant les contraintes dites faibles. A titre d'exemple, considérons le groupe de critères catalyseurs qui s'observe généralement sur des entités destinées à contraindre localement la direction de navigation. La sémantique associée aux entités de type « bouées » sera prioritaire par rapport à la sémantique des entités de type « alignement » qui sont des aides à la navigation

et par conséquent des contraintes faibles.

Ainsi, notre stratégie prend d'abord en compte le groupe de critères décisionnels uniquement associé à la contrainte de profondeur dans le cadre de cette thèse. La prise en compte de la profondeur a été présentée dans la section 5.2.1. Cette prise en compte de la profondeur permet de déterminer une zone NA_i à l'intérieur de laquelle la navigation sera possible. Il peut arriver que nous ayons plusieurs contraintes appartenant au groupe de critères décisionnels. Dans ce contexte, la zone navigation résultante sera celle qui satisfait toutes les contraintes. La prise en compte des critères décisionnels exige de déterminer une séquence de points d'intérêt (voir section 5.2.1) qui permettront éventuellement de contourner les différentes zones de non navigation. Cette séquence de points d'intérêt sera utilisée pour construire la route de navigation. La prise en compte des entités et de leur sémantique influence la direction de navigation. Cela revient à considérer le groupe de critères catalyseurs qui permet d'accélérer le processus, de réduire la taille de la structure de données (voir section 5.3.1) et de garantir la sécurité des biens et des usagers. Enfin, la stratégie prendra en compte le groupe de critères cumulatifs, qui est associé uniquement à la distance euclidienne dans cette thèse. Il permet de déterminer un chemin optimal en terme de longueur de trajectoire.

La prise en compte de ces différents critères a été rendue possible à partir de la formalisation présentée dans la section suivante.

5.3.1 Formalisation des différents critères

L'extension de l'algorithme *BIDA** consiste à évaluer une cellule courante sur plusieurs critères au lieu d'un seul critère comme présenté au chapitre 4. La structure de données ne change pas, seul le mécanisme d'intégration des contraintes est décrit dans les paragraphes suivants.

Considérons ${}^m_n[Hex(i, j)]_S^{d_{pc}}$ la notation d'une cellule hexagonale de la grille itérative générée par *BIDA** (équation 4.14, page 105) et $A^{(i,j)}(l, c)$ une matrice d'adjacence pondérée, appliquée au point central de la cellule d'indices (i, j) . La matrice $A^{(i,j)} = (a_{lc}^{ij})_{l=1\dots L \ c=1\dots 6}$ se compose de L lignes désignant les L critères locaux et C colonnes désignant les 6 directions $\{Nord-Est, Est, Sud-Est, Sud, Sud-Ouest, Ouest, Nord-Ouest\}$ correspondant aux 6 directions de l'hexagone. La direction Nord-Est correspond à la valeur 1, la direction Est à la valeur 2, la direction *Sud - Est* à la valeur 3, la direction *Sud - Ouest* à la valeur 4, la direction Ouest à la valeur 5 et direction *Nord - Ouest* à la valeur 6 comme déjà défini dans la partie 4.3.1 (page 93). La composante (a_{lc}^{ij}) correspond à la contribution du nœud fils du nœud parent (i, j) dans la direction c par rapport au critère l . A chaque évaluation d'un nœud fils, la matrice d'adjacence associée au nœud parent (i, j) est mise à jour.

La mise à jour de la matrice $A^{(i,j)}$ à partir des différents critères est réalisée comme suit :

- la mise à jour de la matrice $A^{(i,j)}$ afin de prendre en compte la direction de navigation i correspondant à l'une des 6 directions de l'hexagone, exige de construire une matrice carrée $D^{(i,j)}$ composée de vecteurs colonnes d_k avec $k = 1 \dots 6$ tels que d_i est un vecteur unitaire pour la i^e

colonne de la matrice et $d_k = 0$ (vecteur nul) pour tous les autres vecteurs colonnes. La prise en compte de plusieurs directions de navigation possibles implique la prise en compte de plusieurs vecteurs unitaires. Ainsi, le choix d'une ou plusieurs de direction de navigation revient donc à multiplier la matrice $A^{(i,j)}$ par $D^{(i,j)}$, c.-à-d. $A'^{(i,j)} = A^{(i,j)} \times D^{(i,j)}$. La nouvelle matrice A' est réduite par rapport à la matrice initiale A dans les directions possibles, d'où le nom de critères catalyseurs. Il en résulte une matrice $A^{(i,j)}$ pondérée (sur l'ensemble des critères) dans les colonnes associées aux directions de navigation possibles et nulle ailleurs. Supposons que l'entité exige d'orienter la navigation vers l'Est qui correspond à la direction 2, les matrices $D^{(i,j)}$ et $A'^{(i,j)}$ associée à cette direction de navigation sont respectivement présentée par les équations 5.2 et 5.3.

$$D^{(i,j)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (5.2)$$

$$A'^{(i,j)} = \begin{pmatrix} 0 & a_{12}^{ij} & 0 & \dots & 0 & 0 \\ 0 & a_{22}^{ij} & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & a_{(L-1)2}^{ij} & 0 & \dots & 0 & 0 \\ 0 & a_{L2}^{ij} & 0 & \dots & 0 & 0 \end{pmatrix} \quad (5.3)$$

- la mise à jour de la matrice $A^{(i,j)}$ afin de prendre en compte la contrainte de distance exige de considérer uniquement les colonnes non nulles de cette matrice. A la ligne de la matrice correspondant à cette contrainte, une heuristique appropriée est utilisée pour identifier le nœud le plus prometteur qui sera choisi et inséré dans la pile de nœuds prometteurs pouvant conduire la solution (voir chapitre 4). L'heuristique définie au chapitre 4 est utilisée dans cette stratégie pour le critère de distance.

En définitive, l'intégration de contraintes dans l'algorithme $BIDA^*$, revient à construire un vecteur $H = (h_1, h_2, \dots, h_L)$ où h_i est une heuristique associée au critère i et qui évalue la ligne i de la matrice $A^{(i,j)}$. Une fois ces critères intégrés et hiérarchisés, l'algorithme $BIDA^*$ permet d'orienter la direction de propagation à partir des contraintes locales et globales.

Les figures 5.5a, 5.5b et 5.6 sont des exemples d'orientation de la navigation par les contraintes globales et locales. Dans la figure 5.5a, la trajectoire de navigation est déviée à la rencontre d'une bouée cardinale ouest de couleur jaune. La prise en compte de cette contrainte locale réoriente la navigation vers une zone plus sécurisée. C'est également dans ce contexte, que la ligne de navigation (ligne en bleu dans la figure 5.5b), considérée comme une contrainte de direction faible, attire la trajectoire de navigation et crée de cette façon une longueur de trajectoire plus importante. La



(a)



(b)

Figure 5.5 – (a) Illustration de la stratégie de routage suite à la rencontre de la bouée cardinale ouest de couleur jaune « Fillettes » et (b) la route finale solution.

figure 5.6 illustre une route de navigation proposée par l'algorithme *BIDA** qui prend en compte l'ensemble des contraintes globales et locales dont les zones d'accès restreint.

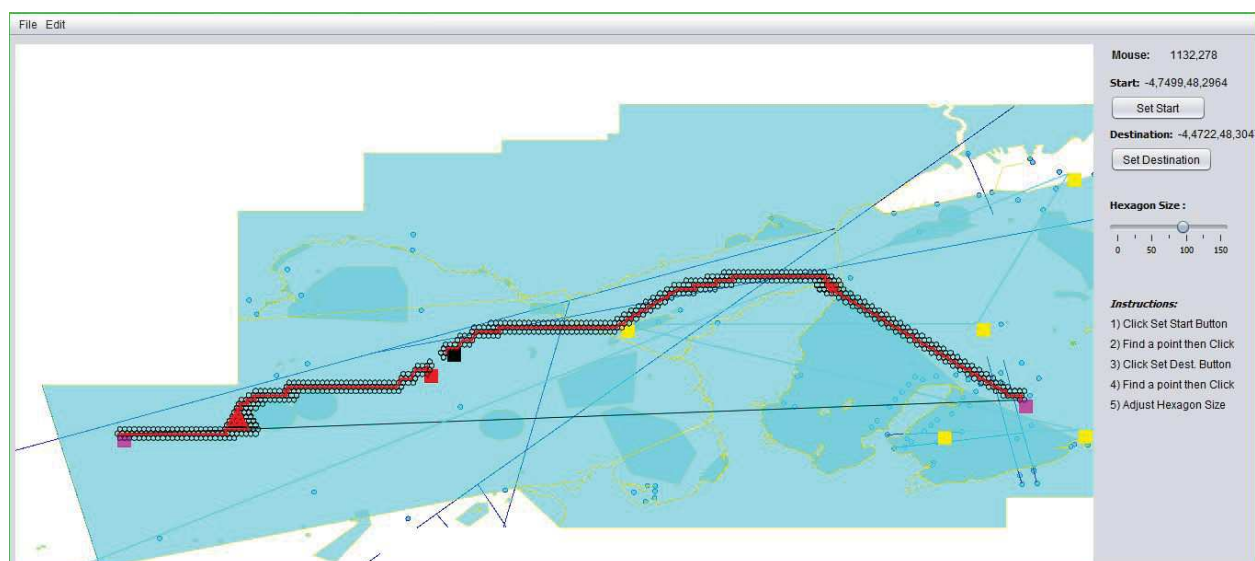


Figure 5.6 – (a) Illustration d'une route de navigation respectant l'ensemble des contraintes locales et globales de l'environnement. En bleu foncé, les zones d'accès restreint.

5.4 Conclusion

Ce chapitre présente tout d'abord l'architecture globale du système développé et qui a permis de stocker l'ontologie computationnelle proposée aux chapitres 2 et 3. Cette architecture se compose de trois grands modules nécessaires pour son implémentation. Les différents modules sont le module de traitement de langage, le module d'édition d'ontologie et le module de stockage et d'utilisation des données. L'initiative de réaliser un stockage et une exploration efficaces des connaissances, nous a orienté vers le choix d'un couplage entre le système fédérateur de bases de données Virtuoso et le serveur de base de données spatiales Parliament. Ce couplage se justifie également par la volonté d'intégrer différents formats de représentation d'information de l'environnement maritime provenant de diverses sources d'acquisition de données (voir les chapitres 2 et 3). La performance de ce couplage, notamment visible lors de l'exploration des données spatio-temporelles, est rendue possible grâce au serveur Parliament qui représente les données spatio-temporelles en triplet par l'utilisation d'une technologie triple store et utilise un langage de requête Geosparql. A partir de ce couplage, il est possible de stocker les données ou les connaissances spatiale, temporelle et sémantique et d'optimiser le temps de réponse des requêtes tout en offrant une capacité de raisonnement importante sur ces connaissances.

Cette architecture a ainsi permis de prendre en compte plusieurs critères dans l'algorithme de routage temps réel initialement proposé dans le chapitre 4. La prise en compte de ces critères par un algorithme de routage exige de mieux catégoriser les différentes contraintes locales et globales qui influencent la navigation maritime. Les contraintes identifiées sont la contrainte locale de direction de navigation et les contraintes globales de distance et de profondeur. La contrainte locale de

direction de navigation est associée à la dimension sémantique de l'entité. Ces deux catégories de contraintes ont été prises en compte dans l'algorithme de routage dynamique *BIDA**.

La stratégie ainsi proposée se compose de deux étapes. La première étape consiste à rechercher une zone de navigation continue qui vérifie la contrainte globale de profondeur sur laquelle sera déployé l'algorithme *BIDA**. Une séquence de points d'intérêt globaux sera utilisée pour orienter l'algorithme de routage dynamique et restreindre la navigation dans cette zone. La seconde étape consiste à rechercher les entités locales susceptibles de contraindre la trajectoire de navigation par l'évaluation de contrainte locale associée à l'entité. Dans notre étude, la contrainte locale est restreinte à la direction de navigation. Ces différentes étapes ont été formalisées et intégrées à l'algorithme de routage dynamique *BIDA**. Les différents cas d'étude ont permis de définir les priorités qui peuvent exister entre les différents critères afin de faciliter et d'optimiser le processus d'évaluation des critères.

Conclusion et perspectives

Contributions

NOTRE thèse est principalement dédiée à l'amélioration des services d'aide à la navigation maritime par l'extraction et la représentation des connaissances de cet environnement. Le service auquel nous nous sommes intéressés est le routage maritime. Le but fondamental d'un routage est de trouver une route de navigation partant d'un point de départ vers un point d'arrivée tout en respectant l'ensemble des contraintes et des règles liées à cet environnement. Ces contraintes ont été catégorisées en deux grands groupes : les contraintes locales et les contraintes globales. Le développement de ce service de routage repose sur la mise en place d'un modèle performant. Le modèle proposé supporte le stockage et la manipulation (extraction, exploration, apprentissage) des connaissances spatiales, temporelles et sémantiques de l'environnement maritime. Dans ce contexte, une architecture a été déployée pour mettre en œuvre le modèle. L'architecture proposée repose sur trois modules : 1) le module de traitement du langage, 2) le module d'édition de l'ontologie et 3) le module de stockage et d'utilisation des données. Pour chacun de ces modules au moins un chapitre lui a été assigné ou a participé à sa réalisation. La répartition de la contribution des chapitres sur les différents modules est présentée dans le tableau ci-dessous :

	Module 1	Module 2	Module 3
Chapitre 1 (page 9)	-	-	-
Chapitre 2 (page 33)	Non	Oui	Non
Chapitre 3 (page 65)	Oui	Oui	Oui
Chapitre 4 (page 85)	Non	Non	Oui
Chapitre 5 (page 119)	Oui	Oui	Oui

Tableau 3 – Contribution des différents chapitres sur les modules de l'architecture du modèle proposé. 1) Module de traitement du langage, 2) Module d'édition de l'ontologie 3) Module de stockage et d'utilisation des données.

Le découpage de l'architecture du modèle en 3 modules indépendants, nous permet d'ajouter facilement de nouvelles fonctionnalités et aussi d'assurer la maintenance du modèle. Bien que les modules soient indépendants, ils peuvent communiquer de façon efficace grâce à l'utilisation de ser-

veurs de bases de données dédiés au web sémantique qui assurent l'interopérabilité et minimisent le temps de réponse aux requêtes sur des données, grâce au langage RDF. Ces données proviennent généralement de plusieurs sources. Afin de pouvoir utiliser ces données, celles-ci doivent être préalablement structurées par des ontologies computationnelles. L'ontologie computationnelle proposée a pour objectifs de représenter l'entité maritime en prenant en compte sa spatialité, sa dynamique et la sémantique (saillance et affordance) qu'elle dégage.

L'ontologie computationnelle proposée au chapitre 2, nous permet de produire une représentation aussi proche que possible de la réalité. La mise en œuvre d'une telle ontologie exige de prendre en compte les dimensions spatiale, temporelle et sémantique qui sont associées à une entité de l'environnement maritime. La modélisation proposée s'est inspirée de la représentation des entités spatio-temporelles basée sur le changement. Cette ontologie computationnelle représente l'entité pendant son cycle de vie et utilise un ensemble d'agrégations d'événements quotidiens compris dans un intervalle délimité par un instant de début et un instant de fin de vie de l'entité.

Chaque agrégation d'événements de l'entité peut connaître une variation de l'empreinte spatiale, une variation de l'emplacement spatial, une variation de la perception et une variation de la durée de vie. Chacune de ces variations a un impact sur la saillance et par conséquent sur l'affordance qui peut être associée à l'entité. La modélisation d'une agrégation d'événements a été considérée différemment de jour et de nuit. A chaque agrégation d'événements, une liste d'états est associée à l'entité. Ces états renseignent sur les différentes étapes que peut traverser une entité au cours de sa vie. Un état renseigne plusieurs informations sur l'entité (la couleur, la direction de navigation, la durée de son activité, la valeur de l'affordance, la visibilité). A l'aide de cette modélisation, il est possible de connaître dans quel état se trouve l'entité. L'intérêt de connaître à chaque instant l'état dans lequel une entité se trouve, est de rechercher les points d'intérêt les plus saillants. Cette modélisation permet ainsi de mieux représenter l'environnement et faciliter la mise en œuvre des applications maritimes dédiées à de l'aide à la décision. L'ontologie computationnelle ainsi proposée nécessite des sources de connaissances ou des données pour son enrichissement. Les connaissances ou les données peuvent provenir de l'expert ou être extraites de gros volumes de données textuelles par des algorithmes issus du traitement automatique du langage. Les connaissances produites par l'expert se retrouvent dans des documents nautiques spécialisés et sont représentées par l'ontologie computationnelle. Les connaissances non identifiées par l'expert sont extraites de façon automatique de volumes de données textuelles conséquents en utilisant des techniques appropriées.

La technique proposée permettant d'extraire des connaissances à partir de données textuelles nautiques est fondée sur le principe du vecteur conceptuel d'un mot. Le vecteur conceptuel d'un mot s'obtient à partir des synsets candidats liés à ce mot. Ce vecteur conceptuel de mot a permis de construire le vecteur conceptuel d'une phrase. Le vecteur conceptuel d'une phrase s'obtient en considérant qu'une phrase est formée d'un ensemble de mots. Ainsi, le vecteur conceptuel d'une phrase a donc été exprimé comme une somme directe des vecteurs conceptuels des différents mots qui la compose. La définition de cette somme directe est présentée dans le chapitre 3. L'utilisa-

tion du vecteur conceptuel d'une phrases a permis de calculer l'affordance associée à un concept donné. Les phrases utilisées sont celles qui décrivent le concept dans un corpus. Une fois les vecteurs conceptuels de ces phrases déterminés, ceux-ci sont projetés dans un espace de décision formé des vecteurs conceptuels des affordances « maneuver, stop, continue, back ». Ces affordances sont couramment utilisées par les marins pour modifier la trajectoire du navire. L'espace de décision fera émerger une des affordances (lorsque celle-ci n'a pas encore été identifiée par l'expert). Cette possibilité de déterminer l'affordance procure à notre modèle la capacité de pouvoir fabriquer ou de déduire lui-même sa propre connaissance et ainsi prendre des décisions raisonnables en toute autonomie. L'affordance ainsi produite est désignée par la propriété (*hasAffordance*) qui fait partie de l'une des propriétés des états d'une entité atomique définie par l'ontologie computationnelle.

L'ontologie computationnelle proposée est utilisée pour représenter l'environnement maritime en tenant compte de sa dynamique et de sa complexité. Il est alors important de pouvoir la stocker et la manipuler de la façon la plus efficace possible. Pour ce faire, nous avons identifié deux serveurs de base de données appropriés pour atteindre ces objectifs. En considérant le fait que les modèles ontologiques utilisent des serveurs de base de données « triple store » pour sauvegarder la connaissance, les deux serveurs qui ont été utilisés sont le serveur Virtuoso, jouant le rôle de fédérateur, et Parliament.

Ce couplage est très performant pour des requêtes spatio-temporelles. Le serveur Virtuoso offre la possibilité de prendre en compte différents formats de données permettant ainsi de représenter les diverses sources d'information de l'environnement maritime. L'optimisation du temps de réponse des requêtes spatio-temporelles par le langage Geosparql a permis à notre algorithme de routage dynamique de prendre en compte plusieurs contraintes. L'intérêt de prendre en compte plusieurs contraintes de navigation maritime est de se rapprocher le plus possible de la réalité physique. Notre algorithme de routage dynamique s'applique de façon bidirectionnelle sur une région donnée en générant de façon itérative une séquence de trois hexagones fils à partir d'un hexagone courant. Les trois hexagones fils ainsi construits itérativement correspondent au facteur de branchement associé à un hexagone. Cet algorithme réduit i) le temps de calcul du chemin optimal qui relie un point de départ à une destination donnée, ii) l'espace mémoire utilisé pour sauvegarder les hexagones composant la trajectoire solution en proposant une structure de données adaptée et iii) le temps d'affichage en évitant la redondance des tracés et des calculs des points et des côtés communs à plusieurs hexagones. Le gain en temps favorise ainsi les stratégies multi-contraintes et les prises de décision en temps réel. L'algorithme de routage a donc été enrichi par l'ajout d'une contrainte locale de direction de navigation issue de la sémantique des entités rencontrées et des contraintes globales de distance et de profondeur.

Perspectives

La représentation du monde par un SIG est une simplification de la réalité physique. L'acquisition des connaissances, leur modélisation et leur exploitation sont essentielles afin d'améliorer cette représentation du monde. La finalité de cette recherche est de tendre vers des systèmes d'information géographiques dits « intelligents », capables d'intégrer des informations hétérogènes, de raisonner et de fournir des résultats en temps réel. C'est dans ce contexte que cette thèse a posé les premières briques d'une architecture dédiée à la navigation maritime. Cependant, de nombreuses perspectives de recherche peuvent contribuer à son enrichissement et son amélioration.

Tout d'abord, le besoin d'une ontologie computationnelle plus complète nécessite d'enrichir la base de connaissances et la base de règles afin de raisonner et d'inférer davantage de connaissances à partir des connaissances originelles. La proposition d'une stratégie d'enrichissement effective exige d'améliorer la stratégie initiale fondée sur le traitement automatique du langage proposée au chapitre 3, en lui permettant :

- d'extraire l'affordance associée à une entité en se basant sur le sens lexicographique de la phrase. Ceci suppose de structurer préalablement les phrases par des techniques de désambiguïsation, sur un corpus varié (wikipédia, etc.) non restreint à des seuls documents textuels nautiques comme cela est réalisé dans cette thèse.
- de déterminer des connaissances ou des propriétés spatiales, temporelles et sémantiques associées aux entités de l'environnement de façon automatique. Ces connaissances seront soit utilisées par le modèle de données (p.ex. la géométrie de l'entité, l'instant de « mise en œuvre » et de « clôture » de l'entité, son nom, etc.), soit utilisées par le modèle de règles de l'ontologie computationnelle (p. ex. la distance minimale de visibilité, la surface minimale de perception, l'information associée à la variation de la durée de vie ou celle associée à la variation de la perception (variation de couleur)).

Une fois ces connaissances extraites, il faudra alors déterminer ou filtrer celles qui apparaissent comme les plus pertinentes pour la modélisation.

Les nouvelles connaissances, ainsi renseignées dans le modèle de l'ontologie computationnelle, permettront d'ajouter de nouvelles contraintes et d'améliorer l'algorithme de routage. Les nouvelles contraintes visées devront être relatives à la prise en compte des entités mobiles présentes dans l'environnement maritime, des données météorologiques et des phénomènes de marée et de courant.

Concernant, les entités mobiles qui se partagent le même environnement, il sera important de développer les règles de navigation en fonction des privilèges accordés aux différentes entités. Ces règles¹⁵ sont utilisées pour éviter les collisions entre les différentes entités mobiles. Pour atteindre cet objectif, le modèle de données de l'ontologie doit intégrer une nouvelle catégorie qui devra modéliser les connaissances provenant des données fournies par les entités mobiles (données AIS, etc.). Dans

15. Ces règles sont transcrites dans un ouvrage nautique appelé le Règlement International pour Prévenir les Abordages en Mer (RIPAM).

le même contexte, les contraintes issues de la modélisation des connaissances météorologiques, de courant, de houle et de marée devront être intégrées en utilisant les champs vecteurs qui leur sont associés. Ces connaissances influenceront sur l'algorithme de routage notamment par la prise en compte d'un critère lié à la « fatigue » du bateau ou à l'impact de l'environnement sur l'entité.

L'intégration de contraintes additionnelles dans l'algorithme de routage va aussi créer une concurrence sur l'ordre d'exécution de ces dernières. Ainsi, le développement d'une approche qui contrôle l'ordonnancement de l'exécution des contraintes devra être mise en œuvre. En ce qui concerne cette thèse, les contraintes que nous avons proposées ont été catégorisées et hiérarchisées suivant un ordre de priorité à cause de leur faible nombre. Cette stratégie doit évoluer afin de gérer un plus grand nombre de contraintes et dont certaines d'entre elles sont en concurrence permanente.

Une fois ces connaissances intégrées au modèle, il est important d'aborder la problématique de la formalisation de la trajectoire solution. Dans le modèle actuel, la trajectoire de navigation est formalisée à l'aide d'un graphe représentant les liens de parenté entre les différents hexagones (éventuellement à différents niveaux de résolution) composant les nœuds solutions. Une telle formalisation est directement exploitable par des entités autonomes et robotisées. En revanche, une telle formalisation n'est pas suffisante en vue d'une exploitation par des humains. Il est alors nécessaire d'introduire une formalisation d'une trajectoire de navigation qui soit plus intuitive (sur la base d'information de cap, distance, points de réorientation (par rapport à un amer), alignement d'amers, etc.) afin de faciliter son exploitation par des humains mais aussi les analyses spatiales qui pourraient être faites, comme la comparaison ou la similarité de trajectoires.

Dans nos travaux, l'évaluation des différents résultats est présentée d'un point de vue scientifique grâce au calcul de complexité algorithmique et sur la base d'une ontologie computationnelle modélisant un échantillon d'entités spatio-temporelles et un corpus réduit à des documents textuels nautiques. Au final, il est indispensable de faire évaluer le modèle et l'algorithme de routage par des experts afin de déterminer leurs performances et leurs impacts dans des cas d'utilisation réels.

Bibliographie

- [Ahmadi *et al.*, 2008] Ahmadi, S., Ebadi, H., et Valadan, Z. (2008). A new method for path finding of power transmission lines in geospatial information system using raster networks and minimum of mean algorithm. *World Applied Sciences Journal*, 3(2) :269–277.
- [Ahn et Snodgrass, 1986] Ahn, I. et Snodgrass, R. (1986). Performance evaluation of a temporal database management system. *SIGMOD Rec.*, 15(2) :96–107.
- [Allen, 1984] Allen, J. F. (1984). Towards a general theory of action and time. *Artif. Intell.*, 23(2) :123–154.
- [Allen et Hayes, 1990] Allen, J. F. et Hayes, P. J. (1990). Moments and points in an interval-based temporal logic. *Comput. Intell.*, 5(4) :225–238.
- [Aubin et Hamon, 2006] Aubin, S. et Hamon, T. (2006). Improving term extraction with terminological resources. In Salakoski, T., Ginter, F., Pyysalo, S., et Pahikkala, T., éditeurs, *Advances in Natural Language Processing*, volume 4139 of *Lecture Notes in Computer Science*, pp 380–387. Springer Berlin Heidelberg.
- [Baader et Sattler, 2001] Baader, F. et Sattler, U. (2001). An overview of tableau algorithms for description logics. *Studia Logica*, 69(1) :5–40.
- [Beckmann *et al.*, 1990] Beckmann, N., Kriegel, H., Schneider, R., et Seeger, B. (1990). The r*-tree : An efficient and robust access method for points and rectangles. *SIGMOD Rec.*, 19(2) :322–331.
- [Beckwith *et al.*, 1990] Beckwith, R., Fellbaum, C., Gross, D., et Miller, K. (1990). Introduction to wordnet : An online lexical database. *International Journal of Lexicography*, 3(4) :235–244.
- [Bizer *et al.*, 2009] Bizer, C., Heath, T., et Berners-Lee, T. (2009). Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3) :1–22.
- [Bonnet, 1989] Bonnet, C. (1989). *Traité de psychologie cognitive*, volume 1, chapter La perception visuelle des formes, pp 1–82. Perception, Action, Langage, Paris, dunod edition.
- [Borrmann *et al.*, 2009] Borrmann, A., Hyvärinen, J., et Rank, E. (2009). Spatial constraints in collaborative design processes. In *Proc. of the Int. Conf. on Intelligent Computing in Engineering (ICE'09)*, Berlin, Germany.

- [Borst *et al.*, 1997] Borst, W. R., Akkermans, J. M., et L, T. J. (1997). Engineering ontologies. *International Journal of Human-Computer Studies*, 46(2-3) :365–406.
- [Chauché, 1990] Chauché, J. (1990). Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance. *TAL Information*, 1(1) :17–24.
- [Claramunt *et al.*, 2005] Claramunt, C., Fournier, S., Li, X., et Peytchev, E. (2005). Real-time geographical information for its. In *the 5th IEEE International Conference in Intelligent Transportation Systems*, pp 237–242. IEEE.
- [Claramunt et Jiang, 2000] Claramunt, C. et Jiang, B. (2000). An integrated representation of spatial and temporal relationships,. *Geographical Systems*, 3(4) :154–159.
- [Claramunt *et al.*, 1997a] Claramunt, C., Parent, C., et Thériault, M. (1997a). Design patterns for spatio-temporal processes. In Spaccapietra, S. et Maryanski, F., éditeurs, *in Searching for Semantics : Data Mining, Reverse Engineering*, pp 415–428. Chapman & Hall.
- [Claramunt et Thériault, 1995] Claramunt, C. et Thériault, M. (1995). Managing time in gis : An event-oriented approach. In Clifford, J. et Tuzhilin, A., éditeurs, *Recent Advances on Temporal Databases*, pp 23–42. Springer-Verlag, Zurich, Switzerland,.
- [Claramunt *et al.*, 1997b] Claramunt, C., Thériault, M., et Parent, C. (1997b). A qualitative representation of evolving spatial entities in two-dimensional spaces. In Carver, S., éditeur, *In Innovations in GIS V*, pp 119–129. Taylor & Francis.
- [Cleuziou *et al.*, 2011] Cleuziou, G., Dias, G., et Levoraga, V. (2011). Acquisiton de structures lexico-sémantiques à partir de textes : un nouveau cadre de travail fondé sur une structuration prétopologique. In Khenchaf, A. et Poncelet, P., éditeurs, *EGC*, volume 20, pp 107–118, Brest.
- [Cleverdon, 1962] Cleverdon, C. W. (1962). *Report on Testing and Analysis of an Investigation into the Comparatie Efficiency of Indexing Systems*. Rapport technique 2, Aslib Cranfield Research Project. <https://dspace.lib.cranfield.ac.uk/handle/1826/836>.
- [Cohn et Hazarika, 2001] Cohn, A. G. et Hazarika, S. M. (2001). Qualitative spatio-temporal continuity. In Montello, D., éditeur, *Spatial Information of Theory Foundations Geographic Information Science (COSIT)*, pp 92–127, Morro Bay, CA, USA,. Springer-Verlag.
- [Cook, 1971] Cook, S. (1971). The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pp 151–158, New York, NY, USA. ACM.
- [Couclelis, 1992] Couclelis, H. (1992). People manipulate objects (but cultivate fields) : Beyond the raster-vector in gis debate in gis. In Frank, A., Campari, I., et Formentini, U., éditeurs, *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, pp 66–77, Pisa, Italy. Springer Berlin Heidelberg.
- [Cui et Shi, 2011] Cui, X. et Shi, H. (2011). A*-based pathfinding in modern computer games. *International Journal of Computer Science and Network Security (JCSNS)*, 11(1) :125–130.

- [Daniel, 2012] Daniel, C. H. (2012). *a WordNet Library for Java Processing*. <http://www.rednoise.org/rita/wordnet/documentation/index.htm>.
- [Dantzig, 1960] Dantzig, G. B. (1960). On the shortest route through a network. *Management Science*, 6(2) :187–190.
- [de Champeaux, 1983] de Champeaux, D. (1983). Bidirectional heuristic search again. *Journal of the ACM (JACM)*, 30(1) :22–32.
- [Diego, 1996a] Diego, C. (1996a). Finite model reasoning in description logics. In *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, Cambridge, Massachusetts, pp 292–303, USA,.
- [Diego, 1996b] Diego, C. (1996b). Reasoning with inclusion axioms in description logics : Algorithms and complexity. In Wahlster, W., éditeur, *12th European Conference on Artificial Intelligence, Budapest, Hungary*, pp 303–307. Wiley, J and Chichester, Sons.
- [Dijkstra, 1959] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1) :269–271.
- [Dillenburg et Nelson, 1994] Dillenburg, J. et Nelson, P. (1994). Perimeter search. *Artificial Intelligence*, 65(1) :165–178.
- [Dolinskaya, 2012] Dolinskaya, I. S. (2012). Optimal path finding in direction, location, and time dependent environments. *Naval Research Logistics (NRL)*, 59(5) :325–339. <http://users.iems.northwestern.edu/dolira/publications.html>.
- [Domini et al., 1997] Domini, F. M., Lenzerini, M., Nardi, D., et Schaerf, D. (1997). Reasoning in description logics.
- [Douglas et Peucker, 1973] Douglas, D. H. et Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2) :112–122.
- [Dreyfus, 1969] Dreyfus, D. (1969). An appraisal of some shortest-path algorithm. *Operations research*, 17(3) :395–412.
- [Drummond et al., 2006] Drummond, N., Rector, A., Stevens, R., Moulton, G., Horridge, M., Wang, H., et Sedenberg, J. (2006). Putting owl in order : Patterns for sequences in owl. In Cuenca Grau, C., Hitzler, P., et Wallace, E., éditeurs, *OWL Experiences and Directions (OWLED 2006)*, volume 216, Athens, Georgia, USA.
- [Dumais et al., 1997] Dumais, S. T., Letsche, T. A., Littman, M. L., et Landauer, T. K. (1997). Automatic cross-language retrieval using latent semantic indexing. In Hull, D. et Oard, D., éditeurs, *AAAI-97 Spring Symposium Series : Cross-language Text and Speech Retrieval*, pp 18–24, Stanford University, California. AAAI Press,.
- [Easterfield et al., 1991] Easterfield, J., Newell, G., et Theriault, D. (1991). Modeling spatial and temporal information,. In *European Conference on GIS (EGIS)*, pp 294–304., Brussels,.

- [Edmonds, 1993] Edmonds, P. G. (1993). A computational model of collaboration on reference in direction-giving. Rapport de Master, University of Toronto (Canada).
- [Egenhofer, 2002] Egenhofer, M. (2002). Toward the semantic geospatial web. In Voisard, A. et Chen, S., éditeurs, *Proceedings of the Tenth ACM International Symposium on Advances in Geographic Information Systems*, pp 1–4, McLean, VA (near Washington, DC), USA. GIS, ACM.
- [Egenhofer, 1989] Egenhofer, M. J. (1989). A formal definition of binary topological relationships. In Litwin, W. et Schek, H. J., éditeurs, *Foundations of Data Organization and Algorithms, 3rd International Conference, (FODO)*, volume 367, pp 457–472, Paris, France., Springer-Verlag New York, Inc.
- [Fargier *et al.*, 2013] Fargier, H., Marquis, P., et Schmidt, N. (2013). Capacité s pratique des diagrammes de dé cision valué s : normalisation, heuristiques et expé rimentations. In *JFPC*, volume 9, Aix-en-Provence, France.
- [Finkel et Bendley, 1976] Finkel, R. et Bendley, J. (1976). Quadrees- a data structure for retrieval on composite keys. *Acta Informatica*, 4 :1–9.
- [Frank, 1998] Frank, A. U. (1998). Formal models for cognition - taxonomy of spatial location description and frames of reference. In Freksa, C., Habel, C., et Wender, K. F., éditeurs, *Spatial Cognition*, volume 1404 of *Lecture Notes in Computer Science*, pp 293–312. Springer Berlin Heidelberg.
- [Freeman, 1975] Freeman, J. (1975). The modelling of spatial relations. *Computer Graphics and Image Processing*, 4(2) :156–171.
- [Freksa, 1992] Freksa, C. (1992). Using orientation information for qualitative spatial reasoning. In Andrew, U. F., Compari, I., et Ubaldo, F., éditeurs, *Spatio-Temporal Reasoning*, volume 639, pp 162–178, Pisa, Italy,.
- [Gadia et Vaishnav, 1985] Gadia, S. K. et Vaishnav, J. H. (1985). A query language for a homogeneous temporal database. In *In Proceedings of the ACM Symposium on Principles of Database Systems*, pp 51–56, Portland, Oregon, (USA). ACM.
- [Genesereth et Nilsson, 1987] Genesereth, M. R. et Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers, inc., Standford, University.
- [Geraldine *et al.*, 2010] Geraldine, D. M., John, G., Claramunt, C., et Thibaud, R. (2010). A graph model for spatio-temporal evolution. *Journal of Universal Computer Science*, 16(11) :1452–1477.
- [Gibson, 1977] Gibson, J. J. (1977). *The Ecological Approach to Visual Perception*, chapter The Theory of Affordances, pp 67–82. Lawrence Erlbaum Associates.
- [Goodchild, 1990] Goodchild, M. F. (1990). Tiling large geographical databases. In B., A. P., Oliver, G., Terence, R., et Yuan-Fang, W., éditeurs, *Proceedings of the First Symposium on Design and Implementation of Large Spatial Databases*, volume 409 of *SSD '89*, pp 137–146, London, UK, UK. Springer-Verlag.

- [Goralski et Gold, 2008] Goralski, R. et Gold, C. (2008). Marine gis : Progress in 3d visualization for dynamic gis. In Ruas, A. et Gold, C. M., éditeurs, *Spatial Data Handling*, pp 401–416, Montpellier, France. Springer.
- [Grosf *et al.*, 2003] Grosf, B. N., Horrocks, I., et Volz, R. (2003). Description logic programs : Combining logic programs with description logic cambridge. In Hencsey, G., white, B., Chen, R., Kovács, L., et Lawrence, S., éditeurs, *WWW2003*, pp 20–24, Budapest, Hungary. ACM.
- [Groth, 1998] Groth, R. (1998). *Data Mining : Hands-On Approach for Business Professionals*. Rapport technique, Prentice Hall, New Jersey.
- [Gruber, 1993] Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2) :199–220.
- [Guarino *et al.*, 2009] Guarino, N., Oberle, D., et Staab, S. (2009). *Handbook on Ontologies*, chapter What Is an Ontology. Springer-Verlag Berlin Heidelberg 2.
- [Guttman, 1984a] Guttman, A. (1984a). R-trees : A dynamic index structure for spatial searching. *SIGMOD Rec.*, 14(2) :47–57.
- [Guttman, 1984b] Guttman, A. (1984b). Rtrees : A dynamic index structure for spatial searching. In Yormark, B., éditeur, *SIGMOD84, Proceedings of Annual Meeting, Boston, Massachusetts*, pp 47–57. ACM Press.
- [Hiemstra, 2001] Hiemstra, D. (2001). *Using Language Models for Information Retrieval*. Rapport de thèse de doctorat, Taaluitgeverij Neslia Paniculata.
- [Hoffart *et al.*, 2013] Hoffart, J., Suchanek, F., Berberich, B., et Weikum, G. (2013). Yago2 : A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence Journal*, 194 :28–61. <http://www.mpi-inf.mpg.de/yago-naga/yago/publications/aij.pdf>.
- [Holland, 1975] Holland, J. (1975). *Adaptation in Natural and Artificial System*. University of Michigan Press, Cambridge, MA, USA.
- [Hornsby et Egenhofer, 2000] Hornsby, K. et Egenhofer, M. (2000). A foundation for spatio-temporal knowledge representation. *International Journal of Geographical Information Science*, 14 :207–224.
- [IHO, 2000] IHO (2000). *S-57 Appendix A IHO Object Catalogue*. International Hydrographic Organization.
- [IHO, 2011] IHO (2011). *Recommended ENC Validation Checks*. International Hydrographic Bureau, MONACO.
- [Jankowski et Nyerges, 2003] Jankowski, P. et Nyerges, T. (2003). *Geographic Information Systems for Group Decision Making : Towards a Participatory, Geographic Information Science*. London and NewYork.
- [Jianping Xu, 1994] Jianping Xu, R. G. L. J. (1994). Improving cost-path tracing in a raster data format. *Computers & Geosciences*, 20(10) :1455–1465.

- [Kaindl et Kainz, 1997] Kaindl, H. et Kainz, G. (1997). Bidirectional heuristic search reconsidered. *Journal of Artificial Intelligence Research*, 7 :283–317.
- [Karp, 1972] Karp, R. M. (1972). Reducibility among combinatorial problems. *Complexity of Computer Computations*, pp 85–103.
- [Kolas et al., 2009] Kolas, D., Emmons, I., et Dean, M. (2009). Efficient linked-list rdf indexing in parliament. In *ISWC*, Washington, DC. BBN Technologie, Arlington, VA 22209, USA.
- [Korf, 1985] Korf, R. E. (1985). Iterative-deepening-A : an optimal admissible tree search. In Joshi, A., éditeur, *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2, pp 1034–1036, Los Angeles, CA,. Morgan Kaufmann Publishers Inc.
- [Koubarakis et al., 2012] Koubarakis, M., Karpathiotakis, M., Kyzirakos, K., Nikolaou, C., et Sioutis, M. (2012). Data models and query languages for linked geospatial data. In Eiter, T. et Krennwallner, T., éditeurs, *Reasoning Web. Semantic Technologies for Advanced Query Answering*, volume 7487 of *Lecture Notes in Computer Science*, pp 290–328. Springer Berlin Heidelberg, Vienna, Austria.
- [Kwa, 1989] Kwa, J. (1989). Bs* : an admissible bidirectional staged heuristic search algorithm. *Artificial Intelligence*, 38(1) :95–109.
- [Kyzirakos et al., 2012] Kyzirakos, K., Karpathiotakis, M., et Koubarakis, M. (2012). Strabon : A semantic geospatial dbms. In Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J., Hendler, J., Schreiber, G., Bernstein, A., et Blomqvist, E., éditeurs, *The Semantic Web - ISWC*, volume 7649 of *Lecture Notes in Computer Science*, pp 295–311, Boston, MA, USA,. Springer Berlin Heidelberg.
- [Lafourcade et al., 2002] Lafourcade, M., Prince, V., et Schwab, D. (2002). Vecteurs conceptuels et structuration émergente de terminologie. *Traitement Automatiques des Langues*, 43(1) :43–72.
- [Landragin, 2004] Landragin, F. (2004). Saillance physique et saillance cognitive. *CORRELA*, 2(2) :7–34.
- [Langran, 1992] Langran, G. (1992). States, events, and evidence : The principle entities of a temporal gis. In *GIS/LIS'92*, pp 416–425. Bethesda, MD, American Congress of Surveying and Mapping.
- [Langran et Chrisman, 1988] Langran, G. et Chrisman, N. (1988). A framework for temporal geographic information. *Cartographical*, 3 :1–14.
- [Lehmann, 1992] Lehmann, F. (1992). *Semantic Networks in Artificial Intelligence*. Elsevier Science Inc, New York, NY, USA.
- [Maedche et al., 2001] Maedche, A., Staab, S., Nedellec, C., et Hovy, H., éditeurs (2001). *Maritime Information Markup and Use in Passage Planning*, volume 38 of *CEUR Workshop Proceedings*, Seattle, USA. CEUR-WS.org.
- [Malyanka, 2002] Malyanka, R. (2002). Vocabulary development for markup languages — a case study with maritime information. In Lassner, D., Roure, D., et Iyengar, A., éditeurs, *WWW2002*, pp 674–685, Honolulu, Hawaii, USA. ACM.

- [Malyankar, 1999] Malyankar, R. M. (1999). Creating a navigation ontology. In *Workshop on Ontology Management*, Orlando, FL,. AAAI-99.
- [Malyankar, 2004] Malyankar, R. M. (2004). Representation and distribution of geospatial knowledge. In *Proceedings of the 2004 Annual National Conference on Digital Government Research*, Seattle, Washington, USA. DG.O, Digital Government Society of North America.
- [Manzini, 1995] Manzini, G. (1995). BIDA* : An improved perimeter search algorithm. *Artificial Intelligence*, 75(2) :347–360.
- [McCarthy, 2000] McCarthy, J. (2000). Concept of logical ai. <http://www-formal.stanford.edu/jmc/whatisai/whatisai.html>.
- [McCarthy, 2007] McCarthy, J. (2007). What is artificial intelligence? <http://www-formal.stanford.edu/jmc/whatisai/whatisai.html>.
- [Minsky, 1974] Minsky, M. (1974). *A Framework for Representing Knowledge*. Rapport technique, MIT-AI Laboratory Memo 306.
- [Moore, 1957] Moore, E. (1957). The shortest path through a maze. In *Proceedings of the International Symposium on Theory of Switching Cambridge*. Harvard University Press.
- [Motschnig-Pitrik et Kaasbøll, 1999] Motschnig-Pitrik, R. et Kaasbøll, J. (1999). Part-whole relationship categories and their application in object-oriented analysis. *IEEE Trans. on Knowl. and Data Eng.*, 11(5) :779–797.
- [Negnevitsky, 2005] Negnevitsky, M. (2005). *Artificial Intelligence : A Guide to Intelligent Systems (2nd Edition)*. Hardcover.
- [Néméta, 2008] Néméta, A. (2008). *Code Vagnon Permis Plaisance : Option cotière*. Vagnon, 26, rue Jacob - 75006 PARIS.
- [Nicholson, 1966] Nicholson, T. A. J. (1966). Finding the shortest route between two points in a network. *The Computer Journal*, 9(3) :275–280.
- [Niveleau, 2006] Niveleau, C. E. (2006). Le concept gibsonien d'áffordance : entre filiation rupture et reconstruction conceptuelle. *Intellectica*, 44 :159–199.
- [NOAA, 2011a] NOAA (2011a). *Navigation Rules : International-Inland*. U.S. Department of Transportation, United States Coast Guard, 2100 Second St., SW United States Coast Guard Washington, DC 20593-0001 Staff Symbol : G-MW.
- [NOAA, 2011b] NOAA (2011b). *United States Coast Pilot*. National Oceanic and Atmospheric Administration, US Department of Commerce, 44 edition.
- [Ogden et Richard, 1923] Ogden, C. et Richard, I. (1923). *The Meaning of the Meaning*. New York, Harcourt, Brace and Word, Inc., 8 edition.
- [Pan, 2005] Pan, F. (2005). Temporal aggregates for web services on the semantic web. In *ICWS*, pp 831–842, Orlando, FL, USA. IEEE Computer Society.
- [Parkes et Thrift, 1980] Parkes, D. et Thrift, N. (1980). *Times, Spaces and Places : A Chronogeographic Perspective*. J. Willey.

- [Pearson, 2008] Pearson, M. (2008). *Mémento Vagnon du Skipper : Moteur et voile*.
- [Perry et Herring, 2011] Perry, M. et Herring, J. (2011). *GeoSPARQL - A geographic Query Language for RDF data A Proposal for an OGC Draft Candidate Standard*. Open Geospatial Consortium.
- [Perry et Herring, 2012] Perry, M. et Herring, J. (2012). *OGC Geosparql - A Geographic Query Language for RDF Data*. Open Geospatial Consortium.
- [Pohl, 1971] Pohl, I. (1971). Bi-directional search. In Meltzer et Michie, éditeurs, *Machine Intelligence*, pp 127–140. Edinburgh University Press.
- [Potthast et al., 2008] Potthast, M., Stein, B., et Anderka, M. (2008). A Wikipedia-based multilingual retrieval model. In Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., et White, R. W., éditeurs, *Proceedings of the 30th European Conference on IR Research, ECIR 2008, Advances in Information Retrieval*, volume 4956 of *LNCS*, pp 522–530, Berlin/Heidelberg. Springer.
- [Pottmann et al., 2007a] Pottmann, H., Brell-Cokcan, S., et Wallner, J. (2007a). Discrete surfaces for architectural design. In Chenin, P., Lyche, T., et Schumaker, L. L., éditeurs, *Curves and Surfaces : Avignon 2006*, pp 213–234. Nashboro Press.
- [Pottmann et al., 2007b] Pottmann, H., Liu, Y., Wallner, J., Bobenko, A., et Wang, W. (2007b). Geometry of multi-layer freeform structures for architecture. *ACM Transactions on Graphics - Proceedings of ACM SIGGRAPH 2007*, 26(3).
- [Pulido et al., 2012] Pulido, F. J., Mandow, L., et Pérez de la Cruz, J. L. (2012). A two-phase bidirectionnal heuristic search algorithm. In *Proceedings of the Sixth Starting AI Researcher's Symposium (STAIRS)*, pp 240–251.
- [Quillan, 1967] Quillan, M. R. (1967). A theorie and simulation of some basic semantic capabilities. *Behaviour Science*, 12 :410–430.
- [Randell et al., 1992] Randell, D., Cui, Z., et Cohn, A. (1992). A spatial logic based on regions and connection. In Nebel, B., Rich, C., et Swartout, W., éditeurs, *Principles of Knowledge Representation and Reasoning : Proceedings of the Third International Conference*, pp 165–176, San Mateo, California. Morgan Kaufmann.
- [Raza, 2001] Raza, A. (2001). *Object-Oriented Temporal GIS for Urban Applications*. Rapport de thèse de doctorat, University of Twente and ITC.
- [Rousselet et Fabre-Thorpe, 2003] Rousselet, G. A. et Fabre-Thorpe, M. (2003). Les mécanismes de l'attention visuelle. *Psychologie Française*, 48 :29–44.
- [Russel et Norvig, 1995] Russel, J. et Norvig, P. (1995). *Artificial Intelligence a Modern Approach (3rd Edition)*. Prentice Hall , Englewood Cliffs, New Jersey 07632.
- [Sagonas et al., 1994] Sagonas, K., Swift, T., et Warren, S. (1994). XSB as an efficient deductive database engine. *SIGMOD*, 23(2) :442–453.
- [Sahr, 2011] Sahr, K. (2011). Hexagonal discrete global grid systems for geospatial computing. *Hexagonal Discrete Global Grid Systems for Geospatial Computing*, 22 :363–376.

- [Sahr *et al.*, 2003] Sahr, K., White, D., et Kimerling, J. A. (2003). Geodesic discrete global grid systems. *Cartography and Geographic Information Science*, 30(2) :121–134.
- [Salton et MacGill, 1983] Salton, G. et MacGill, M. (1983). *Introduction to Modern Information Retrieval*.
- [Samet et Webber, 1985] Samet, H. et Webber, R. E. (1985). Storing a collection of polygons using quadtrees. *ACM Trans. Graph.*, 4(3) :182–222.
- [Schmid, 2012] Schmid, H. (2012). Treetagger - a language independent part-of-speech tagger. <http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>.
- [Schmidt-Schauß, 1989] Schmidt-Schauß, M. (1989). Subsumption in KL-ONE is undecidable. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pp 421–431, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Schmolze *et al.*, 1985] Schmolze, J., Beranek, B., et Inc, N. (1985). An overview of the kl-one knowledge representation system. *Cognitive Science*, 9 :171–216.
- [Schwalb et Vila, 1998] Schwalb, E. et Vila, L. (1998). Temporal constraints : A survey. *Constraints*, 3(2-3) :129–149.
- [Sellis et Faloutsos, 1987] Sellis, T.K. and Roussopoulos, N. et Faloutsos, C. (1987). The r+ - tree : A dynamic index for multi-dimensional objects. In Stocker, M., Kent, W., et Hammersley, P., éditeurs, *VLDB'87, Proceedings of 13th International Conference on Very Large Data Bases.*, Brighton, England.
- [Simone, 2011] Simone, M. (2011). Repenser la notion d'affordance dans ses dynamiques sémiotiques. *Intellectica*, 55 :1–55.
- [Sinclair, 1990] Sinclair, J. M. (1990). *Collins COBUILD essential English dictionary*.
- [Sint et de Champeaux, 1977] Sint, L. et de Champeaux, D. (April 1977). An improved bidirectional heuristic search algorithm. *Journal of the ACM (JACM)*, 24(2) :177–191.
- [Sintek *et al.*, 2002] Sintek, M., Decker, S., et Harth, A. (2002). Triple - a query, inference, and transformation language for the semantic web. In Horrocks, I. et Hendler, J., éditeurs, *In International Semantic Web Conference (ISWC)*, volume 2342 of *Lecture Notes in Computer Science*, Sardinia, Italy. Springer. <http://triple.semanticweb.org/>.
- [Snodgrass et Ahn, 1986] Snodgrass, R. et Ahn, I. (1986). Temporal databases. *IEEE Computer*, 19(9) :35–42.
- [Sorrows et Hirtle, 1999] Sorrows, M. et Hirtle, S. (1999). *The Nature of Landmarks for Real and Electronic Spaces*. Freksa et Mark.
- [Spéry *et al.*, 2001] Spéry, L., Claramunt, C., et Libourel, T. (2001). A spatio-temporal model for the manipulation of lineage metadata. *Geoinformatica*, 5 :51–70.
- [Strzalkowski *et al.*, 1994] Strzalkowski, T., Carballo, J., et Marinescu, M. (1994). Natural language information retrieval : Trec-3-report. In Strzalkowski, T., Carballo, J., et Marinescu, M., éditeurs, *Proceedings of the third Text Retrieval Conference*.

- [Studer *et al.*, 1998] Studer, R., Benjamins, R., et Fensel, D. (1998). Knowledge engineering : Principles and methods. *Data & Knowledge Engineering*, 25 :161–197.
- [Tansel *et al.*, 1993] Tansel, A. U. and Clifford, J., Gadia, S., Jajodia, S., Segev, A., et Snodgrass, R. (1993). *Temporal Databases : Theory, Design, and Implementation*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA.
- [Tong *et al.*, 2013] Tong, X., Ben, J., Wang, Y., Zhang, Y., et Pei, T. (2013). Efficient encoding and spatial operation scheme for aperture 4 hexagonal discrete global grid system. *International Journal of Geographical Information Science*, 27(5) :898–921.
- [Tsatcha *et al.*, 2013a] Tsatcha, D., Saux, E., et Claramunt, C. (2013a). A bi-directional path-finding algorithm and data structure for maritime routing. *International Journal of Geographical Information Science(IJGIS)*, 28(7) :1355–1377., http://www.tandfonline.com/eprint/Ud6sKz55G4UNQhp6QcFr/full#.U234xPl_v09.
- [Tsatcha *et al.*, 2013b] Tsatcha, D., Saux, E., et Claramunt, C. (2013b). A modeling approach for the extraction of semantic information from a maritime corpus. In Timpf, S. et Laube, P., éditeurs, *Advances in Spatial Data Handling*, Advances in Geographic Information Science, pp 175–191, Colone, Germany. Springer Berlin Heidelberg.
- [Tsatsaronis et Panagiotopoulou, 2009] Tsatsaronis, G. et Panagiotopoulou, V. (2009). A generalized vector space model for text retrieval based on semantic relatedness. The 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL-09).
- [Turing, 1950] Turing, A. (1950). Computing machinery and intelligence. *Mind*, 59 :433–460.
- [U.S Coast Guard, 1999] U.S Coast Guard (1999). *Navigation Rules : International-Inland*. Technical report, U.S. Department of Transportation, United States Coast Guard, 2100 Second St., SW United States Coast Guard Washington, DC 20593-0001 Staff Symbol : G-MW.
- [Woods et Schmolze, 1992] Woods, W. et Schmolze, J. (1992). The kl-one family. *Computers & Mathematics with Applications*, 23 :133–177.
- [Worboys, 1994] Worboys, F. (1994). A unified model for spatial and temporal information. *The Computer Journal*, 37(1) :36–34.
- [Yap, 2002] Yap, P. (2002). Grid-based path-finding. In Robin Cohen, R. et Spencer, B., éditeurs, *Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence*, volume 2338, pp 44–45, Calgary, Canada., Springer.
- [Yousefi et Donohue, 2004] Yousefi, A. et Donohue, G. L. (2004). Temporal and spatial distribution of airspace complexity for air traffic controller workload-based sectorization. In *Proceedings of the 4th American Institute of Aeronautics and Astronautics (AIAA) Aviation Technology, Integration and Operations Forum, Chicago, IL*.
- [Yu *et al.*, 2003] Yu, C., Lee, J., et Munro-stasuik, M. J. (2003). Research article : Extensions to least-cost path algorithms for roadway planning. *International Journal of Geographical Information Science*, 17(4) :361–376.

-
- [Yuan, 1999] Yuan, M. (1999). Use of a three-domaine representation to enhance gis support for complex spatio-temporal queries. *Transactions in Gis*, 3 :137–159.
- [Zadeh, 1965] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8 :338–353.

Contribution à l'extraction et à la représentation des connaissances de l'environnement maritime. Proposition d'une architecture dédiée aux applications de navigation

Résumé :

De nos jours, les applications informatiques autonomes sont au centre de grandes préoccupations de la recherche scientifique. Ces dernières sont destinées initialement à des systèmes d'aide à la décision dans des environnements contraints et dynamiques, communément appelés environnements complexes. Elles peuvent dès à présent, à l'aide des avancées de la recherche, permettre de construire et déduire leurs connaissances propres afin d'interagir en temps réel avec leur environnement. Cependant, elles sont confrontées à la difficulté d'avoir une modélisation fidèle du monde réel et des entités qui le composent. L'un des principaux objectifs de nos recherches est de capturer et modéliser la sémantique associée aux entités spatio-temporelles afin d'enrichir leur expressivité dans les SIG ou les systèmes d'aide à la décision. Un service de routage maritime dynamique a été déployé en exploitant cette modélisation. Cet algorithme a été démontré comme optimal en termes d'espace mémoire et de temps de calcul.

La sémantique capturée se compose de l'affordance et de la saillance visuelle de l'entité spatiale. Les connaissances associées à cette sémantique sont par la suite représentées par une ontologie computationnelle qui intègre des approches spatio-temporelles. Ces connaissances sont soit déduites du savoir de l'expert du domaine, soit extraites de gros volumes de données textuelles en utilisant des techniques de traitement automatique du langage.

L'ontologie computationnelle proposée nous a permis de définir un algorithme de routage maritime dynamique (fonction des événements ou objets présents dans l'environnement) fondé sur une heuristique itérative monocritère de plus courte distance et bidirectionnelle. L'algorithme BIDA* proposé s'applique sur un graphe itératif qui est une conceptualisation d'une grille hexagonale itérative recouvrant la zone de navigation. Cet algorithme permet aussi la gestion de différents niveaux de résolution.

Toujours dans l'initiative de produire un modèle aussi proche que possible du monde réel, l'algorithme BIDA* a été enrichi des stratégies multicritères afin de prendre en compte les différentes contraintes de la navigation maritime. Les contraintes globales et locales auxquelles nous nous sommes intéressés sont la profondeur des eaux, la distance de navigation et la direction de navigation.

Le modèle proposé permet ainsi d'enrichir les capacités cognitives des utilisateurs évoluant dans les environnements maritimes et peut aussi être utilisé pour construire des systèmes complètement autonomes explorant ces environnements. Un prototype expérimental de navigation intelligente mettant en œuvre cette modélisation et proposant un service de routage maritime a été développé dans le cadre de cette thèse.

Mots clés : Intelligence artificielle, sciences de l'information géographique, ontologie, traitement automatique du langage, intelligence, routage maritime, géométrie combinatoire, algorithmique combinatoire, systèmes de gestion de bases de données (relationnelles et ontologiques).