



Représentation de la connaissance spatiale pour la planification

Lamia Belouaer

► To cite this version:

Lamia Belouaer. Représentation de la connaissance spatiale pour la planification. Intelligence artificielle [cs.AI]. Université de Caen, 2011. Français. NNT: . tel-01107561

HAL Id: tel-01107561

<https://hal.science/tel-01107561>

Submitted on 21 Jan 2015

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

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

U.F.R. : Sciences

ÉCOLE DOCTORALE : SIMEM

THÈSE

présentée par

Lamia BELOUAER

et soutenue

le 09 Décembre 2011

en vue de l'obtention du

DOCTORAT de l'UNIVERSITÉ de CAEN

spécialité : Informatique et applications

(Arrêté du 7 août 2006)

Représentation de la connaissance spatiale pour la planification

MEMBRES du JURY

Mme Florence SÈDES	Professeur des Universités	IRIT, Toulouse	(rapporteur)
Mr Malek MOUHOU	Professeur des Universités	Université de Régina	(rapporteur)
Mme Anne BOYER	Professeur des Universités	KIWI, Université de Nancy	
Mr Cyrille BERTELLE	Professeur des Universités	Université du Havre	
Mme Maroua BOUZID	Professeur des Universités	Université de Caen	(directrice)
Mr Abdel-illah MOUADDIB	Professeur des Universités	Université de Caen	(co-directeur)

Mis en page avec la classe thloria.

Remerciements

Comme le veut la tradition, je vais tenter de satisfaire au difficile exercice de la page des remerciements, peut-être la tâche la plus difficile de ces années de thèse. Non qu'exprimer ma gratitude envers les personnes en qui j'ai trouvé un soutien soit contre ma nature, bien au contraire. La difficulté tient plutôt dans le fait de n'oublier personne. C'est pourquoi, je remercie par avance ceux dont le nom n'apparaît pas dans cette page et qui m'ont aidé d'une manière ou d'une autre. Ils se reconnaîtront.

Je tiens à remercier, dans un premier temps, mes directeurs de thèse Maroua BOUZID et Abdel-illah MOUADDIB pour leur encadrement et pour la confiance et la liberté qu'ils m'ont accordé. Bien que les débuts de la thèse furent difficile, je me suis vraiment amusée à travailler sur ce sujet. Le travail fourni durant cette thèse m'a permis de toucher à différentes thématiques, ce que j'ai trouvé très intéressant.

Ensuite, je tiens à remercier mon jury d'avoir répondu présent. Je remercie particulièrement mes rapporteurs Florence SÈDES et Malek MOUHOU. Grâce à vos remarques j'ai pu identifier certains points qui permettront d'améliorer ce travail. Je remercie également Anne BOYER et Cyrille BERTELLE pour leur participation.

Je remercie les membres de l'équipe MAD, et en particulier Bruno, Laurent, Laetitia, de leurs disponibilités et leur conseils pour répondre à toutes mes questions. Plus que des collègues au Greyc, ce sont mes amis. Je remercie Jean-Philippe, Cyril, Simon et Hugo de m'avoir initiée au jorky Ball (pendant des semaines j'avais des marques témoignant que j'ai été la bienvenue). Merci à toi Matthieu(B) de m'avoir initiée au DDR ce qui m'a permis de redécouvrir certains morceaux de musique que je n'avais jamais appréciée. Merci à Céline et Eugénie pour les moments entre filles (même si à la fin ils devenaient rares). Je suis reconnaissante à Gtalk et FB de me permettre de rester en contact avec Elsa, Sandra et Guigui. Arnaud, Boris, Nicolas(C), Guillaume(L), Benoît et Mathieu(F), merci d'avoir fait de moi votre « Padawan » (Vous m'avez tellement appris sur la vie ...). Merci à Romain pour son « coaching » (professionnel, mais surtout humain) durant ses trois années et en particulier pendant la phase de rédaction et préparation de la soutenance. Et merci à Grégory d'avoir accepté de relire mon état de l'art. Et surtout d'avoir toujours été encourageant et réconfortant.

Plus personnellement, je tiens à remercier mes professeurs de danse : Laurence et Marion, qui m'ont permis d'oublier un peu la thèse et de mieux m'épanouir dans ce que je fais. Je tiens

également à remercier des personnes qui me sont chères et qui ont toujours sut m'apporter un réconfort : Mado, Nicole, Medi, Hassen, Fouzi et Mohssen.

Je tiens finalement à remercier mes amis les plus proches qui m'ont toujours soutenue : Samia, Aymen, Dhia, Abir, Rym, Salma, Guillaume(Y), Mohamed (Botti) et Hatem. De tout mon cœur un grand MERCI à mes parents dont le soutien, l'amour et la confiance m'ont permis d'en être là où je suis aujourd'hui. Merci à Amani et Rim, les sœurs que j'aime. Et pour finir, Merci à Nicolas qui a su s'adapter à mes états d'âme sans se plaindre. Tu as su me donner le courage et la volonté nécessaire pour avancer.

Benoît : *La thèse m'a permise de me rendre compte à quel point, Toi Benoît m'est important dans le travail et dans la vie. Merci de m'avoir écoutée, soutenue, réconfortée, grondée.*

Romain : *Je te suis reconnaissante, t'as su être un super Coach, même si tu n'avais pas le temps. C'est grâce à toi que j'ai eu un papier nommé à un prix dans une conférence.*

Arnaud : *Arnaud tu as su m'apprendre à avoir confiance en moi. Merci.*

Bobo : *Boris, j'ai adoré toutes nos discussions et j'ai hâte de (re)partager encore une fois une chambre d'hôtel avec toi.*

Prospect : *Comme j'ai joué au foot avec toi, un jour tu danseras la Zumba avec moi.*

Guillaume (L) : *Une chanson : « Je Ne Veux Pas Travailler » de Edith Piaf.*

Abir : *Que dire à part MERCI d'être mon amie. Sans ton soutien, ta présence, je n'aurai pas supporté plein de choses. Encore merci.*

Table des matières

Table des figures	xi
Liste des tableaux	xv
Introduction	1
Introduction	3
I État de l’art	7
Introduction de la partie I	9
1 Planification	11
1.1 Planification en intelligence artificielle	12
1.1.1 Généralités	12
1.1.2 Planification classique	13
1.2 Planification de tâches	15
1.2.1 Représentations des états	15
1.2.2 Représentation des opérateurs et des actions	16
1.2.3 Représentation d’un problème de planification	17
1.3 Algorithmes et techniques de planification	17
1.3.1 Planification dans un espace d’états	18
1.3.2 Techniques de planification	18
1.4 Langages de planification	20
1.4.1 Historique	21
1.4.2 Langage PDDL	22
1.4.3 Utilisation de PDDL	23
1.5 Intégration de la connaissance spatiale dans la planification	25

1.5.1	Planificateurs de chemins	25
1.5.2	Couplage de planificateurs	26
1.5.3	Formalisation de la connaissance spatiale	27
2	Représentation et raisonnement spatial	31
2.1	Connaissance spatiale	32
2.1.1	Représentation spatiale	32
2.1.2	Raisonnement spatial	33
2.2	Représentation d'une entité spatiale	34
2.2.1	Représentation géométrique	34
2.2.2	Représentation hiérarchique	34
2.3	Représentation des relations spatiales	35
2.3.1	Relations topologiques	36
2.3.2	Relation de distance	38
2.3.3	Cadre de référence	39
2.4	Approches quantitatives pour le raisonnement spatial	41
2.4.1	Modèle raster/Modèle vectoriel	41
2.4.2	Types primitifs	42
2.5	Approches pour le raisonnement spatial qualitatif	42
2.5.1	Formalismes inspirées des relations temporelles d'Allen	43
2.5.2	Formalismes fondés sur d'autres techniques	49
2.6	Connaissance spatiale en planification	51
2.6.1	Carte métrique, carte topologique, carte hybride et grille d'occupation	52
2.6.2	Carte sémantique	53
3	Ontologies	57
3.1	Qu'est ce qu'une ontologie?	58
3.1.1	Réseau sémantique	58
3.1.2	Types d'ontologies	59
3.1.3	Utilité des ontologies	60
3.1.4	Construction d'une ontologie	61
3.2	Logiques de description	62
3.2.1	Syntaxe des logiques de description	62
3.2.2	Niveaux de descriptions	63
3.2.3	Inférence dans les logiques de description	65
3.2.4	Logique minimale $\mathcal{AL}$	66
3.3	Outils pour les ontologies	68

3.3.1	Langages	68
3.3.2	Outils	73
3.4	Connaissance spatiale dans les ontologies	74
3.4.1	Ontologies spatiales	75
3.4.2	Modélisation des entités spatiales	76
3.4.3	Modélisation des relations spatiales	77
3.5	Planification et ontologie spatiale	79
3.5.1	Ontologie spatiale au cœur du processus de planification	79
3.5.2	Ontologie spatiale et le bon fonctionnement du processus de planification	80
Conclusion de la partie I		83
II Contributions		85
Introduction de la partie II		87
4	Connaissance spatiale : <i>SpaceOntology</i>	89
4.1	Définition d'une ontologie spatiale	90
4.1.1	Entité spatiale	92
4.1.2	Relations topologiques	94
4.1.3	Relations de distances	96
4.1.4	Relations entre entités spatiales	99
4.2	Raisonnement avec <i>SpaceOntology</i>	102
4.2.1	Raisonnement hiérarchique	102
4.2.2	Raisonnement avec les relations de distance	103
4.2.3	Raisonnement avec les relations topologiques	105
4.3	Mécanisme de composition	107
4.3.1	Représentation des relations topologiques	107
4.3.2	Décomposition des relations topologiques	108
4.3.3	Calcul de la composition	110
4.3.4	Consistance de la composition	111
4.3.5	Maintenir la cohérence pour les relations topologiques	113
5	Modèle et langage pour la planification spatiale	117
5.1	Planification spatiale	118
5.1.1	Approche proposée : Architecture SPOON	119
5.1.2	Raisonnement dans SPOON	120
5.1.3	Aperçu du processus de planification de SPOON	122

5.2	Fonctionnement de SPOON	122
5.2.1	Information spatiale dans le processus de la planification	122
5.2.2	Opérateurs de planification spatiale	127
5.3	De PDDL à Spatial-PDDL	130
5.3.1	Définition des objets dans Spatial-PDDL	132
5.3.2	Définition d'un domaine de planification avec Spatial-PDDL	132
5.3.3	Définition d'un problème de planification avec Spatial-PDDL	137
6	Processus de planification avec des informations spatiales	141
6.1	Exemple d'application	141
6.2	Processus de planification : principes	142
6.3	Description avec Spatial-PDDL	144
6.4	Planification de tâches	145
6.5	Planificateur spatial hiérarchique et floue	146
6.5.1	Génération du graphe de réseaux de passages	146
6.5.2	Définition du chemin vers une cible floue	154
	Conclusion de la partie II	159
III	Expérimentations	161
7	Performances de SPOON	163
7.1	Inférence de l'adjacence	164
7.1.1	Principe	164
7.1.2	Résultats expérimentaux	165
7.1.3	Étude de la complexité	165
7.2	Cohérence de la connaissance spatiale : justification du choix AC-3	165
7.3	Performances du planificateur de chemins	167
7.3.1	Performances du principe de la fonction <i>DijkstraMC</i>	167
7.3.2	Intérêt de la hiérarchisation de l'environnement	168
7.3.3	Recherche de chemins dans un graphe de réseaux de passages	168
7.4	Planification de tâches : étude avec HTN	170
7.5	Performances de SPOON	171
7.5.1	Rôle de la connaissance spatiale dans la planification	171
7.5.2	Comparaison avec d'autres planificateurs	172
7.6	Système complet de la planification	173

Conclusion et perspectives	177
Conclusion et perspectives	179
Annexes	181
Bibliographie	185

Table des figures

1	Problème d'interaction entre un humain et un robot.	4
1.1	Monde des blocs : la représentation graphique des états initial et but	13
1.2	Modèle de planification.	13
1.3	Construction du graphe de planification.	19
1.4	Problème de planification : un robot qui ramasse des blocs	23
2.1	Conséquence d'une mauvaise exploitation de la connaissance spatiale	32
2.2	Exemple d'abstraction en utilisant un AH-graph	36
2.3	Différents types de relations topologiques	37
2.4	Exemple de cadres de référence	39
2.5	Cadres de référence : absolu, intrinsèque et relatif	40
2.6	Hierarchie de types spatiaux dans le standard de l'OGC, extrait de [MIRON, 2009]	42
2.7	Définition d'une relation topologique $\langle m, b \rangle$	44
2.8	Composition entre deux relations topologiques définies par l'algèbre des rectangles.	45
2.9	Scène et Réseau de rectangles	46
2.10	Réseau simplifié des autoroutes de Boston	48
2.11	Relations spatiales topologiques RCC-8.	50
2.12	Exemple de CSP non complet et complet	51
3.1	Exemple de réseau sémantique.	59
3.2	Une base de connaissance de la famille.	64
3.3	Un graphe RDF.	69
3.4	Représentation graphique de Harry Potter qui est inscrit à Poudlard.	69
3.5	Représentation graphique de GeoRSS-Simple [École d'été GDR MAGIS-2009].	76
3.6	Deux niveaux des relations de direction	78
1	Représentation graphique de nos contributions.	88
4.1	Représentation d'un immeuble B_1 avec plusieurs étages.	90
4.2	Deux plans d'étage dans un immeuble	90
4.3	Représentation graphique de <i>SpaceOntology</i>	91

4.4	Représentation graphique des concepts Space et Regions	92
4.5	Représentation géométrique d'une entité spatiale.	93
4.6	Arbre hiérarchique d'un bâtiment et les différents niveaux hiérarchiques	94
4.7	Représentation graphique de l'organisation hiérarchique	94
4.8	Algèbre ABLR	95
4.9	Représentation graphique du concept AxisRelations	96
4.10	Représentation graphique du concept TopologicalRelations	97
4.11	Représentation graphique du concept DistanceRelations	98
4.12	Distance euclidienne	98
4.13	Découpage d'une région en quatre niveaux de proximité	99
4.14	Représentation graphique du concept SpatialRelations	100
4.15	Liens entre concepts et leurs instanciations.	100
4.16	Représentation graphique du concept HasRelation	101
4.17	Définition de zone de proximité selon Hall [Hall, 1966].	104
4.18	Évaluation de la distance	105
4.19	Relations topologiques et composition.	106
4.20	Squelette d'une règle de composition	106
4.21	Scène et graphe de relations associé	108
4.22	Représentation graphique d'un espace de deux manières	109
4.23	Graphe de relations sur un seul axe.	109
4.24	Règle de composition pour un seul axe	110
4.25	Réseau de contraintes modélisant la connaissance topologique.	114
4.26	Réseau de contraintes modélisant les liens topologiques.	114
5.1	Définition d'une mission pour le robot par l'humain.	118
5.2	Architecture du planificateur spatial.	120
5.3	Représentation graphique des différents modules de SPOON.	121
5.4	Illustration des deux définitions 21 et 22.	124
5.5	Illustration des deux propriétés 12 et 13.	124
6.1	Complexe de bâtiments.	142
6.2	Plan partiel du premier étage dans le bâtiment <i>B1</i>	148
6.3	Modélisation d'un environnement.	148
6.4	Requête pour définir la zone d'intérêt hiérarchique	150
6.5	Requête utilisée dans la fonction définir	150
6.6	Suppression des impasses.	151
6.7	Définition d'un arc dans le réseau de passages abstrait Γ^{L_l}	151
6.8	Exemple de graphe de réseaux de passages	153
6.9	Calcul pour la sélection d'un nœud.	155
6.10	Chemin solution hiérarchique.	155

6.11	Application de la règle	157
7.1	Adjacence définies de deux manières	164
7.2	Comparaison entre les méthodes de déduction	166
7.3	Intérêt de l'organisation hiérarchique et de l'élagage	168
7.4	Décomposition de l'action go	170
7.5	Temps de calcul d'un plan par rapport aux nombres d'instances.	171
7.6	Représentation graphique du système complet	173
7.7	Capture d'écran du simulateur.	174

Liste des tableaux

1.1	Comparaison entre les langages STRIPS et ADL [Russell et Norvig, 2006]	21
2.1	Primitives descriptives dans [Levinson, 1996].	39
2.2	Cadres de référence et exemples associés.	40
2.3	Relations d'Allen	44
2.4	Relations topologiques entre deux régions 4IM et 9IM	47
2.5	Relations entre les autoroutes I-90, I-93, I-95 et I-495 [Kurata, 2009].	48
2.6	Composition de relations topologiques entre (<i>LL</i>) et (<i>LP</i>)	48
2.7	Relations topologiques possibles déduites	49
3.1	Syntaxe et interprétation des logiques descriptives.	63
3.2	Constructeurs de $\mathcal{AL}$.	67
3.3	Exemple de constructeurs de rôles et de concepts pour étendre $\mathcal{AL}$.	67
3.4	Prédicats définis pour les classes.	71
3.5	Prédicats définis pour les propriétés.	71
4.1	36 relations possibles avec le formalisme ABLR.	95
4.2	Matrice de relations associée à un graphe de relations	108
4.3	Matrice de relations pour une seule dimension.	109
4.4	Matrice de relations M_1 .	111
4.5	Matrice de relations M_2 .	111
5.1	Actions possibles pour le robot dans ce scenario.	119
5.2	Notations dans chaque langage	131
5.3	Extension des éléments de PDDL avec les éléments de <i>SpaceOntology</i> .	137
7.1	Différentes instanciations de <i>SpaceOntology</i> .	165
7.2	Complexité des algorithmes d'arc-consistance.	166
7.3	Temps de calcul pour la recherche de chemins (en secondes)	169
7.4	Temps de calcul d'un plan solution en fonction des requêtes (en secondes).	172
7.5	Comparaison du temps de calcul (secondes).	172

Introduction

Introduction

L'expression « travail collaboratif » désigne l'interaction entre les membres d'une équipe ainsi que la réalisation d'un objectif. En d'autres termes, ce travail fait interagir plusieurs acteurs pour la réalisation d'une mission qui vise à atteindre un but commun. Le projet de construction d'une maison est un exemple de résultat de travail collaboratif entre un maçon, un électricien et le reste de l'équipe. Le travail collaboratif fait intervenir des agents hétérogènes. C'est-à-dire, des agents ayant des visions différentes. La collaboration entre des agents hétérogènes évoluant dans un même environnement nécessite la gestion et le partage de la connaissance, notamment celle de la **connaissance spatiale** de l'environnement commun. La gestion de la connaissance spatiale correspond à l'ensemble des initiatives, des méthodes et des techniques permettant de percevoir, d'identifier, d'analyser, d'organiser, et de partager ces connaissances entre différents acteurs. Par exemple, sur un chantier, pour que les intervenants puissent collaborer, il faut qu'ils s'échangent des informations spatiales telles que « à gauche de l'escalier » ou « à droite de la porte ».

L'interaction homme-robot est une instance de la collaboration entre acteurs hétérogènes. Dans ce document, le terme acteur est remplacé par le terme agent qui indique un humain ou un robot (ou agent artificiel). L'interaction homme-robot est l'étude des interactions entre humains et robots. Dans notre travail, l'interaction homme-robot consiste en un échange de la connaissance spatiale. Cet échange est réalisé grâce à la possibilité d'interpréter et de comprendre la représentation spatiale de l'autre. Pour que cet échange soit possible, l'interaction homme-robot doit considérer un système permettant de référencer les objets existants dans l'environnement. En effet, il faut que le robot et l'humain établissent une même référence à un objet donné. Cependant, ceci n'est pas aussi simple puisqu'ils envisagent l'espace différemment (figure 1) :

- d'un point de vue de l'humain, l'espace est généralement envisagé d'une manière qualitative et imprécise ;
- d'un point de vue du robot, l'espace est considéré d'une manière numérique. Par exemple, l'utilisation de coordonnées pour indiquer la position d'un objet.

La vision du robot n'est pas souvent claire pour l'humain et *vice versa*. Par conséquent, l'interaction entre eux n'est pas intuitive et parfois elle est vouée à l'échec (comme dans l'exemple décrit dans la figure 1). Pour que l'interaction homme-robot soit plus intuitive et de qualité, la mise en place d'un système de médiation est nécessaire. Ce système collecte les informations de l'humain et du robot, ensuite communique à chacun les informations dont il a besoin dans son propre langage. Le scénario le plus fréquent pour l'interaction homme-robot, est que l'humain

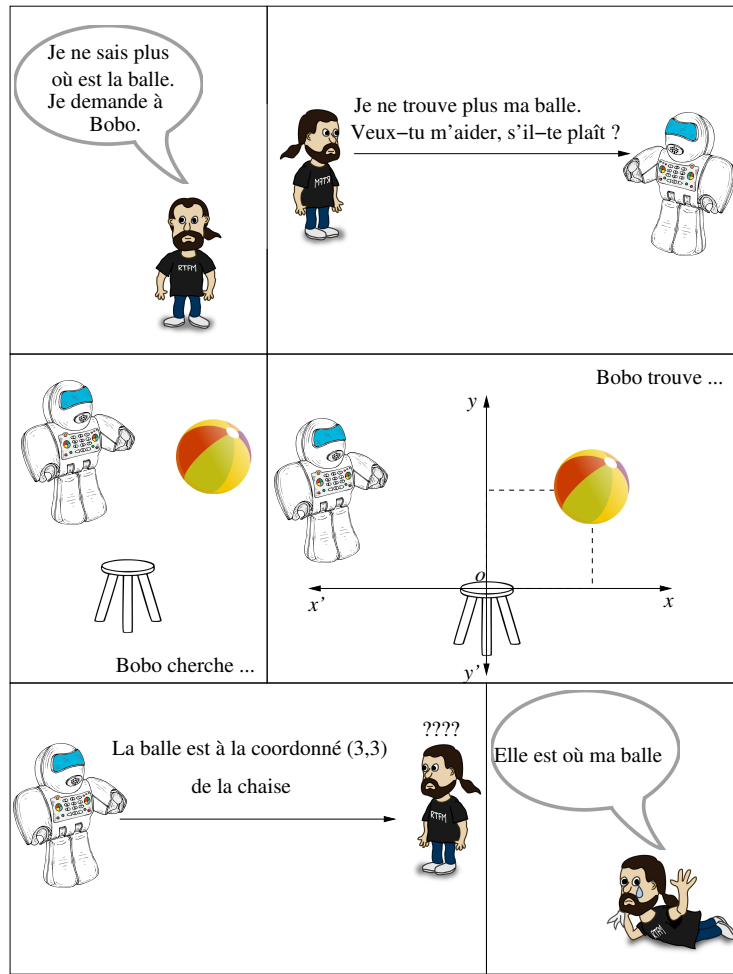


FIGURE 1 – Problème d’interaction entre un humain et un robot.

définisse une mission au robot qui consiste à manipuler certains objets dans l’environnement. Pour accomplir sa mission, le robot doit avoir un plan lui permettant d’exécuter les actions et les déplacements nécessaires.

La planification est une discipline de l’intelligence artificielle qui vise le développement d’algorithmes pour produire des plans, typiquement pour l’exécution par un agent. Bien qu’il existe plusieurs types de planification, ici, nous retenons seulement **la planification de tâches** et **la planification de chemins**.

La planification de tâches est un raisonnement qui a pour objectif de construire un plan d’actions permettant à partir d’un état initial, de résoudre un ensemble de buts en appliquant un ensemble d’actions. La planification de chemins, dite aussi planification de déplacements, est un sous problème de la planification de mouvements. La planification de chemins est un raisonnement qui a pour objectif de définir une séquence de positions définissant un chemin entre une position initiale et une position finale, en considérant certaines contraintes spatiales.

La planification de tâches et la planification de chemins sont liées par le fait que la réalisation d'une tâche (action/déplacement) nécessite la prise en compte de la dimension spatiale de l'environnement dans lequel l'humain et le robot évoluent. En effet, la réalisation d'une tâche peut impliquer une position : par exemple, pour réaliser la tâche « laver la vaisselle », l'agent concerné doit se trouver dans la cuisine. Aussi, la réalisation d'une tâche peut impliquer un ou plusieurs déplacements de l'humain et/ou du robot : par exemple, pour réaliser la tâche « ranger l'appartement », l'agent doit se déplacer entre les pièces et ranger chaque pièce.

Nous nous plaçons dans le cadre d'interaction homme-robot en considérant la composante spatiale, dans le but de planifier des actions pour réaliser une mission. L'objectif global de cette thèse est de proposer une représentation de l'information spatiale pour sa prise en compte et son exploitation dans un processus de planification. Le planificateur spatial qui en découle prend en compte les différentes représentations de la connaissance spatiale fournies par les différents agents.

Le problème de planification spatiale peut être décomposé en deux sous-problèmes, d'une part, **la représentation spatiale** et **le raisonnement spatial** et, d'autre part, **la planification**. Chaque sous-problème implique une expertise appropriée. Nous proposons un modèle unifié pour la représentation et le raisonnement spatial. Ce modèle permet pour une même scène de représenter la connaissance spatiale d'une manière :

- **quantitative**, nécessaire pour gérer les informations fournies par un agent robot. Cette représentation est fondée sur une définition **numérique** (utilisation de coordonnées d'une position, d'un angle, etc.),
- **qualitative**, nécessaire pour gérer les informations fournies par un agent humain. Cette représentation est fondée sur une définition de liens **topologiques** (par exemple l'intersection entre deux régions ou encore une région *a* est à la frontière d'une région *b*) et des liens **flous et ambigus** (utilisation des relations telles que « proche » ou « loin »),
- **hiérarchique**, nécessaire pour **hiérarchiser** un environnement. Cette représentation réduit la quantité d'information à considérer pour une mission donnée.

Les différentes représentations considérées dans ce modèle améliorent la qualité de l'interaction homme-robot. Pour le raisonnement spatial, ce modèle fait correspondre différentes représentations d'une même scène et déduit une connaissance spatiale cohérente. Notre modèle offre une sémantique pour la connaissance spatiale en prenant en compte les différentes représentations spatiales citées ci-dessus. Il fournit un ensemble de concepts et de relations définis par une **ontologie spatiale**.

Nous proposons d'intégrer cette ontologie dans un planificateur qui combine un planificateur de tâches et un planificateur de chemins. Le planificateur de tâches planifie une séquence d'actions. Le planificateur de chemins planifie un chemin entre les différentes zones d'actions. L'intégration d'une ontologie spatiale dans la planification permet de prendre en compte la connaissance spatiale quantitative et qualitative, en particulier floue, lors du processus de planification. Ce qui permet la planification de chemins entre deux positions floues.

Les données manipulées par un planificateur sont codées dans un langage de planification formel. Nous proposons une extension du langage PDDL (*Planning Domain Definition Language*)

en utilisant les éléments de l'ontologie spatiale afin de permettre l'expression de la connaissance spatiale dans un problème de planification.

Ce document est organisé en quatre parties : l'état de l'art, la contribution, l'expérimentation et la conclusion. La première partie présente un ensemble des concepts, techniques et outils nécessaires à nos travaux. Cette partie se compose de trois chapitres. Le chapitre 1 définit d'une manière formelle la planification de tâches et la planification de chemins et présente les différents algorithmes, techniques et langages de planification. Ce chapitre présente une étude bibliographique des travaux intégrant la connaissance spatiale en planification. Le chapitre 2 présente les méthodes de représentation et de raisonnement spatial les plus connus dans le domaine. À partir de ce chapitre, nous constatons qu'il n'y a pas de méthodes permettant de combiner les représentations qualitatives, quantitatives et hiérarchiques en mêmes temps. Le chapitre 3 présente les ontologies, une technique de représentation de l'information utilisée notamment dans les systèmes d'informations géographiques (SIG). En effet, les ontologies peuvent représenter une grande source d'informations géographiques de nature différentes et permettent de déduire des informations spatiales très pertinentes. Ce chapitre présente les langages et outils pour modéliser et exploiter les ontologies.

La deuxième partie présente nos contributions. Cette partie se compose en trois chapitres. Le chapitre 4 décrit la définition d'une ontologie spatiale combinant les représentations qualitatives, quantitatives et hiérarchiques en même temps et permettant un raisonnement spatial en définissant des règles d'inférence. En effet, bien qu'il existe différentes ontologies spatiales dans la littérature (décrit dans le chapitre 3), celles-ci sont définies par rapport aux domaines d'applications et, en général, elles ne permettent pas de gérer l'imprécision. Le chapitre 5 présente un modèle de planification combinant un planificateur de tâches, de chemins et une ontologie spatiale. Il décrit un langage de planification intégrant les éléments de l'ontologie définie dans le chapitre 4. Ceci dans le but d'exprimer la composante spatiale dans un problème de planification et de l'exploiter dans le processus de la planification. Le chapitre 6 détaille ce processus en insistant sur la prise en compte de l'information spatiale lors de la sélection d'une action et la planification d'un chemin.

La troisième partie présente les expérimentations. Cette partie se compose du chapitre 7. Ce chapitre décrit l'apport de la modélisation de la connaissance spatiale dans un problème de planification et son exploitation dans le processus de planification d'un point de vue expérimentation.

Nous terminons ce document par une conclusion dans laquelle nous faisons un bilan de nos contributions et nous y présentons les voies de recherches ouvertes par notre travail.

Première partie

État de l'art

Introduction de la partie I

Cette partie est une étude bibliographique qui présente un panorama non exhaustif des concepts, techniques et outils nécessaires à nos travaux à savoir : le domaine de planification en se focalisant plus sur la représentation que sur les algorithmes ; et la représentation de l'information spatiale grâce aux ontologies comme outil de représentation.

Le premier chapitre introduit la planification classique et donne un aperçu des algorithmes et techniques de planification pour la résolution de ce problème. Nous introduisons aussi l'idée d'intégrer la connaissance spatiale au regard des travaux qui exploitent de telle connaissance.

Le deuxième chapitre expose les différentes approches permettant de représenter et d'exploiter la connaissance spatiale. Nous montrons l'importance de la connaissance spatiale dans la planification. Cela se traduit par une étude bibliographique concernant les travaux en planification intégrant différentes représentations de la connaissance spatiale. Cependant, ces travaux ne considèrent pas des représentations différentes à la fois.

Dans le cadre de nos travaux, nous avons besoins de définir un cadre formel qui permet de combiner certaines représentations de la connaissance spatiale et permettant l'exploitation de cette connaissance. Les ontologies semblent être une bonne technique pour atteindre cet objectif.

Dans le troisième chapitre nous présentons les ontologies comme réponse à notre problème. Nous citons les travaux les plus répandus décrivant la connaissance spatiale à l'aide des ontologies et nous présentons les travaux de planification exploitant ces dernières. Nous terminons par une discussion sur l'ensemble de ces travaux.

Chapitre 1

Planification

Sommaire

1.1	Planification en intelligence artificielle	12
1.1.1	Généralités	12
1.1.2	Planification classique	13
1.2	Planification de tâches	15
1.2.1	Représentations des états	15
1.2.2	Représentation des opérateurs et des actions	16
1.2.3	Représentation d'un problème de planification	17
1.3	Algorithmes et techniques de planification	17
1.3.1	Planification dans un espace d'états	18
1.3.2	Techniques de planification	18
1.4	Langages de planification	20
1.4.1	Historique	21
1.4.2	Langage PDDL	22
1.4.3	Utilisation de PDDL	23
1.5	Intégration de la connaissance spatiale dans la planification	25
1.5.1	Planificateurs de chemins	25
1.5.2	Couplage de planificateurs	26
1.5.3	Formalisation de la connaissance spatiale	27

Ce chapitre introduit les principaux concepts du domaine de planification en utilisant le formalisme de représentation classique STRIPS qui utilise des prédicats logiques du premier ordre [Fikes et Nilsson, 1971]. Bien qu'il existe de nombreux types de planification, nous ne considérons ici que la planification de tâches et la planification de chemins (section 1.1). La planification de tâches est un raisonnement abstrait visant à définir un plan d'actions pour la mission (section 1.2). Nous découvrons quelques unes des nombreuses variantes des langages de type STRIPS (section 1.4). Enfin, nous présentons l'intégration de la connaissance spatiale dans la planification (section 1.5).



1.1 Planification en intelligence artificielle

“ If the organism carries a « small-scale model » of external reality and of its own possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise, utilize the knowledge of past events in dealing with the present and future, and in every way to reach in a much fuller, safer, and more competent manner to the emergencies which face it ”[Craik, 1967].

Nous retenons de cette citation que la planification est une activité cognitive qui implique une formulation mentale des futurs états du monde. L'être humain opère dans ses représentations mentales pour simuler le comportement du monde réel et produire ainsi des prédictions lui permettant en même temps d'éviter les erreurs et les risques possibles.

1.1.1 Généralités

Dans les années 70, le travail [Simon et Newell, 1972] propose un simulateur informatique qui modélise la résolution d'un problème par un homme. Ce simulateur est composé de trois composantes principales : **un état initial**, **un état final** et des **opérateurs** (ou actions).

Un opérateur est un processus de changements de situations ou des mouvements autorisés par des règles traduisant des connaissances expertes du problème. Ces processus vont mener d'un état initial à un état but en passant par des états intermédiaires. Un état (ou situation) est représenté par un ensemble d'objets décrivant le monde¹ et un opérateur par un triplet d'ensembles d'objets représentant :

- les **préconditions** : ce sont les objets qui déterminent les conditions d'application de l'opérateur. Un opérateur peut être appliqué dans un état si et seulement si toutes ses préconditions sont vérifiées dans l'état,
- les **ajouts** : ce sont les objets qui seront ajoutés à l'état courant après application de l'opérateur,
- les **retraits** : ce sont les objets qui seront retirés à l'état courant après application de l'opérateur.

Résoudre un problème se résume à construire un espace d'états, puis à sélectionner les opérateurs adéquats (ou actions) pour suivre un chemin dans cet espace allant de l'état initial à l'état final. La sélection et l'application d'une opération à effectuer va progressivement transformer l'état initial du problème en un état contenant l'état but. Cette modélisation suppose que toutes les informations contenues dans l'énoncé suffisent pour résoudre le problème et qu'il est bien défini. Ce modèle est formalisé à l'aide de STRIPS (*STanford Research Institute Problem Solver*) [Fikes et Nilsson, 1971]. La séquence d'actions permettant de résoudre un problème est appelé **plan solution**.

1. Le terme **monde** est généralement employé en planification en intelligence artificielle (IA) pour désigner une simplification (un modèle) de ce qu'on appelle environnement.

1.1.2 Planification classique

Le « monde des blocs » est l'une des applications de la planification classique les plus connues. Il est constitué d'un ensemble de blocs de forme cubique posés sur une table et d'un bras robotique (figure 1.1(a)). Le but est toujours de construire une ou plusieurs piles de blocs spécifiées, en indiquant quels blocs sont au-dessus de quels autres blocs. Un plan solution possible est défini par deux actions : le bras attrape le bloc C ensuite il le pose sur le bloc B. Nous utilisons cet exemple tout au long de ce chapitre.

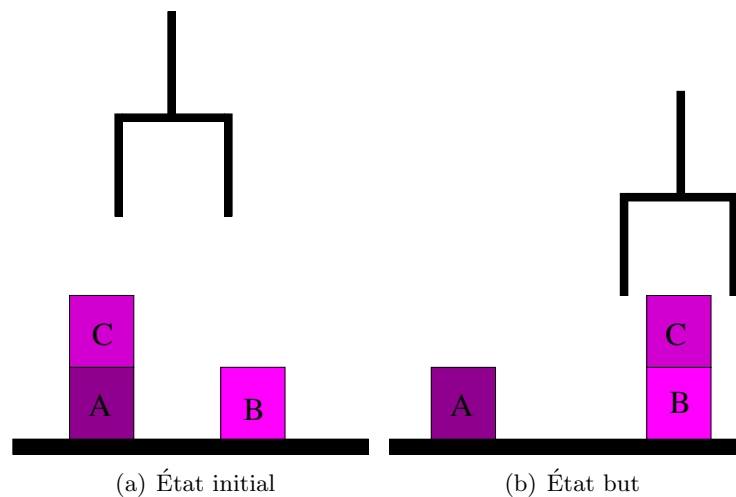


FIGURE 1.1 – Le monde des blocs : la représentation graphique des états initial et but. La figure 1.1(a) décrit l'état initial : le bloc C est sur le bloc A, le bloc A sur la table et le bloc B sur la table. La figure 1.1(b) décrit l'état but : le bloc C sur le bloc B.

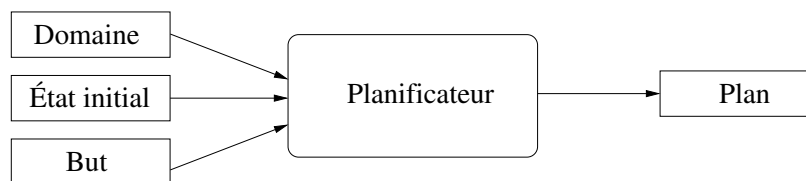


FIGURE 1.2 – Modèle de planification.

Pour résoudre un problème de planification, nous avons besoin d'utiliser un algorithme de planification (**planificateur**). La figure 1.2 décrit les entrées et les sorties d'un planificateur. En entrée, un planificateur prend l'ensemble des données suivant :

- domaine : le modèle du monde est un ensemble d'actions possibles (parfois appelés opérateurs) et d'actions. Chaque opérateur spécifie généralement des préconditions qui doivent être présentes dans l'état actuel pour qu'il puisse être appliqué, et des effets sur l'état actuel ;
- une description de l'état initial du monde ;
- une description d'un but à atteindre.

Ces données forment ce qu'on appelle le **problème de planification**. En sortie, il génère un plan. Ce plan peut prendre différentes formes, la plus commune étant une séquence d'actions.

Dans la littérature, plusieurs familles de planification existent telles que présentées dans [Ghallab *et al.*, 2004] et [Régner et Vidal, 2004]. Nous invitons les lecteurs à les consulter pour plus de détails. Dans cette thèse, nous nous situons dans le cas de la planification classique en considérant les hypothèses évoquées dans [Ghallab *et al.*, 2004] à savoir :

1. observabilité totale : tout ce qui peut être connu sur l'environnement est connu. Par exemple, dans le monde de bloc les positions des blocs et l'état du bras sont connus à tout instant, dans la figure 1.1(a) : le bras est libre, le bloc C sur le bloc A, le bloc B sur la table ;
2. déterministe : lorsqu'on exécute une action le résultat est constant. Autrement dit, l'exécution se déroule dans un monde parfait où les actions ne peuvent pas échouer et leurs effets sont totalement prédéterminés. Par exemple dans le monde de bloc, si le bras pose le bloc C sur le bloc B, après exécution de cette action l'état du monde sera comme illustré dans la figure 1.1(b) : le bloc C sur le bloc B, la surface supérieure du bloc A est libre ;
3. monde statique : il n'y a pas d'événements externes qui modifient l'environnement. Par exemple, dans le monde de bloc, seul l'intervention du bras fait changer l'état du monde ;
4. plans séquentiels : les plans sont des séquences d'actions où les actions s'exécutent l'une à la suite de l'autre. Il n'y a pas d'actions simultanées.
5. discret : les actions n'ont pas de durée, elles sont considérées comme étant instantanées au moment de la planification.

Toutes ces hypothèses ne simplifient pas la complexité du problème de planification mais spécifie seulement le cadre. La quantité des travaux dans la littérature en planification classique prouve la complexité de ce problème. En particulier, lors de l'introduction d'une ressource, d'une dimension temporelle ou d'une dimension spatiale, etc.

Dans la littérature, il existe plusieurs manières pour représenter un problème classique de planification. En effet, nous pouvons citer *Set-theoretic representation* [Green *et al.*, 1969] ou encore *Classical representation* [Fikes et Nilsson, 1971]. Dans la première représentation (respectivement, la deuxième représentation), ce sont les propositions qui sont utilisées (respectivement, les prédicats du premier ordre et des connecteurs logiques). Ainsi, chaque état du monde est un ensemble de propositions (respectivement, un ensemble de prédicats du premier ordre et des connecteurs logiques) et chaque action est une expression spécifiant les propositions (respectivement, une expression spécifiant les prédicats du premier ordre et des connecteurs logiques) qui doivent appartenir à l'état courant pour que l'action soit exécutable ainsi que les propositions à ajouter ou à supprimer de l'état suite à son exécution (respectivement, une expression spécifiant les prédicats du premier ordre et des connecteurs logiques).

1.2 Planification de tâches

La planification de tâches consiste en un raisonnement symbolique qui a pour objectif de construire un plan d'actions pour réaliser une mission dans laquelle les conditions du but final sont valides à partir d'une description de l'état initial, des opérateurs et du but à atteindre.

Les planificateurs nécessitent la représentation du problème de planification à résoudre (figure 1.2), c'est-à-dire la représentation des états, des actions et des buts. Pour cela, il faut trouver un langage suffisamment expressif pour pouvoir décrire une large variété de problèmes, mais aussi suffisamment restrictif pour l'utiliser d'une manière efficace. Nous considérons pour l'instant le formalisme de représentation classique STRIPS [Fikes et Nilsson, 1971] (détails dans la section 3.3.1) qui utilise des notations dérivées du premier ordre : un état est représenté sous la forme d'un ensemble d'atomes logiques qui sont vrais lorsque le système est dans l'état considéré (notons par $\mathcal{S}$ l'ensemble des états). Les actions sont représentées par des opérateurs de planification qui changent les valeurs de ces atomes (notons par $\mathcal{O}$ l'ensemble des opérateurs). Formellement, l'ensemble des états du monde sont représentés par un ensemble de prédicats de la logique de premier ordre $\mathcal{L}$ (définition 1) décrivant le monde. Nous rappelons les principaux concepts du domaine de la planification en utilisant STRIPS.

Définition 1 *Le langage $\mathcal{L}$ est un langage de premier ordre, basé sur des prédicats : $\mathcal{L} = \{p_1, \dots, p_n\}$. Ce langage est utilisé pour représenter un domaine de planification. Le langage $\mathcal{L}$ ne contient ni symboles ni fonctions.*

Définition 2 *Un prédicat p est défini par son symbole et une liste de paramètres. Il représente des relations entre les différents objets du domaine.*

Par exemple, le prédicat $\text{Sur}(\mathbf{C}, \mathbf{A})$ signifie que le bloc $\mathbf{C}$ se trouve au-dessus du bloc $\mathbf{A}$ (figure 1.1(a)).

1.2.1 Représentations des états

Un état est une description du monde. Dans la logique du premier ordre, un état est représenté sous la forme d'une conjonction de littéraux positifs. Par exemple, $\text{SurTable}(\mathbf{A}) \wedge \text{Sur}(\mathbf{C}, \mathbf{A}) \wedge \text{SurTable}(\mathbf{B})$ représente l'état du monde de bloc décrit dans la figure 1.1(a). Aussi, nous considérons les littéraux propositionnels : par exemple, BrasVide pour indiquer que le bras est vide dans la figure 1.1(a). Les variables (par exemple $\text{Sur}(\mathbf{x}, \mathbf{y})$) et les fonctions (par exemple $\text{Sur}(\text{Cube}(\mathbf{D}), \mathbf{E})$) ne sont pas permises dans les littéraux. Formellement, nous avons :

Définition 3 *Un état s est un ensemble d'atomes instanciés du langage $\mathcal{L}$.*

Propriété 1 *Comme $\mathcal{L}$ ne contient pas de fonctions, nous avons alors un ensemble fini d'états.*

Propriété 2 *$\mathcal{S}$ est l'ensemble des états tel que chaque état est un sous-ensemble de $\mathcal{L}$, ainsi $\mathcal{S} \subseteq 2^{\mathcal{L}}$.*

Nous considérons l'hypothèse du monde clos ; tout ce qui n'est pas vrai est supposé faux. Autrement dit, un prédicat, qui n'est pas vérifié dans un état est considéré comme faux dans cet état. Formellement, cette hypothèse se traduit par le fait qu'un prédicat p est vérifié dans un état s si et seulement si p peut être unifié avec des prédicats de s ($p \in s$, par exemple).

Un état but est un état partiellement spécifié. Il est défini par une conjonction de littéraux (sans variable) tel que $\text{Sur}(C,B)$ (figure 1.1(b)). Un état propositionnel s satisfait un but s_g si s contient tous les atomes de s_g (et éventuellement d'autres atomes). Par exemple, l'état $\text{SurTable}(A) \wedge \text{Sur}(C,B)$ satisfait le but $\text{Sur}(C,B)$ (figure 1.1(b)).

1.2.2 Représentation des opérateurs et des actions

Une action est spécifiée par des préconditions qui doivent être respectées avant l'exécution et les effets qui résultent de cette exécution. Un opérateur est un schéma d'action représentant une opération possible dans un domaine donné. Il représente un ensemble d'actions différentes que l'on peut dériver en instanciant les paramètres avec des constantes différentes (définition 4).

Définition 4 *Un opérateur o est défini par un quadruplet tel que :*

$$o = (\text{nom}(o), \text{params}(o), \text{precond}(o), \text{effets}(o)) :$$

- $\text{nom}(o)$: indique le nom de l'opérateur,
- $\text{params}(o)$: liste des paramètres de l'opérateur sous la forme $\{\text{param}_1, \dots, \text{param}_n\}$,
- $\text{precond}(o)$: représente les préconditions de l'opérateur, c'est-à-dire les propriétés du monde nécessaires à son exécution,
- $\text{effets}(o)$: définit l'ensemble des faits à ajouter et à supprimer de l'état après l'exécution de o .

Par exemple, le bras pose un bloc sur un autre est défini par l'opérateur PoserSurBloc , comme suit :

```
PoserSurBloc( $b_1, b_2$ )
Precond: SurfaceLibre( $b_2$ ), Attrape( $b_1$ )
Effets:  Sur( $b_1, b_2$ ), SurfaceLibre( $b_1$ ), BrasLibre,
         $\neg$  SurfaceLibre( $b_2$ ),  $\neg$  Attrape( $b_1$ )
```

CODE 1.1 - Définition de l'opérateur PoserSurBloc .

Pour augmenter la lisibilité, certains travaux (par exemple : [Hioual et Boufaïda, 2010]) distinguent entre les effets positifs et les effets négatifs, en définissant une liste d'insertion pour les littéraux positifs ($\text{Sur}(b_1, b_2)$, $\text{SurfaceLibre}(b_1)$, BrasLibre) et une liste de suppression pour les littéraux négatifs ($\text{SurfaceLibre}(b_2)$, $\text{Attrape}(b_1)$) (code 1.1).

Le principe de la modélisation des actions par des opérateurs consiste à spécifier à l'aide de prédicats, les propriétés du monde nécessaires à son application, c'est-à-dire, les préconditions de

l'action, ainsi que les effets engendrés par son exécution. Les préconditions positives expriment les propriétés qui doivent être vérifiées, les préconditions négatives expriment celles qui doivent être absentes de l'état pour que l'action soit appliquée. Les effets d'un opérateur spécifiant les propriétés du monde modifiées par l'exécution d'une action. D'un point de vue formel, si une action est définie par un opérateur qui transforme un état s_i en un état s_{i+1} , les effets d'une action sont représentés par les prédicats ajoutés (effets positifs) et les prédicats enlevés (effets négatifs) à s_i pour obtenir s_{i+1} . Intuitivement, si le modèle d'actions (opérateurs) est correct et que l'exécution se déroule comme prévu, le fait qu'une action a_i est applicable à l'état s_i signifie qu'elle mènera à l'état s_{i+1} lors de son exécution. Une action a est dite « applicable » à un état s si toutes ses préconditions sont satisfaites dans l'état s .

Définition 5 Une action est une instance d'un opérateur. Si a est une action et s_i est un état tel que $\text{precond}(a) \subseteq s_i$ alors a est applicable à s_i et le résultat de cette application est l'état $s_{i+1} = (s_i - \text{effets}^-(a)) \cup \text{effets}^+(a)$

1.2.3 Représentation d'un problème de planification

La définition 6 présente d'une manière formelle un problème de planification. En planification, un domaine D décrit le cadre dans lequel va être résolu le problème. Il est indépendant de n'importe quel état initial ou but. Intuitivement, un domaine modélise toutes les actions que peut effectuer le système pour lequel on planifie. Dans l'exemple du monde de blocs, le système pour lequel on planifie est le bras robotique. Formellement, le domaine est défini par l'ensemble des types de variables, l'ensemble des prédicats et A l'ensemble des opérateurs instanciés de O (définition 7). Une solution à un problème de planification classique est appelée plan (définition 8).

Définition 6 Un problème de planification pour un domaine de planification D est défini par le couple (s_0, s_g) où : s_0 est l'état initial et s_g est le but à atteindre.

Définition 7 Un domaine de planification D est un triplet (S, A, γ) tel que :

- S : ensemble des états avec $S = 2^{\{\text{atomes instanciés de } \mathcal{L}\}}$,
- A : l'ensemble des opérateurs instanciés de O ,
- $\gamma(s, a) = (s - \text{effets}^-(a)) \cup \text{effets}^+(a)$ si $a \in A$ et a est applicable à $s \in S$. La fonction γ étant la fonction de transitions entre états.

Définition 8 Un plan solution est défini par une séquence d'actions $\{a_1, a_2, \dots, a_n\}$ où chaque action a_i est une instance d'opérateur de O permettant d'aller de s_0 à s_g .

1.3 Algorithmes et techniques de planification

Il existe deux grandes familles d'algorithmes de planification : (1) les algorithmes de planification dans un espace d'états (*State-Space Planning*) [Fikes et Nilsson, 1971, Kautz et Selman,

1996] et (2) les algorithmes de planification dans un espace de plans (*Plan-Space Planning*) [Minton *et al.*, 1994, Kambhampati *et al.*, 1995]. La recherche dans les espaces d'états a été longtemps délaissée car était considérée comme peu efficace. Plusieurs travaux [Bonet *et al.*, 1997, Bonet et Geffner, 1998] ont cependant montré que cette méthode pouvait s'avérer plus performante que la recherche dans les espaces de plans partiels qui était jusque là privilégiée. C'est pour cette raison que nous nous intéressons aux algorithmes de planification dans un espace d'états.

1.3.1 Planification dans un espace d'états

Ici, les algorithmes de recherche considèrent que l'espace de recherche est un sous-ensemble de l'espace d'états. Chaque nœud correspond à un état du monde, chaque arc correspond à une transition entre deux états, et le plan courant correspond au chemin courant dans l'espace de recherche. Les principaux algorithmes de planification dans un espace d'états sont : la recherche par chaînage avant [Hoffmann et Nebel, 2001], la recherche par chaînage arrière [Helmert, 2006], la recherche mixte [LaValle, 2006] et l'algorithme STRIPS [Fikes et Nilsson, 1971].

Le principal problème des algorithmes de planification dans un espace d'états est la taille de l'espace de recherche. L'objectif étant de savoir comment améliorer l'efficacité en réduisant la taille de cet espace de recherche. L'algorithme STRIPS développe un arbre de recherche dont la racine est l'état initial et le but. Les nœuds de l'arbre sont étiquetés par l'état courant et un ensemble de sous-buts à atteindre. Cet arbre permet de combiner le chaînage arrière, par création de nœuds suivants comprenant le même état avec un sous-but supplémentaire et, le chaînage avant, par création de nœuds suivants comprenant l'état résultat d'une action éliminant le premier sous-but de la liste, ainsi que cette liste réduite.

L'algorithme STRIPS réduit une grande partie de l'espace de recherche, mais il n'est ni complet ni optimal. Cette réduction est caractérisée par les deux points :

- à chaque appel récursif de l'algorithme, les sous-buts qui doivent être immédiatement satisfaits sont uniquement les préconditions de la dernière action considérée pour réduire l'écart entre le but et l'état courant. Ainsi le facteur de branchement de l'algorithme est réduit,
- si l'état courant satisfait toutes les préconditions d'un opérateur, alors celui-ci est marqué et, en cas d'échec, le retour arrière ne sont pas remises en cause.

L'utilisation d'heuristiques et des progrès sur l'algorithmique de recherche ont maintenant hissé la technique de recherche dans les espaces d'états au premier rang en terme de taille de problèmes traités et de temps de calcul.

1.3.2 Techniques de planification

Afin d'optimiser le processus de planification, plusieurs techniques de planification ont été proposées telles que la planification graphique, hiérarchique ou par heuristique.

Technique de planification graphique

GraphPlan est un planificateur, le plus connu, qui utilise comme support un graphe [Blum et Furst, 1997]. C'est une technique de planification graphique dans un espace de recherche très puissant s'appuyant sur les espace de recherches dans un espace d'états et les espaces de recherche dans un espace de plans. GraphPlan fonctionne en deux étapes.

Première étape : Tout d'abord, il construit un graphe appelé **graphe de planification** (figure 1.3) qui permet de déterminer les états atteignables à partir de l'état initial et des opérateurs disponibles. Ce graphe est structuré en niveaux qui alternent successivement des nœuds étiquetés par des propositions et des nœuds étiquetés par des actions. La construction de graphe se termine lorsqu'un niveau contient tous les prédicats du but. L'algorithme conserve dans le graphe de planification l problème initial grâce à un mécanisme d'exclusion mutuelle entre certains nœuds du graphe.

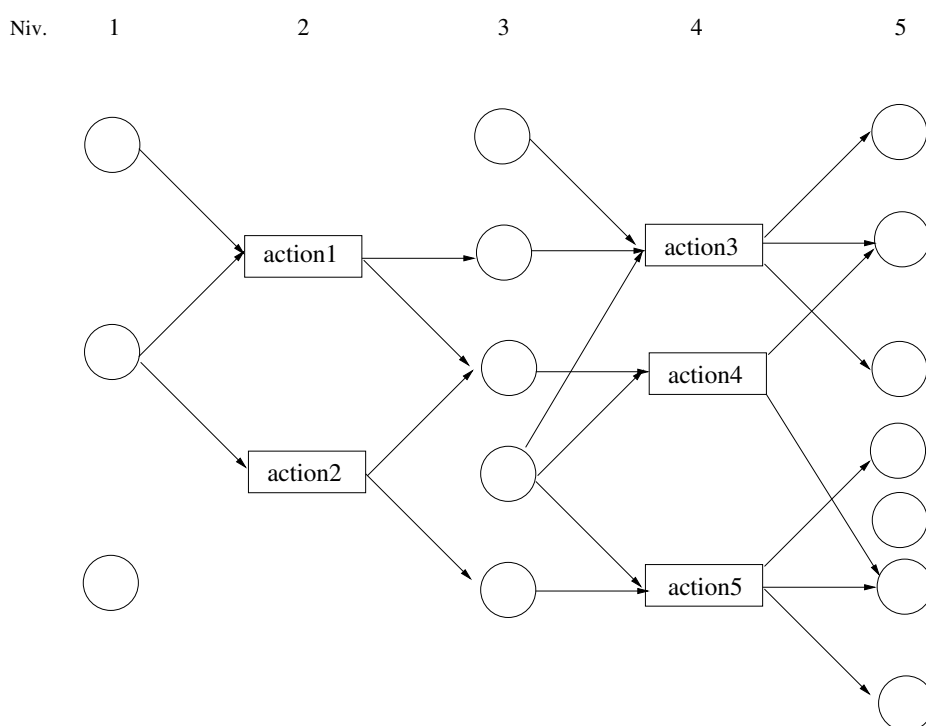


FIGURE 1.3 – Construction du graphe de planification.

Deuxième étape : Dans la deuxième phase, appelée phase d'extraction, l'algorithme parcourt le graphe de planification à la recherche d'un plan solution. Il procède par chaînage à partir de l'état final et remonte le graphe jusqu'à l'état initial en choisissant, à chaque niveau d'actions, une action compatible avec les actions précédemment sélectionnées.

Technique de planification hiérarchique

La technique de planification hiérarchique est une méthode pour la création des plans par la décomposition des tâches. C'est un processus où le système de planification décompose les tâches en sous-tâches jusqu'à ce que des tâches primitives soient retrouvées.

Le planificateur ABSTIPS [Knoblock, 1992] utilise pour la première fois plusieurs niveaux d'abstraction lors du processus de planification pour limiter l'espace de recherche. À chaque niveau d'abstraction, le planificateur effectue son choix parmi un nombre restreint d'opérateurs et de faits. Des sous-buts sont définis et résolus en fonction de l'importance qui leur est accordée. Dans le planificateur NOAH [Chaignaud et Lévy, 2001], ce sont les opérateurs de planification qui sont hiérarchisés. À partir des années 1990, le terme planification hiérarchique fait référence à la planification hiérarchique (Hierarchical Task Network) [Erol, 1996] dans laquelle le domaine de planification spécifie la façon de réaliser les buts. Les planificateurs HTN les plus connus sont UCMP [Erol *et al.*, 1994], SHOP [Nau *et al.*, 1999] et SHOP2 [Nau *et al.*, 2003].

La planification hiérarchique utilise un langage similaire à celui de la planification classique. En effet, chaque état du monde est représenté par un ensemble de prédicats instanciés et chaque action correspond à une transition déterministe entre deux états. Cependant, la planification hiérarchique diffère de la planification classique de l'objectif. En effet, dans un planificateur HTN, l'objectif n'est pas d'atteindre un ensemble de buts mais de résoudre un ensemble de tâches. En plus de l'ensemble d'opérateurs, le domaine est constitué d'un ensemble de méthodes.

Les tâches non primitives sont représentées par des méthodes. Une méthode décrit la manière de décomposer une tâche en un ensemble de sous-tâches. Le principe de ce type de planification est de décomposer une tâche non-primitive en sous-tâches de plus en plus petites jusqu'à obtenir des tâches primitives qui peuvent être résolues en appliquant les opérateurs de planification du domaine.

Technique de planification par recherche heuristique

Les heuristiques sont utilisées en planification de tâches pour accélérer le processus de recherche d'une solution. On parle de planification par recherche heuristique [Bonet et Geffner, 2001]. Dans ce type de planification, le problème est souvent relaxé, c'est-à-dire, les effets négatifs des actions sont ignorés, puis un plan est recherché pour le problème relaxé. L'objectif est de calculer la distance au but depuis l'état initial, c'est-à-dire connaître à priori la longueur du plan solution. Parmi les planificateurs utilisant ce principe on peut citer HSP [Bonet et Geffner, 2000] et FF [Hoffmann et Nebel, 2001].

1.4 Langages de planification

Dans la littérature, trois principaux langages de planification permettent de représenter un problème de planification : STRIPS [Fikes et Nilsson, 1971], ADL [Pednault, 1994] et PDDL [McDermott *et al.*, 1998].

1.4.1 Historique

Le langage STRIPS est considéré comme le langage de représentation basique pour la planification classique. STRIPS s'est révélé insuffisamment expressif pour certains domaines du monde réel [Russell et Norvig, 2006]. Aussi, plusieurs améliorations ont été proposées, par exemple ADL (*Action description Language*). Ce langage permet l'utilisation de quantificateurs et d'expressions conditionnelles dans la description des opérateurs. Le tableau 1.1 dresse une comparaison entre les langages STRIPS et ADL.

STRIPS	ADL
Seulement des littéraux positifs dans les états, par exemple dans la figure 1.1(a) : BrasLibre $\wedge$ SurfaceLibre(B)	Littéraux positifs et négatifs dans les états
Hypothèse du monde clos : les littéraux qui ne sont pas mentionnés sont faux	Hypothèse du monde ouvert : les littéraux non mentionnés sont inconnus
L'effet $P \wedge \neg Q$ signifie ajouter P et supprimer Q	L'effet $P \wedge \neg Q$ signifie ajouter P et $\neg Q$ et supprimer $\neg P$ et Q
Seulement des littéraux clos dans les buts : BrasLibre $\wedge$ SurfaceLibre(B)	Variables quantifiées dans les buts : $\exists x A(P_1, x) \wedge A(P_2, x)$ est le but pour avoir P_1 et P_2 au même endroit
Les buts sont des conjonctions : BrasLibre $\wedge$ SurfaceLibre(B)	Les buts autorisent les conjonctions et les disjonctions : BrasLibre $\wedge$ (SurfaceLibre(B) $\vee$ SurfaceLibre(A))
Les effets sont des conjonctions	Les effets conditionnels sont autorisés lorsque $P : E$ signifie que E est un effet uniquement si P est satisfait
L'égalité n'est pas prise en charge	Le prédicat d'égalité ($x = y$) est prédéfini.
Les types ne sont pas pris en charge	Les variables peuvent être typées par exemple ($b : Bloc$)

TABLE 1.1 – Comparaison entre les langages STRIPS et ADL pour représenter les problèmes de planification [Russell et Norvig, 2006].

Récemment, différents formalismes de planification utilisés en intelligence artificielle ont été unifiés dans une syntaxe standard dite PDDL (*Planning Domain Description Language*). Ce langage permet aux chercheurs de la communauté de planification classique d'échanger des problèmes de tests et de performance et de comparer les résultats. PDDL contient des sous langages pour STRIPS, ADL et les réseaux hiérarchiques de tâches (HTN). Les progrès réalisés ces dernières années permettent au langage PDDL de décrire des domaines complexes et de plus en plus proches des problèmes réels tels que la prise en compte de la dimension temporelle [Fox et Long, 2003], ou la gestion des ressources [Maris, 2009].

1.4.2 Langage PDDL

La définition d'un problème de planification à l'aide de PDDL nécessite : la définition d'un domaine générique commun à tous les scénarios ; une description d'un problème. Dans ce qui suit, nous présentons la description en PDDL d'un domaine, d'un opérateur et d'un problème de planification.

Le code 1.2 présente la description d'un domaine de planification. Nous notons :

- la déclaration des types qui passe par l'utilisation du mot clé **:types** ;
- la déclaration des prédicats qui passe par l'utilisation du mot clé **:predicates** ;
- la définition générique des opérateurs qui passe par l'utilisation du mot clé **:action**.

```
(define (domain DOMAIN_NAME)
  (:requirements :strips :equality ....)
  (:typing T1 T2 T3 ... TN - objects)
  (:predicates (PREDICATE_1_NAME [?A1_1 - T1_1 ... ?AN_1 - TN_1])
               (PREDICATE_2_NAME [?A1_2 - T1_2 ... ?AN_2 - TN_2])
               ...)
  (:action ACTION_1_NAME
   :parameters ([?P1 ?P2 ... ?PN])
   :precondition PRECOND_FORMULA
   :effect EFFECT_FORMULA)
  ...)
```

CODE 1.2 - Définition d'un domaine de planification.

Un opérateur est défini par son nom, ses paramètres, ses préconditions et ses effets (code 1.3).

```
(:action ACTION_1_NAME
 :parameters ([?P1 ?P2 ... ?PN])
 :precondition PRECOND_FORMULA
 :effect EFFECT_FORMULA)
```

CODE 1.3 - Définition d'un opérateur de planification.

Les paramètres (**:parameters**) d'un opérateur sont des objets typés ou des instances des prédicats (ou atomes) spécifiés par les préconditions (**:precondition**) d'application de l'action. Les préconditions contiennent aussi des conjonctions d'atomes. Les effets de l'opérateur sont des conjonctions d'ajouts d'atomes et des conjonctions de retraits d'atomes. Les retraits d'atomes sont précédés du mot **:not** noté $\neg$.

```

(define (problem PROBLEM_NAME)
  (:domain DOMAIN_NAME)
  (:objects OBJ1 OBJ2 ... OBJ_N)
  (:init ATOM1 ATOM2 ... ATOM_N)
  (:goal CONDITION_FORMULA))

```

CODE 1.4 - Définition d'un problème de planification.

Une instance est composée des objets autorisés à être manipulés, une description de l'état initial et une description de l'état final (code 1.4).

1.4.3 Utilisation de PDDL

Pour illustrer, l'utilisation de PDDL, nous étendons le scénario du monde des blocs (figure 1.1, page 13) pour considérer un robot mobile qui doit ramasser différents blocs dans l'environnement (figure 1.4(a)). Cet environnement se compose de bureaux (B1, B2, B3, B4) et de couloirs (C1, C2). La mission du robot, est de ramasser tous les blocs (b1, b2, b3, b4, b5) et de les déposer le couloir C1 comme décrit dans la figure 1.4(b).

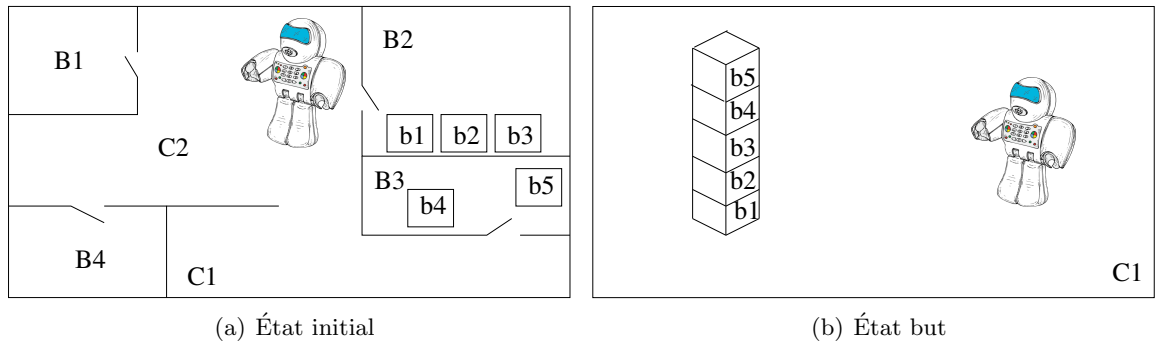


FIGURE 1.4 – Le monde des blocs étendu. Le robot doit se déplacer dans l'espace décrit par la figure 1.4(a) pour ramasser les différents blocs et les mettre dans le couloir C1 selon l'organisation décrite dans la figure 1.4(b).

Le code 1.5 définit le domaine de planification `BlocWorld` qui sera commun à tous les scénarios de ramassage et d'organisation des blocs. En particulier celui présenté dans la figure 1.4 défini par le code 1.6. Le problème de planification `BlocWorld-5-blocs`, définit 5 blocs à ramasser et à déposer dans le couloir C1 comme décrit dans la figure 1.4(b). L'élément `:init` permet de décrire l'état initial du monde. Dans cet exemple, l'état initial concerne la disposition des blocs et du robot (figure 1.4). Par exemple `(at b1 B2)` exprime que le bloc b1 est dans le bureau B2, etc. L'élément `:goal` permet de décrire l'état but, c'est-à-dire le but que nous voulons atteindre : « tous les blocs sont dans le couloir C1 » `((at b1 C1), ..., (at b4 C1)(at`

b5 C1)), tel que « b5 est au-dessus de b4, b4 est au-dessus de b3, b3 est au-dessus de b2 et b2 est au-dessus de b1 » ((on b5 b4), ..., (on b2 b1)) (figure 1.4(b)).

```
define (domain BlocWorld)
  (:types gripper bloc office corridor door)
  (:predicates atRobot at free holding connects on free_sureface)
  (:action pick
    :parameters (?b - bloc ?g - gripper ?o - office)
    :precondition (and (at ?b ?r) (atRobot ?o) (free ?g))
    :effect (and (holding ?g ?b) (not (at ?b ?o)) (not (free ?g))))
  (:action release
    :parameters (?b - bloc ?g - gripper ?o - office)
    :precondition (and (holding ?g ?b) (atRobot ?r))
    :effect (and (at ?b ?o) (not (holding ?g ?b)) (free ?g)))
  (:action move
    :parameters (?o1 ?o2 - office ?d - door)
    :precondition (atRobot ?o1) (connects ?d ?o1 ?o2)
    :effect (and (atRobot ?o2) (not (atRobot ?o1))))
  (:action put
    :parameters (?b1 ?b2 - bloc ?g - gripper)
    :precondition (and (free_sureface ?b1) (holding ?g ?b2) (not (on ?b2 ?b1)))
    :effect (and (on ?b2 ?b1) (free_sureface ?b2)
      (not (holding ?g ?b2)) (not (free_sureface ?b1))))
```

CODE 1.5 - Définition du domaine de planification BlocWorld.

```
(define (problem BlocWorld-5-blocs)
  (:domain CleanWorld)
  (:objects B1 B2 B3 B4 - office
            C1 C2 - corridor
            b1 b2 b3 b4 b5 - target)
  (:init
    (atRobot C2) (at b1 B2) (at b2 B2)
    (at b3 B2) (at b4 B3) (at b5 B3))
  (:goal (and
    (at b1 C1) (at b2 C1) (at b3 C1) (at b4 C1) (at b5 C1)
    (on b5 b4) (on b4 b3) (on b3 b2) (on b2 b1))))
```

CODE 1.6 - Définition du problème de planification CleanWorld-5-targets.

1.5 Intégration de la connaissance spatiale dans la planification

Pour montrer le rôle important que joue l'information spatiale dans la planification. Nous considérons l'exemple par la figure 1.4 (page 23). Pour réaliser la mission, il faut se déplacer dans l'environnement entre les différents bureaux et couloirs en évitant les obstacles, s'il y en a. Des travaux pour résoudre ce genre des problèmes de planification en exploitant la connaissance spatiale ont été proposés dans la littérature. Ces travaux consistent à combiner un planificateur de tâches avec un planificateur de chemins. On y considère l'aspect symbolique (d'un point de vue planification : utilisation des prédicats pour décrire un état) et l'aspect géométrique ce qui correspond aux informations spatiales. Ce type de planification est très exploité pour résoudre les problèmes en robotique.

Tout d'abord, nous allons présenter la planification de chemins. Ensuite, nous évoquons le couplage d'un planificateur de tâches et d'un planificateur de chemins. Enfin, nous abordons la question de la formalisation de la connaissance spatiale dans un problème de planification.

1.5.1 Planificateurs de chemins

La planification de chemins (*Path planning*) dite aussi planification de déplacements, est un sous problème de la planification de mouvements. Globalement, la planification de chemins se définit comme suit : connaissant une description de l'environnement et la position d'un agent au sein de cet environnement, il est possible de calculer une trajectoire pour atteindre un but. La planification de chemins consiste à définir une séquence de configurations définissant un chemin sans collision entre une position initiale et une position finale, en respectant certaines contraintes. La planification de chemins est un domaine de recherche très actif. Les approches qui émergent dans ce domaine sont **la décomposition cellulaire**, **les approches par réduction**, **les champs de potentiel** et **les méthodes aléatoires**. Dans notre travail, nous ne considérons pas les méthodes aléatoires puisque nous sommes dans le cadre d'une planification déterministe. Nous invitons le lecteur à consulter l'annexe A pour plus de détails sur les approches citées.

Méthode par décomposition cellulaire

Les méthodes par décomposition cellulaire permettent de décomposer l'espace de configuration en un ensemble de cellules et un ensemble de relations d'adjacence entre ces cellules [Barraquand et Latombe, 1991]. Chaque cellule correspond à une zone particulière de l'espace de configuration. Nous construisons alors un graphe d'adjacence où chaque nœud représente une cellule et où une arête entre deux nœuds indique que les deux cellules correspondantes sont adjacentes. Ainsi, il existe un chemin entre deux configurations q_i et q_f si et seulement si q_i et q_f appartiennent à des cellules membres d'une composante connexe du graphe d'adjacence. Il ne reste plus qu'à construire le chemin à partir de la séquence de cellules ainsi obtenues. De nombreux travaux ont adopté ce type de méthodes et ont conduit à la mise en œuvre de plu-

sieurs techniques, généralement classées en deux catégories et se distinguent par les méthodes de décomposition utilisées et leur complétude quand à la résolution du problème de planification.

Méthode par réduction

L'idée est de ramener l'espace de recherche à un graphe. Une fois construit, ce graphe est utilisé en tant qu'ensemble de chemins possibles. La planification de chemins est alors réduite à la connexion des configurations initiale et finale du graphe, puis à la recherche d'un chemin solution. Le problème de planification de chemins devient alors un problème de recherche dans un graphe qui peut être résolu par différents algorithmes A^* [Hart *et al.*, 1968], Dijkstra [Dijkstra, 1959], programmation par contraintes, etc. Plusieurs méthodes existent dans la littérature [Tournassoud, 1992], les plus développées parmi elles sont le graphe de visibilité et le diagramme de Voronoï.

Méthode du potentiel

Les champs de potentiel [Khatib, 1986] est une méthode qui assimile l'agent à une particule soumise à un champ de forces répulsives et attractives. Un obstacle génère un champ de potentiel répulsif tandis que l'objectif à atteindre génère un champ de potentiel attractif. À chaque itération, la force induite par la fonction de potentiel permet de choisir la direction la plus prometteuse. L'algorithme calcule donc un vecteur résultant qui indiquera à l'agent comment effectuer son déplacement. Cet algorithme est totalement réactif et peut donc être très facilement implémenté en temps réel. Cependant, comme l'environnement est souvent très peu convexe, cette méthode entraîne facilement l'agent dans des minima locaux. De plus, les problèmes d'oscillation peuvent, dans certain cas, être constatés. Ici non plus, on ne tient pas compte des contraintes de l'agent. Le principal reproche fait à ces méthodes est la possible existence de minima locaux piégeant l'agent.

1.5.2 Couplage de planificateurs

Plusieurs travaux en planification, exploitent le principe de coupler un planificateur de tâches avec un planificateur de chemins, notamment, les problèmes faisant intervenir un robot mobile qui doit se déplacer dans l'environnement tel que [Lamare et Ghallab, 1998], [Moratz *et al.*, 2003], [Guitton, 2008], [Bouguerra *et al.*, 2008], etc. Par exemple, dans [Lamare et Ghallab, 1998], le planificateur de tâches IXTET [Ghallab et Laruelle, 1994] est couplé avec un planificateur d'itinéraires qui s'appuie sur un algorithme de Moore-Dijkstra. Lors du processus de planification, lorsqu'un sous-problème de déplacement est identifié, celui-ci est envoyé au planificateur d'itinéraires. Ces sous-problèmes sont identifiés par le fait qu'il n'y ait plus que des défauts sur un attribut particulier relatif à la position du robot. Cet attribut est défini lors de la modélisation du domaine de planification. Aussi, nous citons le planificateur hiérarchique SHOP2 qui fait appel à des modules extérieurs codées pour calculer les meilleurs chemins sur les problèmes de domaines des rovers.

Cette manière d'améliorer les performances du planificateur est vue comme l'utilisation d'un modèle hybride dans lequel certains sous-problèmes sont résolus par des algorithmes spécifiques. Il existe trois types de couplage : générer les tâches puis le déplacement ([Zacharias *et al.*, 2006], [Choi et Amir, 2009]), générer le déplacement puis les tâches ([Chanthery, 2005], [Baltié *et al.*, 2008]) et la planification hybride ([Kambhampati *et al.*, 1993]).

Pour le premier couplage, c'est-à-dire les tâches puis le déplacement, un plan abstrait permettant de réaliser la mission est défini, dans un premier temps. Puis, ce plan est raffiné, c'est-à-dire que les déplacements du robot sont calculés en tenant compte des informations géométriques de l'environnement et du modèle du robot.

Pour le deuxième couplage, c'est-à-dire générer le déplacement puis les tâches, les architectures incluent des pré-calculs géométriques qui produisent un graphe. Les nœuds sont calculés à partir des objectifs courants et des contraintes géométriques sur les actions. Les arcs sont calculés en utilisant une méthode de planification de déplacements. Le principal inconvénient de cette approche est que les calculs géométriques sont effectués pour toutes les actions possibles même celles qui ne seront pas sélectionnées dans la solution finale.

Le principe d'un modèle hybride est qu'il utilise un ensemble de modules spécialisés pour augmenter le pouvoir d'expression et les capacités de raisonnement du planificateur de tâches. Les différents raisonneurs, dits spécialistes et le planificateur de tâches utilise un modèle central qui contient les données symboliques ainsi que les données géométriques du problème à résoudre. Le système résout le problème d'une manière incrémentale : le planificateur de tâches produit un premier plan d'actions. Puis ce plan est vérifié par les spécialistes qui vont ajouter un ensemble de contraintes entre les actions en cas d'échec. À partir de ces contraintes, le planificateur de tâches révisé le plan puis le re-soumet à l'ensemble de spécialistes.

1.5.3 Formalisation de la connaissance spatiale

Un planificateur nécessite une connaissance représentée à l'aide d'un langage formel comme par exemple le langage PDDL (section 1.4). La connaissance spatiale est nécessaire pour la planification dans certains problèmes.

Avec PDDL nous pouvons formaliser certaines connaissances spatiales, comme, l'expression « l'humain est dans le bureau » à l'aide du prédicat `at` tel que `(at h b)` où `h` est un paramètre de type humain et `b` est un paramètre de type bureau. Certains opérateurs nécessitent l'expression de l'information spatiale dans les préconditions. Par exemple, pour que l'action « prendre un objet cible » (définie par l'opérateur `take`) soit exécutable il faut que l'agent soit à une distance entre 10 et 20 centimètres de la cible, ceci se traduit par `distance(?a,?t) >= 10)(distance(?a,?t) <= 20)`. Des descriptions spatiales sont possibles avec PDDL. Cependant, la sémantique spatiale est absente. En d'autres termes, il n'y a pas de distinction entre la connaissance symbolique et la connaissance spatiale.

Le travail [Guitton, 2008] a permis de faire cette distinction. Dans ce travail, l'opérateur a été étendu en introduisant les pré-conditions et effets géométriques afin de prendre en compte la dimension spatiale (définition 9).

Définition 9 *Un opérateur o est défini par un quadruplet tel que :*

$$o = (name(o), symb_pre(o), geom_pre(o), symb_eff(o), geom_eff(o)) :$$

- $name(o)$ est le nom d'opérateur ;
- $symb_pre(o)$ est l'ensemble des pré-conditions symboliques ;
- $geom_pre(o)$ est l'ensemble des pré-conditions géométriques ;
- $symb_eff(o)$ est l'ensemble des effets symboliques ;
- $geom_eff(o)$ est l'ensemble des effets et références géométriques.

Les différents ensembles d'un opérateur de planification peuvent être vides. Par exemple, celui-ci peut avoir des préconditions géométriques mais pas de préconditions symboliques, ou encore il peut définir des effets uniquement symboliques.

```
;; Opérateur !take
(Operator (!pick ?r ?t)
;; préconditions symboliques
((robot ?r) (target ?t))
;; préconditions d'attitude
((robot ?r)(target ?t) (distance(?r,?t) >= 10)(distance(?r,?t) <= 20))
;; préconditions de comportements
((until(duration,10))(constant(?r.speed)))
;; effet symboliques
(( holding ?r ?o))
;; effets géométriques
()
)
```

CODE 1.7 - Opérateur étendu **take**.

L'opérateur étendu **take** (code 1.7) est exécuté lorsque l'agent r veut prendre un objet. Dans la définition de cet opérateur, les préconditions d'attitude spécifient que pour ramasser une cible, le robot doit se placer à une distance comprise entre 10 et 20 unités de la cible. Aussi, il doit maintenir sa vitesse constante durant le temps de l'action soit 10 unités de temps.

Le problème de ce travail, est qu'il définit son propre langage, ce qui le rend très restrictif pour les chercheurs de la communauté de planification classique. Notre idée est de proposer un formalisme comme PDDL 2.0 (intégrant la dimension temporelle) que toute la communauté scientifique a pu exploiter et enrichir. De plus, exprimer la dimension spatiale dans les

préconditions et les effets géométriques ne permet pas de considérer l'environnement. En effet, les préconditions géométriques permettent de décrire la position et l'orientation que le robot doit avoir pour accomplir son action. En d'autres termes, ces préconditions expriment le comportement que l'agent doit avoir en termes de mouvements, au cours de l'exécution des actions. Les effets géométriques donnent les valeurs émises par le planificateur de mouvement au planificateur de tâches. De plus, les éléments géométriques considèrent seulement les informations spatiales numériques. Ceci n'est pas suffisant. En effet, plusieurs autres représentations sont possibles pour la connaissance spatiale, telles que la représentation topologique, hiérarchique, floue, etc.

Conclusion

Nous avons présenté dans ce chapitre le vocabulaire de la planification nécessaire pour la suite de ce document. Nous nous sommes intéressés à deux types de planificateurs : les planificateurs de tâches et de chemins. Lorsque nous couplons ces deux types de planificateurs, nous pouvons exploiter l'information spatiale pour résoudre des problèmes de planification plus complexes. Plusieurs travaux de planification existent actuellement en robotique et permettent d'intégrer cette connaissance [Hoffmann *et al.*, 2006]. Cependant, ils ne considèrent pas la dimension spatiale. En effet, la connaissance spatiale se limite à la localisation des obstacles, à les éviter et à la définition du comportement de l'agent (vitesse, distance, etc.). De plus, la représentation de cette connaissance est limitée à une représentation numérique. La connaissance spatiale est plus complexe et peut avoir d'autres formes de représentations (chapitre 2). Celles-ci peuvent apporter plus d'expressivité à la planification ce que nous détaillons dans les chapitres 5 et 7. Comme langage de définition, nous avons opté le langage PDDL dont nous présentons une extension tenant compte de la connaissance spatiale (chapitre 5).

Chapitre 2

Représentation et raisonnement spatial

Sommaire

2.1	Connaissance spatiale	32
2.1.1	Représentation spatiale	32
2.1.2	Raisonnement spatial	33
2.2	Représentation d’une entité spatiale	34
2.2.1	Représentation géométrique	34
2.2.2	Représentation hiérarchique	34
2.3	Représentation des relations spatiales	35
2.3.1	Relations topologiques	36
2.3.2	Relation de distance	38
2.3.3	Cadre de référence	39
2.4	Approches quantitatives pour le raisonnement spatial	41
2.4.1	Modèle raster/Modèle vectoriel	41
2.4.2	Types primitifs	42
2.5	Approches pour le raisonnement spatial qualitatif	42
2.5.1	Formalismes inspirées des relations temporelles d’Allen	43
2.5.2	Formalismes fondés sur d’autres techniques	49
2.6	Connaissance spatiale en planification	51
2.6.1	Carte métrique, carte topologique, carte hybride et grille d’occupation	52
2.6.2	Carte sémantique	53

Ce chapitre s’intéresse à la représentation et l’exploitation de la connaissance spatiale. Nous montrons l’utilité de la connaissance spatiale et les différentes représentations que nous considérons (section 2.1). Nous définissons deux éléments pour le raisonnement spatial : les entités spatiales et les relations spatiales (sections 2.2 et 2.3). Nous présentons des approches pour le raisonnement spatial quantitatif (section 2.4) et qualitatif (section 2.5). Enfin, nous abordons la prise en compte de la connaissance spatiale en planification (section 2.6).



2.1 Connaissance spatiale

Représenter l'espace permet de déterminer les informations dont nous avons besoin pour évoluer dans celui-ci. En effet, la représentation spatiale joue un rôle important, notamment, pour nous aider à se situer. Dans la vie de tous les jours, des outils comme des cartes géographiques, GPS ou encore boussole, nous permettent de planifier un chemin entre un point de départ et un point d'arrivée ou de se situer dans une ville, etc.

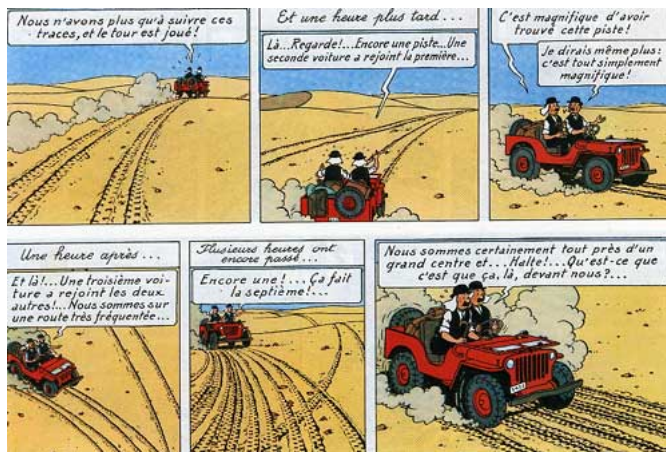


FIGURE 2.1 – Conséquence d'une mauvaise exploitation de la connaissance spatiale dans « Tintin au pays de l'or noir » (pages 29 et 30).

Une mauvaise exploitation (ou un mauvais raisonnement) de la connaissance spatiale peut induire en erreur. Considérons l'exemple décrit dans la figure 2.1. Dans cet exemple, Dupond et Dupont sont perdus dans le désert puisqu'ils ont mal interprété l'information spatiale donnée par les traces de leur propre voiture.

2.1.1 Représentation spatiale

La représentation spatiale consiste à modéliser l'information spatiale afin de pouvoir l'exploiter. Dans tout système de représentation de connaissance, il faut selon les objectifs des recherches conduites, définir en préalable ce que l'on considère comme données que nous voulons créer, analyser et manipuler. En représentation spatiale, pour analyser une scène, nous considérons l'**entité spatiale** et les **relations spatiales**.

La représentation spatiale d'un espace consiste à décrire les entités spatiales qui y se trouvent, ainsi que de décrire les relations spatiales entre ces entités spatiales. Selon [Kuipers et Levitt, 1988], il existe deux types de relations spatiales : les relations topologiques et les relations métriques. Par exemple, « la Seine traverse Paris », « traverse » est une relation spatiale topologique ; ou encore dans « Caen est à 233 km de Paris », « être à 233 km » est une relation spatiale métrique indiquant une distance. Nous retenons ces deux types de relations. Il existe plusieurs techniques de représentations des relations spatiales que nous détaillons dans la suite.

La connaissance spatiale peut être floue. Par exemple, les frontières d'une entité spatiale ne sont pas précises ; ou encore une relation spatiale peut être imprécise (le terme « proche » est une information spatiale imprécise. Son sens dépend du contexte dans lequel les objets sont considérés). Dans la littérature, trois classes de méthodes pour représenter l'information spatiale floue sont proposées. La première classe regroupe les méthodes fondées sur le traitement d'une forme simplifiée déduite de la forme originale de chaque entité spatiale. La forme originale des entités est négligée au profit d'une forme plus simple à traiter. Par exemple, certaines méthodes remplacent le problème d'évaluation des relations spatiales entre entités spatiales par celui du traitement de deux points issus des deux régions (centre de symétrie de gravité, etc.) [Keller et Wang, 1996]. Les résultats de ces méthodes ne sont pas fiables dès qu'il s'agit de traiter des entités spatiales ayant des formes complexes [Bloch, 1996]. La deuxième classe regroupe les méthodes fondées sur le passage par une représentation simplifiée de l'information spatiale. Les différentes relations spatiales sont évaluées directement à partir de cette représentation. La troisième classe regroupe les méthodes qui n'appartiennent ni à la première ni à la deuxième classe. Ces méthodes permettent une évaluation des relations spatiales directement à partir du traitement de la position relative de l'ensemble des points des deux régions. Dans cette thèse, nous considérons des entités de formes simplifiées. Ainsi, le flou est lié uniquement à la définition des relations spatiales.

2.1.2 Raisonnement spatial

Le raisonnement spatial recouvre tous les moyens dont dispose un individu pour traiter les informations spatiales d'un environnement afin de s'orienter dans ce dernier. Les techniques de raisonnement sont étroitement liées aux techniques de représentation. En effet, une technique de raisonnement est conditionnée par le choix de la méthode de représentation. Plusieurs travaux ont mis l'accent sur la grande complexité des mécanismes de raisonnement spatial. Deux familles de méthodes sont distinguées dans [Dehak, 2002] : les méthodes qualitatives souvent liées aux domaines des systèmes d'informations géographiques (SIG) [Frank, 1992] ou traitement automatique des langues (TAL) [Ligozat, 1998] et les méthodes quantitatives qui sont plutôt liées au traitement d'images [Krishnapuram *et al.*, 1993] ou la robotique [Kuipers, 2000]. Le raisonnement spatial soulève des problèmes algorithmiques et est traité de façon très différente dans les approches qualitatives et quantitatives.

Dans certaines applications il est nécessaire de combiner les approches qualitatives et quantitatives. Par exemple, considérons le problème décrit par la figure 1.4 (page 23). Pour réaliser une mission, le robot doit se déplacer dans l'environnement (figure 1.4, page 23). Pour ce déplacement, nous avons la contrainte du plus court chemin. Pour respecter la contrainte de parcourir le plus court chemin, il est nécessaire de connaître la distance d'une manière numérique. Ce genre d'information est donné par des relations métriques décrivant la distance (le côté quantitatif). Cependant, cette information n'est pas suffisante. En effet, il est nécessaire de connaître les points d'accès entre les pièces. Ce genre d'information est donné par les relations topologiques

décrivant l'adjacence (le côté qualitatif). La section 2.6 présente les travaux en planification combinant les approches qualitatives et quantitatives.

2.2 Représentation d'une entité spatiale

Une entité spatiale est un objet de l'espace pouvant être dynamique (humain, robot, etc.) ou statique (une ville, un quartier, etc.). Une entité spatiale est localisable dans un espace donné par un de ses attributs ou par une définition géométrique. Par exemple, une entité spatiale comme une maison peut être localisée soit par son adresse (attribut), soit par un polygone (définition géométrique). Les entités spatiales peuvent être décrites d'une manière hiérarchique. Par exemple, un bâtiment est un ensemble d'étages, un étage est un ensemble de bureaux, etc. Ce mode de représentation, permet une description à plusieurs niveaux de détails. Ce qui permet de raisonner d'une manière plus efficace.

2.2.1 Représentation géométrique

La description d'un espace peut se faire de deux manières. La première manière se base sur la description physique des entités spatiales, par exemple, une ville, un bâtiment, etc. Ce genre de définition est généralement universelle et univoque. La deuxième manière décrit l'espace sans limites claires, c'est-à-dire, les entités spatiales ne possèdent pas des frontières bien limitées. Selon [Green, 1987], une entité spatiale est identifiée par une frontière et une surface, la frontière a une longueur, une largeur, une orientation, etc. Ainsi, nous considérons que toute entité spatiale est identifiée par un ensemble d'attributs qui permettent de la décrire d'une manière géométrique (point, ligne, polygone, etc.).

Selon [Saval *et al.*, 2009], il existe deux types d'entités spatiales :

- entité spatiale absolue : référence qui peut être directement localisées géographiquement ;
- entité spatiale relative : composée d'entités spatiales absolues et de relations spatiales.

Une entité spatiale peut être définie en fonction d'autres entités spatiales. Par exemple : l'entité spatiale « proche de Caen » est une entité relative composée de l'entité absolue « Caen » et de la relation spatiale « proche » [Saval *et al.*, 2009].

2.2.2 Représentation hiérarchique

En informatique, « diviser pour régner » est une méthode de conception d'algorithme réduisant récursivement un problème en sous-problèmes. La hiérarchisation d'un espace donnée est un cas particulier de la stratégie « diviser pour régner ».

La hiérarchisation de l'espace se base sur **l'abstraction**. Par exemple, pour désigner un bâtiment à une autre personne, nous pouvons lui dire qu'il se compose de quatre étages au lieu de lui donner sa taille précise. Dans [Frommberger et Wolter, 2008], l'abstraction est un processus ou un résultat de la réduction d'une observation donnée dans le but de faire une classification qui omet toute la connaissance non nécessaire pour une application donnée.

Pour illustrer l'exploitation d'une telle abstraction, considérons le problème suivant : nous voulons visiter la « cité des arts et des sciences à Valence en Espagne ». Notre point de départ est le « 10 boulevard Maréchal Juin à Caen en France ». Le moyen de transport est la voiture. Pour trouver un chemin entre le point de départ et le point d'arrivée, la recherche se fait en trois étapes :

1. atteindre le périphérique de Caen à partir de l'adresse de départ, à savoir *10 boulevard Maréchal Juin* ;
2. planifier jusqu'au périphérique de Valence ;
3. atteindre la cité des arts et des sciences.

La première et la troisième étape nécessitent une carte de routes détaillée pour chacune des deux villes. La deuxième étape nécessite une carte de routes moins détaillée qui connecte les villes en faisant abstraction de leurs détails. Ainsi, l'abstraction réduit la quantité d'information considérée pour faire face à un monde fortement détaillé et complexe (la carte de l'Europe).

Pour la représentation spatiale, l'abstraction consiste à regrouper les éléments spatiaux dans des éléments plus généraux qui peuvent être abstraits. Le résultat est une **hiérarchie d'éléments** qui se termine quand toute l'information spatiale est modélisée par un seul élément. Cette abstraction fournit différents **niveaux hiérarchiques** qui représentent l'espace du niveau le plus abstrait au niveau le moins abstrait.

[Fernandez et Gonzalez, 2001] définissent un modèle mathématique appelé **AH-graph**. Ce modèle est une représentation graphique de l'environnement, qui le représente hiérarchiquement en se basant sur un mécanisme d'abstraction. Chaque niveau est un graphe (ou *flat graph*). De tels graphes représentent efficacement les informations relationnelles. Ils sont employés pour modéliser les environnements physiques, les sommets représentent les objets de l'espace et leurs arcs représentent des relations entre sommets.

À partir de la vue proposée par la figure 2.2(a), la figure 2.2(b) est un AH-graph définissant une abstraction possible des objets spatiaux dans la pièce. Elle montre les niveaux hiérarchiques modélisant l'espace, chaque niveau est un graphe. La figure 2.2(c) montre la hiérarchie résultante.

Le niveau le plus bas représente le monde avec la quantité maximale de détails disponibles. Le niveau hiérarchique le plus élevé s'appelle le niveau universel (*universal level*) est le niveau le plus abstrait.

Un groupe de sommets d'un niveau hiérarchique peut être abstrait en un seul sommet d'un niveau hiérarchique plus élevé appelé super-nœud. Les sommets originaux s'appellent les sous-nœuds de ce super-nœud. De façon analogue, un groupe d'arcs d'un niveau hiérarchique peut être représenté par un seul arc appelé super-arc au niveau plus haut.

2.3 Représentation des relations spatiales

Les relations spatiales sont au cœur du raisonnement spatial. En effet, la modélisation des relations spatiales est un sujet de recherche qui a attiré l'intérêt des communautés scientifiques

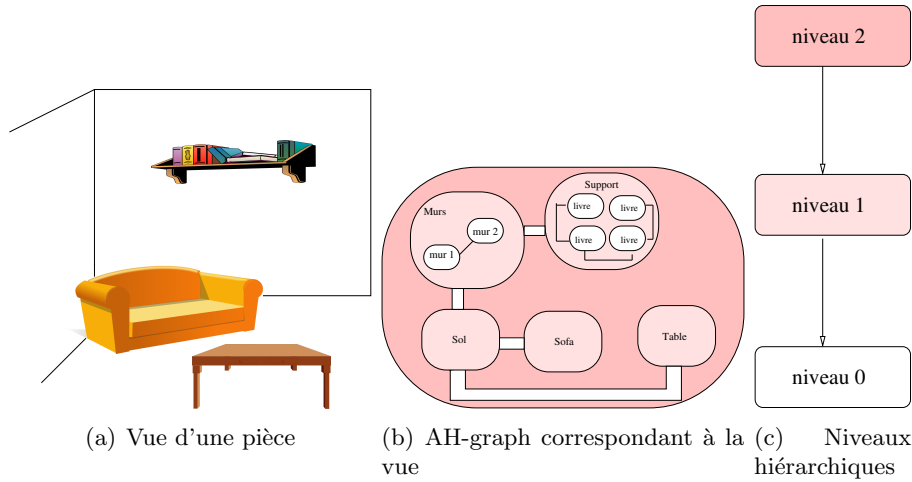


FIGURE 2.2 – Exemple d'abstraction d'une scène en utilisant un AH-graph : de la vue d'une pièce aux niveaux hiérarchiques correspondants. Les éléments du AH-graph sont coloriés différemment en fonction du niveau hiérarchique d'appartenance.

de différents domaines comme la linguistique [Lautenschütz *et al.*, 2007], la philosophie [Casati et Varzi, 1999], la psychologie [Ishikawa et Montello, 2006], etc. En informatique, différents domaines impliquent la recherche sur les relations spatiales, tels que les bases de données spatiales [Güting, 1994], les Systèmes d'Informations Géographiques (SIG) [Egenhofer *et al.*, 1995], le traitement d'image [Bloch, 2002], la géolocalisation, etc.

Les relations spatiales représentent l'ensemble de caractéristiques déterminant la position de deux ou plusieurs objets. Dans nos travaux, nous considérons les relations topologiques (des relations d'adjacence et les relations d'orientation) et les relations métriques. Les relations métriques ce sont des relations qui ont un lien direct avec la distance séparant deux objets [Dehak, 2002]. Les relations de distance peuvent être décrites d'une manière exacte quantitative (numérique) ou d'une manière qualitative pouvant être imprécise et floue.

2.3.1 Relations topologiques

La notion de topologie se rapporte à une démarche de notre esprit pour gérer l'espace. Lorsque nous observons un paysage, un lieu ou une carte notre première vision est globale. Les éléments tels qu'une forêt, un bâtiment ou une agglomération sont considérés dans leur contexte, c'est-à-dire les objets environnants. Par exemple dans « une rivière traverse une agglomération » ou encore « le jardin de M. Canu est adjacent à celui de M. Lesner », les objets sont situés par rapport à ceux qui les entourent. La topologie est donc l'ensemble des relations perçues qui nous permettent de situer les objets les uns par rapport aux autres.

Les relations topologiques sont largement décrites dans la littérature [Clementini *et al.*, 1993, Egenhofer et Herring, 1994, Cohn *et al.*, 1997]. Elles expriment les notions : d'adjacence,

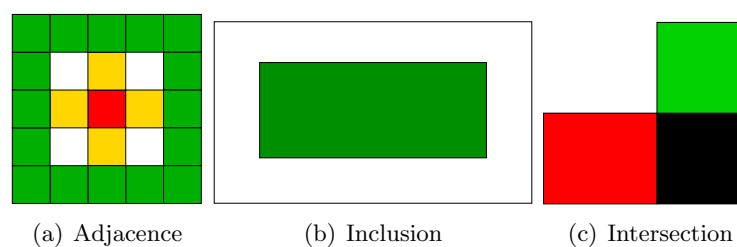


FIGURE 2.3 – Différents types de relations topologiques.

d'inclusion ou encore d'intersection (figure 2.3). Ces relations permettent d'extraire des relations de voisinage (dans une base de données spatiales, de connectivité dans un réseau, etc.).

L'adjacence

Elle implique que les objets spatiaux possèdent en commun un côté ou un sommet. La figure 2.3(a) montre deux types d'adjacence par rapport à une cible (le carré rouge) :

- au sens strict si une seule ligne frontière est commune (c'est le cas des carrés jaunes),
- au sens large si au moins un sommet est commun (c'est le cas des carrés blancs).

L'adjacence donne un ordre à un ensemble d'objets par rapport à une cible appelée « ordre d'adjacence ». Cet ordre intervient, par exemple, dans les transports pour déterminer le nombre de correspondances nécessaires pour se déplacer d'un endroit à un autre. L'ordre d'adjacence est l'ordre dans lequel les objets se rencontrent en s'éloignant de la cible. Dans les SIG, l'adjacence est dite de 1^{er} ordre si les deux entités sont en contact (dans la figure 2.3(a), elle concerne les carrés jaunes et blancs), de 2^{ème} ordre si une entité s'intercale et ainsi de suite (dans la figure 2.3(a), elle concerne les carrés verts).

L'inclusion

Elle est employée dans plusieurs domaines. En mathématiques, l'inclusion est une relation binaire qui indique qu'un ensemble est un sous-ensemble d'un second. D'un point de vue topologique, la notion d'inclusion implique qu'un objet spatial est totalement situé dans un autre. L'inclusion est un cas particulier adjacence (figure 2.3(b)) : le rectangle vert est inclut dans le rectangle blanc).

L'intersection

C'est une notion de géométrie qui désigne le lieu de rencontre de plusieurs formes telles que droites, courbes, plans, surfaces et volumes. Plus généralement, l'intersection est une opération ensembliste qui porte le même nom que son résultat, à savoir l'ensemble des éléments appartenant à plusieurs ensembles. D'un point de vue topologique, une intersection exprime le point ou la surface commune à deux entités spatiales (figure 2.3(c)) : le rectangle rouge et vert s'intersectent, leur intersection est le rectangle noir).

2.3.2 Relation de distance

La distance est une mesure de l'écart spatial entre deux lieux [Pumain *et al.*, 1997]. La relation de distance exprime la connaissance concernant la disposition des objets spatiaux dans un espace. D'un point de vue quantitatif, la distance est définie d'une manière numérique. D'un point de vue qualitatif, elle est définie par des termes tels que « proche », « loin », etc. Ces termes donnent une définition floue de la distance.

Représentation numérique

La distance numérique est quantifiée de manière à permettre des évaluations objectives. On parle de distances numérique. En mathématiques, une distance est une application qui formalise la longueur qui sépare deux points. La distance numérique est une connaissance spatiale très importante. Elle exprime la connaissance concernant la disposition des objets spatiaux dans une scène en estimant numériquement les distances qui les séparent. Sur une carte, la distance numérique est mesurée en unités de longueur, le plus souvent des mètres (des degrés de latitude ou de longitude n'ayant pas toujours la même valeur). La distance est mesurée par rapport à un repère (à un centre, à une voie de circulation, etc.) ou caractérise des couples de lieux (espacement ou distance moyenne).

Représentation floue

Nous nous intéressons aux relations de distance floues telles que « proche », « loin », etc. Les relations de distance floues ont essentiellement été traitées par des approches quantitatives. En effet, différentes méthodes pour la quantification des relations spatiales (en particulier la distance et l'orientation) ont été développées dans la littérature [Koczy, 1988], [Bloch, 1999], [Keller et Wang, 1996], [Dehak, 2002], etc.

La plupart des approches consistent à définir les relations de distance entre des objets flous. Selon [Bloch, 1999], il existe différentes catégories de relations entre des objets flous, notamment ; les distances entre deux points dans un ensemble flou, les distances d'un point à un ensemble flou et les distances entre deux ensembles flous.

Dans nos travaux, nous considérons que les entités spatiales ne sont pas floues. Nous considérons que la relation de distance est elle même floue. Plusieurs approches concernant les relations spatiales floues ont été proposées dans la littérature. Nous pouvons citer : les approches qui s'appuient sur la définition d'une distance qui n'est pas floue et tentent de la généraliser ([Bloch, 1999]), les approches qui déduisent une distance d'une fonction de similarité ([Jacas et Recasens, 2000]), les approches qui déduisent une distance d'un ensemble de relations entre deux ensembles ou d'autres types de relations ([Bouchon-Meunier *et al.*, 1996]) et les approches symboliques ([Schockaert, 2008], [Krishnapuram *et al.*, 1993]) sur lesquelles se fondent nos travaux.

Les approches symboliques ont pour but de définir des variables linguistiques représentant les distances, par exemple « proche », « moyen », « loin », etc. Plusieurs travaux dans la littérature définissent et exploitent les approches symboliques. Par exemple dans [Schockaert, 2008] ou

encore [Krishnapuram *et al.*, 1993], ces approches représentent les relations NEAR et FAR par des degrés calculés à partir de la distance entre deux points et/ou des points de deux régions.

2.3.3 Cadre de référence

Quelque soit le type de la relation spatiale (topologique ou de distance), la notion de **cadre de référence** est nécessaire. Pour souligner l'importance du cadre de référence, prenons l'énoncé suivant : « deux observateurs debout de part et d'autre d'une route sur l'axe nord-sud et une voiture qui roule vers le sud » (figure 2.4). La figure 2.4 montre que pour la personne à l'est de la route, la voiture se déplace vers la droite ; et à gauche pour celle à l'ouest.

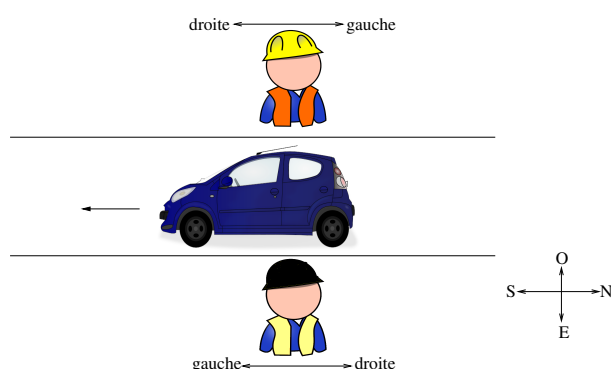


FIGURE 2.4 – Exemple de cadre de référence pour deux agents observant la même voiture, deux représentations propres

Cette différence dans la représentation d'une même scène est due à l'utilisation de différents cadres de référence propres à chaque observateur. Les préférences de l'observateur influant sur le choix du cadre de référence ([Levelt, 1982], [Tenbrink *et al.*, 2007]), une représentation spatiale commune et sans ambiguïté doit être définie selon un cadre de référence commun.

notation	sémantique associée
F	figure ou référent
G	fond ou relatum
V	point de vue
X	origine du système de coordonnées ;
X2	origine secondaire
A	point d'ancrage pour fixer les coordonnées étiquetées
L	repère désigné
A	point d'ancrage, (équivalent à G ou V (en temps normal), à L dans les systèmes utilisant des repères)
Pente	direction définie par une droite ou un système de points

TABLE 2.1 – Primitives descriptives dans [Levinson, 1996].

La notion de cadre de référence désigne la manière dont une scène est structurée et dont les représentations spatiales sont encodées. [Levinson, 1996] propose trois cadres de référence décrits dans la table 2.2. Ces cadres sont exprimés en fonction du formalisme défini dans la table 2.1.

Cadre de référence	Caractéristiques			Exemple
	Relation	Origine	Ancré par	
<i>cadre absolu</i>	binaire	relatum G	« Pente »	L'homme est au nord de la maison (figure 2.5(a))
<i>cadre intrinsèque</i>	binaire	relatum G	A sur G	L'homme est devant la maison (figure 2.5(b))
<i>cadre relatif</i>	ternaire	point de vue V	A sur V	L'homme est à gauche de la maison (figure 2.5(c))

TABLE 2.2 – Cadres de référence et exemples associés.

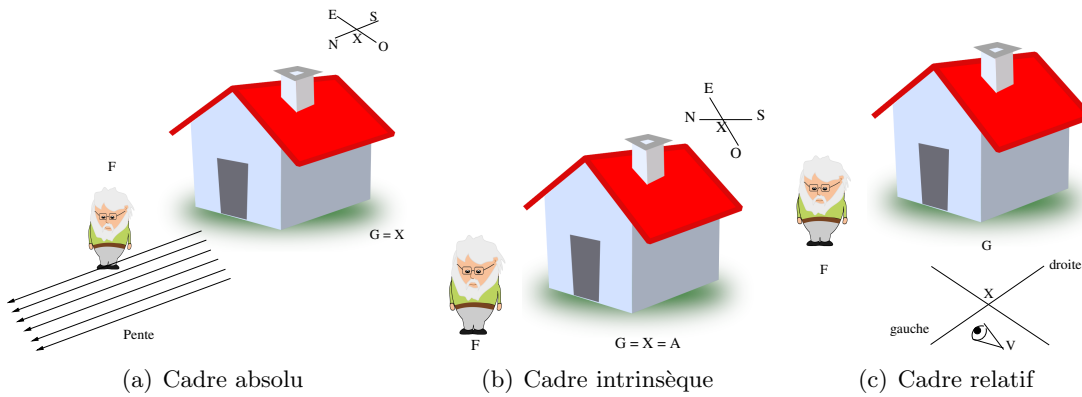


FIGURE 2.5 – Exemples illustrant le cadre de référence absolu, intrinsèque et relatif pour une même scène.

les cadres de références absolus permettent la représentation de relations binaires entre un référent (**F**) et un *relatum* (**G**) (figure 2.5(a)). L'origine est fixée sur le *relatum*. Le système de coordonnées est ancré par des relations fixes à l'environnement (points cardinaux, direction définie par la gravité, etc.).

Les cadres de références intrinsèques permettent la représentation d'une relation binaire entre un référent (**F**) et un *relatum* (**G**) (figure 2.5(b)). L'origine est fixée sur le *relatum*. Le système de coordonnées est ancré par les propriétés intrinsèques de ce *relatum*. Ces propriétés intrinsèques dépendant des propriétés géométriques des objets traités (par exemple : pour définir le « devant » d'une chaise).

Les cadres de références relatifs font intervenir une relation binaire entre un référent (**F**), un *relatum* (**G**) et un point de vue (**V**) (figure 2.5(c)). Le système primaire de coordonnées

a toujours son origine centrée sur un point de vue. Il peut exister un système de coordonnées secondaires avec une origine centrée sur le relatum. Le point de vue peut être fixe ou mobile.

Les cadres de référence par rapport à trois paramètres (l'observateur, le destinataire et une entité spatiale) peuvent être d'après [Trafton *et al.*, 2005] : centré sur soi (*egocentric*), centré sur l'interlocuteur (*addressee-centered*) ou centré sur l'objet décrit (*object-centered*).

2.4 Approches quantitatives pour le raisonnement spatial

En terme de représentation spatiale, les systèmes d'information géographique (SIG) offrent une longue tradition d'intégration, de gestion et d'analyse efficaces des données spatiales quantitatives, modélisées dans un espace de coordonnées cartésien [MIRON, 2009]. La représentation quantitative de la connaissance spatiale permet de représenter tous les éléments qui peuvent être exprimés d'une manière numérique en respectant les mesures prédéfinies. Par exemple, une entité spatiale est à 46 degré par rapport à une ligne de vue et à 192 mètres de ma position.

L'un des problèmes les plus connus nécessitant un raisonnement spatial quantitatif, est la recherche du chemin (*Pathfinding*). Plusieurs algorithmes ont été proposés pour ce genre de problème. En effet, le raisonnement quantitatif a plusieurs avantages. Par exemple, les méthodes quantitatives permettent de réduire les grands ensembles de données à une plus petite quantité d'information plus significative. Ceci est important dans l'analyse des données de plus en plus grandes obtenues à partir de diverses sources telles que les satellites dans les systèmes d'informations géographiques.

Dans ce qui suit, nous présentons deux modes de raisonnement spatial quantitatif : le modèle vectoriel [Gilbert, 2007], le modèle raster [Gisler *et al.*, 2011], ainsi que les types primitifs quantitatifs.

2.4.1 Modèle raster/Modèle vectoriel

Le modèle raster (ou modèle continu) représente la surface de la région d'intérêt par une succession régulière de cellules, selon un maillage défini. À chaque cellule sont associées une ou plusieurs valeurs d'attributs qui caractérisent la cellule en question. En conséquence, le modèle continu représente le modèle réel à travers un nombre fini de variables, chacune définit une cellule particulière. Le modèle raster est particulièrement utilisé comme fond de carte car il offre un rendu similaires aux cartes conventionnelles et a un grand impact visuel en réussissant à communiquer beaucoup d'informations [Longley, 2005].

Le modèle vectoriel (ou modèle discret) propose la modélisation des objets à travers la spécification de leurs contours en termes de points, de lignes ou de polygones, définis en coordonnées réelles (x,y). Chaque ligne est définie par une succession de points, et chaque surface (ou polygone) est définie par les arcs qui tracent ses limites. Ce modèle représente le monde comme un ensemble d'objets ayant des frontières très bien définies.

Dans [Hernández, 1994], les auteurs montrent l'intérêt du raisonnement spatial qualitatif à l'aide d'un exemple en décrivant une même scène d'une manière quantitative ensuite qualitative, comme suit :

- La description quantitative consiste en une boîte rectangulaire de 50 unités de large et de 40 unités de hauteur, contenant les objets suivants :
 1. un objet O_1 de 15 unités de large et 10 unités de hauteur à $35,30^\circ$;
 2. un objet O_2 de 6 unités de large et 5 unités de hauteur à $39,20^\circ$;
 3. un objet O_3 de 17 unités de large et 4 unités de hauteur à $8,36^\circ$;
 4. un objet O_4 de 6 unités de large et 16 unités de hauteur à $0,8^\circ$.
- La description qualitative est comme suit : dans une pièce, il y a un bureau au fond sous la fenêtre. À sa droite, une table d'ordinateur, à sa gauche un autre bureau et une bibliothèque.

La description qualitative permet : de définir le contexte (une pièce qui est un bureau) ; de définir les relations entre objets : à droite, à gauche ; de considérer différents niveaux de granularité (la pièce est décomposée en sous-régions par exemple « au fond sous la fenêtre ») .

Les relations topologiques sont de nature qualitative. En effet, elles n'ont pas besoin de mesures précises pour être expliquées [Egenhofer *et al.*, 1995]. La littérature du raisonnement spatial qualitatif est riche en travaux. Nous pouvons citer [Egenhofer et Franzosa, 1991], [Pigot, 1991], [Pullar et Egenhofer, 1988] et [Balbiani *et al.*, 1998] qui proposent des formalismes pour représenter les relations topologiques. Ces travaux peuvent être classés en deux catégories. Une première catégorie de formalismes s'inspirent des formalismes temporels [Pullar et Egenhofer, 1988], [Balbiani *et al.*, 1998] ; une autre regroupe des alternatives à ce type de représentation [Clementini *et al.*, 1993], [Randell *et al.*, 1992].

2.5.1 Formalismes inspirées des relations temporelles d'Allen

Plusieurs travaux sur les relations spatiales topologiques s'inspirent de l'algèbre des intervalles [Allen, 1983]. L'algèbre des intervalles [Allen, 1983] définit 13 relations atomiques qui constituent la liste exhaustive des relations qui peuvent exister entre deux intervalles (table 2.3).

Algèbre des rectangles

C'est un formalisme dédié au raisonnement spatial qualitatif [Balbiani *et al.*, 1998]. Dans ce formalisme, les objets spatiaux sont considérés comme des rectangles dont les côtés sont parallèles aux axes d'une base orthogonale d'un espace euclidien de dimension deux. Une relation atomique $\langle A, B \rangle$ est satisfaite entre deux rectangles x et y si les projections orthogonales de x et y sur le premier axe satisfont A et si les projections orthogonales de x et de y sur le deuxième axe satisfont B . L'ensemble des relations est l'ensemble $A_{rec} = \{\langle A, B \rangle ; A, B \in A_{int}\}$. Ces relations sont mutuellement exclusives. Chaque relation peut être considérée comme la disjonction des relations atomiques qui la composent. Étant donnés deux rectangles x et y et une relation

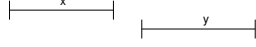
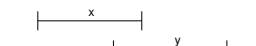
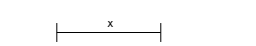
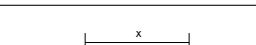
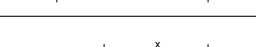
Relation	Symbole	Inverse	signification
précède	b	bi	
rencontre	m	mi	
chevauche	o	oi	
commence	s	si	
pendant	d	di	
termine	f	fi	
égale	eq	eq	

TABLE 2.3 – Les 13 relations d’Allen.

atomique R , xRy dénotera le fait que x et y satisfont R . La figure 2.7 présente deux régions x et y . Ces deux régions sont définies par des rectangles. La relation spatiale R entre x et y est définie par le couple $\langle m, b \rangle$.

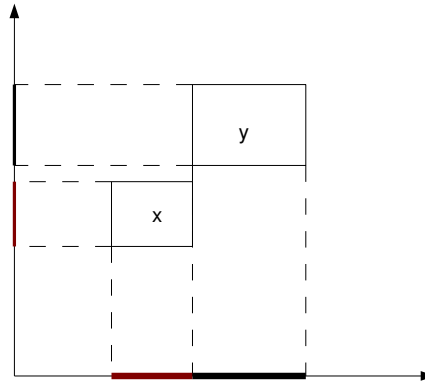


FIGURE 2.7 – Définition d’une relation topologique $\langle m, b \rangle$.

Le traitement des relations topologiques est facilité par la décomposition du problème en considérant les relations d’Allen sur chaque axe séparément. L’algèbre des rectangles est un formalisme qui permet d’exprimer 169 relations topologiques (13×13) (annexe B). Au delà de l’expression topologique, cette algèbre exprime aussi les relations d’orientation telles que « à droite », « en haut », « au même niveau », etc.

Un des aspects important dans le raisonnement est l'inférence, c'est-à-dire la déduction de nouvelles informations à partir d'informations existantes. Les relations topologiques définies dans l'algèbre des rectangles se basent sur les relations temporelles définies par Allen.

Pour déduire une nouvelle relation topologique à partir des relations existantes, l'algèbre des rectangles utilise la table de composition définie par Allen (annexe B). La figure 2.8 montre la relation inférée, en exploitant la table de composition (annexe B), entre le rectangle z et x à partir des relations existantes entre les rectangles z et y et les rectangles y et x .

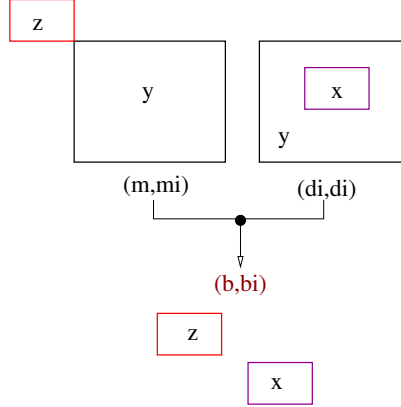


FIGURE 2.8 – Composition entre deux relations topologiques définies par l'algèbre des rectangles.

L'information spatiale entre plusieurs rectangles peut être représentée par un CSP binaire particulier : **un réseau de rectangles** [Balbiani *et al.*, 1998]. Un réseau de rectangles $\mathcal{N}$ est une structure de la forme $(\mathcal{V}, \mathcal{R})$, où :

1. $\mathcal{V}$ ($n = |\mathcal{V}|$ et $n \geq 1$) : est l'ensemble de variables $\{v_1, v_2, \dots, v_n\}$ représentant des rectangles,
2. $\mathcal{R}$: est l'ensemble des contraintes binaires entre deux rectangles. On peut voir $\mathcal{R}$ comme une application de $\mathcal{V} \times \mathcal{V}$ à un ensemble de relations définies par $2^{A_{rec}}$ tel que pour chaque $v_i \in \mathcal{V}$ $\mathcal{R}_{ii} = \{(eq, eq)\}$ et pour chaque $(v_i, v_j) \in \mathcal{V}$, $\mathcal{R}_{v_j v_i}$ est l'inverse de $\mathcal{R}_{v_i v_j}$.

Soit $(\mathcal{V}, \mathcal{R})$ un réseau de rectangles :

- il est dit *atomique* si et seulement si chacune de ses contraintes est composée d'une seule relation atomique ;
- il est *chemin consistant* lorsque pour chaque $v_i, v_j \in \mathcal{V}$, $\mathcal{R}_{v_i v_j} \neq \emptyset$ et pour chaque $v_i, v_j, v_k \in \mathcal{V}$, $\mathcal{R}_{v_i v_j} \circ \mathcal{R}_{v_j v_k} \supseteq \mathcal{R}_{v_i v_k}$;
- il est *consistant* lorsqu'il existe une application a de $\mathcal{V}$ à l'ensemble des relations entre deux rectangles, telle que pour chaque $v_i, v_j \in \mathcal{V}$, il existe une relation atomique A dans $\mathcal{A}_{int}$ et il existe une relation atomique B dans $\mathcal{A}_{int}$ tel que $(A, B) \in \mathcal{R}_{v_i v_j}$ et a_{v_i} et a_{v_j} satisfont (A, B) dans le modèle de relations de rectangles.

Le problème de savoir si un réseau de rectangles est consistant ou non est un problème **NP-complet** dans le cas le plus général. Dans [Balbiani *et al.*, 1998], les auteurs montrent

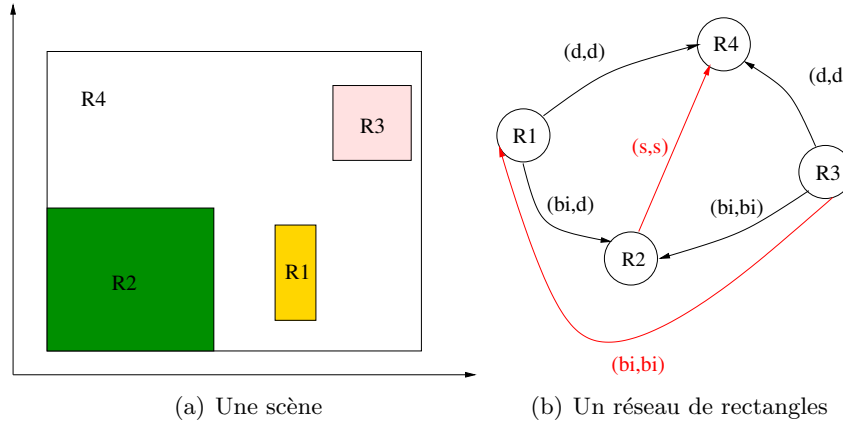


FIGURE 2.9 – Une scène définissant un ensemble d’entités spatiales représentées par des rectangles et son réseau de rectangles correspondant.

que pour la classe de relations PRECONVEX, le problème de vérification de la consistance est polynomial.

Méthode des intersections

En éliminant la notion d’ordre de l’algèbre d’intervalles [Allen, 1983], [Pullar et Egenhofer, 1988] ont défini un ensemble minimal de six relations entre deux entités spatiales. Une entité spatiale est définie par son intérieur et sa frontière. Les relations possibles sont : intersections entre frontières, entre intérieurs, entre frontière d’une entité et intérieur de l’autre, chacune de ses intersections pouvant être vide ou non vide. Ces relations ont permis la mise en place d’une approche pour représenter et raisonner sur des relations topologiques entre des régions : **la méthode de n-intersections** [Egenhofer et Franzosa, 1991, Mark et Egenhofer, 1994]. Il existe deux versions pour cette méthode : la méthode des 4-intersections et la méthode des 9-intersections.

Dans la méthode des 4-intersections (4IM), les relations topologiques entre deux objets X et Y sont définies en termes de quatre intersections entre la frontière de X (∂X), l’intérieur de X (X°) avec la frontière de Y (∂Y) et l’intérieur de Y (Y°) [Egenhofer et Franzosa, 1991]. Ce modèle est défini d’une manière concise avec une matrice 2×2 dite **4-intersections** ou M_4 . Il existe 16 ($= 2^4$) relations possibles définies par la matrice M_4 . Parmi, ces relations seulement 8 relations topologiques concernent des objets à deux dimensions [Egenhofer *et al.*, 1993] (table 2.4).

$$M_4 = \begin{pmatrix} \partial X \cap \partial Y & \partial X \cap Y^\circ \\ X^\circ \cap \partial Y & X^\circ \cap Y^\circ \end{pmatrix}$$

Dans la méthode des 9-intersections (9IM), les relations topologiques sont aussi définies entre un segment et une région. Chaque objet (O) étant représentée par son intérieur (O°), sa frontière (∂O) et son extérieur (O^-) [Mark et Egenhofer, 1994]. Ce modèle est défini d’une manière concise

par une matrice 3×3 dite **9-intersection** ou M_9 . Il existe 512 ($= 2^9$) relations possibles définies par la matrice M_9 . Parmi, ces relations seulement 8 relations topologiques concernent des objets à deux dimensions [Egenhofer *et al.*, 1993] (table 2.4).

$$M_9 = \begin{pmatrix} X^\circ \cap Y^\circ & X^\circ \cap \partial Y & X^\circ \cap Y^- \\ \partial X \cap Y^\circ & \partial X \cap \partial Y & \partial X \cap Y^- \\ X^- \cap Y^\circ & X^- \cap \partial Y & X^- \cap Y^- \end{pmatrix}$$

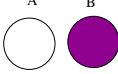
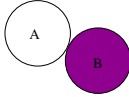
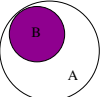
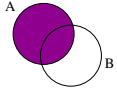
Relation	disjoint	contient	dans	égal
Matrice	$\begin{matrix} X^\circ & \\ \partial X & \end{matrix} \begin{pmatrix} Y^\circ & \partial Y \\ \oslash & \oslash \end{pmatrix}$	$\begin{matrix} X^\circ & \\ \partial X & \end{matrix} \begin{pmatrix} Y^\circ & \partial Y \\ \neq \oslash & \neq \oslash \end{pmatrix}$	$\begin{matrix} X^\circ & \\ \partial X & \end{matrix} \begin{pmatrix} Y^\circ & \partial Y \\ \neq \oslash & \oslash \end{pmatrix}$	$\begin{matrix} X^\circ & \\ \partial X & \end{matrix} \begin{pmatrix} Y^\circ & \partial Y \\ \neq \oslash & \neq \oslash \end{pmatrix}$
Sens				
Relation	contact	couvre	est couvert	chevauche
Matrice	$\begin{matrix} X^\circ & \\ \partial X & \end{matrix} \begin{pmatrix} Y^\circ & \partial Y \\ \oslash & \neq \oslash \end{pmatrix}$	$\begin{matrix} X^\circ & \\ \partial X & \end{matrix} \begin{pmatrix} Y^\circ & \partial Y \\ \neq \oslash & \neq \oslash \end{pmatrix}$	$\begin{matrix} X^\circ & \\ \partial X & \end{matrix} \begin{pmatrix} Y^\circ & \partial Y \\ \neq \oslash & \oslash \end{pmatrix}$	$\begin{matrix} X^\circ & \\ \partial X & \end{matrix} \begin{pmatrix} Y^\circ & \partial Y \\ \neq \oslash & \neq \oslash \end{pmatrix}$
Sens				

TABLE 2.4 – Les 8 relations topologiques entre deux régions.

Nous nous intéressons principalement au formalisme 9-intersections. Pour pouvoir faire un raisonnement efficace en utilisant le formalisme 9-intersections, [Kurata, 2009] propose deux variantes de cette approche qui se basent sur la classification des objets (points, lignes, régions, ...) pris en compte dans la relations : *homogenous 9-intersection calculi* qui cible un couple d'objets de même type et le *heterogenous 9-intersection calculi* qui cible des objets différents.

Dans ce qui suit un exemple tiré de [Kurata, 2009], pour illustrer l'application du formalisme 9-intersections dans le raisonnement spatial qualitatif.

Exemple 1 On considère la ville de Boston, avec quatre autoroutes : I-90, I-93, I-95 et I-495 (figure 2.10(a)). Le tableau³ 2.5 énumère les relations topologiques entre ces autoroutes.

Nous considérons ces quatre autoroutes et nous observons leurs connexions aux deux autres. Par exemple dans la figure 2.10(a), l'autoroute I-95 (représentée par une ligne L) traverse l'autoroute I-90 (représentée par une ligne L), ce qui explique la relation $cross_{LL}$. Nous considérons

3. Nous gardons les relations en anglais.

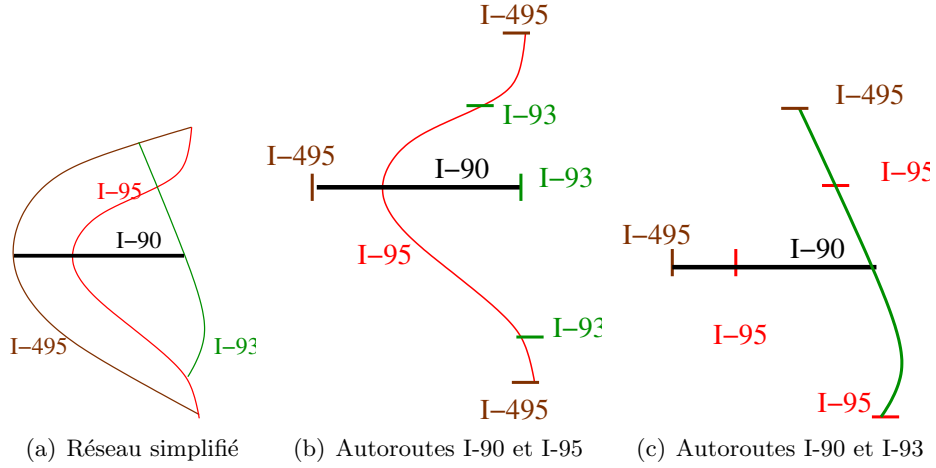


FIGURE 2.10 – Réseau simplifié des autoroutes de Boston et différente connaissance spatiale.

	I-90	I-93	I-95	I-495
I-90	-	diverges _{LL}	cross _{LL}	diverges _{LL}
I-93	divergedBy _{LL}	-	diverges _{LL} & cross _{LL}	diverges _{LL}
I-95	cross _{LL}	divergedBy _{LL} & cross _{LL}	-	meet-at-both-ends _{LL}
I-495	divergedBy _{LL}	divergedBy _{LL}	meet-at-both-ends _{LL}	-

TABLE 2.5 – Relations entre les autoroutes I-90, I-93, I-95 et I-495 [Kurata, 2009].

que la figure 2.10(b) illustre la connaissance acquise à partir de la conduite sur I-90 et I-95 et que la figure 2.10(c) illustre la connaissance acquise à partir de la conduite sur I-90 et I-93. Sur la base de ces connaissances, que pouvons-nous dire au sujet des relations entre les deux autres routes I-95 et I-93 ?

Le détails du raisonnement est décrit comme suit. Tout d'abord, toutes les relations entre des autoroutes non visitées sont remplacées par U_{LL} . Ensuite, nous exploitons la table de composition décrite dans la table 2.6.

	disjoint _{LP}	meet _{LP}	contains _{LP}
equal _{LL}	disjoint _{LP}	meet _{LP}	contains _{LP}
disjoint _{LL}	U_{LP}	disjoint _{LP}	disjoint _{LP}
meet _{LL}	U_{LP}	disjoint _{LP} / meet _{LP}	disjoint _{LP}
cross _{LL}	U_{LP}	U_{LP}	U_{LP}
covers _{LL}	disjoint _{LP}	disjoint _{LP} / meet _{LP}	U_{LP}
coverdBy _{LL}	U_{LP}	meet _{LP} / contains _{LP}	contains _{LP}
contains _{LL}	disjoint _{LP}	disjoint _{LP}	U_{LP}

TABLE 2.6 – La table de composition de relations topologiques entre des relations entre lignes (LL) et des relations entre une ligne et un point (LP).

Le résultat du calcul est affiché dans le tableau 2.7. Pour déduire les relations inconnues (table 2.7), ils utilisent SparQ [Dylla *et al.*, 2006] ils calculent tous les scénarios cohérents dans le cadre du réseau de contraintes en se basant selon le tableau 2.5.

Relation inconnue	Solution dérivée
(I-95,I-495)	All but equal, covers _{LL} , coveredBy _{LL} , contains _{LL} , inside _{LL}
(I-93,I-495)	disjoint _{LL} , cross _{LL} , diverges _{LL} , diverges _{LL} & cross _{LL}
(I-93,I-95)	disjoint _{LL} , cross _{LL} , diverges _{LL} , diverges _{LL} & cross _{LL}
(I-90,I-495)	disjoint _{LL} , cross _{LL} , diverges _{LL} , diverges _{LL} & cross _{LL}
(I-90,I-95)	disjoint _{LL} , cross _{LL} , diverges _{LL} , diverges _{LL} & cross _{LL}
(I-90,I-93)	disjoint _{LL} , cross _{LL} , coveredBy _{LL} , cross _{LL} , meet _{LL} , meet _{LL} & cross _{LL} , diverges _{LL} , diverges _{LL} & cross _{LL} , divergedBy _{LL} , divergedBy _{LL} & cross _{LL} , diverges _{LL} & meet _{LL} , diverges _{LL} & cross _{LL} & meet _{LL} , divergedBy _{LL} & meet _{LL} , divergedBy _{LL} & cross _{LL} & meet _{LL} , diverges _{LL} & divergedBy _{LL} , diverges _{LL} & cross _{LL} & divergedBy _{LL} , diverges _{LL} & divergedBy _{LL} & meet _{LL} , diverges _{LL} & cross _{LL} & divergedBy _{LL} & meet _{LL}

TABLE 2.7 – Les relations topologiques possibles déduites.

Pour le scénario de la figure 2.10, il existe quatre relations possibles ente I-93 et I-495 : disjoint_{LL}, cross_{LL}, diverges_{LL}, diverges_{LL}& cross_{LL} (tableau 2.7). Cette solution semble raisonnable, puisque :

1. les deux extrémités de la I-495 ne se croisent pas avec les I-93 ;
2. une extrémité de I-93 ne se croise pas avec I-495.

Pour le scénario de la figure 2.10(c), il existe 28 relations possibles (tableau 2.7) entre la I-95 et I-495. Puisqu'à partir de la connaissance donnée, nous pouvons dire que nous avons obtenu que jusqu'à 28 relations possibles entre la I-95 et I-495. Cependant, nous pouvons seulement dire que I-95 n'est ni contenu, ni couverts par la I-495 et inversement.

2.5.2 Formalismes fondés sur d'autres techniques

L'utilisation des intervalles est généralement inadaptées aux objets de formes complexes ou extrêmement proches les uns des autres. Ainsi, des travaux tel que [Clementini *et al.*, 1993] et [Randell *et al.*, 1992] ont proposé des relations topologiques entre régions composites et/ou complexes.

Méthode RCC-8

Le modèle de régions [Randell *et al.*, 1992] est le modèle le plus utilisé pour le raisonnement spatial. Les objets sont des régions de l'espace.

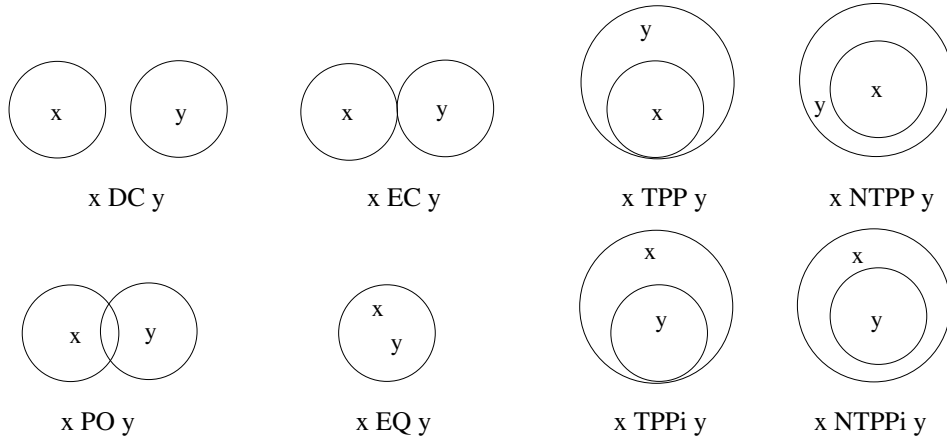


FIGURE 2.11 – Relations spatiales topologiques RCC-8.

Les relations spatiales entre deux régions sont données par la figure 2.11, où

- DC (disjonction) : les deux régions sont disjointes ;
- EC (contact extérieur) : les deux régions ne sont pas incluses l'une dans l'autre et leurs bordures sont en contact ;
- PO (recouvrement partiel) : les deux régions se chevauchent l'une avec l'autre ;
- TPP (partie propre tangente) : les deux régions sont incluses l'une dans l'autre et leurs bordures ne sont pas en contact ;
- TPP_i : la relation inverse de TPP.
- NTPP : (partie propre non tangente) : les deux régions sont incluses l'une dans l'autre et leurs bordures ne sont pas en contact ;
- NTPP_i : la relation inverse de NTPP ;
- EQ (identité) : les deux régions se confondent l'une avec l'autre.

Ce modèle, ne permet pas de raisonner sur des relations cardinales telles qu'un objet est au nord d'un autre objet, un objet est situé à droite d'un autre objet, etc. Ce qui est possible avec d'autres formalismes qualitatifs tel que l'algèbre des rectangles vu précédemment.

Le raisonnement avec le formalisme RCC-8 est difficile, notamment pour des raisons de complexité [Clementini et Di Felice, 1995a]. Ces difficultés peuvent en partie être contournées par l'utilisation de tables de composition pré-calculées ou encore en utilisant un treillis de prédicats comme dans RCC [Clementini et Di Felice, 1995a]. Dans ce qui suit, nous définissons un exemple permettant d'illustrer la méthode de raisonnement avec le formalisme RCC-8.

Exemple 2 *Considérons deux maisons M_1 et M_2 , chacune dans une parcelle P_1 pour M_1 et P_2 pour M_2 et une rue piétonne R_p tel que :*

- La maison M_1 est DC avec la maison M_2
- La maison M_1 est $\{TPP \text{ ou } NTPP\}$ avec la parcelle P_1
- La maison M_1 est $\{DC \text{ ou } EC\}$ avec la parcelle P_2
- La maison M_1 est EC avec la rue piétonne R_p

- La maison M_2 est $\{DC \text{ ou } EC\}$ avec la parcelle P_1
- La maison M_2 est $NTPP$ avec la parcelle P_2
- La maison M_2 est EC avec la rue piétonne R_p
- La parcelle P_1 est $\{DC \text{ ou } EC\}$ avec la parcelle P_2

Que pouvons-nous déduire sur la relation entre P_1 et R_p et entre P_2 et R_p ?

Nous pouvons représenter cet exemple à l'aide du réseau de contraintes défini dans la figure 2.12(a).

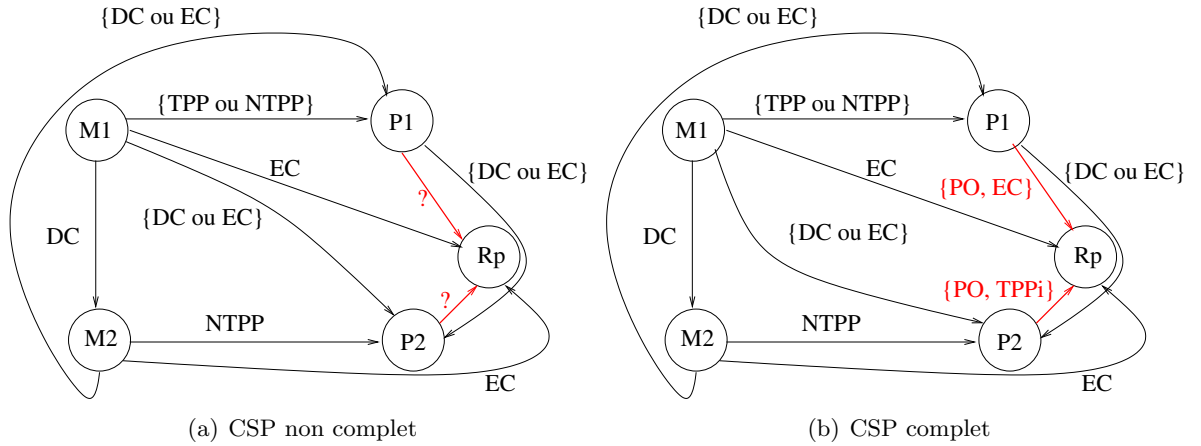


FIGURE 2.12 – Exemple de CSP non complet et complet.

Les relations possibles entre la parcelle P_1 et la rue piétonne R_p et entre la parcelle P_2 et la rue piétonne R_p sont $\{DC \text{ ou } EC \text{ ou } TPP \text{ ou } TPPI \text{ ou } PO \text{ ou } EQ \text{ ou } NTPP \text{ ou } NTPPI\}$, ce qui fait plusieurs possibilités qui n'apportent pas de solution forte et cohérente. En appliquant la table de composition de RCC-8 (voir annexe B) et les algorithmes de consistance de chemins [Chmeiss, 1996] (nous n'avons pas besoin de connaître le détail de ces algorithmes), nous pouvons donner une vision plus fine de la relation entre la rue piétonne R_p avec les parcelles P_1 et P_2 . La figure 2.12(b), illustre le réseau consistant donnant une solution à notre problème :

- entre la parcelle P_1 et la rue piétonne R_p : $\{PO \text{ ou } EC\}$
- entre la parcelle P_2 et la rue piétonne R_p : $\{PO \text{ ou } TPPI\}$

2.6 Connaissance spatiale en planification

En planification (de tâches, de chemins), en particulier en planification pour les robots mobiles, le raisonnement spatial s'avère très utile. En effet, la connaissance spatiale est nécessaire pour effectuer des opérations de base telles que la navigation ou la localisation, etc. Par exemple, étant donnée la tâche « prendre une bouteille de lait », un planificateur de tâche peut produire une séquence d'actions comme « aller à la cuisine, être en face du réfrigérateur, l'ouvrir, prendre

le lait et fermer le réfrigérateur » [Galindo *et al.*, 2008]. Pour exécuter ces actions, les connaissances spatiales suivantes sont nécessaires pour le planificateur :

- les connaissances qualitatives, par exemple, le lait se trouve dans le réfrigérateur, le réfrigérateur se trouve dans la cuisine ;
- les connaissances quantitatives, par exemple, pour calculer le chemin le plus court pour atteindre la cuisine.

Différents travaux en planification ont utilisé des cartes pour modéliser la connaissance spatiale, par exemple, MERCATOR [Davis, 1986] et RPLAN [Kortenkamp, 1993] et plus récemment StealthBot [Kennedy *et al.*, 2007] et [Vasudevan *et al.*, 2007]. Il existe des cartes qui permettent de modéliser les connaissances qualitatives, d'autres modélisent les connaissances quantitatives et certaines qui combinent les deux. Ces cartes sont spécialement conçues pour la localisation et la navigation. En effet, elles ne permettent pas d'exprimer la sémantique de l'information spatiale. D'autres cartes sont dédiées à ce type d'objectif. Les paragraphes suivants sont un état de l'art sur ces différentes cartes.

2.6.1 Carte métrique, carte topologique, carte hybride et grille d'occupation

Les connaissances sur la structure de l'espace considéré est généralement encodée sous forme d'une carte. La représentation, la construction et l'entretien des cartes a été l'un des domaines le plus actif de la recherche en planification pour la robotique mobile [Thrun, 2002]. Il existe plusieurs types de cartes, notamment : les cartes métrique (*metric map*) ([Arras, 2003]), les cartes topologiques (*topological map*) ([Angeli *et al.*, 2008]) et les cartes dites hybrides (*hybrid map*) ([Park *et al.*, 2009]).

Les cartes métrique représentent les caractéristiques géométriques de l'environnement. Des informations telles que la longueur, la distance, la position etc., apparaissent explicitement dans les cartes métriques et sont en général définies dans un référentiel unique [Thrun, 2002]. Par contre, elles sont moins adaptés pour représenter des informations qualitatives et pour représenter des grands environnements (pour cause du degré de détails).

Les cartes topologiques permettent une représentation plus abstraite décrivant les relations entre les éléments de l'environnement, sans utiliser un repère de référence [Fabrizi et Saffiotti, 2002]. Elles se présentent sous forme de graphes, dont les sommets correspondent à des lieux, souvent associés à des informations observées et dont les arêtes indiquent l'existence d'un chemin possible entre les lieux associés aux deux sommets de l'arête. Elles sont plus adaptées que les cartes métriques pour représenter des grands environnements et pour rajouter des informations qualitatives. Cependant, la plus grande limite de ces cartes est le manque d'information métrique. En effet, lors de la construction de ces cartes, la définition des différentes composantes nécessite l'utilisation d'information métrique pour décrire la géométrie des objets [Dufourd, 2005]. Aussi, dans le cas de la navigation, le manque d'information sur la longueur des chemins (sur les arcs) peut empêcher un agent de choisir entre deux branches du graphe qui mènent au même endroit.

Nous pouvons constater que les représentations métriques et topologiques s'avèrent souvent complémentaires. Une combinaison de ces deux représentations est susceptible de favoriser l'exploitation de la connaissance spatiale résultante. Les cartes hybrides exploitent les cartes métriques pour une navigation précise dans environnement local et exploitent les cartes topologiques pour se déplacer entre des régions et mener un raisonnement de haut niveau.

L'idée des cartes hybrides est d'utiliser deux (ou plusieurs) types de représentation ensemble, ce qui va permettre de profiter des points forts de chaque représentation, et pourrait aider à surmonter les points faibles. Par exemple, plusieurs auteurs ont proposé des méthodes hybrides métriques-topologiques dans l'intention de combiner la précision des cartes métriques avec l'extensibilité des cartes topologiques [Duckett et Saffiotti, 2000], [Tomatis, 2001], etc. Un exemple de carte hybride, la carte métrique et hybride (*HYbrid Metric Maps* (HYMMs)) [Nieto *et al.*, 2004]. La carte est représentée par un graphe de systèmes de coordonnées, telle que chaque sommet dans le graphe représente un repère local, et les arêtes représentent les transformations entre les repères adjacents.

Un autre mode de représentation de l'espace : la grille d'occupation. Les grilles d'occupation constituent une représentation surfacique simple et populaire. La grille d'occupation est une approche pour représenter une carte sous forme d'une grille régulière en cellules carrés ou rectangulaires de même taille. Chaque cellule contient un indice indiquant si l'espace correspondant est plutôt libre ou occupé [Baba, 2007]. Ainsi, elles sont souvent utilisées lorsque l'application visée repose sur la connaissance de l'espace libre, en particulier pour la planification de trajectoires.

Ce type de cartes est suffisant pour fournir la fonctionnalité de base : navigation. Ce qui n'est pas suffisant dans le cadre d'application que nous considérons. La carte sémantique (*semantic map*) est une première solution permettant la navigation et offrant d'autres fonctionnalités telle que l'organisation des objets dans l'espace [Nüchter et Hertzberg, 2008].

2.6.2 Carte sémantique

Plusieurs travaux se sont intéressés aux cartes sémantiques en particulier pour le développement de robots intelligents et pour améliorer l'interaction homme-robot, par exemple : [Meger *et al.*, 2008]. Dans ces travaux, le terme carte sémantique fait référence à une représentation spatiale de l'environnement qui inclut les types et les emplacements des objets qui s'y trouvent. Par exemple, une « cuisine » est un espace dans lequel se trouve un réfrigérateur.

D'autres travaux utilisent les cartes sémantiques pour améliorer l'interaction homme-robot en inférant de l'information sémantique. Par exemple le travail décrit dans [Galindo *et al.*, 2005] présente une carte sémantique hybride organisée en deux structures hiérarchiques. Cette carte permet au robot de représenter les pièces et les objets simples (par exemple une cuisinière) et d'en déduire des informations implicites.

Nous citons aussi les travaux de [Kruijff *et al.*, 2007], [Zender *et al.*, 2007], [Vasudevan *et al.*, 2007], consacrés à la définition des cartes sémantiques et qui ont développés des relations sémantiques et des mécanismes solides pour la construction de cartes sémantiques semi-

automatique. Cependant, aucun de ces travaux exploitent toutes les possibilités offertes par ces cartes sémantiques. En effet, ils utilisent l'information sémantique pour déduire des informations sur les objets acquis en fonction de leurs propriétés.

Dans ce type de carte, on apporte beaucoup d'importance à la connaissance sémantique (qualitative). En planification, il est nécessaire pour atteindre une mission de gérer la connaissance spatiale floue. Par exemple, certaines instructions à exécuter utilisent des indications spatiales floues exprimant des directions [Moratz *et al.*, 2003]. Cependant, la connaissance sémantique ne considère pas la connaissance spatiale floue. En effet, ce genre de carte est exploité dans des problèmes de connaissance et de localisation des objets, et ne permet pas d'exploiter réellement la connaissance spatiale.

Conclusion

Ce chapitre présente un panorama non exhaustif des méthodes pour représenter et exploiter la connaissance spatiale. La représentation de l'espace, consiste à décrire à la fois l'ensemble des objets (entités spatiales) qui s'y trouvent, ainsi que les relations existant entre eux (relations spatiales). Une entité spatiale, dans notre travail est décrite par des attributs permettant sa définition géométrique (plus de détails dans le chapitre 4). Pour une stratégie de simplification du raisonnement, nous présentons l'espace d'une manière hiérarchique. Nous considérons les relations topologiques et les relations de distance (numérique et floue).

Dans nos travaux, une relation topologique décrit des relations d'orientation ainsi qu'entre autre des relations d'adjacence et de recouvrement. Pour cela, nous utilisons l'algèbre des rectangles comme formalisme de représentation et raisonnement spatial. En effet, les approches que nous avons présentées ne sont pas aussi expressives que l'algèbre des rectangles (par exemple ces méthodes n'expriment pas l'orientation). La définition des entités spatiales avec l'algèbre des rectangles est plus simple. Chaque entité spatiale est alors définie par un rectangle (plus de détails dans le chapitre 4). Ainsi, nous retenons l'algèbre des rectangles pour le raisonnement spatial. Dans ce formalisme les informations sont généralement représentées par des réseaux de contraintes où les variables représentent les objets considérés et les contraintes consistent en des relations qualitatives. Les méthodes de raisonnement employées sont des méthodes dérivées de celles utilisées dans le cadre général des problèmes de satisfaction de contraintes (*Constraint Satisfaction Problems* (CSP)) (plus de détails dans le chapitre 4).

Une relation de distance peut être représentée d'une manière numérique précise ou d'une manière floue. Pour la représentation numérique, nous utilisons la distance euclidienne. Pour la représentation floue, nous nous basons sur des approches symboliques qui ont pour objectif de quantifier des variables linguistiques.

Pour certaines applications, il est nécessaire de combiner ces approches qualitatives et quantitatives pour avoir un raisonnement plus complet et plus performant. Ceci a donné naissance aux cartes hybrides et aux cartes sémantiques. Dans ce type de cartes, la connaissance spatiale floue n'est pas considérée. La représentation hiérarchique n'est pas considérée dans les cartes

hybrides, mais elle l'est dans les cartes sémantiques, et ne permet pas de faire un raisonnement hiérarchiques. Ces cartes, ne permettent pas d'exploiter le potentiel qui découle de l'information spatiale (et principalement de la connaissance qualitative).

Dans cette thèse, nous définissons un cadre formel qui concerne une grande classe de problèmes en particulier en planification. Ce formalisme tient compte de l'aspect topologique, métrique et hiérarchique de l'espace mais aussi floue. Il se fonde sur les ontologies. Le chapitre suivant présente une étude bibliographique sur les ontologies. Le chapitre 4 présente nos travaux utilisant les ontologies comme technique de représentation et raisonnement spatial.

Chapitre 3

Ontologies

Sommaire

3.1	Qu'est ce qu'une ontologie ?	58
3.1.1	Réseau sémantique	58
3.1.2	Types d'ontologies	59
3.1.3	Utilité des ontologies	60
3.1.4	Construction d'une ontologie	61
3.2	Logiques de description	62
3.2.1	Syntaxe des logiques de description	62
3.2.2	Niveaux de descriptions	63
3.2.3	Inférence dans les logiques de description	65
3.2.4	Logique minimale $\mathcal{AL}$	66
3.3	Outils pour les ontologies	68
3.3.1	Langages	68
3.3.2	Outils	73
3.4	Connaissance spatiale dans les ontologies	74
3.4.1	Ontologies spatiales	75
3.4.2	Modélisation des entités spatiales	76
3.4.3	Modélisation des relations spatiales	77
3.5	Planification et ontologie spatiale	79
3.5.1	Ontologie spatiale au cœur du processus de planification	79
3.5.2	Ontologie spatiale et le bon fonctionnement du processus de planification	80

Nous avons montré l'importance de combiner différents aspect de la connaissance spatiale. Les ontologies semblent être un bon outil pour cela. Ce chapitre présente un état de l'art sur les ontologies. La section 3.1 présente une définition de la notion d'ontologie. La section 3.2 présente les logiques de description qui forment la base théorique des langages d'ontologie comme OWL. La section 3.3 présente différents langages et outils pour modéliser et exploiter les ontologies. La section 3.4 présente la prise en compte de la dimension spatiale dans les ontologies. La section 3.5 est une étude bibliographique des ontologies spatiales intégrées en planification.



3.1 Qu'est ce qu'une ontologie ?

Le développement des technologies d'Internet ouvre de nouveaux horizons dans le domaine du partage d'informations. L'information spatiale n'échappe pas à cette tendance et les besoins de modèles, de méthodes et d'outils adaptés pour représenter, manipuler et partager l'information spatiale sur le web, deviennent cruciaux. Ce partage d'informations passe par leur compréhension et une automatisation des processus de navigation, d'extraction et de mise à jour de ces informations. Les ontologies paraissent un outil très approprié pour atteindre cet objectif. Elles ont pour objectif de supporter la gestion des connaissances et le raisonnement sur ces dernières dans le but de partager la même sémantique entre agents humains et/ou artificiels.

Les ontologies sont actuellement au cœur de plusieurs techniques de l'intelligence artificielle [Bourigault *et al.*, 2004], de l'informatique biomédicale [Falasconi *et al.*, 1997], de l'ingénierie des connaissances [Studer *et al.*, 1998], du web sémantique [Charlet *et al.*, 2003]. Formellement, une ontologie est un vocabulaire commun et une compréhension du sujet décrit [Gandon, 2002]. Généralement, une ontologie décrit :

- des concepts : une idée abstraite, un objet, un principe ;
- des individus : des objets de base, des instances ;
- des attributs : des propriétés, des fonctionnalités, des caractéristiques ou des paramètres que les objets peuvent posséder et partager ;
- des relations, des prédicats ou bien des rôles : les liens que les objets peuvent avoir entre eux.

Considérons l'exemple décrit dans la figure 1.1(b) (chapitre 1, page 13) : le bloc A est sur la table (noté par $\text{Sur}(\text{A}, \text{Table})$), le bloc B est sur la table (noté par $\text{Sur}(\text{B}, \text{Table})$) et le bloc C est sur le bloc B (noté par $\text{Sur}(\text{C}, \text{B})$). En se basant sur ce qui a déjà été mentionné, on a :

- $\text{Bloc}(\text{X})$: est un concept spécifiant qu'une entité X est un cube avec des angles droits et des cotés de longueurs égales ;
- Table : est un objet global qui est composé d'un plan horizontal ;
- $\text{Sur}(\text{Bloc} : \text{X}, \text{Bloc Y}, \text{Table})$: est une relation exprimant qu'un bloc X peut être placé sur un autre bloc Y ou sur la table.

3.1.1 Réseau sémantique

L'objectif premier d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné. Formellement, une ontologie est un ensemble structuré sous la forme d'un graphe orienté dit **réseau sémantique** [Lindley *et al.*, 1994]. Un réseau sémantique est défini comme suit :

- un ensemble de nœuds représentant les concepts ;
- un ensemble d'arcs représentant les relations (ou prédicats) entre les concepts. Ces relations peuvent être des relations sémantiques ou des relations de composition et d'héritage (au sens objet).

À partir de cette structure, la sémantique de chaque mot est déduite par les relations que ce mot possède dans l'ontologie, ce qui permet de restreindre les interprétations possibles.

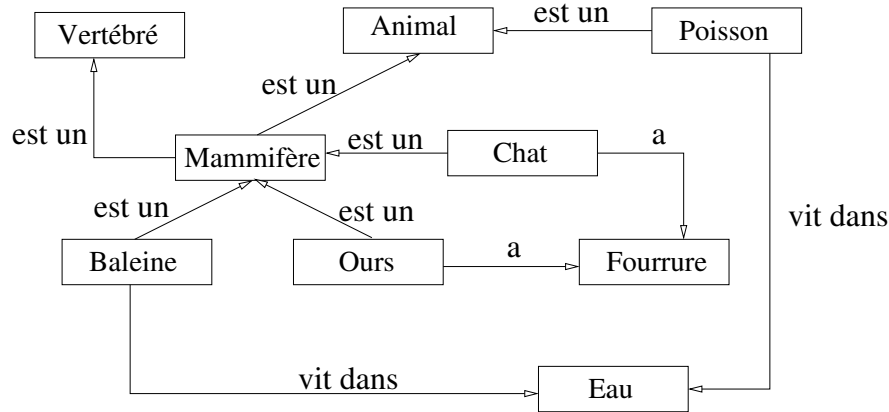


FIGURE 3.1 – Exemple de réseau sémantique.

La figure 3.1 est un exemple de réseau sémantique. Les nœuds représentent les concepts, par exemple Fourrure, Chat. Les arcs représentent les relations, par exemple vit dans, est un. À partir de ce réseau sémantique nous pouvons exprimer qu'un chat est un mammifère qui a de la fourrure.

3.1.2 Types d'ontologies

Différents types d'ontologies peuvent être considérés selon leur contenu et leurs utilisation. Nous en présentons les plus notables dans ce qui suit.

Les ontologies de types *thesaurus* : elles sont dites aussi *taxonomiques*. Elles visent à définir des termes pour fournir un vocabulaire de référence pour harmoniser les noms désignant des données dans différentes applications. L'ontologie Wordnet⁴ est sans doute la plus représentative de type *thesaurus*. Elle fournit un ensemble de définitions, de termes, structurés en arborescence et des relations entre les termes. On peut considérer ce type d'ontologie comme la première génération d'ontologie. Elles sont essentiellement utilisées de façon exploratoire pour apporter une compréhension commune d'un vocabulaire. Mais les besoins de modélisation et de partage d'informations plus complexes ont conduit à une évolution des ontologies vers des modèles plus riches.

Les ontologies descriptives : au-delà de la définition d'une taxonomie, elles visent à modéliser un domaine ou une activité. Elles sont plus proches de la définition de schémas conceptuels de base de données et s'attachent à modéliser les informations à l'aide de concepts et de relations sémantiquement riches. L'ontologie apporte alors une connaissance sur un domaine ou une activité qui peut être utilisée par un concepteur pour modéliser des applications

4. <http://www.w3.org/2001/sw/BestPractices/WNET/wordnet-sw-20040713.html>.

particulières. Ces ontologies peuvent être utilisées comme un outil de compréhension des structures de données décrivant les concepts du domaine, ou bien si elles possèdent des instances, elles peuvent être utilisées comme des bases de données notamment en étant accessibles au public via le web. De nombreux environnements ont été proposés pour modéliser, interroger et gérer des ontologies.

Les ontologies géographiques : elles recouvrent les ontologies de l'espace, les ontologies de domaines géographiques et les ontologies spatialisées. Les ontologies de l'espace, plus spécifiquement, dédiées à la description des concepts qui caractérisent l'espace comme une ligne, un point, etc. Ces ontologies sont typiquement élaborées par de grands organismes de normalisation. L'OpenGIS (www.opengis.net) propose par le biais de GML (Geography Markup Language) un langage pour structurer et échanger des informations géographiques [Burggraf, 2006]. En particulier GML apporte un quasi standard sur la définition des géométries des objets. Les ontologies de domaines géographiques sont des ontologies « métier » développées par une communauté d'utilisateur du domaine concerné. Les ontologies spatialisées (ou spatio-temporelles) qui sont des ontologies dont les concepts sont localisées dans l'espace. Une composante temporelle est souvent nécessaire en complément pour la modélisation de l'information géographique, car les applications géographiques manient aussi très souvent des données temporelles, voir spatio-temporelles. Les ontologies spatio-temporelles sont typiquement descriptives et peuvent s'appliquer à tout contenu nécessitant une modélisation localisée dans l'espace et le temps.

3.1.3 Utilité des ontologies

Nous pouvons résumer l'utilité des ontologies en quatre points :

1. modéliser la sémantique d'une connaissance riche, complexe et hétérogène,
2. raisonner : être capable d'inférer/classifier des informations et vérifier la cohérence de la connaissance,
3. prendre en compte les instances dans le but de gérer les données d'une ontologie,
4. interroger l'ontologie par des requêtes.

Besoin de modélisation et de raisonnement

Une ontologie est par définition une représentation d'un consensus entre une communauté de personnes qui peuvent être amenées à compléter ou à modifier cette ontologie même si elle comporte déjà des instances. Contrairement à une base de données, dont le schéma est défini avant la définition des instances et qui impose un certain nombre de critères de cohérence et de règles d'intégrité incontournables lors de l'insertion, la suppression et la mise à jour des données, la modélisation d'une ontologie doit permettre une certaine flexibilité dans sa définition et son évolution. Ceci signifie notamment que les informations (schéma ou instances) peuvent être incomplètes et que les mises à jour peuvent être faites sans avoir une connaissance complète

des informations de l'ontologie. Pour ce faire, il est nécessaire de mettre en œuvre des outils qui permettent de vérifier la cohérence du schéma mais aussi des instances de façon dynamique. Ces spécificités expliquent l'engouement pour les modèles logiques et notamment les logiques de description qui offrent en plus des aspects purement modèle, des techniques d'inférence et de vérification de la validité et de la cohérence d'un ensemble d'axiomes.

Besoin de requêtes

Les systèmes de gestion de bases de données apportent toutes les fonctionnalités nécessaires à la gestion des données d'une base. Les ontologies qui gèrent des instances doivent s'appuyer sur ces outils mais la frontière entre schéma et instances est moins clairement définie pour une ontologie. En effet, l'évolution du schéma et des instances est soumise à des contraintes plus souples et l'exploration de l'ontologie (requêtes) se fait aussi bien au niveau des instances qu'au niveau du schéma. Les langages de requêtes doivent donc être suffisamment riches pour permettre ce type d'interrogation sur le schéma lui-même.

3.1.4 Construction d'une ontologie

Il existe plusieurs méthodes pour définir une ontologie. La définition d'une ontologie se fait d'une manière itérative. Elle se fait en quatre étapes [Bachimont, 2000] :

- la spécification : cette étape répond à la question suivante, pourquoi définir une ontologie et qui sont ces utilisateurs ;
- la conceptualisation : cette étape permet de définir un domaine de connaissance structuré ;
- la formalisation : cette étape transforme le modèle conceptuel en un modèle formel ;
- l'implémentation : cette étape implémente le modèle formel.

La construction d'une ontologie demande un travail approfondi d'analyse et de compréhension du domaine et des utilisateurs du domaine. Par exemple, le projet *Cyc*⁵ lancé par Douglas Lenat développe une ontologie globale. Cette ontologie contient des faits et des règles sur la vie de tous les jours, par exemple : « tout arbre est une plante » et « les plantes finissent par mourir ». Lorsqu'on lui demande si les arbres meurent, le moteur d'inférence est en mesure de tirer la conclusion évidente et de répondre correctement à la question. Cette ontologie contient plus d'un million d'assertions, de règles et d'idées relevant du sens commun, définies par des êtres humains. De plus, bien que la conception des ontologies doit être la plus objective possible, elle est néanmoins affectée par la subjectivité du concepteur et des informations contenues dans le corpus. Ainsi deux ontologies d'un même domaine, définies par deux concepteurs différents, présenteront très certainement des décompositions différentes. Il est donc probable de voir l'apparition de plus en plus d'ontologies différentes, concernant des domaines proches ou identiques, chacune apportant une spécificité ou un point de vue particulier.

5. <http://sboisse.free.fr/technique/info/cyc.php>

3.2 Logiques de description

Dans la section 3.1, nous avons présenté la définition de la notion d'ontologie en spécifiant les besoins et en énonçant les étapes de construction. Cette section présente les logiques de description qui forment la base théorique des langages d'ontologie.

La logique de description (*Description Logics* noté DL) est une famille de formalisme conçue pour décrire et raisonner sur les connaissances d'un domaine [Baader, 2003]. Par exemple, la formule pour caractériser un cétacé comme un « mammifère qui vit dans un milieu aquatique » peut être représentée en utilisant :

- les concepts : `Mammifere`, `Milieu_Aquatique`,
- un prédicat : `vit`,
- et la description suivante : `Mammifere  $\sqcap$   $\exists$  vit.Milieu_Aquatique`.

Les logiques de description ont pour but d'obtenir une représentation plus proche de la représentation humaines que les formules de la logique de prédicats (PL) [Borgida, 1996]. Cependant, toute formule écrite en DL peut aussi être écrite en PL. Par exemple :

- en DL : `Mammifere  $\sqcap$   $\exists$  vit.Milieu_Aquatique`,
- en PL : `Mammifere(x)  $\wedge$   $\exists y$  (vit(x,y)  $\wedge$  Milieu_Aquatique(y))`.

De nombreuses recherches portent sur l'étude du rapport expressivité/performance des différentes LD [Nardi *et al.*, 2003]. La principale qualité des LD réside dans leurs algorithmes d'inférence dont la complexité est souvent inférieure aux complexités des démonstrateurs de preuves de la logique de premier ordre [Tsarkov et Horrocks, 2003].

3.2.1 Syntaxe des logiques de description

Les logiques de description sont une famille de systèmes de représentation fondée sur la connaissance caractérisée principalement par un ensemble de constructeurs qui permettent de construire des concepts et des prédicats complexes à partir des concepts et prédicats atomiques. Les logiques de description sont de parfaits candidats pour les ontologies, ceci est dû à leur sémantique bien définie et à leurs outils de raisonnement très puissants [Baader, 2003].

Nous adoptons la notation suivante : R dénote un rôle, C, D des concepts composés et A, B des concepts atomiques. Les concepts correspondent à des classes. Un concept C représente un ensemble d'individus (un sous-ensemble du domaine d'interprétation). Les rôles (prédicats) sont les relations binaires entre les objets, les lettres a et b représentent les individus.

Une fonction d'interprétation assigne à chacun des objets a , un individu a^I tel que $a^I \in \Delta^I$. Les concepts atomiques et rôles atomiques constituent les entités élémentaires. Les noms débutant par une lettre majuscule désignent les concepts, alors que ceux débutant par une lettre minuscule dénomment les rôles. Les concepts et rôles atomiques peuvent être combinés au moyen de constructeurs pour former respectivement des concepts et des rôles composés. Les différentes DL se distinguent par les constructeurs qu'elles proposent. Plus les DL sont expressives, plus les chances sont grandes que les problèmes d'inférence soient non décidables ou de complexité

Constructeur	Syntaxe	Exemple	Sémantique
Concept atomique	A	Homme	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Individu	a	Maria	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
Top	$\top$	Thing	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
Bas	$\perp$	Nothing	$\perp^{\mathcal{I}} = \emptyset$
prédicat atomique	r	aEnfant	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
Conjonction	$C \sqcap D$	Personne $\sqcap$ Femelle	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Disjonction	$C \sqcup D$	Male $\sqcup$ Femelle	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
Négation	$\neg C$	$\neg$ Femelle	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
Restriction existentielle	$\exists r.C$	$\exists$ aEnfant.Homme	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
Restriction universelle	$\forall r.C$	$\forall$ aEnfant.Personne	$\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}} : (x,y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$
Restriction de valeur	$\ni r.\{a\}$	$\ni$ aEnfant.{Maria}	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x,y) \in R^{\mathcal{I}} \Rightarrow y = a^{\mathcal{I}}\}$
Restriction de nombre	$(\geq nR)$ ou $(\leq nR)$	$(\geq 2$ aEnfant) ou $(\leq 1$ aEnfant)	$\{x \in \Delta^{\mathcal{I}} \mid \ \{y \mid (x,y) \in R^{\mathcal{I}}\} \mid \geq n\}$ ou $\{x \in \Delta^{\mathcal{I}} \mid \ \{y \mid (x,y) \in R^{\mathcal{I}}\} \mid \leq n\}$
Subsomption	$C \sqsubseteq D$	Personne $\sqsubseteq$ Thing	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
Définition de concept	$C \equiv D$	Parent $\equiv$ Papa $\sqcup$ Maman	$C^{\mathcal{I}} = D^{\mathcal{I}}$
Instanciation de concept	$a : D$	Paul : Homme	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
Instanciation de prédicat	$(a,b) : R$	(Paul,Maria) : aEnfant	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

TABLE 3.1 – Syntaxe et interprétation des logiques descriptives.

très élevée. Par contre, les DL trop peu expressives démontrent une inaptitude à représenter des domaines complexes.

En DL, une sémantique est associée à des concepts, des rôles (prédicats) et des individus qui utilisent une interprétation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, où $\Delta^{\mathcal{I}}$ est un ensemble non vide et $\cdot^{\mathcal{I}}$ est une fonction d'interprétation qui fait correspondre un concept C à un sous ensemble $C^{\mathcal{I}}$ de $\Delta^{\mathcal{I}}$ ou un rôle r à un sous ensemble $R^{\mathcal{I}}$ de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Le tableau 3.1 décrit les principaux constructeurs et une syntaxe pour DL. Dans la DL il existe la subsomption (notée $\sqsubseteq$) qui est une relation transitive et hiérarchique entre les concepts.

3.2.2 Niveaux de descriptions

Les logiques de descriptions permettent de définir des bases dans connaissances. Une base de connaissance dans la logique de description est répartie en une TBox (*Terminological Box*) et une ABox (*Assertional Box*).

Niveau terminologique : TBox

Une TBox contient des déclarations définissant des concepts, c'est-à-dire des classes d'éléments, et les relations, appelées rôles, entre ceux-ci. Un concept peut être défini sur la base d'autres concepts de la TBox et il existe des concepts atomiques, c'est-à-dire qui ne sont pas définis sur la base d'autres concepts. Les DL prédéfinissent au minimum deux concepts atomiques :

- le concept $\top$: c'est le concept le plus général,
- le concept $\perp$: c'est le concept le plus spécifique⁶.

Les concepts et prédicats atomiques peuvent être combinés au moyen de constructeurs pour former respectivement des concepts et des prédicats composés. Nous donnons la définition formelle d'une TBox. Une TBox contient des axiomes terminologiques de la forme :

- $C \sqsubseteq D$ sert à énoncer des relations d'inclusion (subsumption) ;
- $C \equiv D$ sert à énoncer des relations d'équivalence.

```

1 : Personne
2 : Femme  $\equiv$  Personne  $\sqcap$  Femelle
3 : Homme  $\equiv$  Personne  $\sqcap$  Male
4 : Maman  $\equiv$  Femme  $\sqcap \exists$  aEnfant.Personne
5 : Papa  $\equiv$  Homme  $\sqcap \exists$  aEnfant.Personne
6 : Parent  $\equiv$  Papa  $\sqcup$  Maman
7 : Grand_Mere  $\equiv$  Maman  $\sqcap \exists$  aEnfant.Parent

```

FIGURE 3.2 – Une base de connaissance de la famille.

La figure 3.2 présente un exemple illustrant une TBox. Les concepts portent les noms en majuscule, les prédicats portent des noms commençant par une minuscule. Dans cet exemple, la ligne 6 exprime qu'un parent peut être un papa ou une maman.

Propriété 3 Une interprétation $\mathcal{I}$ satisfait un axiome $C \sqsubseteq D$ si et seulement si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

Propriété 4 Une interprétation $\mathcal{I}$ satisfait un axiome $C \equiv D$ si et seulement si $C^{\mathcal{I}} = D^{\mathcal{I}}$

Propriété 5 Une interprétation $\mathcal{I}$ satisfait une TBox (est un modèle de la TBox) si et seulement si l'interprétation satisfait tous les axiomes de la TBox.

Niveau assertionnel : ABox

Une ABox contient des informations sur les individus, appelées assertions. Chaque ABox doit être associée à une TBox. En effet, les assertions s'expriment en terme de concepts et de prédicats de la TBox. L'exemple ci-dessous comprend les individus nommés Béatrice, Lætitia et Maria :

6. Le concept vide.

Béatrice : Maman
aEnfant(Béatrice, Lætitia)
aEnfant(Lætitia, Maria)

Cet exemple de la ABox exprime que Béatrice a un enfant qui s'appelle Lætitia. Lætitia a elle-même un enfant Maria. Ce qui permet de déduire que Béatrice est une grand mère.

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

Propriété 6 *Une interprétation satisfait une assertion d'appartenance $C(a)$ si et seulement si $a^{\mathcal{I}} \in C^{\mathcal{I}}$.*

Propriété 7 *Une interprétation satisfait une assertion de rôle $R(a, b)$ si et seulement si $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.*

Propriété 8 *Une interprétation satisfait une ABox A (est un modèle d'ABox) si et seulement si $\mathcal{I}$ satisfait toutes les assertions de A .*

Monde ouvert

Dans les logiques descriptives, l'interprétation des connaissances est faite dans un monde ouvert (Open-World Assumption). Le monde ouvert suppose qu'il peut exister des instances ou des relations qui ne sont pas définies dans la ABox. En d'autres termes, une information dont on ignore l'existence n'est pas forcément fausse : elle est inconnue.

3.2.3 Inférence dans les logiques de description

L'inférence s'effectue au niveau de la TBox ou la ABox. Il existe de nombreux raisonneurs capable de calculer les inférences. Selon [Baader, 2003], il y a quatre principaux problèmes d'inférence au niveau terminologique (c'est-à-dire au niveau de la TBox) :

1. « satisfaisabilité » : trouver tous les concepts insatisfaisables. Un concept C d'une terminologie $\mathcal{T}$ est satisfaisable si et seulement si existe un modèle $\mathcal{I}$ de $\mathcal{T}$ tel que $C^{\mathcal{I}} \neq \emptyset$;
2. « subsomption » : calculer la hiérarchie. Un concept C est subsumé par un concept D pour une terminologie si et seulement si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ pour tout modèle $\mathcal{I}$ de $\mathcal{T}$;
3. « équivalence » : trouver tous les concepts équivalents. Un concept C est équivalent à un concept D pour une terminologie si et seulement si $C^{\mathcal{I}} \equiv D^{\mathcal{I}}$ pour chaque modèle $\mathcal{I}$ de $\mathcal{T}$;
4. « disjonction » : trouver tous les concepts disjoints. Deux concepts C et D sont disjoints par rapport à une terminologie si et seulement si $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ pour chaque modèle $\mathcal{I}$ de $\mathcal{T}$.

Les moteurs d'inférence actuels tirent généralement profit du fait que les quatre types de problèmes d'inférence peuvent être réduits à des problèmes de subsomption ou à des problèmes de satisfaisabilité.

Propriété 9 *Réduction des problèmes d'inférence d'une TBox à des problèmes de subsomption :*

- 1 : C est insatisfaisable $\iff C$ est subsumé par $\perp$
- 2 : C et D sont équivalents $\iff C$ est subsumé par D et D par C
- 3 : C et D sont disjoints $\iff C \sqcap D$ est subsumé par $\perp$

Propriété 10 *Réduction des problèmes d'inférence d'une TBox à des problèmes de satisfaisabilité :*

- 1 : C est subsumé par $D \iff C \sqcap \neg D$ est insatisfaisable
- 2 : C et D sont équivalents $\iff \neg C \sqcap D$ et $C \sqcap \neg D$ sont insatisfaisables
- 3 : C et D sont disjoints $\iff C \sqcap D$ est insatisfaisable

Les propriétés 9 et 10 illustrent cette idée qui implique, que les moteurs d'inférence des LD ne nécessitent souvent qu'un seul algorithme d'inférence pour raisonner au niveau terminologique. D'ailleurs dans [Baader, 2003], les deux grandes classes d'algorithmes de raisonnement⁷ pour les LD correspondent aux façons de réduire respectivement des problèmes d'inférence à des problèmes de subsomption et de satisfaisabilité [Baader, 2003].

Pour l'inférence au niveau de la ABox, les moteurs d'inférence pour LD adoptent généralement l'hypothèse de noms uniques (HNU), c'est-à-dire que pour tout individu nommé a et b , $a = b$ (Baader et Nutt, 2003).

3.2.4 Logique minimale $\mathcal{AL}$

Il existe de nombreuses logiques de descriptions qui se distinguent par les constructeurs qu'elles proposent. Plus les DL sont expressives, plus les problèmes de complexité sont élevés. Par contre, les DL trop peu expressives ne permettent pas de représenter des connaissances de domaines complexes. Parmi les différentes logiques de description, il existe une logique minimale nommée $\mathcal{AL}$ [Schmidt-Schauß et Smolka, 1991]. Cette logique a été introduite par Schmidt-Schaub et Smolka (1991) et qui revêt d'une grande importance dans le domaine. Cette logique est minimale, dans le sens où c'est une logique moins expressive [Baader, 2003], c'est-à-dire qu'elle propose le moins de constructeurs (tableau 3.2).

Le constructeur $C \sqcap D$ permet de faire la conjonction de deux concepts composés, ce qui représente l'ensemble des individus, membres à la fois du concept C et du concept D pour une interprétation. Le constructeur $\neg C$ est utilisé pour évoquer la négation d'un concept atomique, c'est-à-dire les individus pour une interprétation qui n'appartiennent pas au concept atomique

7. Les deux grandes classes d'algorithmes de raisonnement sont les algorithmes de subsomption de type normalisation ou comparaison et les algorithmes de vérification de satisfaisabilité à base de tableaux.

Syntaxe	Constructeur
concept atomique	C
concept universel	$\top$
concept vide	$\perp$
négation atomique	$\neg C$
intersection	$C \sqcap D$
union	$C \sqcup D$
quantification existentielle limitée	$\exists R. \top$
quantification universelle complète	$\forall R. C$

TABLE 3.2 – Constructeurs de $\mathcal{AL}$.

C . Le constructeur $C \sqcup D$ permet de faire la disjonction de deux concepts, ce qui représente l'ensemble des individus, membres du concept C ou du concept D . Le quantificateur existentiel non typé $\exists R. \top$ désigne l'ensemble des individus, membres du domaine d'un rôle R pour une interprétation donnée. Le quantificateur universel $\forall R. C$ évoque l'ensemble des individus du domaine d'un rôle R qui sont en relation, par le biais de R , avec un individu du concept C , pour une interprétation donnée.

$[\mathcal{O}]$	$\{a_1, a_2, \dots, a_n\}$	$\{a_1^{\mathcal{I}}, a_2^{\mathcal{I}}, \dots, a_n^{\mathcal{I}}\}$
$[\mathcal{U}]$	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$[\mathcal{E}]$	$\exists R. C$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b. (a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$
$[\mathcal{C}]$	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$[\mathcal{I}]$	R_1^-	$\{(y, x) \mid (x, y) \in R_1^{\mathcal{I}}\}$
$[\mathcal{H}]$	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$
$[\mathcal{F}]$	$= 1R, \geq 2R$	$\{x \in \Delta^{\mathcal{I}} \mid \{y \in \Delta^{\mathcal{I}} \mid (x, y) \in R^{\mathcal{I}}\} \models 1\}, \{x \in \Delta^{\mathcal{I}} \mid \{y \in \Delta^{\mathcal{I}} \mid (x, y) \in R^{\mathcal{I}}\} \models \geq 2\}$
$[\mathcal{N}]$	$\geq nR, \leq nR, = nR$	$\{a, b \in \Delta^{\mathcal{I}} \mid \{(a, b) \in R^{\mathcal{I}}\} \models \geq n\}, \{a, b \in \Delta^{\mathcal{I}} \mid \{(a, b) \in R^{\mathcal{I}}\} \models \leq n\}, \{a, b \in \Delta^{\mathcal{I}} \mid \{(a, b) \in R^{\mathcal{I}}\} \models = n\}$
$[\mathcal{Q}]$	$\geq nR. C, \leq nR. C, = nR. C$	$\{a, b \in \Delta^{\mathcal{I}} \mid (a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}} \models \geq n\}, \{a, b \in \Delta^{\mathcal{I}} \mid (a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}} \models \leq n\}, \{a, b \in \Delta^{\mathcal{I}} \mid (a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}} \models = n\}$

TABLE 3.3 – Exemple de constructeurs de rôles et de concepts pour étendre $\mathcal{AL}$.

Plusieurs extensions possibles d' $\mathcal{AL}$ existent. Le nom de chaque logique de description permet d'identifier, les constructeurs de rôles de celle-ci. Il existe trois façons d'étendre $\mathcal{AL}$ détaillées dans ce qui suit.

Extension par ajout des constructeurs de concepts ou de rôles : le tableau 3.3 illustre des exemples de constructeurs pour étendre $\mathcal{AL}$. La première colonne contient la lettre qui désigne le constructeur, la deuxième représente sa syntaxe d'utilisation et la dernière

concerne sa sémantique. Chaque logique de description se base au minimum sur $\mathcal{AL}$ qui forme le préfixe le nom de toutes les logiques de description, les lettres suivantes forment l'extension. Par exemple la logique $\mathcal{AL}$, enrichie de l'union $\mathcal{U}$ et de la quantification existentielle complète $\mathcal{E}$, se nomme $\mathcal{ALUE}$.

Extension par ajout de contraintes sur l'interprétation des rôles : la spécification d'un ensemble de rôles transitifs $\mathcal{N}_{\mathcal{R}+}$, constitue $\mathcal{R}+$ une extension par ajout de contraintes sur l'interprétation des rôles (désignée par la lettre $\mathcal{R}+$), qui permet l'expression de rôles transitifs [Baader, 2003]. La lettre $\mathcal{S}$ désigne la logique $\mathcal{ALC}$ additionnée de $\mathcal{R}+$. OWL (*Ontology Web Language*, section 3.3) fait partie de cette famille de DL, puisqu'il est basé sur la logique de description $\mathcal{SHOIN}(\mathcal{D})$.

Extension des types primitifs : une dernière extension, symbolisée par la lettre $\mathcal{D}$, ajoute le support des types primitifs [Horrocks *et al.*, 2003]. Cette extension augmente $\mathcal{AL}$ d'un second domaine d'interprétation $\Delta_{\mathcal{D}}^{\mathcal{I}}$ disjoint avec $\Delta^{\mathcal{I}}$ et qui représente l'ensemble des valeurs de type primitif. Le domaine $\Delta_{\mathcal{D}}^{\mathcal{I}}$ définit plusieurs sous-domaines tels que les entiers, les chaînes de caractères, les entiers positifs, etc. Les éléments de ces domaines se nomment individus primitifs.

3.3 Outils pour les ontologies

Dans la section 3.2, nous avons présenté le fondement théorique sur lequel se base les différents langages pour créer des ontologies d'un point de vue formel. Dans cette section, nous présentons différents langages pour créer des ontologies d'un point de vue informatique et les outils permettant de les manipuler.

3.3.1 Langages

Le W3C recommande un ensemble de langage permettant de définir des ontologies. Il s'agit de RDF (*Resource Description Framework*) [Manola *et al.*, 2004], RDFS (*Resource Description Framework Schema*), OWL (*Ontology Web Language*) et SWRL (*Semantic Web Rule Language*).

RDF

Le langage RDF permet de décrire l'information dans le web sémantique en encodant de manière formelle les liens entre les ressources Web. RDF est un langage qui a pour but de permettre à une communauté d'utilisateurs de partager les mêmes données pour des ressources partagées. Ce langage permet ainsi de décrire le contenu d'une ressource au travers de méta-données telles que le nom de la ressource ou son type.

La connaissance est représentée sous forme d'un tuple $\langle \text{sujet}, \text{prédicat}, \text{objet} \rangle$, où :

- **sujet** : la ressource décrite par un tuple,
- **prédicat** : la propriété de cette ressource,

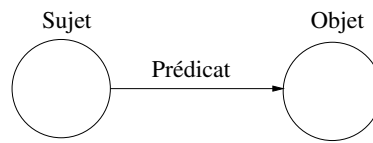


FIGURE 3.3 – Un graphe RDF.

- **objet** : la valeur prise par la propriété. L'objet peut être une valeur littérale, permettant de représenter des valeurs numériques ou des dates, ou alors une autre ressource.

Un tel tuple peut être représenté par un graphe. Le sujet et l'objet sont des nœuds du graphe tandis que le prédicat est un label d'un arc directionnel, orienté du sujet vers l'objet, indiquant ainsi une relation entre ces derniers (figure 3.3). Un ensemble de triplets constitue un graphe RDF. L'objet peut être une ressource/classe, un littérale, une date, etc.

```
<rdf :RDF>
  <rdf :Description about = "Harry">
    <rdf :Property about = "nom">Potter</rdf :Property>
    <rdf :Property about = "est inscrit">Poudlard</rdf :Property>
  </rdf :Description>
</rdf :RDF>
```

EXTRAIT 3.1 - Représentation RDF de Harry Potter inscrit à Poudlard

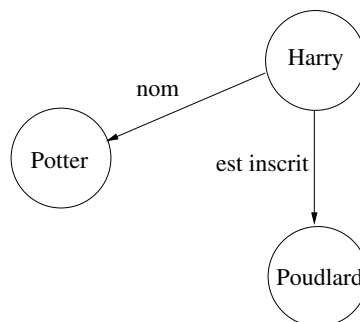


FIGURE 3.4 – Représentation graphique de Harry Potter qui est inscrit à Poudlard.

La figure 3.4 présente un graphe RDF illustrant la représentation RDF décrite dans l'extrait 3.1. À partir de ces représentations, nous avons le triplet (Harry, est inscrit, Poudlard) indique que la ressource « Harry » possède une propriété « est inscrit » ayant la valeur « Poudlard », représentant ainsi que « Harry étudie à Poudlard ».

RDFS

Le langage RDFS est un langage de description de vocabulaire RDF permettant la description formelle des hiérarchies et des relations entre les ressources [Manola *et al.*, 2004]. Toutes les

références de RDFS sont regroupées dans l'espace de noms `http://www.w3.org/2000/01/rdf-schema#` auquel le préfixe `rdfs` est associé.

Ce langage permet également de définir des restrictions sur le domaine et l'espace d'arrivée des propriétés [Manola *et al.*, 2004]. Grâce à ce langage, il est possible de réaliser des raisonnements et de définir des ontologies simples.

```
<rdf :RDF>
  <rdfs :Class rdf :about = "Sorcier">
    <rdfs :subClassOf rdf :resource= "Personne">
  </rdfs :Class>
  <rdf :Property about = "nom">
    <rdfs :domain rdf :resource = "Sorcier">
    <rdfs :range rdf :resource = "xsd :string">
  </rdf :Property>
  <rdf :Property about = "est inscrit">
    <rdfs :domain rdf :resource = "Sorcier">
    <rdfs :range rdf :resource = "xsd :string">
  </rdf :Property>
</rdf :RDF>
```

EXTRAIT 3.2 - Représentation RDFS du concept Sorcier

L'extrait 3.2 décrit le concept `Sorcier`. Ce concept est défini par des types et des objets de chaque propriété.

```
<Sorcier rdf :ID="Harry">
  <nom rdf :resource="Potter"/>
  <est inscrit rdf :resource="Poudlard"/>
</Sorcier>
```

EXTRAIT 3.3 - Représentation RDFS de l'instance Harry

Considérons le triplet (`Harry`, `est_un`, `Sorcier`) indique que la ressource `Harry` est un sorcier (extrait 3.3). Connaissant les deux premières déclarations suivantes `Harry est_un Sorcier` et `Sorcier rdfs:subClassOf Personne`, la troisième déclaration peut être inférée par un moteur de raisonnement en appliquant la règle suivante :

`Harry est_un Sorcier ∧ Sorcier rdfs:subClassOf Personne → Harry est_un Personne`

En effet, il existe une relation hiérarchique entre la classe `Personne` et la classe `Sorcier` exprimée par la propriété `rdfs:subClassOf` qui indique qu'une classe est sous-classe d'une autre. Ainsi, nous déduisons qu'un sorcier est une personne. Un exemple de restriction, décrit dans l'extrait 3.2, la propriété `nom` ne peut s'appliquer qu'à des personnes (également des sorciers) (il

s'agit de son domaine : `rdfs:domain`) et sa valeur doit être une chaîne de caractères (`rdfs:range` permet de définir la classe de ses objets) comme indiqués dans l'extrait 3.2.

Nous avons vu que RDF et RDFS permettent de définir sous la forme d'un graphe des triplets de données. Cependant, il est impossible de raisonner sur ces représentations car la sémantique (sauf la subsomption) reste très limitée. C'est un manque de sémantique que OWL comble par l'apport d'un vocabulaire plus riche.

OWL

Le langage OWL définit de nombreuses relations permettant aux applications de réaliser des raisonnements plus complexes sur les données représentées [McGuinness *et al.*, 2004]. Ce langage est basé sur la logique de description qui est un formalisme logique permettant la représentation de la connaissance [Baader, 2003] (décrit dans voir la section 3.2).

Prédicat	Définition
<code>owl:equivalentclass</code>	équivalence entre deux classes
<code>owl:disjointWith</code>	disjonctions entre deux classes
<code>owl:unionOf</code>	l'union
<code>owl:intersectionOf</code>	l'intersection
<code>owl:complementOf</code>	le complément d'une autre classe

TABLE 3.4 – Prédicats définis pour les classes.

Prédicat	Définition
<code>owl:equivalentProperty</code>	équivalence entre deux propriétés
<code>owl:inverseOf</code>	inverse d'une propriété
<code>owl:SymetricProperty</code>	symétrie d'une propriété
<code>owl:transitiveProperty</code>	transitivité d'une propriété
<code>owl:FunctionalProperty</code>	fonctionnalité d'une propriété

TABLE 3.5 – Prédicats définis pour les propriétés.

OWL offre un ensemble de prédicats permettant beaucoup plus d'expressivité que RDFS. Les tables 3.4, 3.5 présentent quelques extensions. Pour plus de détails sur la définition du langage OWL le lecteur est invité à consulter l'annexe C. Le langage OWL définit trois sous-langages allant du moins au plus expressif :

- le langage **OWL Lite** propose des fonctionnalités pour la classification hiérarchiques et la définition de contraintes simples ;
- le langage **OWL DL** (*Description Logic*) permet une expressivité maximale sans perdre la complétude du calcul⁸ et la décidabilité⁹ des systèmes de raisonnement. Ce langage inclut

8. C'est-à-dire que toutes les inférences sont prises en compte.

9. C'est-à-dire tous les calculs seront terminés dans un intervalle de temps fini.

toutes les structures du langage OWL avec des restrictions comme la séparation des types (une classe ne peut pas être un individu ou une propriété ne peut être un individu ou une classe). La section 3.2 présente les logiques de description qui forment la base théorique des langages d'ontologie comme OWL ;

- le langage **OWL Full** propose une expressivité maximale et la liberté syntaxique de RDF sans garantie de calcul. Par exemple, dans OWL Full, une classe peut se traiter simultanément comme une collection d'individus et comme un individu à part entière. Il est peu probable qu'un système de raisonnement puisse mettre en œuvre toutes les caractéristiques de OWL Full.

```
<owl:Class rdf:ID="VinRouge">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Vin" />
    <owl:Restriction>
      <owl:onProperty rdf:resource="#decouleur" />
      <owl:hasValue rdf:resource="#Rouge" />
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

EXTRAIT 3.4 - Représentation OWL du concept VinRouge

L'extrait 3.4 indique que la classe **VinRouge** est exactement l'intersection de la classe **Vin** et l'ensemble des choses qui sont de couleur rouge (utilisation de la propriété `owl:intersectionOf`). Ceci signifie que si quelque chose est rouge et est un vin, il est alors une instance de **VinRouge**. Dans une telle définition, on peut savoir que les vins rouges sont des objets de types vins et sont de couleur rouge, mais pas *vice-versa*. C'est un outil important pour classer les individus.

OWL2 [Motik *et al.*, 2009] une extension qui ajoute de nouvelles fonctionnalités à OWL, parmi lesquelles la possibilité de définir des propriétés asymétriques, réflexives et disjointes. OWL2 permet également d'associer des clés à des instances de classes pour identifier ces individus de manière univoque. Pour plus de détails sur le langage OWL2, nous vous invitons à consulter l'article [Motik *et al.*, 2009].

SWRL

Le langage SWRL enrichit la sémantique d'une ontologie définie en OWL [Horrocks *et al.*, 2004]. En effet SWRL permet contrairement à OWL de manipuler des instances par des variables (`?x, ?y, ?z`). SWRL ne permet pas de créer ni des concepts ni des relations. Il permet seulement d'ajouter des relations suivant les valeurs des variables et la satisfaction de la règle. Les règles SWRL sont construites suivant ce schéma : antécédent $\rightarrow$ conséquent. L'antécédent et le conséquent sont des conjonctions d'atomes. Un atome est une instance de concept, une relation OWL ou une des deux relations SWRL (`same-as(?x, ?y)` ou `different-from(?x, ?y)`).

Le fonctionnement d'une règle est basé sur le principe de satisfaisabilité de l'antécédent ou du conséquent. Pour une règle, il existe trois possibilités :

- l'antécédent et le conséquent sont définis. Si l'antécédent est satisfait alors la conséquence doit l'être ;
- l'antécédent est vide, cela équivaut à un antécédent satisfait ce qui permet de définir des faits ;
- la conséquence est vide, cela équivaut à un conséquent insatisfait, l'antécédent ne doit pas être satisfiable.

Considérons¹⁰ les relations suivantes : `aUnOncle`, `aParent` et `aFrere`. La combinaison des relations `aParent` et `aFrere` implique la propriété `aUnOncle`, en définissant la règle suivante :

$$\text{aParent}(?x1, ?x2) \wedge \text{aFrere}(?x2, ?x3) \rightarrow \text{aUnOncle}(?x1, ?x3)$$

En respectant la syntaxe, la règle est définie comme suit :

```
Implies(Antecedent(aParent(I-variable(x1) I-variable(x2))
                  aFrere(I-variable(x2) I-variable(x3)))
Consequent(aUnOncle(I-variable(x1) I-variable(x3))))
```

À partir de cette règle, si John a Mary comme parent et Mary a un frère Bill alors John a Bill comme oncle.

3.3.2 Outils

Cette partie présente les moteurs d'inférence et les langages d'interrogation utilisés dans les ontologies.

Moteurs d'inférence

La plupart de ces moteurs sont conçus pour raisonner sur les logiques de descriptions, mais acceptent en entrée des fichiers OWL. Une fois l'ontologie chargée, ces moteurs effectuent les inférences sur la TBox et la ABox. Parmi les raisonneurs existants, citons RacerPro¹¹, FaCt++¹² et Pellet¹³, Racer [Haarslev et Möller, 2002]. Nous considérons Pellet et Racer (pour plus de détails voir l'annexe C) qui sont à l'heure actuelle les deux moteurs permettant le raisonnement sur la ABox et la TBox en exploitant des ontologies possédant un niveau d'expressivité en DL et en OWL satisfaisant (annexe C).

Le moteur Racer [Haarslev et Möller, 2002] est le moteur d'inférence sans doute le plus connu et l'un des plus utilisés dans le domaine pour ces performances et sa stabilité. Racer travaille

10. Cet exemple est tiré de <http://www.daml.org/2003/11/swrl/examples.html>.

11. www.racer-systems.com

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

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

sur les ontologies représentées à l'aide de son langage, mais il accepte des ontologies décrites en RDF ou en OWL, ces dernières étant traduites vers le langage utilisé par Racer. Ce langage possède également son propre langage de requête nRQL (*new Racerpro Query Language*) pour interroger les ontologies sur la ABoc et la TBox (annexe C).

Le moteur Pellet [Sirin *et al.*, 2007] est beaucoup plus récent. Pellet travaille sur des ontologies décrites en RDF ou en OWL et permet les requêtes avec RDQL et SPARQL sur la ABox et la TBox (annexe C).

Langages d'interrogation

Il existe plusieurs langages d'interrogation notamment SPARQL (*Protocol And RDF Query Language*) [Prud'Hommeaux *et al.*, 2006] et nRQL [Haarslev et Möller, 2002]. SPARQL est le langage d'interrogation pour Pellet et nRQL est le langage d'interrogation pour Racer. Ces langages d'interrogation sont basés principalement sur la reconnaissance de graphe RDF. Le principe est de décrire le graphe RDF à l'aide de variables. Les résultats d'une requête étant les valeurs des variables pour lesquelles il existe un graphe RDF dans l'ontologie correspondant au graphe défini par la requête.

SPARQL permet l'interrogation des triplets des documents RDF et ne réalise aucune inférence. Les requêtes de ce langage contiennent généralement un ensemble de modèles de triplets appelé modèle de graphe élémentaire (*Basic Graph Pattern*). Un modèle de graphe élémentaire correspond à un sous-graphe du graphe RDF sur lequel s'applique la requête SPARQL lorsque les termes RDF de ce sous graphe peuvent être substitués aux variables et le résultat est un graphe RDF équivalent à ce sous-graphe. Le langage SPARQL permet également de réaliser des correspondances en utilisant des littéraux ou des valeurs numériques dans le modèle de graphe élémentaire. Il est également possible d'agir sur les résultats d'une requête (restreindre les résultats à certaines valeurs en utilisant des filtres ou de donner un ordre).

nRQL [Haarslev et Möller, 2002] est le langage d'interrogation de Racer. Comme SPARQL, nRQL est basé sur la recherche de graphes RDF. Sa syntaxe est proche de SPARQL sauf pour sa notation préfixée des opérateurs.

3.4 Connaissance spatiale dans les ontologies

Dans les sections précédentes, nous avons abordé la définition des ontologies, les langages pour les définir et les outils pour les exploiter. Cette section est consacrée à l'intégration de la connaissance spatiale dans une ontologie dite **ontologie spatiale**. Plusieurs ontologies qui tentent de combiner différentes approches pour le raisonnement spatial ont été développées [Bateman et Farrar, 2004].

3.4.1 Ontologies spatiales

Les ontologies spatiales combinent plusieurs approches en un ensemble d'outils gérant l'espace. Cet ensemble a pour but de représenter et de raisonner dans divers domaines d'applications où la dimension spatiale joue un rôle important. Par exemple, les ontologies spatiales sont très exploitées par les SIG [Casati *et al.*, 1998], [Klien et Lutz, 2005], [Macario et Medeiros, 2009]. En effet, elles représentent une grande source d'informations permettant de déduire des informations spatiales très importantes. Par exemple, à l'aide d'une ontologie spatiale exploitée dans un SIG, on peut établir les régions géographiques les plus menacées par la montée des eaux de par leur proximité d'un océan et leur faible altitude moyenne.

Considérons l'ontologie définie dans [Grenona et Smitha, 2004]. Cette ontologie purement spatiale s'appuie sur des vues instantanées du monde à des instants successifs du temps, ce qui définit une ontologie spatiale dynamique. Nous citons aussi *Suggested Merged Upper Ontology* (SUMO) [Pease *et al.*, 2002]. L'axiome de base de l'emplacement de SUMO est ¹⁴ :

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

EXTRAIT 3.5 - Fragment de l'ontologie SUMO.

Le deuxième paramètre (?LOC) du prédicat *located* doit être une région. Les régions respectent des centaines de propriétés et permettent la définition de quelques relations spatiales telles que *partlyLocated*, *exactlyLocated* ainsi que la connexion, le chevauchement et l'inclusion (extrait 3.5).

Selon [Goodwin *et al.*, 2008], les ontologies spatiales peuvent être classées en sept catégories : « les ontologies géospatiales pour la description des éléments », « les ontologies pour la définition de types d'éléments », « les ontologies pour la définition de relations spatiales », « les ontologies de noms », « les ontologies définissant des systèmes de coordonnées », « les méta-ontologies géospatiales », « les ontologies géospatiales pour les services Web ».

Dans ce travail, nous nous concentrons principalement sur « les ontologies géospatiales pour la description des éléments » et « les ontologies pour la définition de relations spatiales ». En effet, les ontologies géospatiales pour la description des éléments (ou *features*) permettent une représentation formelle pour la définition des éléments dans leur contexte. Quant aux ontologies spatiales pour la définition de relations spatiales, elles sont utiles pour la description des relations spatiales topologiques et de distance. Une de nombreuses ontologies définissant les relations spatiales est l'ontologie *SpatialRelations*¹⁵

14. Les descriptions de SUMO sont données selon la version 1.60.

15. <http://www.ordnancesurvey.co.uk/ontology/SpatialRelations.owl>

3.4.2 Modélisation des entités spatiales

De nombreux travaux proposent des ontologies pour la description d'éléments (ou *features*), dans lesquelles les concepts sont des types spatiaux (point, ligne, etc.). Parmi celles-ci, nous citons l'ontologie GeoRSS¹⁶. Il existe deux éléments du standard GeoRSS : le GeoRSS-GML et le GeoRSS-Simple. Nous présentons l'encodage de GeoRSS-Simple.

GeoRSS-Simple est un format très léger pour la description des données spatiales que les développeurs et les utilisateurs peuvent rapidement et facilement intégrer dans leurs applications, ou bien utiliser pour créer des annotations sémantiques avec peu d'efforts.

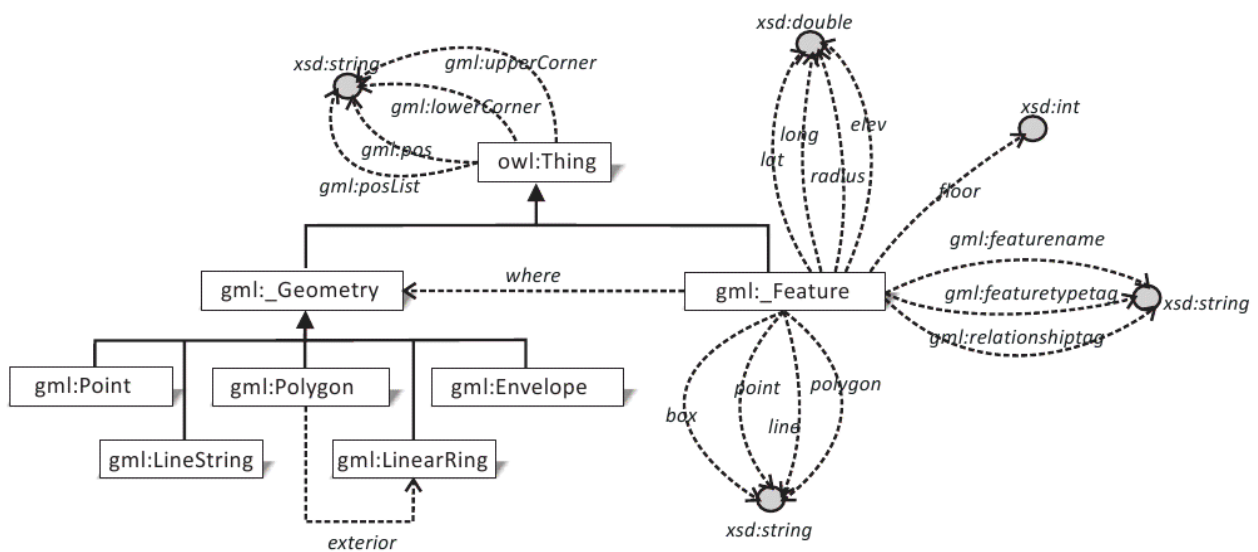


FIGURE 3.5 – Représentation graphique de GeoRSS-Simple [École d'été GDR MAGIS-2009].

La figure 3.5 est une représentation graphique de l'ontologie GeoRSS-Simple. Le concept `gml:_Feature` regroupe les types spatiaux de base (point, ligne, polygone, boîte). D'autres spécialisations de `gml:_Feature` décrivent des attributs utilisés habituellement pour caractériser les éléments (*features*), tels que le nom (`gml:featurename`), le type (`gml:featuretypetag`), etc. Le concept `gml:_Feature` peut être relié au concept `gml:_Geometry` par la relation¹⁷ `where`. La relation `where` est utilisée pour associer une entité spatiale à un type de géométrie. En effet, le concept `gml:_Geometry` permet une description géométrique des éléments. L'ontologie définit les géométries de base (`gml:_Point`, `gml:_Polygon`, etc.) comme spécialisations de la classe abstraite (`gml:_Geometry`) (figure 3.5). Ces géométries permettent d'associer une représentation quantitative aux éléments et facilitent l'analyse et la visualisation des données.

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

17. Une propriété-lien.

```

<_Feature rdf:about="tourEiffel">
  <georss:point rdf:datatype="&xsd:string"> 48.8583 2.2945</georss:point>
</_Feature>
<_Feature rdf:about="Pentagon">
  <georss:polygon rdf:datatype="&xsd:string">
    -77.0579537082376 38.87258672915797
    -77.0584754972063 38.87005770844568
    -77.0555999760046 38.86886750371786
    -77.0532658178173 38.87064560343153
    -77.0546559466248 38.87292421787603
    -77.0579537082376 38.87258672915797
  </georss:polygon>
  <featurename rdf:datatype="&xsd:string"> Pentagon</featurename>
</_Feature>

```

EXTRAIT 3.6 -

Fragment OWL décrivant la Tour Eiffel et le Pentagone [École d'été GDR MAGIS-2009].

L'extrait 3.6 illustre la définition des coordonnées géographiques de la tour Eiffel et une description possible du bâtiment du Pentagone à l'aide de GeoRSS-Simple. Dans cet extrait, la tour Eiffel est définie par un point : `<georss:point rdf:datatype="&xsd:string"> 48.8583 2.2945</georss:point>`. Le bâtiment du Pentagone est définie par un polygone.

3.4.3 Modélisation des relations spatiales

Plusieurs travaux se sont concentrés sur la définition des relations spatiales par des ontologies et ceci dans divers domaines d'applications, par exemple la robotique [Dominey *et al.*, 2007], la médecine [Donnelly *et al.*, 2006], le traitement d'images [Dasiopoulou *et al.*, 2005], etc. Dans nos travaux, nous considérons deux types de relations (chapitre 2) :

- de distance, deux représentations possibles : numérique précise ou qualitative floue,
- topologique, dans le chapitre 2, nous avons considéré que les relations directionnelles sont incluses dans les relations topologiques puisque nous considérons l'algèbre de rectangles comme formalisme de représentation (page 43). Ainsi, nous pouvons exprimer les relations topologiques (telles que « chevauche », « se touchent », etc.), les relations d'orientations (telles que « à droite de », « à gauche de », etc), et même des relations de la forme : « au même niveau », « complètement à droite de », etc.

Dans la littérature, il existe différentes ontologies représentant et exploitant les relations spatiales topologiques, nous citons [Hudelot *et al.*, 2008], [Reitsma et Hiramatsu, 2006], [Cullot *et al.*,], [Grenona et Smitha, 2004] et [Martin, 2007].

Dans [Reitsma et Hiramatsu, 2006], les auteurs présentent une ontologie spatiale définissant les relations spatiales¹⁸. Les relations topologiques, de direction et de distance sont définies comme des classes. Leur association avec les entités spatiales se fait *via* la propriété `hasSpatialRelation`. Dans cette ontologie, les relations topologiques sont présentées par l’approche de 9-intersections étendue (*Extended 9-Intersection model* DE-9IM [Clementini et Di Felice, 1995b]). D’un point de vue topologique, cette ontologie inclut les relations suivantes : *equals*, *disjoint*, *intersects*, *touches*, *crosses*, *within*, *contains*, *overlaps*. Les relations de direction sont décrites par le modèles de huit secteurs afin d’exprimer les différentes directions cardinales : le nord, le nord est, l’est, le sud est, le sud, le sud ouest, l’ouest et le nord ouest. Ces relations sont transitives et sont réparties en deux niveaux (figure 3.6). Le premier niveau comprend des relations telles que `isNorthOf` et `isNorthEastOf`. La relation `A isNorthOf B` est valide lorsque le point le plus au nord de la région A est situé au nord du point le plus au nord de la région B (figure 3.6(a)). Le second niveau comprend des sous-propriétés du premier niveau, respectivement `isCompletelyNorthOf` et `isCompletelyNorthEastOf`. Dans la figure 3.6(b), la relation `A isCompletelyNorthOf B` est plus restrictive et elle n’est valide que si la région A est dans son entièreté plus au nord que la région B, c’est-à-dire, le point le plus au sud de la région A étant plus au nord que le point le plus au nord de la région B.

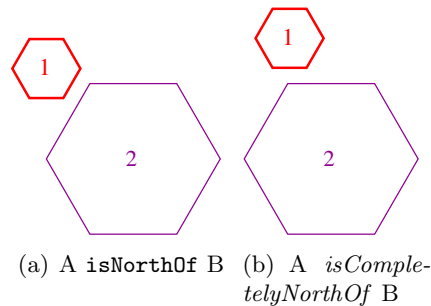


FIGURE 3.6 – Deux niveaux des relations de direction.

Dans cette ontologie, les éléments nécessaires pour établir une relation à distance ne sont pas spécifiés. En d’autres termes, les éléments définissant un système de référence ne sont pas définies dans l’ontologie. Cette ontologie inclut, une spécification numérique pour la distance. Elle inclut également une représentation non numérique de la distance restreinte aux termes suivants : « proche », « moyen » et « loin ».

Plusieurs ontologies, dans le cas des relations spatiales de proximité emploient des termes tels que « proche », « loin », etc. Ceci génère une imprécision et rend l’information spatiale floue. La limite la plus notable de la plupart des ontologies existantes est leur incapacité à représenter et raisonner avec des informations floues [Hudelot *et al.*, 2008].

¹⁸. Les relations spatiales intégrées dans l’ontologie sont topologiques, direction, distance, et relations méréologique.

L'ontologie des relations spatiale décrite dans [Hudelot *et al.*, 2008] permet de réduire le fossé entre les informations numériques contenues dans l'image et des concepts représentant l'information spatiale floue. Dans ce travail, les auteurs considèrent que les entités spatiales sont de forme floues liées par des relations spatiales floues. L'intégration du flou consiste à relier les concepts de l'ontologie floue par rapport à leur représentation spatiale physique dans le domaine de l'image. Par exemple, pour la relation floue définie par l'expression « à droite de X », on s'intéresse à la région de l'image qui satisfait la relation « à droite de » de l'élément X.

Plusieurs autres approches ont été proposées pour tenir compte du flou et l'imprécision. Nous citons : les approches utilisant les logiques de description floues (*fuzzy DL*) ([Bobillo et Straccia, 2008], [Straccia, 2011], etc.) et les approches basées sur les extensions probabilistes du langage standard OWL en utilisant des réseaux bayésien (par exemple : [da Costa *et al.*, 2008]).

3.5 Planification et ontologie spatiale

Dans le chapitre 2, nous avons montré que dans plusieurs applications et en particulier en planification il est nécessaire de combiner l'information spatiale qualitative (topologique et floue) avec l'information spatiale quantitative (numérique). L'ontologie semblent être une technique très adéquate pour combiner différents aspects de la connaissance spatiale et permet leur exploitation.

La section précédente a présenté quelques travaux sur les ontologies spatiales développées jusqu'à aujourd'hui. Plusieurs travaux s'intéressent aux ontologies spatiales dans certains domaines tels que la robotique [Dominey *et al.*, 2007], la médecine [Donnelly *et al.*, 2006], le traitement d'images [Dasiopoulou *et al.*, 2005] ont été proposés dans la littérature. Toutes ces ontologies se sont intéressées à la représentation des concepts spatiaux selon les domaines d'applications. Par exemple en médecine, les ontologies sont limitées aux concepts de la théorie de méréologie [Donnelly *et al.*, 2006]. Une grande limite des ontologies usuelles est leur incapacité à représenter et à raisonner avec des imprécisions. Un travail intéressant défie cette limite [Hudelot *et al.*, 2008].

Cette section évoque l'utilisation d'une ontologie spatiale dans un système de planification. L'idée d'exploiter l'ontologie pour représenter les connaissances spatiales dans un système de planification a été explorée dans plusieurs travaux [Car et Frank, 1994], [Gribble *et al.*, 1998], [Przytula-Machrouh *et al.*, 2004], [Bateman et Farrar, 2005], [Bouguerra *et al.*, 2008], etc. Certaines de ces ontologies sont utilisées comme un outil de contrôle dans le processus de planification. D'autres sont réellement exploitées dans le processus de planification.

3.5.1 Ontologie spatiale au cœur du processus de planification

Dans le chapitre 2, nous avons retenu trois aspects de la connaissance spatiale : la représentation hiérarchiques, qualitative (topologique et floue) et quantitative (numérique). La résolution de certains problèmes de planification nécessite la prise en compte de tous ces aspects. Certains travaux en planification utilisent l'ontologie spatiale comme base de connaissance, et surtout ex-

exploitent cette connaissance grâce aux mécanismes d'inférence offerts par l'ontologie. Par exemple, les mécanismes d'inférence permettent de gérer un manque d'informations sur l'environnement.

Dans [Car et Frank, 1994], les auteurs présentent le problème de planification : « s'orienter dans un grand réseau routier ». Leur approche de résolution se base sur la définition d'une ontologie permettant une description hiérarchique de l'espace. Cette ontologie définit un ensemble de règles hiérarchiques pour effectuer un raisonnement sur les niveaux hiérarchiques. Les résultats expérimentaux décrits dans [Car et Frank, 1994] montrent que l'exploitation du raisonnement hiérarchique proposée par cette ontologie simplifie la résolution du problème d'orientation dans un grand réseau routier par rapport à une résolution classique où le réseau est considéré comme un tout. Cependant, la représentation hiérarchique seule n'est pas suffisante pour la résolution d'un problème de planification.

La carte sémantique présentée dans [Galindo *et al.*, 2008] est définie comme une ontologie. En effet, elle se compose de deux parties la S-Box (*spatial box*) et la T-Box (*terminological box*). La S-Box contient des données factuelles concernant les états de l'environnement et les objets qui y sont. Elle met l'accent sur les relations spatiales (topologique et métrique). La T-Box contient la connaissance sémantique concernant le domaine en donnant un sens aux entités dans la S-Box en termes de concepts et de relations. Cette carte a été utilisée pour permettre une planification et une exécution des tâches qui ne pouvaient pas être réalisées sans l'aide de l'information topologique et métrique.

3.5.2 Ontologie spatiale et le bon fonctionnement du processus de planification

En planification, certains travaux exploitent les ontologies comme bases de connaissances spatiales, sans réellement exploiter l'aspect raisonnement. Par exemple dans le problème d'évacuation décrit dans [Lydia Kraus et Mijovi, 2011], l'ontologie est utilisée d'une manière statique pour décrire un aéroport en fonctions de régions (bureaux, couloirs, etc) et relations reliant ces régions (porte, ascenseurs, etc).

D'autres travaux exploitent l'ontologie spatiale comme un outil de contrôle du processus de planification. Dans [Bouguerra *et al.*, 2008], les auteurs proposent une nouvelle approche de contrôle de l'exécution d'un plan en se basant sur une ontologie spatiale appelée LOOM [MacGregor, 1999]. Cette ontologie est utilisée pour une classification des régions et une vérification de l'exécution du plan.

D'autres ontologies sont définies par l'extension d'une ontologie ou la combinaison de plusieurs ontologies comme décrit dans [Bateman et Farrar, 2005]. Nous pouvons citer l'extension de l'ontologie DOLCE (*Descriptive Ontology for Linguistic and Cognitive Engineering*) [Masolo *et al.*,] par des éléments de l'ontologie BFO (*Basic Formal Ontology*) [Smith et Grenon, 2002] pour gérer la connaissance spatiale dans le cas de planification de chemins.

Conclusion

De ce chapitre, nous retenons qu'une ontologie est une description explicite d'un domaine. Une ontologie est composée d'un ensemble de concepts, de leurs définitions et des relations entre ces concepts. Une ontologie spatiale est exprimée *via* un réseau sémantique. Nous avons présenté trois type d'ontologies. Tout d'abord, les ontologies de type *thesaurus* qui sont la première génération d'ontologies. Elles apportent une compréhension commune d'un vocabulaire. Ensuite, les ontologies de descriptives qui permettent de modéliser un domaine ou une activité. Elles sont utilisées comme un outil de compréhension des structures des données ou comme des bases de données. Enfin, les ontologies géographiques qui recouvrent les ontologies de l'espace, les ontologies de domaine géographiques et les ontologies spatialisées. Pour notre travail, nous retenons les ontologies spatialisées en considérant seulement la dimension spatiale. Ces ontologies s'appliquent à tout contenu nécessitant une modélisation localisée dans l'espace. Par abus de langage nous parlerons d'ontologies spatiales.

Plusieurs travaux s'intéressent aux ontologies spatiales dans certains domaines tels que la robotique [Dominey *et al.*, 2007], la médecine [Donnelly *et al.*, 2006], le traitement d'images [Dasiopoulou *et al.*, 2005] ont été proposés dans la littérature. Toutes ces ontologies se sont intéressées à la représentation des concepts spatiaux selon les domaines d'applications. En planification, plusieurs travaux exploitent les ontologies spatiales, certains d'une manière descriptive comme une base de connaissance pour stocker l'information spatiale, d'autres, au-delà de la fonctionnalité de stockage, exploitent les mécanismes d'inférence pour faire une classification des régions, afin de compléter la connaissance spatiale. La plupart des travaux en planification utilisent l'ontologie comme outil de contrôle pour le bon fonctionnement du processus de planification. Cependant, ceux qui intègrent l'ontologie spatiale dans la planification ne permettent pas de gérer tous les aspects de la connaissance spatiales que nous considérons.

C'est pour cette raison que nous nous sommes fixés de définir une ontologie spatiale permettant de gérer la représentation hiérarchique, qualitative (topologique et floue) et quantitative (numérique) en même temps. Cette ontologie sera ensuite couplée avec le langage de planification PDDL afin de fournir une représentation riche pour la planification.

L'ontologie que nous présenterons dans le chapitre 4 aura comme langage formel le langage OWL DL puisqu'il possède des propriétés de représentation très expressive et des propriétés de calcul très avantageuses pour les moteurs d'inférence.

Conclusion de la partie I

La partie état de l'art a présenté tous les éléments nécessaires pour nos travaux. Aussi, cette partie évoque certains travaux traitant la même thématique dans laquelle s'inscrit cette thèse. Tout d'abord, nous avons abordé la planification comme domaine visé pour y intégrer l'information spatiale. En particulier de la planification de tâches et la planification de chemins (chapitre 1). Nous avons montré l'importance de la connaissance spatiale dans la planification. Pour résoudre un problème de planification en considérant la dimension spatiale, il faut tout d'abord coupler un planificateur de tâches avec un planificateur de chemins. Ensuite, il faut formaliser la connaissance spatiale dans un langage comme PDDL. PDDL ne permet pas d'exprimer une connaissance spatiale complexe. Plusieurs extensions gérant différents types de connaissance ont été proposées dans la littérature. À notre connaissance, il n'y a pas eu d'extensions de PDDL permettant de gérer une connaissance spatiale. Dans le chapitre 5, nous présentons une extension de PDDL qui permet de répondre à cette limite en considérant les aspects quantitatives, qualitatives et hiérarchiques.

Notre objectif est de définir un cadre formel combinant ces différentes représentations et permettant l'exploitation de l'information spatiale. Pour cela, nous utilisons les ontologies comme technique de représentation et raisonnement spatial. Dans ce travail, nous nous concentrons principalement sur les ontologies géospatiales pour la description des éléments et les ontologies pour la définition de relations spatiales.

Deuxième partie

Contributions

Introduction de la partie II

Après l'étude de l'état de l'art, nous nous intéressons à la représentation des entités spatiales et des relations spatiales. D'un point de vue géométrique, une entité spatiale est représentée par des formes géométriques simples. La représentation hiérarchique d'une entité spatiale permet une structure pour une description multi-niveaux de l'environnement. La représentation des relations spatiales se base sur la représentation des relations spatiales entre deux entités spatiales. Nous considérons, la représentation topologique, métrique et floue.

Cette partie présente les contributions de cette thèse au problème de planification tenant compte de la dimension spatiale dans le cadre d'interaction homme-robot. Ces contributions sont résumées par la figure 1.

1. Définition d'une ontologie spatiale combinant différentes représentations de la connaissance spatiale, à savoir ; qualitative, quantitative et hiérarchique. De plus, cette ontologie permet un raisonnement spatial en définissant des règles d'inférence (chapitre 4).
2. Introduction d'un planificateur qui combine un raisonnement symbolique et un raisonnement spatial en intégrant l'ontologie définie dans le chapitre 4 (chapitre 5).
3. Extension de l'opérateur classique de planification afin de prendre en compte la composante spatiale (chapitre 5).
4. Extension de PDDL afin de formaliser la connaissance spatiale dans un problème de planification (chapitre 5).
5. Introduction d'un ensemble de requêtes utilisées lors du processus de planification (chapitre 5).
6. Définition d'un problème de planification de chemins entre deux positions connues d'une manière imprécises ou floues (chapitre 6).

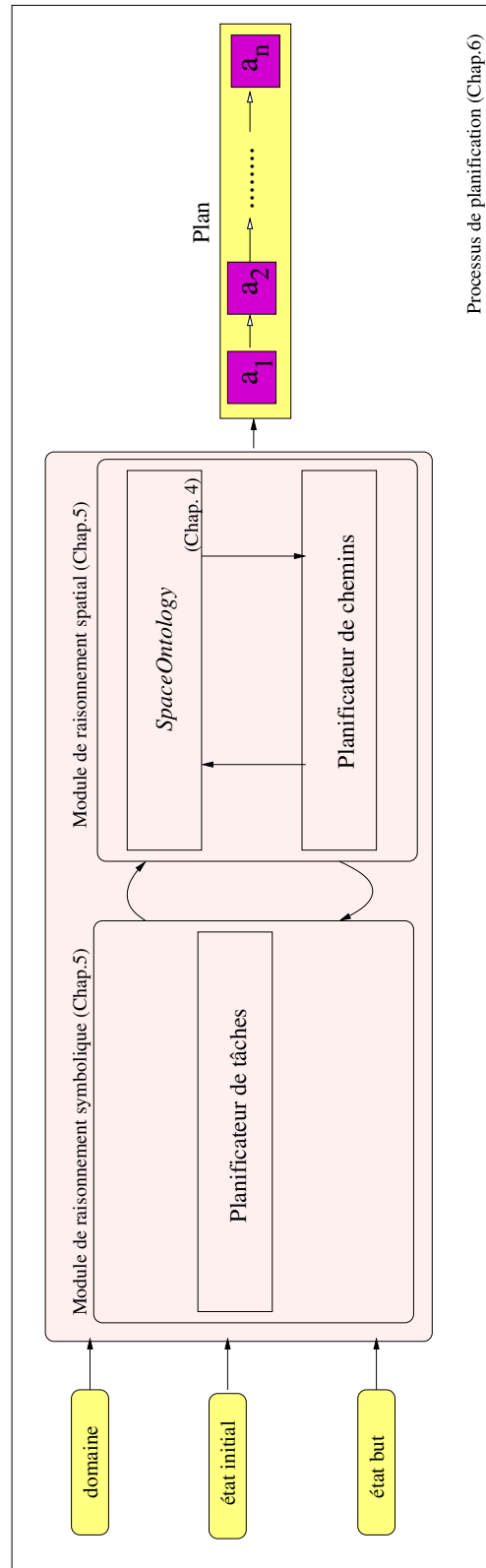


FIGURE 1 – Représentation graphique de nos contributions.

Chapitre 4

Connaissance spatiale :

SpaceOntology

Sommaire

4.1	Définition d'une ontologie spatiale	90
4.1.1	Entité spatiale	92
4.1.2	Relations topologiques	94
4.1.3	Relations de distances	96
4.1.4	Relations entre entités spatiales	99
4.2	Raisonnement avec <i>SpaceOntology</i>	102
4.2.1	Raisonnement hiérarchique	102
4.2.2	Raisonnement avec les relations de distance	103
4.2.3	Raisonnement avec les relations topologiques	105
4.3	Mécanisme de composition	107
4.3.1	Représentation des relations topologiques	107
4.3.2	Décomposition des relations topologiques	108
4.3.3	Calcul de la composition	110
4.3.4	Consistance de la composition	111
4.3.5	Maintenir la cohérence pour les relations topologiques	113

Nous nous intéressons à différentes représentations spatiales, à savoir : les représentations hiérarchique, qualitative et quantitative. À notre connaissance, il n'existe pas un travail combinant tous ces aspects à la fois. Ce chapitre présente notre première contribution : la définition d'une ontologie spatiale nommée *SpaceOntology* [Belouaer *et al.*, 2010b, Belouaer *et al.*, 2010a]. Elle permet la représentation et le raisonnement spatial. Ce chapitre est organisé de la manière suivante. La section 4.1 présente *SpaceOntology* en définissant les concepts, les relations entre les concepts. La section 4.2 présente le mécanisme de raisonnement offert par cette ontologie. La section 4.3 détaille la composition des relations topologiques qui est une opération primordiale et importante pour le raisonnement.



4.1 Définition d'une ontologie spatiale

Nous avons choisi l'ontologie comme technique de représentation et de raisonnement spatial. Cette technique permet une description complexe et riche des connaissances. Elle fournit une structure facilement extensible, offre des mécanismes d'inférence permettant d'enrichir et de compléter la connaissance spatiale sur un environnement. Surtout, elle structure l'information spatiale en vue d'être exploitée. Avant d'aller plus loin, nous présentons un exemple permettant d'illustrer nos définitions.

Exemple 3 *Considérons l'espace global décrit dans la figure 4.1. Il s'agit d'un bâtiment qui se compose de quatre étages $\{E_0, E_1, E_2, E_3\}$.*

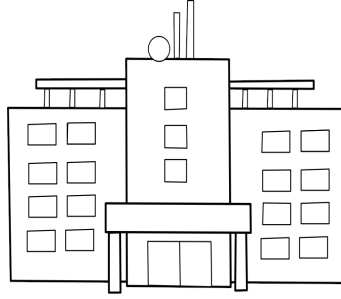


FIGURE 4.1 – Représentation d'un immeuble B_1 avec plusieurs étages.

Chaque étage se compose à son tour d'un ensemble de bureaux, de couloirs, etc. Nous considérons, les deux étages E_1 et E_2 (figure 4.2). Pour la suite, nous notons par $b_{i,j}$ $j^{\text{ème}}$ bureau dans le $i^{\text{ème}}$ étage. Par exemple, le bureau $b_{1,2}$ identifie le deuxième bureau au premier étage. De la même manière, tout élément $e_{i,j}$ correspond au $j^{\text{ème}}$ élément de l'étage i (par exemple : $c_{i,j}$ pour les couloirs, $p_{i,j}$ pour les portes, etc).

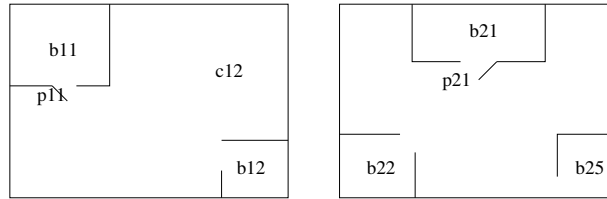


FIGURE 4.2 – Deux plans d'étage dans un immeuble : à gauche le plan de l'étage E_1 , à droite celui de l'étage E_2 .

Nous avons mis en place l'ontologie spatiale *SpaceOntology* [Belouaer et al., 2010b]. Elle se décompose en deux parties. La première est consacrée à la représentation de caractéristiques géographiques telles que : une ville, un bâtiment, ou encore un bureau. La seconde partie permet la représentation de relations spatiales qui sont des relations topologiques et des relations de

distance [Belouaer *et al.*, 2010b]. D'un point de vue formel, nous optons pour le formalisme OWL DL décrit dans le chapitre 3, page 57.

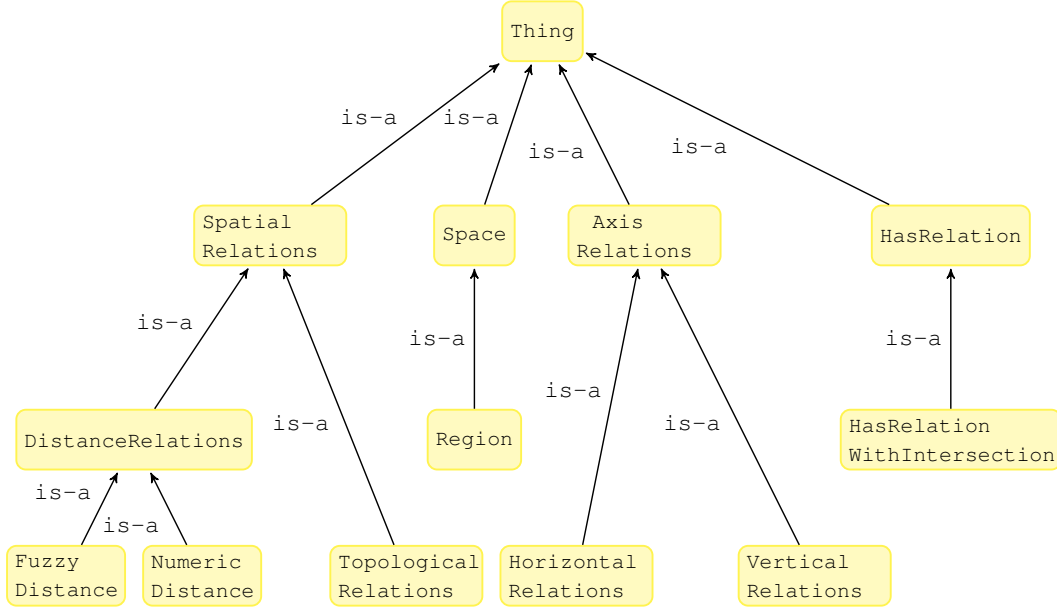


FIGURE 4.3 – Représentation graphique de *SpaceOntology*.

La figure 4.3 présente l'ensemble des concepts définissant *SpaceOntology*. Ces concepts sont liés par un lien d'héritage « est un » (« is-a ») défini par l'opérateur $\sqsubseteq$ (tableau 3.1, page 63). Le concept de racine est le concept **Thing** défini par le constructeur $\top$.

Définition 10 *SpaceOntology* est un couple (Φ, R) tel que :

- Φ est l'ensemble des concepts spatiaux ;
- R est l'ensemble des relations entre les concepts.

La définition 10 permet d'introduire formellement *SpaceOntology*. L'ensemble Φ correspond à tous les concepts définis dans la figure 4.3 (**SpatialRelations**, **Regions**, etc.). L'ensemble R correspond à tous les liens entre les concepts (**is-a**, **label**, etc.). Les liens entre les concepts sont décrits et détaillés dans l'annexe D.

Définition 11 Une instance de *SpaceOntology* est un triplet (Φ, R, I) tel que :

- Φ est l'ensemble des concepts spatiaux ;
- R est l'ensemble des relations entre ces concepts ;
- I est l'ensemble des individus instanciant les concepts Φ et décrivant le domaine tel que I est un sous-ensemble de ce dernier.

La définition 11 permet d'introduire formellement une instance de *SpaceOntology* pour décrire un domaine donné. Dans ce qui suit, nous détaillons la TBox (ou niveau terminologique) c'est-à-dire, les ensembles Φ et R . Nous ne parlerons pas de la ABox (ou niveau assertionnel)

car elle varie selon le domaine. L'annexe D, présente une instance de *SpaceOntology* décrivant l'espace présenté dans l'exemple 3.

4.1.1 Entité spatiale

Une entité spatiale e est une entité localisable dans un espace donné. Elle peut être :

statique : tout élément fixe dans l'espace dont la position n'évolue que par une action accomplie par un agent (par exemple, une pièce ou un meuble).

dynamique : toute entité spatiale dont la position évolue au cours du temps dans l'espace (un humain, un robot, etc.).

Pour décrire une entité spatiale, nous définissons deux concepts : le concept **Space** et le concept **Regions** (figure 4.4).

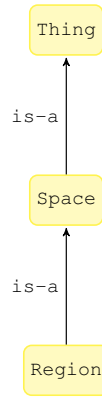


FIGURE 4.4 – Représentation graphique des concepts **Space** et **Regions**.

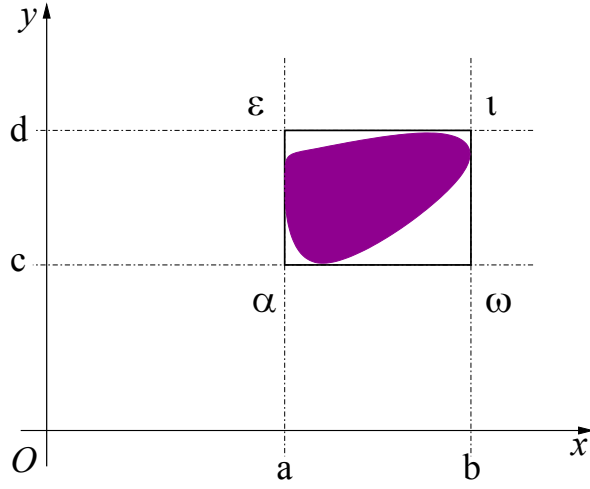
Le concept **Space** permet de représenter un espace global donné (un pays, une ville, un quartier, un immeuble, etc.). Ce concept admet un ensemble d'attributs qui sont nécessaires pour définir une sémantique spatiale (annexe D). Plus particulièrement, l'attribut **label** permet d'identifier un espace. Par exemple, B_1 est une instance du concept **Space** identifié par le label « *le bâtiment B_1* » (exemple 3). Le concept **Regions** permet de représenter des entités spatiales statiques ou dynamiques dans l'espace. Il est un sous concept de **Space** : $\text{Regions} \sqsubseteq \text{Space}$ (figure 4.4). Ce concept permet une définition géométrique de toute entité spatiale.

Représentation géométrique

Nous considérons une représentation en deux dimensions définie par $(O, \vec{x}, \vec{y})$. D'un point de vue géométrique, on associe à une entité spatiale le plus petit rectangle qui l'englobe. δ, ζ, λ et η sont les sommets du rectangle délimitant l'entité spatiale e (figure 4.5).

Les attributs définissant le concept **Region** sont (d'après la figure 4.5) :

- L'attribut **point** correspond à l'origine du rectangle, ici le point δ ;
- L'attribut **length** correspond à la longueur du rectangle $|b - a|$;



- a et b sont les coordonnées minimale et maximale de la projection de e sur l'axe (ox) ;
- c et d sont les coordonnées minimale et maximales de la projection de e sur l'axe (oy) ;
- δ est la donnée de (a, c) ;
- ζ est la donnée de (b, c) ;
- λ est la donnée de (b, d) ;
- η est la donnée de (a, d) .

FIGURE 4.5 – Représentation géométrique d'une entité spatiale.

- L'attribut **width** correspond à la largeur du rectangle $|d - c|$.

Considérons le premier étage E_1 une instance du concept **Regions** : « E_1 : **Regions** » (exemple 3). D'un point de vue géométrique, la région E_1 est délimitée par le rectangle dont l'origine est donnée par le point δ tel que $\delta = (2, 1)$, de longueur 5 et de largeur 3 (annexe D).

Représentation hiérarchique

Le niveau le plus abstrait de la représentation (*universal level* [Fernandez-Madriral, 2004]) correspond au niveau 0. Ce niveau est représenté par le concept **Space**. Le niveau le plus détaillé (*ground level* [Fernandez-Madriral, 2004]) correspond au niveau l que nous fixons. Le concept **Regions** représente toute entité spatiale appartenant aux différents niveaux hiérarchiques intermédiaires et final. Une région est considérée comme un espace d'un niveau hiérarchique supérieur au niveau hiérarchique courant. Par exemple, un immeuble, qui est l'espace considéré, se compose de quatre étages. Un étage est une région se composant de plusieurs pièces, elles mêmes régions (exemple 3). Pour représenter l'appartenance d'une région à un niveau hiérarchique, nous définissons l'attribut **level**. Par exemple, le bâtiment B_1 est une région du premier niveau hiérarchique : **level**= 1 (figure 4.6).

Pour exprimer le lien hiérarchique entre régions, nous avons défini deux liens transitifs et symétrique entre les concepts **Space** et **Regions** (figure 4.7). Le lien **isPartOf** exprime qu'une région peut appartenir à une ou plusieurs régions (flèche verte) : **Regions** $\sqsubseteq$ **Space** $\sqcap$ $\exists$ **isPartOf.Space** ≥ 1 **isPartOf**. Le lien **consistsOf** exprime qu'une région peut être décomposée en plusieurs sous régions (flèche rouge) : **Space** $\sqsubseteq$ **Thing** $\sqcap$ $\exists$ **consistsOf.Regions** ≥ 1 **consistsOf**.

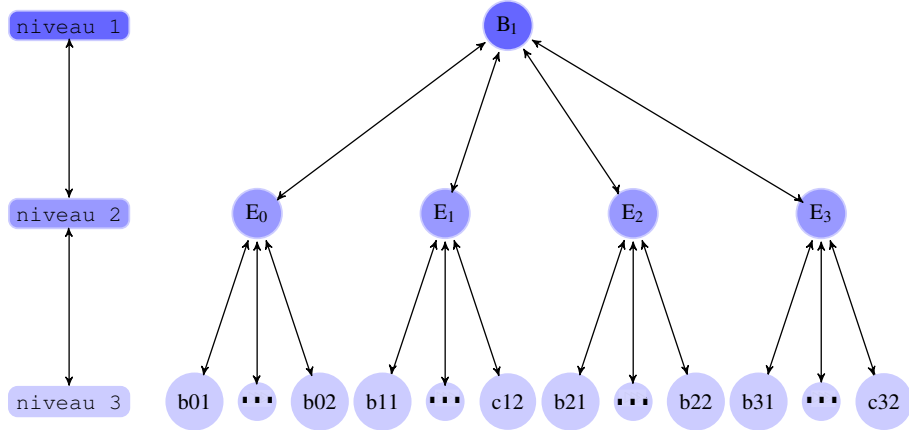


FIGURE 4.6 – Arbre hiérarchique du bâtiment décrit dans la figure 4.2 et ses différents niveaux hiérarchiques.

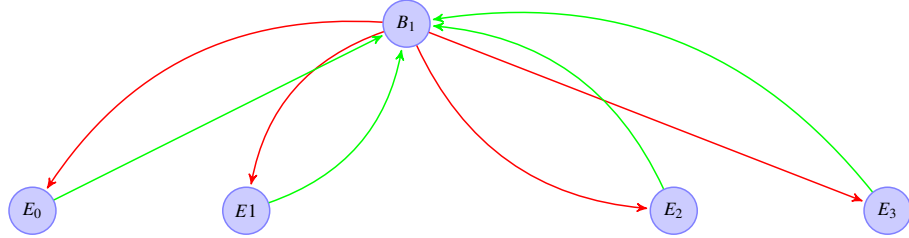


FIGURE 4.7 – Représentation graphique de l’organisation hiérarchique entre l’espace B_1 et les régions $\{E_0, E_1, E_2, E_3\}$ à l’aide des liens *isPartOf* (en vert) et *consistsOf* (en rouge).

4.1.2 Relations topologiques

Pour manipuler les relations spatiales entre entités, nous exploitons l’algèbre de rectangles [Baldani *et al.*, 1998] (chapitre 2, page 43). Cependant, le nombre conséquent de relations (169 relations spatiales) rend complexe le raisonnement. Le formalisme ABLR (pour *Above Below Left Right*) présente un compromis entre l’expressivité et le nombre de relations [Laborie *et al.*, 2006]. Cette représentation consiste à réduire le nombre de relations tout en préservant la caractéristique directionnelle de la représentation dans l’algèbre d’Allen [Allen, 1983]. La figure 4.8 donne l’ensemble des relations topologiques par axe. Nous notons X le rectangle définissant une région et ces projections sur chaque axe (ox et oy).

Définition 12 Une relation topologique ABLR est un couple $\langle r_x, r_y \rangle$ tel que $r_x \in \{L, O_L, C_x, I_x, O_R, R\}$ et $r_y \in \{A, O_A, C_y, I_y, O_B, B\}$.

D’après la définition 12, une relation topologique est le résultat d’un mécanisme de combinaison entre une relation horizontale r_X et une relation verticale r_Y ($r_X \oplus r_Y$). Ainsi, il existe 36 relations spatiales topologiques au lieu des 169 initialement (tableau 4.1).

$\frac{X}{X} \quad \frac{Y}{Y}$	$\frac{X}{X} \quad \frac{Y}{Y}$	$\frac{X}{Y}$	$\frac{X}{Y} \quad \frac{Y}{Y}$	$\frac{Y}{Y} \quad \frac{X}{X}$	$\frac{Y}{Y} \quad \frac{X}{X}$
X à gauche de Y <i>left</i> (L)	X chevauche l'extrémité gauche de Y <i>overlaps-left</i> (O_L)	X contient Y <i>contains</i> (C_x)	X est contenu dans Y <i>inside</i> (I_x)	X chevauche l'extrémité droite de Y <i>overlaps-right</i> (O_R)	X à droite de Y <i>right</i> (R)
X au dessus Y <i>above</i> (A)	X chevauche l'extrémité supérieure de Y <i>overlaps-above</i> (O_A)	X contient Y <i>contains</i> (C_y)	X est contenu dans Y <i>inside</i> (I_y)	X chevauche l'extrémité inférieure de Y <i>overlaps-below</i> (O_B)	X au dessous Y <i>below</i> (B)

FIGURE 4.8 – Groupement des relations d'Allen se fait en groupant les relations avec les propriétés directionnelles communes [Laborie *et al.*, 2006].

$r_x \oplus r_y$	A	O_A	C_y	I_y	O_B	B
L	$\langle L, A \rangle$	$\langle L, O_A \rangle$	$\langle L, C_y \rangle$	$\langle L, I_y \rangle$	$\langle L, O_B \rangle$	$\langle L, B \rangle$
O_L	$\langle O_L, A \rangle$	$\langle O_L, O_A \rangle$	$\langle O_L, C_y \rangle$	$\langle O_L, I_y \rangle$	$\langle O_L, O_B \rangle$	$\langle O_L, B \rangle$
C_x	$\langle C_x, A \rangle$	$\langle C_x, O_A \rangle$	$\langle C_x, C_y \rangle$	$\langle C_x, I_y \rangle$	$\langle C_x, O_B \rangle$	$\langle C_x, B \rangle$
I_x	$\langle I_x, A \rangle$	$\langle I_x, O_A \rangle$	$\langle I_x, C_y \rangle$	$\langle I_x, I_y \rangle$	$\langle I_x, O_B \rangle$	$\langle I_x, B \rangle$
O_R	$\langle O_R, A \rangle$	$\langle O_R, O_A \rangle$	$\langle O_R, I_y \rangle$	$\langle O_R, C_y \rangle$	$\langle O_R, O_B \rangle$	$\langle O_R, B \rangle$

TABLE 4.1 – 36 relations possibles avec le formalisme ABLR.

Ces relations expriment les relations topologiques comme par exemple « se chevauche » ($\langle O_L, O_B \rangle$), les relations d'orientation comme par exemple « à droite » ($\langle L, B \rangle$) et d'autres relations d'orientation telles que « au même niveau » ($\langle L, C_y \rangle$).

Pour définir les relations topologiques du formalisme ABLR dans *SpaceOntology*, nous nous basons sur la définition 12 dans *SpaceOntology*.

Nous définissons le concept **AxisRelations** (**AxisRelations** $\sqsubseteq$ **Thing**) qui englobe les concepts **HorizontalRelations** et **VerticalRelations** (figure 4.9). Le concept **HorizontalRelations** définit les six relations décrite dans le formalisme ABLR pour l'axe horizontal :

$$\text{HorizontalRelations} \sqsubseteq \text{AxisRelations} \sqcap \text{HorizontalRelations} \equiv \{L, O_L, C_x, I_x, O_R, R\}$$

Le concept **VerticalRelations** définit les six relations décrite dans le formalisme ABLR pour l'axe vertical :

$$\text{VerticalRelations} \sqsubseteq \text{AxisRelations} \sqcap \text{VerticalRelations} \equiv \{A, O_A, C_y, I_y, O_B, B\}$$

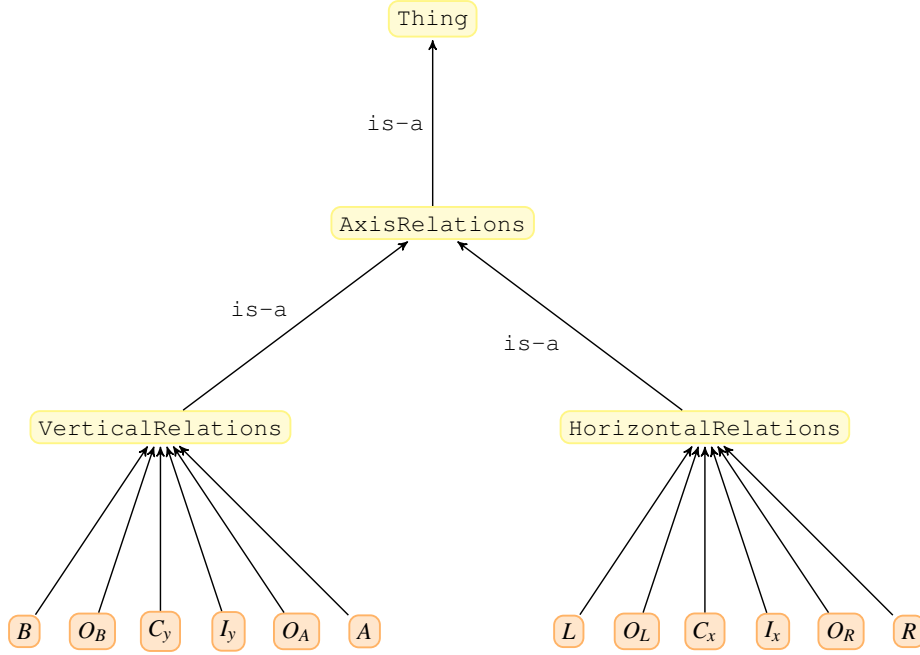


FIGURE 4.9 – Représentation graphique du concept **AxisRelations**.

Ainsi, formellement nous définissons le concept **TopologicalRelations** dans *SpaceOntology* comme suit :

$$\text{TopologicalRelations} \sqsubseteq \text{SpatialRelations}$$

$$\sqcap \exists \text{ isAnHorizontalRelation.}\{L, O_L, C_x, I_x, O_R, R\} = 1 \text{ isAnHorizontalRelation}$$

$$\sqcap \exists \text{ isAVerticalRelation.}\{A, O_A, C_y, I_y, O_B, B\} = 1 \text{ isAVerticalRelation}$$

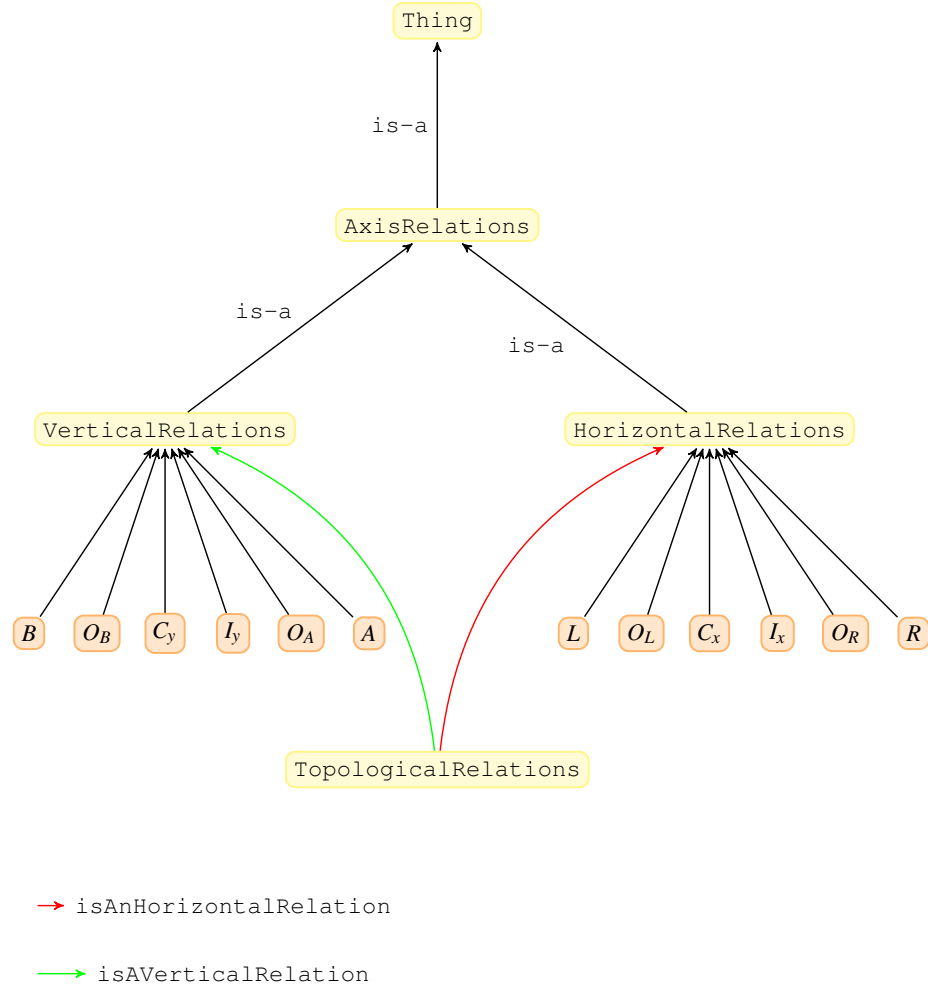
4.1.3 Relations de distances

Nous représentons une relation de distance de deux manières : numérique à l'aide de la distance euclidienne ou bien floue à l'aide de variables linguistiques. Dans *SpaceOntology*, nous définissons le concept **DistanceRelations** pour définir une relation de distance (figure 4.11).

Représentation numérique de la distance

Les distances numériques sont aussi utilisées pour décrire une configuration spatiale. Dans ce qui suit, chaque rectangle est réduit à son centre de symétrie. Ainsi, une entité spatiale est présentée par le centre de symétrie du rectangle qui la représente. Un centre est identifié par le nom de l'entité qu'il représente (figure 4.12).

Pour la distance numérique entre deux entités spatiales E1 et E2, nous nous intéressons à la distance euclidienne entre les centres de symétries de celles-ci (figure 4.12). Le concept

FIGURE 4.10 – Représentation graphique du concept `TopologicalRelations`.

`NumericDistance` est un sous concept du concept `DistanceRelations` qui permet de décrire une relation de distance d'une manière numérique (figure 4.11). Ce concept définit l'attribut `distance` qui permet de donner la distance numérique. La définition formelle de ce concept dans *SpaceOntology* est comme suit :

$$\text{NumericDistance} \sqsubseteq \text{DistanceRelations}$$

Représentation floue de la distance

Nous considérons quatre variables linguistiques pour décrire une distance d'une manière floue : « proche », « assez proche », « assez loin » et « loin » (figure 4.13).

Nous définissons alors trois cercles dont la référence est le centre, chaque cercle correspond à un niveau de proximité défini par une des variables linguistique (figure 4.13). Le dernier niveau de proximité est au-delà du dernier cercle (figure 4.13). Ces trois cercles C_1 , C_2 et C_3 de

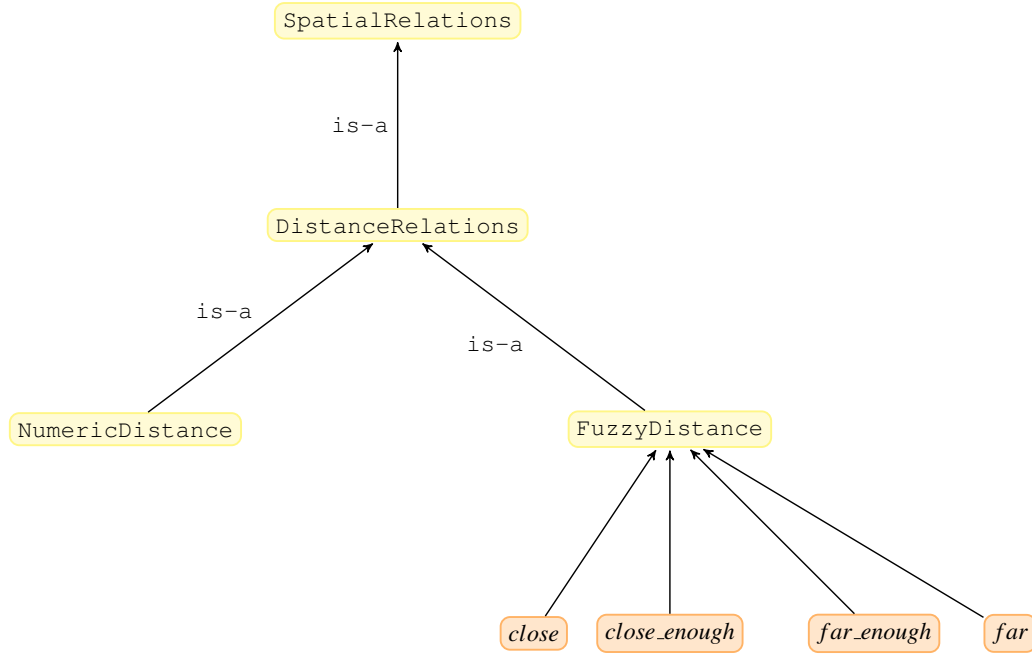


FIGURE 4.11 – Représentation graphique du concept **DistanceRelations**.

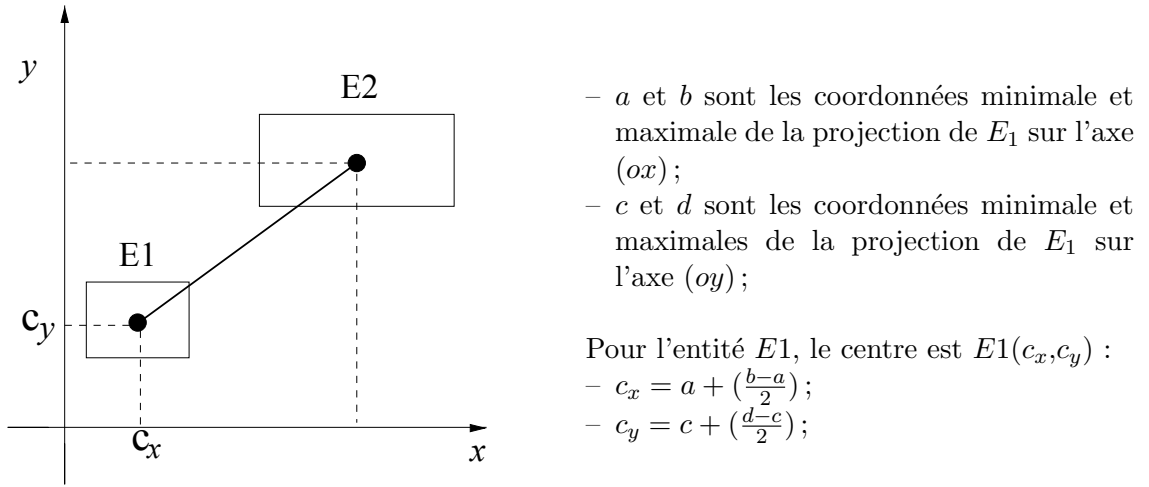


FIGURE 4.12 – Distance euclidienne

même centre ont les rayons respectifs r_1, r_2, r_3 tel que $r_1 \leq r_2 \leq r_3$ (figure 4.13). Le concept **FuzzyDistance** est un sous concept du concept **DistanceRelations** qui permet de décrire une relation de distance d'une manière floue (figure 4.11). Sa définition formelle est comme suit :

FuzzyDistance $\sqsubseteq$ **DistanceRelations** $\sqcap$ **FuzzyDistance** $\equiv \{\text{close}, \text{close_enough}, \text{far_enough}, \text{far}\}$

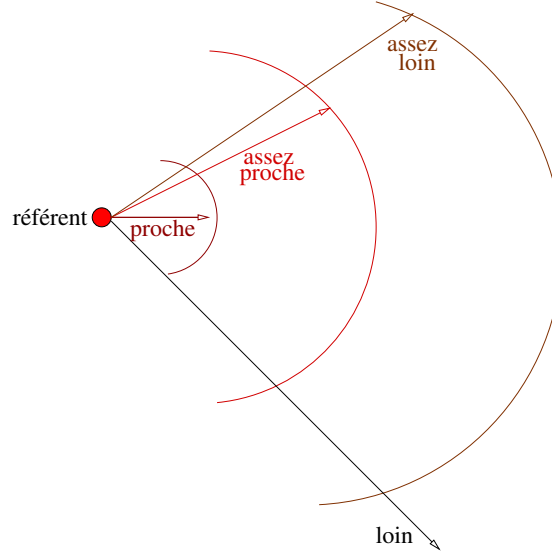


FIGURE 4.13 – Découpage d'une région en quatre niveaux de proximité : « proche », « assez proche », « assez loin » et « loin ».

Les éléments de l'ensemble $\{\text{close}, \text{close_enough}, \text{far_enough}, \text{far}\}$ correspondent respectivement aux variables linguistiques « proche », « assez proche », « assez loin » et « loin ». Ces éléments sont des individus qui permettent de définir le concept **FuzzyDistance**.

4.1.4 Relations entre entités spatiales

Afin de caractériser la position d'une entité spatiale par rapport à une autre, nous utilisons les relations spatiales comme par exemple « à droite du bâtiment », « traverse la ville », « à 1 mètres de la table », etc. Dans notre ontologie, une relation spatiale est considérée comme un concept et non comme un lien entre deux régions (figure 4.14) : **SpatialRelations** $\sqsubseteq$ **Thing**. Le concept **SpatialRelations** englobe :

- le concept **TopologicalRelations** permet la description des relations topologiques :

$$\text{TopologicalRelations} \sqsubseteq \text{SpatialRelations}$$

- le concept **DistanceRelations** permet la description des relations de distance :

$$\text{DistanceRelations} \sqsubseteq \text{SpatialRelations}$$

Pour exprimer une relation spatiale entre deux ou plusieurs entité spatiales, nous utilisons le concept **HasRelation**. Ce concept permet de spécifier : la région de référence, la relation spatiale et la ou les régions cibles. Formellement, le concept **HasRelation** est défini comme suit :

$$\begin{aligned} &\text{HasRelation} \sqsubseteq \text{Thing} \\ &\sqcap \exists \text{ concernsSpatialRelation} . \{ \text{SpatialRelations} \} \sqcap = 1 \text{ concernsSpatialRelation} \end{aligned}$$

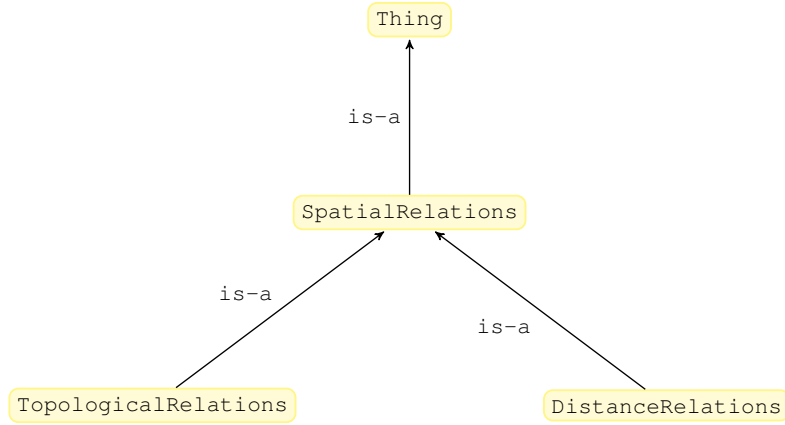


FIGURE 4.14 – Représentation graphique du concept **SpatialRelations**.

$\sqcap \exists \text{hasReferent}.\{\text{Regions}\} \sqcap \geq 1 \text{hasReferent} \sqcap \exists \text{hasTarget}.\{\text{Regions}\} \sqcap = 1 \text{hasTarget}$

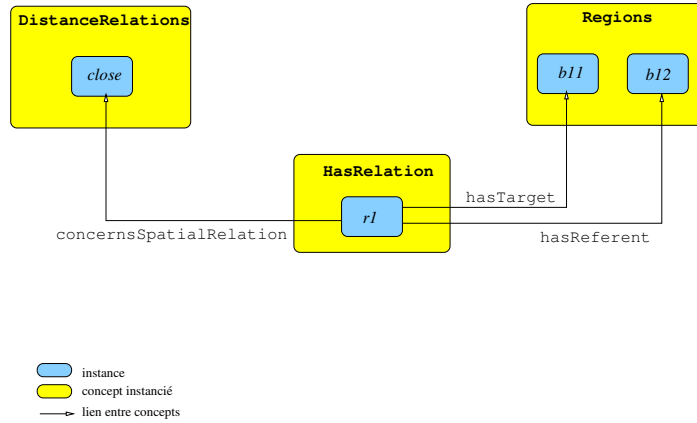


FIGURE 4.15 – Liens entre concepts et leurs instanciations.

La figure 4.15 illustre comment représenter une relation spatiale. Une instance du concept **HasRelation** *r1* relie deux instance du concept **Regions** : *b11* et *b12* tel que *b12* est la région de référence (le lien **hasReferent**) et la région *b11* est la région cible (le lien **hasTarget**) par la relation de distance *close* instance du concept **DistanceRelations**.

Nous avons présenté les trois types de cadres de référence : absolu, relatif et intrinsèque (page 39). Dans ce travail, nous ne considérons pas le cadre de référence absolu. En effet, l'utilisation de ce type de cadre n'est pas adéquat dans des espaces de petites tailles (par exemple, un bâtiment, un bureau, etc.) [Liu *et al.*, 2010]. De plus, ce type de cadre est basé sur les points de référence abstraits et prédéfinis comme les notions du « nord » ou du « sud ». Dans le cas, où on a plusieurs agents qui décrivent une même scène, la définition de ces points peut être ambiguë. En effet, Levinson a démontrée par une série de tests sur différents individus que la

définition de ces points dépend de certains critères propre à l'individu testé (dans ces travaux les tests ont porté sur la langue maternelle) [Levinson, 1996].

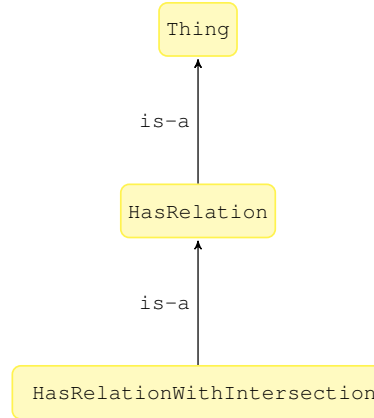


FIGURE 4.16 – Représentation graphique du concept `HasRelation`.

Le concept `HasRelation` englobe un concept `HasRelationWithIntersection` (figure 4.16). Ce concept permet de décrire la relation spatiale entre deux ou plusieurs régions ainsi que l'ensemble de régions qui forment l'intersection entre les deux ou plusieurs régions reliées par cette relation spatiale. Formellement, le concept `HasRelationWithIntersection` est défini comme suit :

$$\text{HasRelationWithIntersection} \sqsubseteq \text{HasRelation}$$

Considérons la figure 4.2, les deux régions *b11* et *c12* sont reliées par la relation spatiale : *In* défini par le couple $\langle I_x, I_y \rangle$ (extrait 4.1) : *b11* $\langle I_x, I_y \rangle$ *c12*, tel que *b11* est la région cible et *c12* la région de référence.

```

<HasRelationWithIntersection:ID="HasRelationWithIntersection_10">
  <concernsRelation>
    <TopologicalRelations rdf:ID="In">
      <isAVerticalRelation rdf:resource="#I_y"/>
      <isAnHorizontalRelation rdf:resource="#I_x"/>
    </TopologicalRelations>
  </concernsRelation>
  <labelhasrelation rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    HasRelationWithIntersection_10</labelhasrelation>
  <hasTarget rdf:resource="#b11"/>
  <hasReferent rdf:resource="#c21"/>
</HasRelationWithIntersection>
  
```

EXTRAIT 4.1 - Relation topologique avec intersection décrite avec *SpaceOntology*.

4.2 Raisonnement avec *SpaceOntology*

La section 4.1 à présenté *SpaceOntology* comme technique de représentation de la connaissance spatiale. Cette section présente cette ontologie d'un point de vue raisonnement.

Les méthodes de raisonnement permettent d'inférer de nouvelles informations à partir de ce qui existe dans le but d'enrichir la description de l'environnement avec une nouvelle connaissance spatiale ou pour gérer un manque d'information. Les méthodes d'inférence sont traduites par un ensemble de règles définies dans *SpaceOntology*. Dans cette section, nous présentons quelques unes pour plus de détails, nous vous invitons à regarder le lien suivant : <http://lbelouae.perso.info.unicaen.fr/SpaceOntology>. Dans ce travail, le raisonnement spatial concerne le raisonnement hiérarchique, l'interprétation des relations de distance floues et le raisonnement topologique.

4.2.1 Raisonnement hiérarchique

Considérer l'espace dans sa globalité avec les différents niveaux hiérarchiques n'est pas toujours nécessaire et rend le raisonnement spatial très coûteux (chapitre 2). L'organisation hiérarchique permet de simplifier le raisonnement, selon le niveau de détails. Pour ce faire, nous définissons la notion d'une *zone d'intérêt hiérarchique* (définition 13).

Définition 13 Une zone d'intérêt hiérarchique notée $\mathcal{Z}_{rg}^l$ est une région de l'espace à définir ou à localiser dans un niveau hiérarchique l et englobant la région rg .

Considérons l'exemple 3. La hiérarchie de l'espace décrite dans la figure 4.6, permet d'identifier des zones d'intérêts hiérarchiques. Par exemple :

- $\mathcal{Z}_{b11}^1 = B_1$, c'est-à-dire que la région qui englobe le bureau $b11$ au niveau hiérarchique $l = 1$ est le bâtiment B_1 ;
- $\mathcal{Z}_{b11}^2 = E_1$, c'est-à-dire que la région qui englobe la porte $b11$ au niveau hiérarchique $l = 2$ est le premier étage E_1 .

Nous avons défini dans *SpaceOntology* deux liens `isPartOf` et `consistsOf` permettant de définir les liens hiérarchiques entre régions (paragraphe « Représentation hiérarchique », page 93). Ces deux liens sont symétriques et transitives.

$$\text{isPartOf}(rg_1, rg_2) \sqcap \text{isPartOf}(rg_2, rg_3) \longrightarrow \text{isPartOf}(rg_1, rg_3) \quad (\mathcal{H}_{\mathcal{R}}.1)$$

$$\text{consistsOf}(rg_1, rg_2) \sqcap \text{consistsOf}(rg_2, rg_3) \longrightarrow \text{consistsOf}(rg_1, rg_3) \quad (\mathcal{H}_{\mathcal{R}}.2)$$

Les règles hiérarchiques ($\mathcal{H}_{\mathcal{R}.1$ et $\mathcal{H}_{\mathcal{R}.2$) permettent pour n'importe quelle région rg au niveau l de connaître ses antécédents (un antécédent est toute région incluant rg au niveau hiérarchique l_a tel que $l_a \in [0, l[$) et de connaître tout ses successeurs (un successeur est toute région incluse dans rg au niveau hiérarchique l_s tel que $l_s \in]l, n]$ avec n est le dernier niveau hiérarchique). Ces règles sont exécutées par le raisonneur Pellet (chapitre 3).

L'organisation hiérarchique de l'espace a un impact sur le raisonnement concernant les relations spatiales et en particulier sur les relations de distance.

4.2.2 Raisonnement avec les relations de distance

Le raisonnement exploitant les relations de distance permet de délimiter une entité spatiale en interprétant une relation floue en une relation numérique et inversement. Ce raisonnement dépend de plusieurs critères, notamment :

- du contexte hiérarchique des déclarations. En effet, dans une ville, une distance de 3 mètres est considérée comme « proche », cependant dans un bureau cette même distance est considérée comme « loin » ;
- du type de l'action à exécuter. En effet si on considère l'action « poser un objet sur une table », il faut que l'agent qui exécute cette action soit suffisamment « proche » de la table pour pouvoir poser l'objet. Un cas particulier, est celui où cette action concerne un humain. Par exemple, pour l'action « donner un objet à un humain », l'agent doit être « suffisamment proche » de l'humain pour que l'interaction soit réalisée sans que ce dernier se sente « agressé ». Pour cela, nous exploitons l'étude des échanges entre les humains réalisée par Hall [Hall, 1966]. Cette étude a permis de définir différents zones de proximité pour les humains (figure 4.17) :

1. la zone intime ne doit pas être franchie sinon l'humain est dans une situation inconfortable, et refuse d'interagir,
2. la zone personnelle respecte la zone intime de l'humain et permet des interactions informelles (par exemple, prendre un objet, donner un objet),
3. la zone sociale les interactions sont plus formelles,
4. la zone publique ne permet pas des interactions directes avec l'humain.

L'idée est de trouver un moyen pour effectuer un raisonnement général. Pour cela, nous fixons des seuils permettant de définir des zones de proximité. Nous découpons l'espace en trois cercles C_1, C_2 et C_3 , de rayons respectifs r_1, r_2 et r_3 , puisque nous nous intéressons à quatre relations floues : « proche », « assez proche », « assez loin » et « loin » (figure 4.13). Le raisonnement concernant la distance est une interprétation d'une relation numérique vers une relation floue et inversement. Dans [Schockaert, 2008], les auteurs ont défini deux degrés : $N_{(\alpha, \beta)}$ et $F_{(\alpha, \beta)}$.

$$N_{(\alpha, \beta)}(p, q) = \begin{cases} 1 & \text{si } d(p, q) \leq \alpha \\ 0 & \text{si } d(p, q) \geq \alpha + \beta \\ \frac{\alpha + \beta - d(p, q)}{\beta} & \text{sinon } \beta \neq 0 \end{cases} \quad (4.1)$$

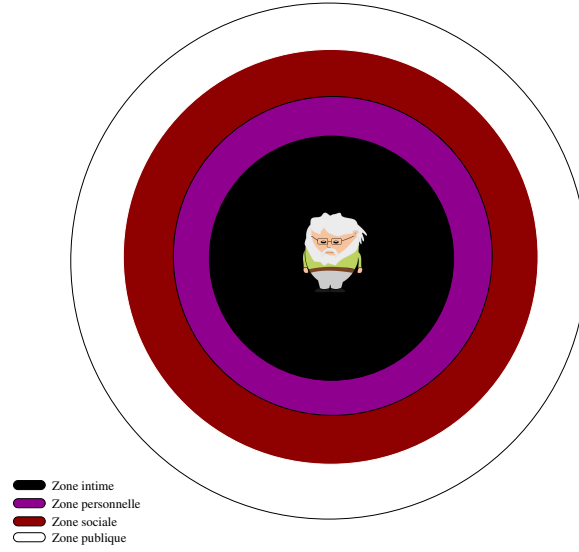


FIGURE 4.17 – Définition de zone de proximité selon Hall [Hall, 1966].

Le degré $N_{(\alpha,\beta)}(p,q)$ pour « proche » (*Near*) est un degré qui mesure de combien deux points p et q sont proches. À partir de l'équation 4.1, si $N_{(\alpha,\beta)}(p,q) = 1$ alors la distance entre p et q est qualifiée de « proche » si $N_{(\alpha,\beta)}(p,q) = 0$ alors la distance entre p et q est qualifiée de « loin ».

$$F_{(\alpha,\beta)}(p,q) = \begin{cases} 1 & \text{si } d(p,q) > \alpha + \beta \\ 0 & \text{si } d(p,q) \leq \alpha \\ \frac{d(p,q) - \alpha}{\beta} & \text{sinon } \beta \neq 0 \end{cases} \quad (4.2)$$

Le degré $F_{(\alpha,\beta)}(p,q)$ pour « loin » (*Far*) est une probabilité qui mesure de combien deux points p et q sont éloignés l'un par rapport à l'autre. À partir de l'équation 4.2, si $F_{(\alpha,\beta)}(p,q) = 0$ alors la distance entre p et q est qualifiée de « proche » et si $F_{(\alpha,\beta)}(p,q) = 1$ alors la distance entre p et q est qualifiée de « loin ».

Dans notre cas, p et q sont des points qui représentent des centres de rectangles représentant des régions où p est le point cible et q est le point référent. Les paramètres α et β sont les seuils que nous fixons pour l'évaluation d'une distance (selon le besoin), tel que :

$$\begin{cases} \alpha \leftarrow r_1 \\ \alpha + \frac{\beta}{2} \leftarrow r_2 \\ \alpha + \beta \leftarrow r_3 \end{cases}$$

À partir de la figure 4.18, il est facile de déduire que $F_{(\alpha,\beta)}(p,q) = 1 - N_{(\alpha,\beta)}(p,q)$. L'évaluation de la distance se fait comme suit :

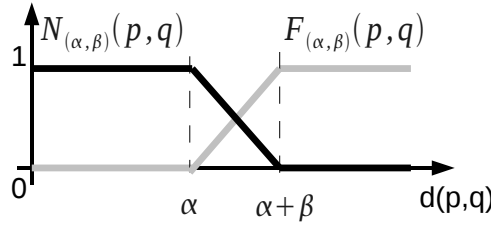


FIGURE 4.18 – Évaluation de la distance

$$\left\{ \begin{array}{l} d \in [0, \alpha] \text{ alors la distance est proche} \\ d \in]\alpha, \alpha + \frac{\beta}{2}] \text{ alors la distance est assez proche} \\ d \in]\alpha + \frac{\beta}{2}, \alpha + \beta] \text{ alors la distance est assez loin} \\ d \in]\alpha + \beta, +\infty[\text{ alors la distance est loin} \end{array} \right.$$

Dans ce travail, la représentation floue concerne seulement les relations de distance. En effet, nous utilisons les variables linguistiques pour évaluer une distance (proche, assez proche, assez loin et loin). Notre objectif est de donner une représentation numérique pour chaque variable linguistique et inversement. *SpaceOntology* permet cette interprétation en définissant un ensemble de règles basées sur les fonctions $N_{\alpha,\beta}$ et $F_{\alpha,\beta}$ [Schockaert, 2008]. La figure 4.18 illustre l'interprétation de la distance d'une définition numérique à une définition symbolique et inversement. Les paramètres α et β permettent d'évaluer une distance numérique. Ces paramètres dépendent du niveau hiérarchique courant et/ou du type de l'action.

4.2.3 Raisonnement avec les relations topologiques

Dans notre travail, les relations topologiques sont définies par le formalisme ABLR. Ce formalisme offre un compromis entre nombre de relations, expressivité et flexibilité de présentation [Laborie *et al.*, 2006]. Le raisonnement topologique se base sur le principe de la composition.

Principe de la composition

La composition des relations définies dans le formalisme ABLR se base sur la composition des relations d'intervalles temporels [Allen, 1983]. Le mécanisme de la composition permet de déduire des nouvelles informations. Formellement, à partir de deux relations R_1 et R_2 il est possible d'obtenir une nouvelle relation $R_3 = R_1 \otimes R_2$. Soient trois rectangles $rect_1$, $rect_2$ et $rect_3$ représentant respectivement trois régions rg_1 , rg_2 et rg_3 . À partir de la figure 4.19, on a :

1. $R_1 = \langle r_{x_1}, r_{y_1} \rangle$ est la relation topologique entre $rect_1$ et $rect_2$,
2. $R_2 = \langle r_{x_2}, r_{y_2} \rangle$ est la relation topologique entre $rect_2$ et $rect_3$.

L'idée est de déduire la relation R_3 qui relie $rect_1$ et $rect_3$ (figure. 4.19). Pour se faire, nous utilisons le mécanisme de composition défini comme suit :

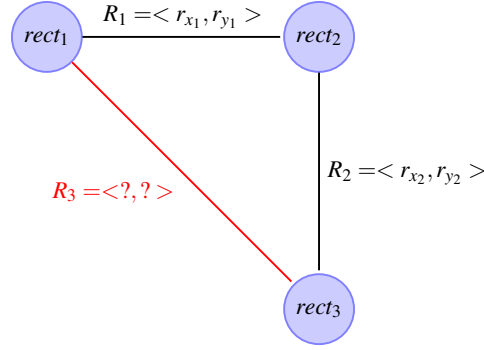


FIGURE 4.19 – Relations topologiques et composition.

$$\langle r_{x_3}, r_{y_3} \rangle = \langle r_{x_1}, r_{y_1} \rangle \otimes \langle r_{x_2}, r_{y_2} \rangle \quad (4.3)$$

$$= \langle r_{x_1} \otimes r_{x_2}, r_{y_1} \otimes r_{y_2} \rangle \quad (4.4)$$

Le calcul des résultats de la composition de $(r_{x_1} \otimes r_{x_2})$ et $(r_{y_1} \otimes r_{y_2})$ est décrit dans la section 4.3. Dans ce paragraphe nous détaillons ce mécanisme.

Composition dans *SpaceOntology*

Formellement, nous définissons dans *SpaceOntology* un ensemble de règles traduisant la table de composition décrite dans l'annexe D. Ces règles sont dites « règles de composition ». La figure 4.20 présente un squelette d'une règle de composition entre deux relations topologiques $r_1 = \langle r_{x_1}, r_{y_1} \rangle$ et $r_2 = \langle r_{x_2}, r_{y_2} \rangle$.

$$\begin{aligned} & \{ \text{isATopologicalRelation}(?r_1) \sqcap \text{isAnHorizontalRelation}(?r_1, "r_{x_1}") \\ & \quad \sqcap \text{isAVerticalRelation}(?r_1, "r_{y_1}") \} \\ & \quad \sqcap \\ & \{ \sqcap \text{isATopologicalRelation}(?r_2) \sqcap \text{isAnHorizontalRelation}(?r_2, "r_{x_2}") \\ & \quad \sqcap \text{isAVerticalRelation}(?r_2, "r_{y_2}") \} \\ & \quad \longrightarrow \\ & \{ \text{isATopologicalRelation}(?r_3) \sqcap \text{isAnHorizontalRelation}(?r_3, "r_{x_3}") \\ & \quad \sqcap \text{isAVerticalRelation}(?r_3, "r_{y_3}") \} \end{aligned}$$

FIGURE 4.20 – Squelette pour définir une règle de composition entre deux relations topologiques.

Décortiquons ce squelette. Nous avons deux relations topologiques :

- r_1 ($\text{isATopologicalRelation}(?r_1)$) définit par le couple $\langle r_{x_1}, r_{y_1} \rangle$:
 $\text{isAnHorizontalRelation}(?r_1, "r_{x_1}") \sqcap \text{isAVerticalRelation}(?r_1, "r_{y_1}")$,
- r_2 ($\text{isATopologicalRelation}(?r_2)$) définit par le couple $\langle r_{x_2}, r_{y_2} \rangle$:
 $\text{isAnHorizontalRelation}(?r_2, "r_{x_2}") \sqcap \text{isAVerticalRelation}(?r_2, "r_{y_2}")$.

Le résultat de la composition est une disjonction de relations topologiques :

$$r_3 = \langle r_{x_3}, r_{y_3} \rangle = \{ \langle r_{x_3}^1, r_{y_3}^1 \rangle, \langle r_{x_3}^2, r_{y_3}^2 \rangle, \dots, \langle r_{x_3}^k, r_{y_3}^k \rangle \}$$

, telle que :

- $r_{x_3} = (r_{x_1} \otimes r_{x_2}) = \{r_{x_3}^1, r_{x_3}^2, \dots, r_{x_3}^k\}$,
- et $r_{y_3} = (r_{y_1} \otimes r_{y_2}) = \{r_{y_3}^1, r_{y_3}^2, \dots, r_{y_3}^k\}$.

Toutes les relations inférées ne sont pas valides. Pour réduire le nombre de relations possibles et ne garder que les relations valides, il faut s'assurer que la matrice de relations déduites est valide (section 4.3, définition 15). Toutes les règles de compositions sont consultables sur le lien suivant : <https://belouaer.users.greyc.fr/>. Le mécanisme de composition est décrit dans la section suivante.

4.3 Mécanisme de composition

Le mécanisme de composition est au cœur du raisonnement topologique. En effet, l'application des règles de composition est basée sur ce mécanisme. Celui-ci se fait en plusieurs étapes.

4.3.1 Représentation des relations topologiques

La description d'une scène utilisant les relations du formalisme ABLR peut être représentée à l'aide d'un « graphe de relations » [Van Beek, 1992]. La définition 14 décrit formellement un graphe de relations.

Définition 14 Soient $\mathcal{Rect}$ un ensemble de rectangles représentant des régions et $\mathcal{R}$ un ensemble de relations entre les éléments de $\mathcal{Rect}$. Une spécification d'une scène $s = \langle \mathcal{Rect}, \mathcal{R} \rangle$ peut être représentée par un graphe orienté étiqueté $g_s = \langle N, E, \lambda \rangle$ nommé graphe de relations, tel que :

- les éléments de $\mathcal{Rect}$ correspondent aux nœuds N du graphe g_s ,
- les éléments de $\mathcal{R}$ correspondent aux arcs E du graphe g_s ,
- une fonction λ qui à chaque arc associe un ensemble de relations topologiques.

Considérons un ensemble de rectangles A, B, C et D (figure 4.21(a)). Le rectangle B chevauche l'extrémité gauche du rectangle D (O_L) et contenu verticalement dans cette zone (I_y). Ainsi, la relation entre B et D est la relation $\langle O_L, I_y \rangle$. La figure 4.21(b) présente un graphe de relations pour la scène décrite dans la figure 4.21(a).

Pour effectuer le raisonnement, il faut définir une « matrice de relations » à partir du graphe de relations. Notons par $M_{i,j}$ cette matrice. Celle-ci est de taille $n \times n$ avec n correspondant au nombre de nœuds de graphe de relations. Chaque indice de ligne ($i \in [0, n[$) ou colonne ($j \in [0, n[$) correspond à un nœud du graphe tandis que chaque case pointée par ces indices correspond à l'étiquette d'un arc du graphe. Par exemple, la table 4.2 correspond au graphe de relations décrits dans la figure 4.21(b).

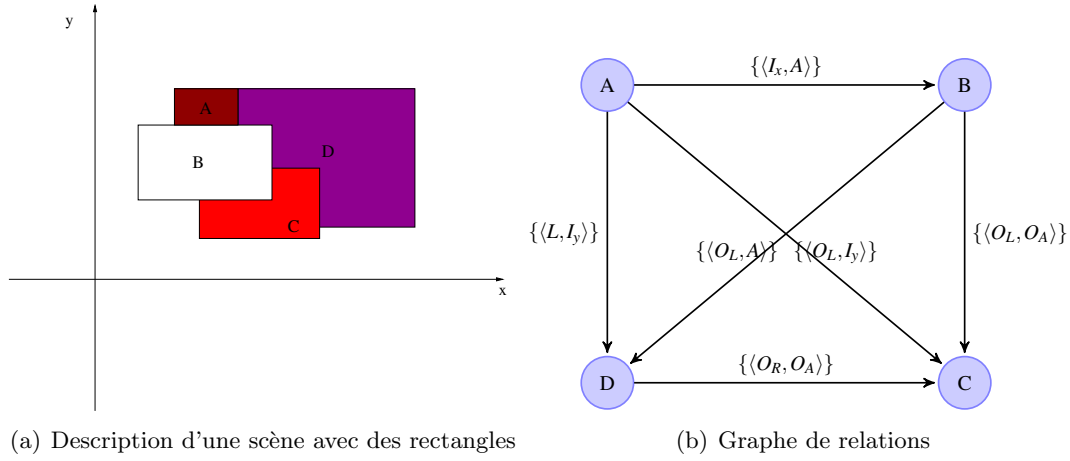


FIGURE 4.21 – Description d'une scène définissant un ensemble d'entités spatiales représentées par des rectangles et son graphe de relations correspondant.

$M_{i,j}$	A	B	C	D
A	$\{e\}$	$\{< I_x, A >\}$	$\{< O_L, A >\}$	$\{< L, I_y >\}$
B	$\{< C_x, B >\}$	$\{e\}$	$\{< O_L, O_A >\}$	$\{< O_L, I_y >\}$
C	$\{< O_R, B >\}$	$\{< O_R, O_B >\}$	$\{e\}$	$\{< O_L, O_B >\}$
D	$\{< R, C_y >\}$	$\{< O_R, C_y >\}$	$\{< O_R, O_A >\}$	$\{e\}$

TABLE 4.2 – Matrice de relations associée au graphe de relations présenté dans la figure 4.21(b).

Dans cette table, lorsque nous voulons définir une relation entre le rectangle lui même, nous notons cette relation par e (« equals » dans la relation d'Allen). Nous considérons les relations inverses (propriété 11). Par exemple, pour le rectangle A et le rectangle C on a la relation $\{< O_L, A >\}$ et son inverse est la relation $\{< O_R, B >\}$ (tableau 4.2). *SpaceOntology* définit un ensemble de règles permettant de traduire les relations inverses.

Propriété 11 *L'ensemble des relations de $M_{i,j}$ est équivalent à l'inverse de l'ensemble de relations de $M_{i,j}$, c'est-à-dire $M_{i,j} = M_{i,j}^{-1}$.*

4.3.2 Décomposition des relations topologiques

La composition pour les relations topologiques est facilitée par la décomposition du problème en considérant les relations ABLR sur chaque axe séparément. Ceci réduit le raisonnement à une dimension comme dans le cadre temporel (figure 4.22).

Soient trois rectangles A, B et C situés dans l'espace comme décrit dans la figure 4.22(a). La figure 4.22(b) représente les projections des rectangles A, B et C sur un seul axe :

1. l'intervalle A chevauche l'intervalle B : (A, O_R, B) ,
2. l'intervalle C suit immédiatement l'intervalle B : (C, R, B) ,

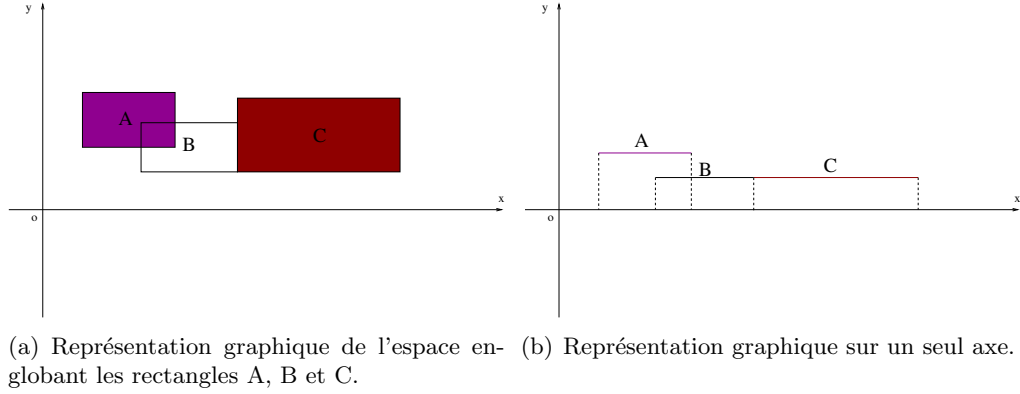


FIGURE 4.22 – Représentation graphique d'un espace de deux manières : sur deux axes et sur un seul axe.

3. L'intervalle A débute et se termine avant l'intervalle C : (A, L, C).

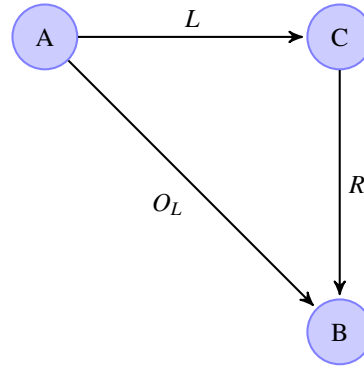


FIGURE 4.23 – Graphe de relations sur un seul axe.

La figure 4.23 présente un graphe de relation d'une seule dimension décrivant la figure 4.22(b). Le tableau 4.3 présente une matrice de relations qui correspond au graphe de relations défini par la figure 4.23.

$C_{i,j}$	A	B	C
A	$\{e\}$	$\{O_L\}$	$\{L\}$
B	$\{O_R\}$	$\{e\}$	$\{L\}$
C	$\{R\}$	$\{R\}$	$\{e\}$

TABLE 4.3 – Matrice de relations pour une seule dimension.

La matrice $C_{i,i}$ exprime les relations topologiques sur un seul axe. Elle est de taille $n \times n$ telle que n correspond au nombre du nœuds du graphe de relations. Chaque indice de ligne ($i \in [0, n[$) ou colonne ($j \in [0, n[$) correspond à un nœud du graphe tandis que chaque case pointée par

ces indices correspond à l'étiquette d'un arc du graphe. La matrice, décrite par la table 4.3, définit aussi les relations inverses. Par exemple, les projections des rectangles A et C sur l'axe (ox) sont liées par la relation $\{L\}$ et son inverse est la relation $\{R\}$. La matrice $C_{i,i}$ vérifie la propriété 11. Dans *SpaceOntology*, nous avons également des règles permettant de déduire les relations inverses. Par exemple, les règles $\mathcal{I}_{\mathcal{R}}.1$ et $\mathcal{I}_{\mathcal{R}}.2$ permettent de déduire que la relation $\langle L, O_B \rangle$ est l'inverse de la relation $\langle R, O_A \rangle$.

$$\{\text{inverseAxe}(\text{HorizontalRelations}.R)\} \longrightarrow \{\text{inverseAxe}(\text{HorizontalRelations}.L)\} \quad (\mathcal{I}_{\mathcal{R}}.1)$$

$$\{\text{inverseAxe}(\text{VerticalRelations}.O_A)\} \longrightarrow \{\text{inverseAxe}(\text{VerticalRelations}.O_B)\} \quad (\mathcal{I}_{\mathcal{R}}.2)$$

4.3.3 Calcul de la composition

Formellement pour appliquer la composition, nous procédons comme suit : tout d'abord, nous formalisons les informations nécessaires et utiles pour cette composition sous forme d'un graphe. C'est-à-dire, nous formalisons les relations topologiques que nous connaissons et qui sont nécessaires pour cette composition sous forme d'un graphe de relations, tel que :

- les nœuds de ce graphe correspondent à des rectangles représentant des régions ;
- chaque arc de ce graphe est étiqueté par un lien topologique donné par les prédicats `isAnHorizontalRelation` et `isAVerticalRelation`.

Puis, nous appliquons les règles de composition pour le calcul de la composition. Ensuite, nous traduisons ce graphe en une matrice de relations. Enfin, nous vérifions la consistance de cette matrice.

Pour réduire la complexité du raisonnement, nous appliquons le mécanisme de composition sur un seul axe. Tout d'abord, nous considérons les relations topologiques horizontales ensuite les relations topologiques verticales. Le raisonnement est le même sur les deux axes. Dans ce qui suit, nous présentons seulement le raisonnement sur l'axe horizontal.

$$\begin{aligned} \{\text{isAnHorizontalRelation}(?r_1, "r_{x_1}") \sqcap \text{isAnHorizontalRelation}(?r_2, "r_{x_2}")\} \\ \longrightarrow \\ \{\text{isAnHorizontalRelation}(?r_3, "r_{x_3}")\} \end{aligned}$$

FIGURE 4.24 – Squelette pour définir une règle de composition entre deux relations horizontales.

Tout d'abord, le graphe de relations global est réduit pour ne présenter que les relations topologiques horizontales entre les nœuds. Ensuite, nous appliquons des règles de composition sur une seule dimension (figure 4.24). Ces règles traduisent la table de composition sur l'axe

horizontal décrite dans l'annexe D. En s'appuyant sur cette table, nous remarquons que la composition sur ces relations retourne dans certains cas une relation disjonctive. Par exemple, nous considérons deux relations $r_1 = \{C_x\}$ et $r_2 = \{L\}$. Nous voulons déduire la relation r_3 , telle que $r_3 = r_1 \otimes r_2$. À partir de la table de composition, décrite dans l'annexe D, nous remarquons que la relation r_3 est disjonctive puisque r_3 peut être C_x , ou L ou O_L ou bien I_x ($r_3 = \{C_x, L, O_L, I_x\}$). Dans le cas où r_1 et r_2 sont elles-mêmes disjonctives. Par exemple, $r_1 = \{L, I_x\}$ et $r_2 = \{I_x, L\}$. Déduire r_3 est simple :

1. décomposer r_1 et r_2 ont une disjonction de relations atomiques ;

$$r_1 = \{L\} \vee \{I_x\} \text{ et } r_2 = \{I_x\} \vee \{L\}$$

2. développer tous les termes en appliquant la distributivité de l'intersection ($\cap$) par rapport à l'union ($\cup$). On obtient ainsi des clauses qui sont des paires conjonctives ;

$$r_3 = (\{L\} \wedge \{I_x\}) \vee (\{L\} \wedge \{L\}) \vee (\{I_x\} \wedge \{I_x\}) \vee (\{I_x\} \wedge \{L\})$$

3. calculer la composition en se basant sur la table de composition sur l'axe horizontal décrite dans l'annexe D ;

$$r_3 = \{I_x, O_L, L\} \vee \{L\} \vee \{I_x\} \vee \{L\}$$

4. les conjonctions ayant disparues, composer les clauses disjonctives en une seule relation composée ; $r_3 = \{I_x, O_L, L\}$.

Nous complétons le graphe de relations que nous considérons en définissant les arcs non étiquetés par les résultats de la composition. La consistance d'un graphe de relations est une étape très importante de la composition. Nous détaillons cette étape dans le paragraphe suivant.

4.3.4 Consistance de la composition

Pour qu'un graphe de relations soit consistant il faut que la matrice de relations qui lui est associée soit consistante (définition 15).

Définition 15 Une matrice de relations est dite consistante si et seulement si un scénario correspondant à cette matrice peut être produit. Dans le cas contraire, on dira que la matrice de relations est inconsistante et que par conséquent aucun scénario ne pourra satisfaire l'ensemble des relations considérées.

$C_{i,j}$	A	B	C
A	$\{e\}$	$\{O_L\}$	$\{L\}$
B	$\{O_R\}$	$\{e\}$	$\{L\}$
C	$\{R\}$	$\{R\}$	$\{e\}$

 TABLE 4.4 – Matrice de relations M_1 .

$C_{i,j}$	A	B	C
A	$\{e\}$	$\{O_L\}$	$\{R\}$
B	$\{O_R\}$	$\{e\}$	$\{L\}$
C	$\{L\}$	$\{R\}$	$\{e\}$

 TABLE 4.5 – Matrice de relations M_2 .

Considérons l'exemple décrit dans la figure 4.22(b). Soient deux matrices de relations M_1 et M_2 décrites respectivement par les tables table 4.4 et table 4.5. La matrice de relations M_1 est consistante car le scénario décrit par la figure 4.22(b) est satisfait cette matrice. La matrice de relations M_2 est inconsistante car le scénario décrit par la figure 4.22(b) n'est pas satisfait par cette matrice.

Dans [Allen, 1983], un algorithme polynomial permettant de tester la consistance d'une matrice de relations est défini. Malheureusement, cet algorithme n'est complet que lorsque toutes les relations de la matrice sont atomiques. En effet, les résultats présentés dans [Vilain et Kautz, 1986] ont montré que cet algorithme pouvait ne pas détecter certaines incohérences et que le problème d'existence d'une solution avec l'algèbre d'intervalles de Allen est NP-complet. Par conséquent, pour trouver une solution, c'est-à-dire un ensemble de relations topologiques vérifiant la consistance de la matrice de relations, des méthodes de recherche exhaustives sont utilisées [Vilain et Kautz, 1986].

Algorithme 1 : testerConsistance

Données : Une matrice de relations $C_{i,j}$ avec $\leq i < n$ et $\leq j < n$

Résultat : Un booléen indiquant si la matrice de relations est consistante ou non

début

```

 $\mathcal{R}_A$  : ensemble de relations du formalisme ABLR sur un axe ;
définirConsistance( $C_{i,j}$ ) ;
si  $C_{i,j}$  contient une relation vidée alors
    | retourner faux ;
sinon
    Choisir une relation  $C_{i,j}$  non-sélectionnée précédemment et décomposer  $C_{i,j}$  en
     $r_1, \dots, r_k$  telle que  $r_1 \in \mathcal{R}_A$  ;
    si  $C_{i,j}$  peut être décomposée alors
        | retourner vrai ;
    fin
    pour chaque  $r_l (1 \leq l \leq k)$  faire
         $C_{i,j} \leftarrow r_l$  ;
        si testerConsistance( $C_{i,j}$ ) alors
            | retourner vrai ;
        fin
    fin
fin
fin
```

L'algorithme 1 est une méthode de recherche exhaustive déjà décrit dans [Nebel, 1997]. Cet algorithme permet d'assurer la consistance de la matrice de relations $C_{i,j}$. Cet algorithme utilise la fonction **définirConsistance** en tant que méthode de propagation de contraintes, pour réduire l'espace de recherche. Cette fonction est décrite par l'algorithme 2.

L'algorithme 2 est défini dans [Allen, 1983]. Nous l'utilisons pour les relations topologiques du formalisme ABLR (pour chaque axe). Le fonctionnement de cet algorithme est comme suit : à partir de la table de composition, on calcule toutes les relations induites d'une matrice de rela-

Algorithme 2 : définirConsistance

Données : Une matrice de relations $C_{i,j}$ avec $i < n$ et $j < n$
Résultat : Une matrice de relations consistante ou une inconsistance
début

```

  Q est une pile ; pour  $i, j \in [0, n-1], i \neq j$  faire
  |    $Q \leftarrow \{i, j\}$  ;
  fin
  tant que  $Q \neq \emptyset$  faire
  |    $\{i, j\} \leftarrow \text{Dépiler}(Q)$  ; pour  $k \in [0, n-1], k \neq i, j$  faire
  |   |    $C'_{i,k} \leftarrow (C_{i,j} \otimes C_{j,k}) \cap C_{i,k}$  ;
  |   |    $C'_{k,j} \leftarrow (C_{k,i} \otimes C_{i,j}) \cap C_{k,j}$  ;
  |   fin
  |   si  $C'_{i,k}$  ou  $C'_{k,j} = \perp$  alors
  |   |   STOP ;
  |   fin
  |   si  $C'_{i,k} \neq C_{i,k}$  alors
  |   |   Empiler  $\{i, k\}$  dans  $Q$  ;
  |   |    $C_{i,k} \leftarrow C'_{i,k}$  ;
  |   fin
  |   si  $C'_{k,j} \neq C_{k,j}$  alors
  |   |   Empiler  $\{k, j\}$  dans  $Q$  ;
  |   |    $C_{k,j} \leftarrow C'_{k,j}$  ;
  |   fin
  fin
fin

```

tions et on les confronte à ses relations existantes à l'aide d'une intersection. Si une intersection est vide, on détecte une incohérence, ce qui signifie que la matrice de relations est inconsistante. Sinon on remplace les relations existantes par les valeurs des intersections. L'algorithme propage ces nouvelles relations par application d'une série de transitivité et ainsi de suite jusqu'à ce que plus rien ne change. Cet algorithme permet donc de préciser de plus en plus la matrice de relations en éliminant des disjonctions. Il s'arrête s'il n'y a plus de relations apportant une information nouvelle ou bien il y a une relation composée qui devient vide, auquel cas on a détecté l'incohérence de la matrice de relations.

Nous avons présenté le mécanisme de composition pour les relations topologiques. Ce mécanisme permet de générer une nouvelle information topologique que nous ajoutons d'une manière explicite dans la base de connaissance définie par *SpaceOntology*. Lors de cet ajout, une question importante se pose : « est-ce que après cet ajout la base de connaissance reste cohérente ? ».

4.3.5 Maintenir la cohérence pour les relations topologiques

Nous considérons que pour les nouvelles relations topologiques, nous avons deux sources. Le raisonnement permet de générer des relations topologiques assurant la consistance locale.

L'observation de l'environnement par un agent permet d'ajouter de nouvelles relations. Nous considérons que ces observations sont valides. Pour gérer les conflits, nous considérons que l'observation de l'environnement priment sur le raisonnement. Dans ce qui suit, nous expliquons comment maintenir la cohérence pour les relations topologiques à chaque ajout d'une nouvelle relation.

Tout d'abord, nous formalisons toute la connaissance topologique sous la forme d'un « réseau de contraintes binaires » (définition 16).

Définition 16 *Un réseau de contraintes est un triplet (X, D, C) , tel que :*

- $X = \{x_1, \dots, x_n\}$ l'ensemble des variables,
- $D = \{d_1, \dots, d_n\}$ l'ensemble des domaines. Chaque domaine d_i porte sur la variable x_i ,
- $C = \{c_1, \dots, c_n\}$ l'ensemble des contraintes.

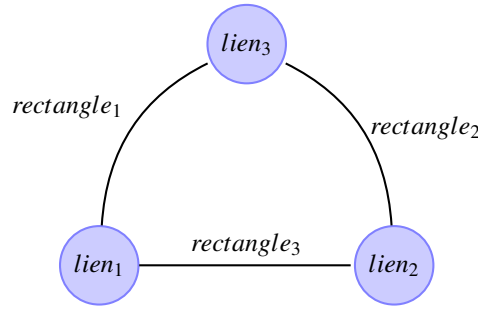


FIGURE 4.25 – Réseau de contraintes modélisant la connaissance topologique.

Intuitivement, le réseau de contraintes représente la description topologique fournie par *SpaceOntology*. Chaque nœud dans ce réseau représente un lien topologique (figure 4.25). Chaque arc dans ce réseau est étiqueté par un rectangle définissant une région. Un arc connectant deux sommets se traduit par le fait que le rectangle représenté par cet arc est concerné par deux liens topologiques (figure 4.25). Dans le cas où un rectangle n'est concerné que par un seul lien topologique, nous ajoutons un nœud au réseau de contraintes. Ce nœud représente le lien topologique ayant la seule valeur « égale ».

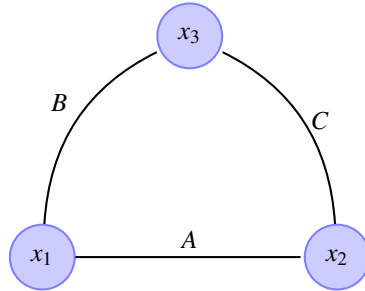


FIGURE 4.26 – Réseau de contraintes modélisant les liens topologiques.

La figure 4.26 présente un réseau de contraintes modélisant les liens topologiques décrits dans la figure 4.22(a). Les nœuds x_1 , x_2 et x_3 représentent les trois liens topologiques existant entre les trois rectangles A, B et C. Ces nœuds ont les domaines respectives : $d_1 = \langle O_A, O_L \rangle$, $d_2 = \langle O_A, L \rangle$ et $d_3 = \langle O_B, R \rangle$.

Définition 17 Une contrainte c est arc-consistante si et seulement si $\forall x_i \in X(c), d_i \neq \emptyset$ et $\forall a \in d_i$ a est consistante avec c .

Définition 18 Un réseau de contraintes $\mathcal{N} = (X, D, C)$ est arc-consistant si et seulement si $\forall x_i \in X, d_i \neq \emptyset$ et $\forall c, \forall x_i \in X(c), \forall a \in d_i, (x_i, a)$ est arc-consistant avec c .

L'idée est de s'assurer que le réseau de contraintes considéré reste cohérent après un ajout. Pour atteindre cette propriété, il est nécessaire de vérifier les couples de valeurs possibles. En d'autres termes, il faut enlever toutes les relations spatiales qui n'assurent pas la cohérence de ce réseau (définition 17). Pour cela, il faut que le réseau vérifie la propriété de l'arc-consistance : un réseau de contraintes est arc-consistant si tous ses arcs sont arc-consistants (définition 18).

Algorithme 3 : reviserArc

Données : Une contrainte $c_{i,j}$
Résultat : $c_{i,j}$ une contrainte arc-consistante
début
 $supprimer = \text{faux}$;
 pour chaque valeur a dans d_i **faire**
 si $\forall b \in d_j, (a,b)$ n vérifie pas $c_{i,j}$ **alors**
 Supprimer a de d_i ;
 $supprimer = \text{vrai}$;
 fin
 fin
 retourner $supprimer$;
fin

La procédure **reviserArc** (voir l'algorithme 3), extraite de [Mackworth, 1977], permet d'atteindre pour un arc donné $c_{i,j}$ la propriété de consistance. Dans un réseau de contraintes, pour faire en sorte que chaque arc soit consistant une seule application de cette fonction n'est pas suffisante. En effet, la suppression d'une valeur d'un domaine pour une contrainte donnée peut entraîner la suppression d'autres valeurs dans d'autres domaines. D'où la nécessité de réviser à nouveau des arcs déjà visité. Il existe plusieurs algorithmes polynomiaux et efficaces pour rendre un réseau de contraintes binaire arc-consistant tels que les algorithmes MAC [Likitvivatanavong *et al.*, 2004], AC-3 [Bessière *et al.*, 2005], AC-4 [Mohr et Henderson, 1986], CAC [Régis, 2004], etc. Dans le chapitre 7, nous justifierons le choix d'utiliser l'algorithme AC-3.

De cette manière, nous pouvons conclure, qu'en appliquant le principe de l'arc-consistance au réseau de contraintes formalisant la connaissance topologique décrite dans la base de connaissance de *SpaceOntology*, celle-ci reste cohérente.

Conclusion

De ce chapitre, nous retenons la définition d'une ontologie spatiale dite *SpaceOntology*. Celle-ci permet de décrire l'espace en s'intéressant à deux représentations. La première consiste à décrire l'environnement à l'aide des caractéristiques géométriques, ceci en définissant des concepts et des liens permettant de décrire toute entité spatiale d'une manière géométrique et d'une manière hiérarchique. La deuxième consiste à représenter les relations topologiques et de distance (floue et numérique). Ensuite, nous avons décrit le raisonnement dans *SpaceOntology*. Celui-ci permet d'inférer une nouvelle information spatiale.

Le mécanisme d'inférence dans *SpaceOntology* se traduit par la définition d'un ensemble de règles d'inférences, telles que, les règles hiérarchiques, les règles de composition topologiques, les règles d'inversion, etc. Le raisonnement concerne la définition hiérarchique, la définition topologique et l'interprétation des relations de distances.

Le raisonnement concernant les relations topologiques se base sur le mécanisme de composition. Celui-ci se fait en plusieurs étapes. La plus importante est de s'assurer que la connaissance spatiale reste cohérente. Il faut savoir que l'inférence concerne aussi la structure de *SpaceOntology* pour vérifier si celle-ci est valide ou non. Cette inférence se fait au niveau de la TBox et de la ABox. Au niveau de la TBox, elle consiste à :

- détecter tous les concepts insatisfiables ;
- calculer la hiérarchie de subsomption des concepts ;
- trouver les concepts équivalents ;
- trouver les concepts disjoints.

Au niveau de la ABox, elle consiste à :

- vérifier la cohérence de la ABox en vérifiant qu les assertions restent cohérentes avec la TBox ;
- vérifier que chaque instance respecte la définition de son concept ;
- vérifier que chaque les rôles de l'instance sont correctement utilisés.

Cette inférence est faite à l'aide du raisonneur Pellet [Sirin *et al.*, 2007] qui en cas d'incohérence lors de l'instanciation de *SpaceOntology* propose des corrections possibles. Le chapitre suivant présente comment exploiter *SpaceOntology* dans un système de planification.

Chapitre 5

Modèle et langage pour la planification spatiale

Sommaire

5.1 Planification spatiale	118
5.1.1 Approche proposée : Architecture SPOON	119
5.1.2 Raisonnement dans SPOON	120
5.1.3 Aperçu du processus de planification de SPOON	122
5.2 Fonctionnement de SPOON	122
5.2.1 Information spatiale dans le processus de la planification	122
5.2.2 Opérateurs de planification spatiale	127
5.3 De PDDL à Spatial-PDDL	130
5.3.1 Définition des objets dans Spatial-PDDL	132
5.3.2 Définition d'un domaine de planification avec Spatial-PDDL	132
5.3.3 Définition d'un problème de planification avec Spatial-PDDL	137

À partir de l'étude bibliographique nous déduisons qu'il existe différents travaux, dans plusieurs domaines d'applications et notamment en planification, qui s'intéressent à la connaissance spatiale. Ce chapitre montre comment *SpaceOntology* peut être exploitée pour faire l'objet d'un système de médiation entre un agent humain et un agent robot afin de planifier la mission assignée à ce dernier tout en tenant compte de la composante spatiale. Pour cela, nous présentons nos deux contributions. La première consiste à définir un modèle de planification combinant deux modules de raisonnements (section 5.1 et section 5.2). La deuxième présente l'extension de PDDL avec des éléments de *SpaceOntology* pour qu'il prenne en compte la dimension spatiale (section 5.3).



5.1 Planification spatiale

Dans des applications telle que l'interaction homme-robot, le scénario typique inclue un humain chargeant un robot d'effectuer certaines actions sur certains objets dans l'environnement. Pour réaliser cette mission, l'humain et le robot établissent chacun une description de l'environnement qui doivent correspondre. Pour se faire, un système de médiation est nécessaire. En effet, la représentation de l'espace de l'humain et celle du robot sont différentes. La représentation spatiale du robot est quantitative et la représentation spatiale humaine, souvent qualitative, utilise des relations topologiques, par exemple, la rivière traverse le champ ; ou bien des expressions floues comme par exemple, le livre proche de la bouteille.

L'exemple 4 permet de montrer l'importance de l'information spatiale (et en particulier l'information spatiale qualitative) lors de la planification. Nous nous basons sur cet exemple pour illustrer notre étude.

Exemple 4 *Considérons l'environnement décrit dans l'exemple 3 (figures 4.1 et 4.2). Une équipe, formée d'un humain et d'un robot, se trouve dans le bureau $O_{11}^{B_1}$ au premier étage du bâtiment B_1 (figure 5.1). Notons par p_h la position de l'humain et par p_r celle du robot. L'humain demande au robot : « de prendre le livre qui se trouve dans le bureau proche de la machine à café ». La machine à café se trouve dans le deuxième étage du bâtiment B_1 .*

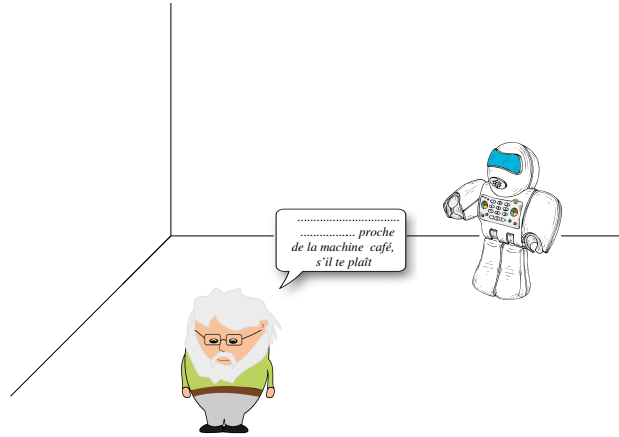


FIGURE 5.1 – Définition d'une mission pour le robot par l'humain.

L'objectif du robot est de réaliser sa mission. Nous supposons que le robot a des capacités de se déplacer, de reconnaître les objets, de communiquer et d'interpréter les ordres. Pour cette mission, le robot peut exécuter l'ensemble des actions décrites dans le tableau 5.1

Un planificateur classique de tâches ne permet pas au robot de réaliser la mission définie dans l'exemple 4. En effet, deux problèmes nécessitant la connaissance spatiale sont identifiés. Le premier concerne la définition des positions et/ou régions floues. Par exemple, l'action $go(p_r, \tilde{p})$, tel que $\tilde{p}$ est une position floue définie par l'expression « le bureau proche de la machine à

Opérateur	Paramètres	Signification
$go(p_1, p_2)$	p_1 et p_2 : deux positions/régions	se déplacer entre p_1 et p_2
$take(o)$	o : un objet	prendre un objet o
$give(o, ag)$	o : un objet et ag : un agent	donner un objet o à un agent ag

TABLE 5.1 – Actions possibles pour le robot dans ce scenario.

café », ne peut ni être exprimée par l’opérateur de planificateur classique, ni être sélectionnée par le planificateur classique. Le second problème concerne le déplacement entre les différentes zones d’actions (celles-ci peuvent être précises ou imprécises). Autrement dit, il est nécessaire de définir la trajectoire du robot durant sa mission.

Cette section présente un planificateur qui prend en compte la connaissance spatiale pour définir des plans solutions.

5.1.1 Approche proposée : Architecture SPOON

La résolution des problèmes de planification faisant intervenir la composante spatiale nécessite de combiner les aspects symboliques et les aspects spatiaux. Tout d’abord, nous citons quelques planificateurs permettant de traiter les aspects mentionnés ci-dessus. Ensuite, nous proposons notre approche.

Travaux dans le domaine

Les planificateurs SHAPER [Guere et Alami, 2001], ASYMOV [Gravot *et al.*, 2005] et celui présenté dans [Guitton, 2008] permettent de traiter les aspects symboliques et les aspects géométriques (ou aspects spatiaux).

Le planificateur SHAPER exploite la connaissance spatiale en apprenant la topologie d’un graphe d’accessibilité entre états. Ce planificateur utilise le langage de planification STRIPS augmenté par des faits numériques, pour représenter les aspects géométriques du problème. Le planificateur SHAPER considère seulement les représentations topologiques et numériques.

Le planificateur ASYMOV permet la production de plans pour les problèmes de manipulation dans lesquels l’exécution d’une action a une forte répercussion sur la représentation géométrique du problème. Par exemple, lorsqu’un robot doit transporter un objet, la forme de l’ensemble robot plus objet est différente de celle du robot à vide. Le planificateur ASYMOV s’intéresse uniquement aux aspects numériques.

Le planificateur présenté dans [Guitton, 2008] se compose de deux modules de raisonnement entrelacés : le module de raisonnement symbolique supporté par le planificateur de tâches et le module de raisonnement géométrique supporté par le planificateur de chemins. Cependant, les aspects géométriques concernent seulement la connaissance spatiale quantitative.

La limite de ces trois planificateurs est de ne pas effectuer un raisonnement spatial. C’est-à-dire, il n’y a pas de mécanismes d’inférence qui génèrent de nouvelles informations pour mieux

appréhender l'environnement et ainsi optimiser le calcul de la solution. En effet, le module géométrique, dans ces trois planificateurs, considère l'environnement comme une carte dans lequel il faut calculer un déplacement en évitant les obstacles et en respectant les contraintes cinématiques de l'agent effectuant un déplacement.

Notre approche

Dans cette thèse, notre objectif est de gérer les aspects symboliques et les aspects spatiaux (quantitatifs et qualitatifs). Pour cela, nous combinons la planification de tâches, la planification de chemins et la gestion de la connaissance spatiale (c'est-à-dire, la représentation et le raisonnement spatial). Nous proposons un planificateur [Belouaer *et al.*, 2011], appelé SPOON (*Spatial Planner based On Ontology*). Celui-ci se compose d'un planificateur de tâches, d'un planificateur de chemin et de *SpaceOntology* (figure 5.2). Il prend en entrée un problème de planification décrit par un domaine, un état initial et un état but. Ce problème tient compte de l'aspect symbolique et de l'aspect spatial. Il retourne un plan permettant de résoudre le problème d'entrée.

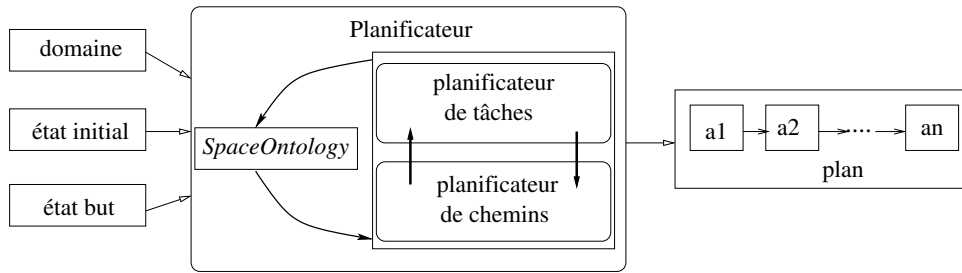


FIGURE 5.2 – Architecture du planificateur spatial.

SPOON s'inspire du planificateur proposé dans [Guitton, 2008]. En effet, notre approche se base sur le principe du raisonnement hybride en combinant un module de raisonnement symbolique et un module de raisonnement spatial. Cependant, nous ne définissons pas les différents modules de raisonnement de la même manière que [Guitton, 2008]. De plus, notre planificateur gère des informations spatiales quantitatives et qualitatives contrairement à [Guitton, 2008] qui ne gère que les informations quantitatives.

5.1.2 Raisonnement dans SPOON

Le planificateur SPOON combine deux modules de raisonnement (figure 5.3) : un module de raisonnement symbolique pour la planification de tâches et un module de raisonnement spatial pour la planification de chemins en utilisant *SpaceOntology*.

Pour, le planificateur de tâches nous avons testé deux planificateurs à savoir GraphPlan [Blum et Furst, 1997] et HTN [Erol, 1996].

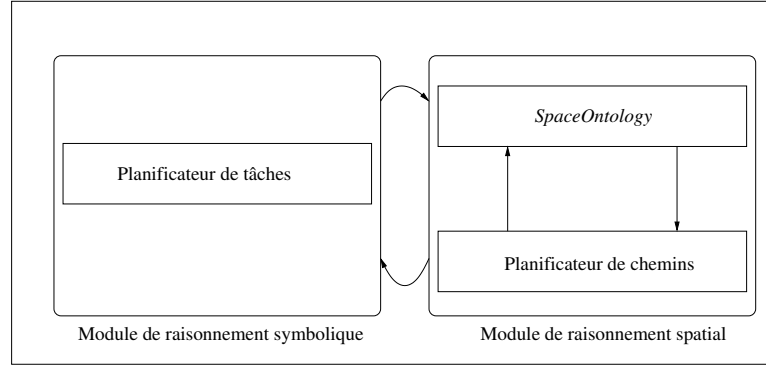


FIGURE 5.3 – Représentation graphique des différents modules de SPOON.

Le module de raisonnement spatial permet de définir les trajectoires entre les différentes zones d’actions nécessaires à la réalisation d’une mission, de gérer la connaissance spatiale et de vérifier la cohérence de celles-ci. Il prend en entrée :

1. la définition spatiale de l’environnement dans lequel la mission doit être effectuée,
2. les contraintes cinématiques de chaque agent intervenant dans la mission (définition 19).

Définition 19 Une contrainte cinématique (*kinematic constraint*) d’un agent Ag , noté $kc(Ag)$, est une contrainte qui réduit la liberté du mouvement de celui-ci.

Pour un agent Ag l’ensemble de contraintes cinématiques est noté $KC(Ag)$. Soit un agent Ag_1 ayant l’ensemble de contraintes cinématiques suivant : $KC(Ag_1) = \{step : false, width : 30cm\}$. La contrainte, $kc_1(Ag_1) = step : false$ permet d’éliminer tous les passages sous forme des escaliers. La contrainte, $kc_2(Ag_1) = width : 30cm$, permet d’éliminer tous les passages qui sont de largeur inférieurs à 30 cm.

Le planificateur de chemins a pour but de calculer les trajectoires nécessaires à la réalisation d’une action suite à une demande du module de raisonnement symbolique. Il trouve un chemin entre deux positions pouvant être précises ou imprécises.

L’exploitation de *SpaceOntology* dans SPOON permet d’améliorer l’efficacité de la planification. En effet, grâce à *SpaceOntology* il est possible d’éliminer toutes les entités spatiales qui ne sont pas en lien avec la mission courante. Par exemple, si seule la partie gauche de la pièce nous intéresse, le système de planification peut ignorer toutes les instances restantes, améliorant ainsi son efficacité de calcul en réduisant la taille de l’environnement à considérer. Le chapitre 7 présente les résultats expérimentaux qui illustrent nos propos. De plus, *SpaceOntology* permet de déduire de nouvelles informations spatiales à partir de celles explicitement connues. La déduction est nécessaire pour accomplir la planification (de tâches et/ou de chemins). En effet, la déduction permet de :

- déduire les propriétés d’éléments implicites qui peuvent être utilisées par le planificateur. Par exemple, déduire une relation d’adjacence entre deux pièces car l’agent a observé un passage entre celles-ci ;

- déduire l'existence d'objets nécessaires pour résoudre un objectif donné. Par exemple, l'existence de points de passage entre deux pièces, même si cela n'a pas été observé ;
- déduire des objets candidats potentiels pour satisfaire un objectif. Par exemple, sélectionner les points de passage permettant une navigation rapide, etc ;
- déduire de nouvelles informations pour compléter la description de l'environnement. Par exemple, utilisation de la composition des relations topologiques pour avoir une vision plus complète de l'environnement.

Le système de gestion de la connaissance spatiale exploite *SpaceOntology*. Pour des raisons de simplicité, ce système porte le même nom.

5.1.3 Aperçu du processus de planification de SPOON

Les trois composants de SPOON, à savoir, le planificateur de tâches, le planificateur de chemins et *SpaceOntology* sont organisés en deux modules (figure 5.3). En entrée, SPOON (figure 5.2) prend un problème de planification spatiale décrit dans un langage formel (section 5.3). En sortie, il génère un plan, c'est-à-dire une séquence d'actions.

Brièvement, le processus de planification se déroule comme suit. Pour appliquer une action, le planificateur de tâches vérifie que les préconditions sont satisfaites. Certaines actions, nécessitent une certaine définition de l'environnement ou un déplacement particulier dans celui-ci. Lors de la sélection de ces actions, le planificateur de tâches demande la satisfaction de ces préconditions au module de raisonnement spatial. Celui-ci détermine la connaissance spatiale nécessaire ou calcule les déplacements nécessaires et retourne la réponse. Si la réponse satisfait les préconditions de l'action sélectionnée alors celle-ci peut être appliquée. Dans le cas contraire, une autre action devra être trouvée pour réaliser la mission.

5.2 Fonctionnement de SPOON

Le processus de planification nécessite la définition de requêtes et la redéfinition des opérateurs de planification afin de prendre en compte les différents aspects de la connaissance spatiale.

5.2.1 Information spatiale dans le processus de la planification

Les planificateurs de tâches et de chemins manipulent l'information spatiale. Le planificateur de tâches a besoin de l'information spatiale pour vérifier si les conditions spatiales de l'exécution de la tâche sont valides. Le planificateur de chemins a besoin des informations spatiales pour définir des relations de proximité ou d'adjacence et parfois une représentation hiérarchique. C'est pour cette raison que nous définissons des requêtes envoyées à *SpaceOntology*. Ces requêtes peuvent être : des « requêtes de satisfaction de préconditions spatiales » (*Spatial Preconditions Satisfaction Query*), des « requêtes d'adjacence » (*Adjacency Query*), des « requêtes de proximité » (*Proximity Query*) ou bien des « requêtes pour définir une zone d'intérêt hiérarchique » (*Hierarchical Interest Zone Query*).

Une requête peut concerner la cohérence de la connaissance spatiale après l'ajout d'une nouvelle information. Ces requêtes, dites « requêtes de communication » (*Communication Query*), sont exprimées par les effets des opérateurs de planification (plus de détails dans la section 5.2.2). Après chaque exécution d'une action, un changement spatial peut se produire. Il doit être pris en compte par le module de raisonnement spatial et en particulier *SpaceOntology* pour vérifier la cohérence de la connaissance spatiale.

Requêtes de satisfaction de préconditions spatiales

Le planificateur de tâches formule des requêtes vers le module de raisonnement spatial afin de vérifier la validité des préconditions spatiales. Ce type de requêtes permet :

- d'exprimer une requête de recherche de chemins entre deux positions définies d'une manière précise ou imprécise ;
- de définir une position (une distance et/ou une orientation) nécessaire pour l'exécution d'une action.

Définition 20 Une requête de satisfaction de préconditions spatiales, notée SP_Q , est définie par un quadruplet : $\langle Action, Param(Action), Agent, KC(Agent), Spatial-Precond(Action) \rangle$, telle que :

- *Action* indique le nom de l'action concernée par la requête. Ceci permet de maintenir la cohérence lors du processus de planification entre les deux modules de raisonnement ;
- *Param(Action)* définit l'ensemble des paramètres de l'action concernés par la requête ;
- *Agent* définit l'agent qui va exécuter l'action *Action* ;
- *KC(Agent)* définit l'ensemble de contraintes cinématiques de l'agent ;
- *Spatial-Precond(Action)* définit l'ensemble des préconditions spatiales qui doivent être satisfaites durant l'exécution de l'action.

$$\langle go, \{\tilde{p}_1, \tilde{p}_2\}, robot, \{step : false, width : 30cm\}, path(\tilde{p}_1, \tilde{p}_2) \rangle \quad (\mathcal{S}_{P_Q}.1)$$

La requête $\mathcal{S}_{P_Q}.1$ exprime les préconditions spatiales à satisfaire pour l'exécution de l'action *go*. Cette requête signifie que l'agent robot doit se déplacer entre deux positions floues $\tilde{p}_1$ et $\tilde{p}_2$. Ce déplacement nécessite l'existence d'un chemin entre $\tilde{p}_1$ et $\tilde{p}_2$ ($path(\tilde{p}_1, \tilde{p}_2)$), tel que ce chemin respecte les contraintes cinématiques du robot ($KC(robot) = \{step : false, width : 30cm\}$) : le robot ne peut ni emprunter des escaliers, ni passer par des régions dont la largeur est inférieure à 30 centimètres.

Requête d'adjacence

Une relation d'adjacence est une relation topologique qui permet de définir les relations de voisinage, c'est-à-dire les points d'accès entre les régions. Un point d'accès peut être une porte,

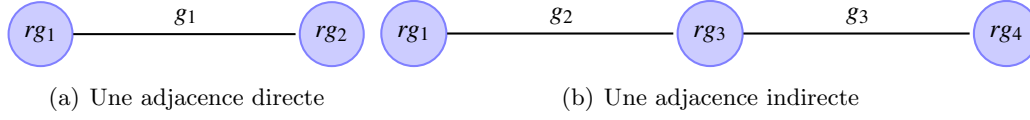


FIGURE 5.4 – Illustration des deux définitions 21 et 22.

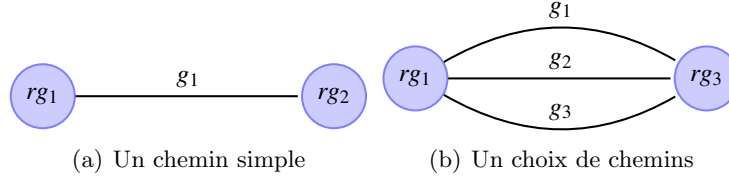


FIGURE 5.5 – Illustration des deux propriétés 12 et 13.

une intersection de couloirs, etc. Nous appelons ces points d'accès « passerelles » [Belouaer *et al.*, 2010b]. Une passerelle permet des transitions entre des régions adjacentes. Ces régions peuvent être du même niveau hiérarchique ou de niveaux hiérarchiques différents. Nous considérons l'adjacence directe et l'adjacence indirecte (figure 5.4).

Définition 21 *Deux régions sont adjacentes si il y a une adjacence directe ou indirecte entre ces deux régions.*

Définition 22 *L'adjacence directe se traduit par l'existence d'un chemin de longueur 1 entre deux régions, tel que le nombre de passerelles existant entre ces deux régions est 1. L'adjacence indirecte se traduit par l'existence d'un chemin de longueur l , tel que $l > 1$ et l correspond au nombre de passerelles séparant ces deux régions.*

Propriété 12 *Un chemin entre deux régions se traduit par l'existence d'une seule passerelle entre ces deux régions.*

Propriété 13 *Un choix multiple de chemins se traduit par l'existence de plusieurs passerelles entre ces deux régions.*

Remarque 1 *Dans SpaceOntology, une passerelle est une instance du concept *Regions*. En d'autres termes, une passerelle est définie comme une région.*

Une requête d'adjacence permet d'identifier pour une région donnée l'ensemble des passerelles, l'ensemble des régions voisines. Elle permet aussi d'identifier, pour un couple défini par une région et une passerelle (rg, p) donné, l'ensemble de régions voisines de rg par la passerelle p [Belouaer *et al.*, 2011].

Définition 23 Une requête d'adjacence, notée $\mathcal{A}_Q$, est définie par un quadruplet :

$$\langle label, rg, rg_1, type \rangle$$

, tel que :

- *label* indique la sémantique du résultat attendu. Nous définissons deux sens pour les requêtes d'adjacence : le premier, donné par le mot clé **neighbors**, indique que le résultat attendu est un ensemble de régions voisines et le deuxième, donné par le mot clé **gates**, indique que le résultat est un ensemble de passerelles ;
- le paramètre *rg* indique la région ou la passerelle de référence pour cette requête ;
- le paramètre *rg₁* est un paramètre optionnel qui indique une région ou une passerelle permettant de préciser la demande ;
- le paramètre *type* est un paramètre optionnel qui indique le type d'adjacence directe ou indirecte. Par défaut, cette requête retourne toutes régions adjacentes à *rg*, c'est-à-dire, l'adjacence directe et l'adjacence indirecte.

Pour répondre à une requête d'adjacence, *SpaceOntology* fait appel aux règles d'inférence. Par exemple, la requête $\mathcal{A}_Q.1$ prend en paramètre la région O_{11}^{B1} et cherche à définir toutes les régions voisines d'une manière directe ou indirecte à cette région (O_{11}^{B1} est notée O.B1.11). Cependant, la requête $\mathcal{A}_Q.2$ cherche à définir toutes les régions voisines d'une manière directe à la région O_{11}^{B1} .

$$\langle \text{neighbors}, O_B1_11 \rangle \quad (\mathcal{A}_Q.1)$$

$$\langle \text{neighbors}, O_B1_11, direct \rangle \quad (\mathcal{A}_Q.2)$$

La requête $\mathcal{A}_Q.3$ prend en paramètre la région O_{11}^{B1} et la passerelle *gate_5* et cherche à définir toutes les régions voisines d'une manière directe à la région O_{11}^{B1} via la passerelle *gate_5*.

$$\langle \text{neighbors}, O_B1_11, gate_5 \rangle \quad (\mathcal{A}_Q.3)$$

Requête pour définir une zone d'intérêt hiérarchique

Une requête concernant une zone d'intérêt hiérarchique permet de définir pour une région *rg*, l'ensemble des régions parents noté *F* et l'ensemble des régions descendants noté *C* (définitions 24 et 25). Cette requête se base sur la définition hiérarchique de l'espace. Pour répondre à cette requête, *SpaceOntology* fait appel aux règles d'inférence permettant de déduire les liens hiérarchique pour une région.

Définition 24 Soit une région rg du niveau hiérarchique l . Toute région rg_i de niveau hiérarchique l_i , tel que $l_i < l$ et rg est incluse dans rg_i alors rg_i est une région « parent » de la région rg .

Définition 25 Soit une région rg du niveau hiérarchique l . Toute région rg_i de niveau hiérarchique l_i , tel que $l_i > l$ et rg_i est incluse dans rg alors rg_i est une région « enfant » de la région rg .

Définition 26 Une requête pour définir une zone d'intérêt hiérarchique, notée Z_Q , est définie par un triplet : $\langle \text{label}, rg, \text{level} \rangle$, tel que :

- *label* indique la sémantique du résultat attendu : parent donné par le mot clé **fathers** ou bien enfant donné par le mot clé **children**;
- le paramètre rg indique la région de référence;
- le paramètre *level* est un paramètre optionnel qui indique le niveau hiérarchique qui nous intéresse.

$$\langle \text{fathers}, O_B1_11, 1 \rangle \quad (Z_Q.1)$$

Soit le bureau O_{11}^{B1} de niveau hiérarchique 3. La requête $Z_Q.1$ est utilisée pour définir les parents de la région O_{11}^{B1} au niveau hiérarchique 1.

Requête de proximité

Pour exprimer une proximité, nous nous basons sur la définition des relations de distance définies dans *SpaceOntology*. Ainsi, nous pouvons exprimer la proximité d'une manière numérique ou d'une manière floue. Une requête de proximité permet de définir une position floue en la délimitant. Aussi, elle permet de vérifier si deux régions respectent un lien de proximité donné [Belouaer et al., 2011].

Définition 27 Une requête de proximité, notée P_Q , est définie par un triplet : $\langle \text{relation}, rg, rg_1 \rangle$, tel que :

- *relation* indique la relation de proximité pouvant être numérique ou bien floue (proche, loin, etc.);
- le paramètre rg indique la région de référence;
- le paramètre rg_1 est un paramètre optionnel qui vérifie si la relation de proximité entre rg , et rg_1 est respectée ou non.

$$\langle \text{close}, O_B1_11 \rangle \quad (P_Q.1)$$

La requête $P_Q.1$ prend comme paramètre la relation proche et délimite une zone dans laquelle toute région appartenant à cette zone vérifie la relation proche du bureau O_{11}^{B1} .

Requête de communication

Une requête de communication concerne la validité de la cohérence de la connaissance spatiale après l'ajout ou la modification d'une information spatiale.

Définition 28 Une requête de communication, notée C_Q , est définie par le triplet :

$$< Action, param, Spatial-Effect(Action) >$$

tel que :

- *Action* indique le nom de l'action concernée par la requête ;
- *param* indique l'ensemble des paramètres de l'action ;
- *Spatial-Effect(Action)* définit l'ensemble des effets spatiaux qui expriment un changement de situation spatiale dans l'environnement.

$$< \text{move}, \{O, p_1, p_2\}, \text{adjacent}(p_1, p_2), \text{isnorth}(p_1, p_2) > \quad (C_Q.1)$$

La requête $C_Q.1$, définie par le planificateur de tâches pour le module de raisonnement spatial exprime les effets spatiaux déduits après l'exécution de l'action **move**. Cette requête demande de vérifier la validité des relations topologiques d'adjacence et d'orientation entre les deux pièces p_1 et p_2 .

5.2.2 Opérateurs de planification spatiale

Nous considérons que toute action exécutée nécessite une connaissance spatiale. En effet, pour se déplacer entre deux positions l'agent a besoin de connaître la configuration de l'espace, pour attraper un objet l'agent doit être à une certaine position (distance et/ou orientation) par rapport à l'objet, etc. Nous proposons une extension de l'opérateur de planification classique (définition 4) à la composante spatiale. Dans ce qui suit, nous présentons les différents éléments que nous avons introduits pour définir un opérateur de planification spatiale.

Intervalle de validité

Intuitivement, lors de l'exécution d'une action, il est nécessaire d'être à la bonne distance pour pouvoir l'exécuter. Par exemple, un agent Ag_1 donne un livre à un agent Ag_2 . L'agent Ag_1 qui exécute cette action ne peut pas être à 100000 cm de l'agent Ag_2 et il ne peut pas être à 0 cm de l'agent Ag_2 . Ainsi, nous considérons, pour chaque action, un intervalle de validité (*Execution Interval Validity*) devant être satisfait pour l'exécution de celle-ci.

Définition 29 Un intervalle de validité défini pour un opérateur O est l'intervalle $I_O = [d_O^{min}, d_O^{max}]$, tel que d_O^{min} est la distance minimale requise pour exécuter l'action définie par cet opérateur et d_O^{max} est la distance maximale à partir de laquelle on ne peut plus exécuter l'action.

L'intervalle de validité pour un opérateur dépend de plusieurs paramètres. Nous considérons les paramètres cités dans le chapitre précédent (paragraphe 4.2.2) : le contexte hiérarchique des déclarations et le type de l'action à exécuter.

Pour le contexte hiérarchique des déclarations ; Ceci concerne principalement les actions exécutant des instructions spatiales faisant intervenir la distance floue. Par exemple, l'action « prendre l'objet proche de la table » est une action exécutant une instruction spatiale « proche ». La réalisation de cette action nécessite que la distance séparant l'objet et le référent (la table) vérifie la propriété spatiale « proche ». Pour cela, il faut considérer le niveau hiérarchique de la région courante. Ainsi, selon la relation de distance évoquée, les bornes de l'intervalle de validité vont être définies en fonction de α et β de la région courante. Rappelons que pour chaque région nous avons défini α et β dépendant du niveau hiérarchique courant de cette région afin d'évaluer les relations de distance floues (page 103). Considérons la région courante le bureau O_{11}^{B1} . Ce bureau appartient au troisième niveau hiérarchique. Aussi, cette région a un α de 50 centimètres et un $\alpha + \beta$ de 100 centimètres. Pour une action exécutant une instruction spatiale de distance telle que « proche », l'intervalle de validité est définie comme suit : $[0, \alpha] = [0, 50]$.

Nous distinguons trois types d'actions :

- les actions impliquant une interaction avec un autre agent ;
- les actions n'impliquant pas une interaction avec un autre agent. Celles-ci regroupent les actions impliquant une interaction avec des objets. Par exemple attraper le livre, poser le livre sur la table ;
- Et les actions qui n'impliquent aucune interaction. Par exemple, aller d'une position initiale à une position finale.

Dans le premier cas, l'action peut engendrer une interaction informelle ou formelle. Selon, le type d'interaction, la définition de l'intervalle de validité pour l'opérateur définissant cette action dépend de la définition des zones de proximité de l'agent impliqué dans cette interaction. Dans le deuxième cas la définition de l'intervalle de validité pour l'opérateur dépend des contraintes cinématiques de l'agent qui exécute l'action définie par l'opérateur. Par exemple, la même action « poser un livre sur une table » ne se fait pas à la même distance pour un enfant et pour un adulte. En effet, l'enfant s'approche d'avantage que l'adulte pour effectuer cette action. En d'autres termes, cette action nécessite une distance séparant l'enfant de la table plus petite que la distance séparant l'adulte de la table. Pour cela, selon l'action chaque agent indique son propre intervalle de validité.

Préconditions et effets spatiaux

Une action est une instanciation d'un opérateur de planification. Pour être exécutée, une action sélectionnée doit satisfaire les préconditions symboliques et les préconditions spatiales.

Les préconditions spatiales sont exprimées par le planificateur de tâches. La satisfaction de ces préconditions est en charge du module de raisonnement spatial. Toute précondition spatiale est exprimée via une requête de satisfaction de préconditions spatiales. Par exemple, l'action

$go(i_p, t_p)$ a comme précondition spatiale l'existence d'un chemin entre i_p et t_p respectant les contraintes cinématiques de l'agent exécutant cette action. Cette précondition est formulée sous forme de requête pour le module de raisonnement spatial. Ce dernier calcule le chemin entre i_p et t_p et retourne la valeur vrai permettant ainsi la sélection de cette action (ainsi que le chemin à emprunter), la valeur « faux » qui implique la sélection d'une autre action.

L'application d'une action implique un changement de situation symbolique et/ou spatiale. Le changement spatial est exprimée par les effets spatiaux. Ces effets permettent de mettre à jour les contraintes cinématiques de l'agent, d'exprimer tout changement de situation spatiale et communiquer au module de raisonnement spatial ce changement. Chaque effet exprimant un changement de situation spatiale est formalisé sous forme de requête de communication (par exemple la requête $C_Q.1$).

Intégration de la connaissance floue

La connaissance spatiale floue concerne la définition des relations spatiales avec les termes proche, assez proche, assez loin et loin. Par exemple, l'expression « proche de la machine à café » décrit une région floue. En planification classique on ne peut pas exprimer de telles expressions. Par conséquent, nous considérons trois types de paramètres ; les paramètres symboliques (par exemple, bloc, porte, etc.), les paramètres spatiaux (par exemple, bureau, région, pays, etc.) et les paramètres spatiaux flous (définition 30).

Définition 30 *Un paramètre spatial flou est noté par $pa\tilde{r}am$ est une expression définie par un couple (rel, ref) , tel que :*

- *rel indique une relation spatiale de distance floue définie par le concept **FuzzyDistance** ;*
- *ref indique une référence pour évaluer la relation rel.*

Les paramètres spatiaux flous sont définis par des expressions utilisant les concepts dans *SpaceOntology*. Un paramètre spatial flou est exprimé par le module de raisonnement symbolique et il est défini par le module de raisonnement spatial. L'expression d'un tel paramètre se fait par une requête qui nécessite une réponse du module de raisonnement spatial pour l'exécution de l'action.

Opérateur de planification spatiale

Un opérateur de planification spatiale est une extension de l'opérateur de planification classique. Cette extension permet d'exprimer toute connaissance spatiale nécessaire lors de l'exécution de l'action qui correspond à une instanciation d'un opérateur. Dans les paragraphes précédents, nous avons évoqué les différents éléments que nous considérons pour la définition d'un opérateur de planification spatiale. La définition 31 résume d'une manière formelle l'extension d'un opérateur de planification classique.

Définition 31 Un opérateur de planification spatial o_s est défini par un n -uplet :

$$o_s = (nom(o_s), params(o_s), interval(o_s), precond(o_s), precond_sp(o_s), queries(o_s), effects(o_s), effects_sp(o_s))$$

- $nom(o_s)$: indique le nom de l'opérateur spatial ;
- $params(o_s)$: liste des paramètres de l'opérateur sous la forme de $\{param_1, \dots, param_n\}$. Cette liste se compose d'éléments symboliques et d'éléments spatiaux ;
- $interval(o_s)$: indique l'intervalle de validité pour exécuter l'opérateur ;
- $precond(o_s)$: représente les préconditions de l'opérateur, c'est-à-dire les propriétés du monde nécessaires à son exécution. Ces préconditions expriment des propriétés symboliques ;
- $precond_sp(o_s)$: représente les préconditions spatiales de l'opérateur, c'est-à-dire les propriétés spatiales du monde nécessaires à son exécution. Ces préconditions expriment des propriétés spatiales ;
- $queries(o_s)$: représente l'ensemble de requêtes ;
- $effects(o_s)$: définit l'ensemble des faits symboliques à ajouter et à supprimer de l'état après l'exécution de o_s ;
- $effects_sp(o_s)$: définit l'ensemble des faits spatiaux à ajouter et à supprimer de l'état après l'exécution de o_s .

Chaque élément de cet n -uplet peut être vide ou non. Dans le cas où les éléments suivants : $interval(o_s)$, $precond(o_s)$, $precond_sp(o_s)$, $queries(o_s)$ et $effects_sp(o_s)$ sont vides alors nous sommes dans le cas d'un opérateur de planification tel que défini dans la définition 4. Avec ces éléments, dans un opérateur de planification nous exprimons la connaissance spatiale qualitative et quantitative.

5.3 De PDDL à Spatial-PDDL

Le planificateur SPOON nécessite un ensemble de données symboliques et spatiales qui décrivent un problème de planification à résoudre. Ces données doivent être exprimées à l'aide d'un langage formel de planification. PDDL permet d'exprimer des informations spatiales simples par :

- des prédicats symboliques, par exemple, le prédicat $AT(Ag, p)$ indique qu'un agent Ag est à la position p ;
- des expressions numériques. En général, celles-ci sont une comparaison entre deux expressions numériques. Par exemple, l'expression $(= (< (surface ?t) (surface ?o)))$ est une comparaison entre deux expressions numériques $(surface ?t)$ et $(surface ?o)$ qui stipule que la surface de l'objet t doit être supérieure ou égale à la surface de l'objet o .

Ces éléments sont adaptés pour décrire le comportement d'un agent pour débiter et exécuter une action. En d'autres termes, ils sont adéquats pour décrire des informations spatiales simples et surtout connues.

Dans notre travail, la connaissance spatiale concerne principalement la description de l'environnement dans lequel on se place. La prise en compte d'une telle connaissance a une influence sur la qualité du plan solution (chapitre 7). Cependant, décrire un environnement en PDDL à l'aide des éléments classiques (prédicats symboliques, expressions numériques et axiomes) n'est pas suffisant. En effet, PDDL ne permet pas :

- d'exprimer une connaissance spatiale floue, par exemple, $\tilde{p}_1$: « proche de la chaise » ;
- de définir des règles d'inférences pour la connaissance spatiale. Les axiomes en PDDL permettent d'appliquer des règles de classification mais pas des règles d'inférence ;
- de vérifier la cohérence et la validité de la connaissance spatiale. Il n'y a aucun moyen permettant de faire ces vérifications ;
- d'exprimer les opérateurs spatiaux (définition 31).

De plus, la description d'un environnement avec PDDL peut être ardue pour n'importe quel utilisateur. Notre idée est de trouver un moyen de stocker la connaissance spatiale et d'utiliser les informations nécessaires à chaque mission et obtenues à partir de cette base de connaissance. Aussi, pour les relations qualitatives chaque utilisateur peut développer autant de prédicats symboliques que de relations. L'idée est de définir des prédicats spatiaux communs à toute une communauté d'utilisateurs. En résumé, notre objectif est de proposer un langage permettant de donner une sémantique spatial à un problème de planification, en proposant une syntaxe pour les expressions spatiales.

Nous proposons une extension de PDDL appelée Spatial-PDDL permettant de prendre en compte la connaissance spatiale. Cette section décrit l'ensemble des éléments définissant le langage Spatial-PDDL¹⁹. Le tableau 5.2 présente l'ensemble des éléments que nous utilisons dans ce travail. Dans Spatial-PDDL nous utilisons un ensemble de symbole défini dans PDDL [McDermott *et al.*, 1998] et d'autres définis dans Web-PDDL [Dou, 2008].

Langage	Symbole	Sémantique
PDDL	A	ensemble d'opérateurs
	D	le domaine
	O	ensemble d'objets
	T	ensemble de type
	Π	ensemble de prédicats (relations entre objets)
<i>SpatceOntology</i>	Φ	ensemble de concepts
	R	ensemble de relations entre les concepts
Spatial-PDDL	$\mathcal{A}$	ensemble d'opérateurs
	$\mathcal{D}$	le domaine
	$\mathcal{O}$	ensemble d'objets
	$\mathcal{T}$	ensemble de type
	Π	ensemble de prédicats (relations entre objets)

TABLE 5.2 – Notations dans chaque langage

19. Le détail de la syntaxe est en ligne à l'adresse : <https://belouaer.users.greyc.fr/>.

Pour assurer une comptabilité avec les langages PDDL et PDDL 2.1, nous garantissons que tout ce qui existe dans PDDL et PDDL 2.1 sont des domaines valides dans Spatial-PDDL. Ainsi, les plans valides dans PDDL et PDDL 2.1 sont valides dans Spatial-PDDL.

5.3.1 Définition des objets dans Spatial-PDDL

Dans Spatial-PDDL, nous notons $\mathcal{T}$ l'ensemble de tous les types symboliques et tous les types spatiaux. Cet ensemble est construit à partir :

1. des types symboliques (T). Par exemple, dans l'extrait donné par le code 5.1 les types **object**, **actor** sont des types symboliques,
2. et des concepts qui sont définis dans *SpaceOntology*. Par exemple, le concept **Regions** devient le type **region** dans Spatial-PDDL (code 5.1).

Pour définir un type dans Spatial-PDDL, nous notons comme suit : $t_1, t_2, t_3 - \mathcal{T}_0$ avec $\mathcal{T}_0 \in \mathcal{T}$. Pour la déclaration des types dans Spatial-PDDL :

- l'élément de déclarations `<< :types >>` (défini en PDDL) déclare les types symboliques ;
- l'élément de déclarations `<< :spatial-types >>` déclare les types spatiaux (code 5.1).

```
(define (domain bookmission)
  (:requirements :typing :fluents)
  (:types actor - object
           item - object
           human robot - actor)
  (:spatial-types
    region - http://www.owl-ontologies.com/SpaceOntology.owl:Regions)
  ...)
...)
```

CODE 5.1 - Un exemple de définition avec l'élément de déclaration `<< :spatial-types >>`.

Par conséquent, dans Spatial-PDDL, nous considérons deux types d'objets : les objets symboliques et les objets spatiaux. Nous notons par $\mathcal{O}$ l'ensemble de tous les objets possibles.

Définition 32 *Un objet o tel que $o \in \mathcal{O}$ est un couple $\langle t, nom \rangle$, indiquant respectivement son type et son nom.*

5.3.2 Définition d'un domaine de planification avec Spatial-PDDL

Définition formelle d'un domaine de planification

Formellement, un domaine de planification permet de décrire un ensemble d'actions possibles pour réaliser une mission. Chaque action est exprimée par un opérateur de planification spatial dans le domaine de planification. Aussi, il permet de décrire le monde d'un point de vue

symbolique et d'un point de vue spatial. La description du monde se fait par la définition des relations entre les différents objets (symboliques et spatiaux) de celui-ci.

Définition 33 *Un domaine de planification $\mathcal{D}$ est un couple :*

$$\mathcal{D} = (\mathcal{A}, \Pi)$$

1. $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ l'ensemble des opérateurs de planification spatiale, avec a_i est un opérateur de planification spatiale,
2. Π ensemble des relations entre les objets.

Définition d'un opérateur de planification spatiale avec Spatial-PDDL

Un opérateur de planification spatiale est exprimée par l'élément « **:spatial-action** ». Ensuite, les attributs symboliques sont exprimés par « **:precondition** » pour les préconditions symboliques et par « **:effects** » pour les effets symboliques. Pour chaque attribut spatial définissant cet opérateur (définition 31), nous avons défini un élément dans Spatial-PDDL pour l'exprimer :

- « **:interval-validity** » un élément qui déclare l'intervalle de validité ;
- « **:queries** » un élément qui déclare l'ensemble des requêtes nécessitant une réponse pour l'exécution de l'action ;
- « **:spatial-precondition** » un élément qui définit l'ensemble de préconditions spatiales ;
- « **:spatial-effect** » un élément qui décrit les effets spatiaux de l'opérateur une fois exécuté.

Le code 5.2 décrit l'opérateur **Enter** dans Spatial-PDDL. Cet opérateur permet d'exprimer une action comme par exemple « entrer dans le bureau proche de la machine à café ».

L'opérateur **Enter** fait intervenir un ensemble de paramètres définissant des objets :

1. un paramètre symbolique : **ag** - actor,
2. des paramètres spatiaux :
`rg_I ref - http://www.owl-ontologies.com/SpaceOntology.owl:Regions`
`et rel - http://www.owl-ontologies.com/SpaceOntology.owl:FuzzyDistance,`
3. un paramètre spatial flou :
`rg~T - http://www.owl-ontologies.com/SpaceOntology.owl:Regions.`

La variable « $?rg_{\sim T}$ » est une variable de requête (*Query Variable*) [Dou, 2008]. Celle-ci est utilisée dans une requête qui doit être définie pour l'exécution de l'action. La requête définie dans l'opérateur **Enter** (code 5.2), est formulée par le module de raisonnement symbolique vers le module de raisonnement spatial. Cette requête cherche à définir la région $rg_{\sim T}$. Pour cela, il faut sélectionner toute région rg tel que la relation spatiale entre rg et **ref** vérifient la relation **rel**. Pour que l'action entrer définie par l'opérateur **Enter** (code 5.2) soit exécutée, les préconditions

doivent être satisfaites. Intéressons nous à la précondition spatiale : $(\text{path } rg_I \tilde{rg_T})$. Celle-ci se traduit par une requête de satisfaction de préconditions spatiales :

$$\langle \text{Enter}, \{rg_I, \tilde{rg_T}\}, ag, KC(ag), path(rg_I, \tilde{rg_T}) \rangle \quad (\mathcal{S}_{P_Q}.2)$$

```
(:spatial-action Enter
:parameters ( ag - actor
              rg_I - http://www.owl-ontologies.com/SpaceOntology.owl:Regions
              rg_T- http://www.owl-ontologies.com/SpaceOntology.owl:Regions
              ref - http://www.owl-ontologies.com/SpaceOntology.owl:Regions
              rel - http://www.owl-ontologies.com/SpaceOntology.owl:FuzzyDistance)

:queries (and
(:query
  (freevars ?rg_T- http://www.owl-ontologies.com/SpaceOntology.owl:Regions
            ?hr - http://www.owl-ontologies.com/SpaceOntology.owl:HasRelation)
  (and (http://www.owl-ontologies.com/SpaceOntology.owl:isPartOf
        rg_T Zlevel(ref)-1(ref))
        http://www.owl-ontologies.com/SpaceOntology.owl:concernsRelation
        ?hr rel)
        (http://www.owl-ontologies.com/SpaceOntology.owl:hasTarget ?hr ?rg_T
          (http://www.owl-ontologies.com/SpaceOntology.owl:hasReferent ?hr ref))
  )
)
:precondition (and (AT ag rg_I) (not (AT ag rg_T)))
:spatial-precondition (and (path rg_I rg_T))
:effect (and (At ag ?rg_T) (not (AT ag rg_I)))
:spatial-effect (and(define direct_gates rg_I rg_T)(define rg_T))
)
```

CODE 5.2 - Déclaration d'un opérateur de planification spatial : **Enter**.

Les effets spatiaux forment la nouvelle connaissance spatiale sous forme de requête au module de raisonnement spatial. La requête $\mathcal{C}_Q.2$ exprime la connaissance spatiale décrite par les effets spatiaux. Elle communique au module de raisonnement spatial la définition de la région $\tilde{rg_T}$ et les points d'accès entre les deux régions rg_I et $\tilde{rg_T}$. Ceci grâce à la fonction **define** qui permet d'affecter à son paramètre la valeur qui le définit.

$$\langle \text{Enter}, \{rg_I, \tilde{rg_T}\}, \{define(\tilde{rg_T}), define(< gatesrg_Irg_T, direct >)\} \rangle \quad (\mathcal{C}_Q.2)$$

Définition des prédicats avec Spatial-PDDL

Un prédicat π ($\pi \in \Pi$) dans PDDL exprime une relation. Chaque relation r ($r \in R$) dans *SpaceOntology* exprime une information spatiale. Ainsi dans Spatial-PDDL, nous définissons l'ensemble de prédicats noté Π tel que $\Pi \leftarrow \Pi \cup R$. Dans Spatial-PDDL, pour la déclaration des prédicats spatiaux, nous avons défini un élément de déclarations « `:spatial-predicates` ». Ces prédicats sont les relations traduites à partir de *SpaceOntology*.

Les prédicats symboliques ayant une sémantique spatiale (par exemple, le prédicat `At` qui indique d'un objet dynamique est inclus dans une région dynamique ou statique) ne sont pas définis en tant que de prédicats spatiaux mais il sont définis en tant que prédicats symboliques. Dans des travaux futurs, ils seront intégrés à l'ensemble de prédicats spatiaux.

```
(define (domain bookmission)
  (:requirements :typing :fluents)
  (:types ...)
  (:spatial-types ...)
  (predicates ...)
  (:spatial-predicates
    http://www.owl-ontologies.com/SpaceOntology.owl:labelhasrelation
    hasrelation - http://www.owl-ontologies.com/SpaceOntology.owl:HasRelation
    http://www.w3.org/2001/XMLSchema#string

    http://www.owl-ontologies.com/SpaceOntology.owl:hasReferent
    hasrelation - http://www.owl-ontologies.com/SpaceOntology.owl:HasRelation
    region - http://www.owl-ontologies.com/SpaceOntology.owl:Regions
    ...
  )
  ...)
...)
```

CODE 5.3 - Déclaration avec « `:spatial-predicates` ».

Le code 5.3 est une traduction d'un extrait de *SpaceOntology* définissant le concept `HasRelation`. Ce code illustre comment définir des prédicats dans Spatial-PDDL. Un exemple de prédicat : `http://www.owl-ontologies.com/SpaceOntology.owl:hasReferent` qui relie tout objet de type `hasrelation - http://www.owl-ontologies.com/SpaceOntology.owl:HasRelation` a un objet de type `hasrelation - http://www.owl-ontologies.com/SpaceOntology.owl:Regions`.

Expression d'un domaine de planification avec Spatial-PDDL

Un domaine décrit les actions, les objets en spécifiant leur types et les relations entre ces objets. Tout élément spatial est défini à partir de la TBox de *SpaceOntology*. Tout élément dans une ontologie a toujours un préfixe. Par exemple, le concept *Regions* est défini par :

```
http://www.owl-ontologies.com/SpaceOntology.owl#Regions.
```

Dans [Dou, 2008], l'auteur définit un opérateur de déclaration « **:extends** ». Cet opérateur exprime qu'un domaine est étendu par une ou plusieurs ontologies définies par les URIs. Ainsi, à chaque ontologie est associé un préfixe pour indiquer l'espace de noms.

```
(define (domain bookmission)
  (:extends
    (uri "http://www.owl-ontologies.com/SpaceOntology.owl#" :prefix sponto)
    ...))
```

CODE 5.4 - Définition du domaine dans Spatial-PDDL.

Le code 5.4 illustre une définition d'un domaine de planification nommé **bookmission** avec *SpaceOntology*. Par conséquent , chaque élément dans le domaine sera préfixé par le préfixe **sponto**.

Le code 5.5 décrit un domaine de planification **bookmission** avec l'ontologie *SpaceOntology.1*. Dans la définition de ce domaine, nous avons des éléments symboliques tels que la définition des types (par exemple **actor - object**) , ou bien la définition des prédicats (par exemple **Empty a - actor**). Aussi, nous avons des éléments spatiaux tels que la définition des types spatiaux (par exemple **region - @sponto:Regions**) ou la définition des prédicats spatiaux (par exemple le prédicat **@sponto:hasTarget**). Aussi, dans cet extrait nous avons des actions spatiales telles que l'action **go** ayant comme précondition spatiale l'existence d'un chemin entre deux régions. Cette précondition utilise la fonction **path** qui cherche à trouver un chemin entre les deux régions données en paramètres. Cette fonction se traduit par la définition d'une requête de satisfaction de préconditions spatiales.

```

(define (domain bookmission)
  (:requirements :typing :fluents)
  (:extends
    (uri ".../SpaceOntology_1.owl" :prefix sponto))
  (:types actor - object
    item - object
    human robot - actor)
  (:spatial-types region - @sponto:Regions ....)
  (:predicates (Empty a - actor)(Holding a - actor i - item)
    (on i - item t - item)(At a - actor rg - @sponto:Regions) ...)
  (:spatial-predicates
    (@sponto:hasTarget hr - @sponto:HasRelation rg - @sponto:Regions) ...)
  (:functions (distance ip tp - position)(maxdistance)(mindistance)
    (maxhdistance)(minhdistance)(path irg trg - region))
  (:spatial-action go
    :parameters(r - robot irg - region trg - region )
    :precondition (and(At r irg)(not(At r trg)))
    :spatial-precondition (and(path irg trg))
    :effect(and(At r trg) (not(At r irg))))
  (:spatial-action pick-up ...)
  (:spatial-action put-down ...)
  (:spatial-action give ...)
  (:spatial-action take ...))

```

CODE 5.5 - Définition du domaine dans Spatial-PDDL avec des actions et des prédicats.

5.3.3 Définition d'un problème de planification avec Spatial-PDDL

Le tableau 5.3 résume les différentes extensions qui ont permis de définir Spatial-PDDL.

Notions	PDDL	Spatial-PDDL
Prédicats	Π	$\Pi \leftarrow \Pi \cup R$
Type	T	$\mathcal{T} \leftarrow T \cup \Phi$
Domaine	D	$\mathcal{D}$ utilisation des espaces de noms(<i>name space</i>) avec « @ns » utilisation de « extends »
Objets	O	$\mathcal{O}$ objets spatiaux/non-spatiaux
Opérateurs	A	$\mathcal{A}$ opérateurs spatiaux/non-spatiaux

TABLE 5.3 – Extension des éléments de PDDL avec les éléments de *SpaceOntology*.

Formellement, un problème de planification spatiale est décrit par un ensemble d'objets (spatiaux et non spatiaux), par un état initial et par un état but qu'on souhaite atteindre.

Définition 34 Un problème de planification spatiale $\mathcal{P}$ est un triplet $\mathcal{P} = \langle \mathcal{O}, \mathcal{S}_i, \mathcal{S}_g \rangle$, où ;

1. $\mathcal{O}$ un ensemble d'objets spatiaux et non spatiaux décrivant un domaine,
2. $\mathcal{S}_i$ un état initial qui peut être défini d'une manière symbolique et/ou spatiale,
3. $\mathcal{S}_g$ un état but qui peut être défini d'une manière symbolique et/ou spatiale.

```
(define (problem bookmissionpb_1)
  (:domain bookmission)

  (:objects Arnaud - human Bobo - robot
            book table -item
            initial_region referent_region - region
            relation - fuzzy_distance
            target- region ;; fuzzy region)

  (:init
    (on book table)
    (@sponto:labelregion initial_region "o_11.b1")
    (@sponto:labelregion referent_region "o_22.b1")
    (@sponto:isA relation "Close")
    (At Arnaud initial_region)
    (At Bobo initial_region)
    (At table target)
    (Empty Arnaud)
    (Empty Bobo)
    (not(Holding Arnaud book))
    (FormulateQueries( target relation referent_region))
  )
  (:goal
    (and ( not (on book table)) (Hodling Arnaud book)))
  )
)
```

CODE 5.6 - Définition d'un problème de planification dans Spatial-PDDL.

Le code 5.6 décrit un problème de planification avec Spatial-PDDL. Le problème défini utilise le domaine `bookmission` du code 5.5 (page 137). Les objets définis dans ce problème sont deux agents (`Arnaud - human` `Bobo - robot`) : un agent humain et un agent robot, des régions et une relation spatiale.

L'état initial est décrit d'une manière symbolique c'est-à-dire le livre est sur une table ((`on book table`)), l'humain n'a pas le livre (`not(Holding Arnaud book)`), etc. Aussi, il décrit la situation spatiale initiale. En effet, l'état initial identifie les régions connues par leurs labels.

Pour cela, nous utilisons le prédicat `@sponto:labelregion` défini dans *SpaceOntology* qui pour chaque région l'identifie avec son nom. Dans cet état nous spécifions également le besoin de requêtes pour définir la zone cible. Ceci est exprimé par la fonction `FormulateQueries` qui définit la région cible floue exprimée en fonction de la relation proche et la région de référence. L'état but représente l'état que nous voulons atteindre qui est : « Arnaud a le livre »

Conclusion

Ce chapitre présente la définition d'un langage de planification Spatial-PDDL. Ce langage est une extension du langage de planification PDDL intégrant des éléments spatiaux. Dans le chapitre 7, nous donnons les résultats expérimentaux d'une description d'un problème de planification à l'aide de Spatial-PDDL. Aussi, nous avons présenté un aperçu du planificateur que nous avons mis en place : le planificateur SPOON. Ce planificateur se compose d'un planificateur de tâches, d'un planificateur de chemins et de *SpaceOntology* permettant un processus de planification fondé sur un raisonnement symbolique pour la planification de tâches et un raisonnement spatial pour la planification de chemins. Ce processus se base sur une interaction entre les différents composants de SPOON. Cette interaction est formalisée sous la forme de requête/réponse. Nous avons présenté différents types de requêtes possibles au sein de SPOON. Dans le chapitre suivant, nous présentons en détails le processus de planification en spécifiant l'utilité des requêtes et comment les utiliser.

Chapitre 6

Processus de planification avec des informations spatiales

Sommaire

6.1	Exemple d'application	141
6.2	Processus de planification : principes	142
6.3	Description avec Spatial-PDDL	144
6.4	Planification de tâches	145
6.5	Planificateur spatial hiérarchique et floue	146
6.5.1	Génération du graphe de réseaux de passages	146
6.5.2	Définition du chemin vers une cible floue	154

Le chapitre précédent a présenté le système de planification SPOON, ainsi que le langage Spatial-PDDL. Ce langage décrit les données (symboliques et spatiales) exprimant un problème de planification spatiale. Ce chapitre détaille le processus de planification dans SPOON. Tout d'abord, nous introduisons un exemple que nous utilisons pour illustration (section 6.1). Ensuite, nous décrivons le principe du processus de planification (section 6.2). Nous donnons la définition d'un problème de planification avec le langage Spatial-PDDL (section 6.3). Deux contributions sont présentées dans ce chapitre : la première présente une modification dans les algorithmes de planification lors de l'instanciation d'un opérateur de planification spatiale (section 6.4) et la deuxième présente une méthode de recherche de chemins entre des positions floues (section 6.5).



6.1 Exemple d'application

Nous étudions l'interaction homme-robot d'un point de vue spatial. Les scénarios typiques de cette application incluent un homme ordonnant un robot de réaliser une mission. Celle-ci consiste

à effectuer des actions sur certains objets dans un environnement. La définition et la réalisation de la mission considère la composante spatiale avec ses facettes quantitatives, qualitatives et hiérarchiques. Pour réaliser ce type de mission, l'humain et le robot doivent partager la même description de l'environnement. Cela exige que les descriptions créées par l'humain et le robot correspondent, mais aussi qu'un système qui permet de faire une médiation entre la vision de l'humain et celle du robot est nécessaire. Pour mieux illustrer nos propos, voici un exemple d'interaction homme-robot dans un environnement donné.

Exemple 5 *Considérons l'environnement décrit par la figure 6.1. Cet environnement est composé de six bâtiments reliés par six couloirs. Une description de cet environnement en utilisant SpaceOntology est donnée en annexe D.*

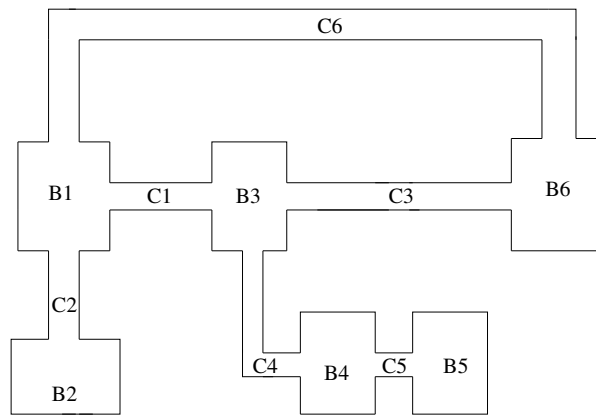


FIGURE 6.1 – Complexe de bâtiments.

L'humain et le robot se trouvent dans le bureau O_{11}^{B1} (figure 5.1, page 118). Celui-ci se trouve dans le premier étage dans le bâtiment B1. L'humain demande au robot d'aller dans le bureau proche de la machine à café et de lui ramener le livre qui se trouve sur la table. La machine à café se trouve dans le hall au deuxième étage du bâtiment B6.

Pour résoudre des problèmes de planification comme celui de l'exemple 5, il faut utiliser le processus de planification décrit par la figure 1 (page 88). La section suivante donne le principe de ce processus.

6.2 Processus de planification : principes

Tout d'abord, nous exprimons le problème (domaine, état initial, état but) avec le langage Spatial-PDDL (figure 1, page 88). Lors de la sélection d'un opérateur de planification spatiale, il faut vérifier que l'ensemble des préconditions symboliques et spatiales sont satisfaites et que les requêtes spatiales sont valides. La définition 31 (page 129) a introduit l'opérateur de planification spatiale. Par conséquent, il est nécessaire de faire une modification des algorithmes de planification afin de prendre en compte les éléments spatiaux (préconditions, effets et requêtes)

lors de la sélection d'un opérateur. Dans certains cas, les préconditions spatiales concernent l'existence d'un chemin entre deux régions qui peuvent être définies d'une manière floue. Ainsi, nous définissons un « problème de planification de chemins entre positions floues » (*fuzzy path planning problem* dans [Belouaer et al., 2011]).

Définition 35 *Considérons deux positions floues ($\tilde{i}_p$ et $\tilde{t}_p$) et un agent Ag qui doit se déplacer dans un environnement décrit par une instance de *SpaceOntology*. Un chemin solution p_{th} entre $\tilde{i}_p$ et $\tilde{t}_p$ est une séquence continue de positions ou de régions qui permettent à l'agent Ag de se déplacer entre la position initiale $\tilde{i}_p$ et la position but $\tilde{t}_p$ en évitant les obstacles et en respectant les contraintes cinématiques de l'agent $\mathcal{CC}(Ag)$.*

Dans ce qui suit, nous considérons que la position initiale est définie d'une manière précise et que la position finale est définie d'une manière floue. Le travail présenté ici peut aussi être appliqué dans le cas de positions précises.

Algorithme 4 : rechercheDeChemins

Données : $\Phi, R, I, i_p, \tilde{t}_p, \mathcal{CC}(Ag)$

Résultat : p_{th} : un chemin selon différents critères entre i_p et $\tilde{t}_p$,
 t_p : une définition plus précise de $\tilde{t}_p$

début

$\Gamma_G \leftarrow \text{développer}(\Gamma_G(\Phi, R, I, i_p, \tilde{t}_p, \mathcal{CC}(Ag)));$
 $p_{th} \leftarrow \text{trouverUnChemin}(\Gamma_G, i_p, \tilde{t}_p);$
 $t_p \leftarrow \text{définir}(\tilde{t}_p);$

fin

L'algorithme 4 présente une synthèse des étapes permettant de résoudre un problème de planification de chemins entre positions floues. En entrée, la procédure **rechercheDeChemins** prend une ontologie décrite par *SpaceOntology* et est définie par le triplet (Φ, R, I) , la position initiale i_p , la position cible floue $\tilde{t}_p$ et les contraintes cinématiques de l'agent $\mathcal{CC}(Ag)$ réalisant la mission. Cette procédure retourne un chemin entre i_p et $\tilde{t}_p$ et donne une délimitation de la zone cible floue $\tilde{t}_p$ que nous notons t_p . Nous résumons les étapes de recherche de chemins en trois points :

1. la génération d'un graphe appelé « graphe de réseaux de passages » (*Crossing Network Graph*) et noté Γ_G en utilisant les relations topologiques et la définition hiérarchique à partir de *SpaceOntology*. En effet, la réalisation de la mission nécessite le déplacement du robot entre différentes régions de différents niveaux hiérarchiques ;
2. la recherche de chemin entre i_p et $\tilde{t}_p$ dans le graphe Γ_G . Nous cherchons un chemin entre deux positions i_p (une position bien définie) et une position $\tilde{t}_p$ (une position floue) dans le graphe de réseaux de passages. La recherche de chemins dans un graphe de réseaux de passages consiste à chercher un chemin dans chaque niveau hiérarchique. Un chemin solution est une liste de régions que l'agent doit traverser pour atteindre sa position cible.

C'est une séquence de chemins $\{pth_0, \dots, pth_k\}$ telle que chaque pth_i contient des nœuds du i^e niveau hiérarchique et c'est une spécialisation des nœuds appartenant à pth_{i-1} ;

3. la définition de la position $\tilde{t}_p$ en utilisant les relations de distance floues, puisqu'il y a une zone d'action définie d'une manière imprécise : « le bureau proche de la machine à café ».

Dans l'exemple 5, l'équipe se trouve dans le bureau O_{11}^{B1} et la cible se trouve à l'étage E_2^{B6} . Ces régions ne sont pas de même niveau hiérarchique. De plus, ces régions ne sont pas adjacentes directement. Ainsi, nous cherchons à définir un chemin entre les régions O_{11}^{B1} et la région où se trouve la machine à café dans la région E_2^{B6} .

6.3 Description avec Spatial-PDDL

Le code 6.1 décrit la mission formulée dans l'exemple 5 (page 142). Dans cette définition du problème, nous avons des objets symboliques (`book table - item`, `Arnaud - human`, `Bobo - robot`) et des objets spatiaux précis (`initial_region referent_region - region`) et imprécis (`target - region`) .

```
(define (problem bookmissionpb_2)
  (:domain bookmission)

  (:objects Arnaud - human Bobo - robot
            book table - item
            initial_region referent_region - region
            relation - fuzzy_distance
            target- region ) ;; fuzzy region

  (:init
    (on book table)
    (@sponto:labelregion initial_region "o_11_b1")
    (@sponto:labelregion referent_region "coffe_machine")
    (@sponto:isA relation "Close")
    (At Arnaud initial_region)
    (At Bobo initial_region)
    (At table target)
    (Empty Arnaud)
    (Empty Bobo)
    (not(Holding Arnaud book))
    (FormulateQueries( target relation referent_region))
  )
  (:goal
    (and ( not (on book table)) (Hodling Arnaud book)))
  )
)
```

CODE 6.1 - Définition d'un problème de planification dans Spatial-PDDL.

L'état initial est décrite d'une manière symbolique ((on book table, not(Holding Arnaud book))) et d'une manière spatiale (@sponto:labelregion initial_region "o_11.b1"). De plus, l'état initial spécifie le besoin de requêtes pour définir des variables en définissant la fonction `FormulateQueries`. Par exemple, délimiter la zone cible définie d'une manière floue est exprimée en fonction de la relation proche et la région de référence. L'état but à atteindre est : « Arnaud a le livre » ((Hodling Arnaud book)). Le planificateur SPOON résout ce problème.

6.4 Planification de tâches

Le planificateur de tâches cherche à définir la séquence d'actions permettant de réaliser une tâche, comme prendre un objet sur une table, en tenant compte de la composante spatiale. Pour le planificateur de tâches, nous avons retenu les planificateurs GraphPlan [Blum et Furst, 1997] et HTN [Erol, 1996]. Le premier pour sa capacité à gérer l'exclusion mutuelle pour les actions parallèles (par exemple : avancer et pousser l'objet) et HTN pour sa capacité à gérer une représentation hiérarchique des tâches. Cependant, introduire des opérateurs de planification spatiale nécessite de modifier les algorithmes correspondants afin de prendre en compte les éléments spatiaux (algorithme 5).

Algorithme 5 : instantiationOperateur

Données : Un opérateur o , l'état courant s , ensemble de contraintes $\mathcal{C}$
Résultat : L'ensemble A de toutes les instances possibles de l'opérateur o
début
 $\mathcal{V} \leftarrow \text{variables}(o)$;
 $\tilde{\mathcal{V}} \leftarrow \text{variablesDeRequetes}(o)$;
 pour chaque $\tilde{v} \in \tilde{\mathcal{V}}$ **faire**
 $v \leftarrow \text{definir}(\tilde{v})$;
 $\mathcal{V} \leftarrow \mathcal{V} \cup v$;
 fin
 pour chaque $v \in \mathcal{V}$ **faire**
 instancier(precond_sp, v) ;
 $rep \leftarrow \text{envoyerRequeteASatisfaire}(\text{precond_sp}, \mathcal{C})$;
 si la réponse de la requête est positive **alors**
 $a \leftarrow \text{instancier}(o, v)$;
 récupérer($effects_sp$) ;
 $A \leftarrow A \cup a$;
 fin
 fin
fin

L'algorithme 5 prend en entrée un opérateur de planification o , l'état courant du monde s et un ensemble de contraintes $\mathcal{C}$. Ces contraintes concernent principalement les contraintes cinématiques de l'agent réalisant la mission. Il retourne en sortie un opérateur instancié.

$\mathcal{V}$ est l'ensemble de couples (paramètre, valeur) satisfaisant les préconditions, les variables v de l'ensemble $\mathcal{V}$ représentent des informations symboliques et spatiales (précises ou floues). L'instanciation des variables se fait en deux étapes :

1. les variables symboliques sont instanciées par le module de raisonnement symbolique ;
2. les variables spatiales sont ensuite instanciées par le module de raisonnement spatial qui traite à la fois leur aspect précis et flou.

Algorithme 6 : application

Données : Une action a , l'état courant s

Résultat : Le nouvel état courant s'

début

$effects_a \leftarrow \text{calculer}(effects(a), effects_sp(a)) ;$

$s' \leftarrow \text{appliquer}(effects_a) ;$

fin

Les effets de l'action sont calculés à partir des effets symboliques et des effets spatiaux. Ils sont appliqués à l'état courant du monde produisant ainsi un nouvel état. Ce calcul est effectué en unifiant les variables des effets symboliques avec ceux des effets spatiaux (algorithme 6).

6.5 Planificateur spatial hiérarchique et flou

L'algorithme 4 (page 143) a présenté les différentes étapes pour planifier un chemin. Tout d'abord, il faut définir à partir de *SpaceOntology* le « graphe de réseaux de passages » par un raisonnement hiérarchique et topologique. Ensuite, il faut chercher le chemin entre les différentes zones d'actions. Enfin, s'il y a besoin, il faut délimiter les zones imprécises en se fondant sur la définition des relations de distance floues.

6.5.1 Génération du graphe de réseaux de passages

Un graphe de réseaux de passages, noté Γ_G , est un graphe orienté défini dans [Belouaer et al., 2010b]. Les nœuds de ce graphe sont organisés d'une manière hiérarchique. Les arcs représentent les liens hiérarchiques entre les nœuds. Un nœud dans ce graphe est un réseau de passages (*Crossing Network*) ou une région selon le niveau de détails que nous nous fixons (définition 36)

Définition 36 *Un réseau de passages, noté Γ^l , est un graphe $\Gamma^l = (V^l, E^l)$, tel que $V^l = \{v_1^l, \dots, v_n^l\}$ est l'ensemble de sommets appartenant au niveau hiérarchique l , chaque v_i^l est un réseau de passages ou une région et $E^l = \{e_1^l, \dots, e_m^l\}$ est l'ensemble d'arcs, chaque arc e_j^l exprime une relation d'adjacence entre deux régions appartenant au niveau hiérarchique l et représentées par des sommets.*

Algorithme 7 : développer Γ_G **Données :** $\Phi, R, I, i_p, \tilde{t}_p, \mathcal{CC}(Ag)$ **Résultat :** Un graphe de réseaux de passages Γ_G **début***/** Génération du réseau de passages le plus abstrait **/* $\Gamma^{L_l} \leftarrow \text{définirLeRéseauDePassagesAbstrait}(\Phi, R, I, i_p, \tilde{t}_p, \mathcal{CC}(Ag));$ */** Initialisation du graphe de réseaux de passages avec le réseau le plus abstrait **/* $\Gamma_G \leftarrow \Gamma^{L_l};$ */** Récupérer le niveau hiérarchique le plus abstrait **/* $L_l \leftarrow \text{définirLeNiveauAbstrait}(\Gamma_G);$ */** Initialisation d'un compteur l qui parcourt les différents niveaux hiérarchiques **/* $l \leftarrow L_l;$ */** Récupérer le niveau hiérarchique le plus détaillé **/* $H_l \leftarrow \text{définirLeNiveauDétaillé}(\tilde{t}_p);$ **tant que $l < H_l$ faire***/** Définir le réseau de passages au niveau hiérarchique l **/* $\Gamma^l \leftarrow \text{définirLeRéseau}(\Gamma_G, l);$ */** Définir le réseau de passages au niveau hiérarchique $l + 1$ à partir du réseau* Γ^l ***/* $\Gamma^{l+1} \leftarrow \text{développer}(\Gamma^l, \Phi, R, I, \mathcal{CC}(Ag));$ */** Ajouter le réseau de passages au niveau hiérarchique $l + 1$ au graphe de*réseaux de passages au niveau $l + 1$ ***/* $\text{ajouter}(\Gamma^{l+1}, \Gamma_G, l + 1);$ */** Incrémenter le compteur l qui parcourt les différents niveaux hiérarchique **/* $l \leftarrow l + 1;$ **fin****fin**

L'algorithme 7 décrit la fonction `développer. $\Gamma_G$` qui construit un graphe de réseaux de passages pour les positions i_p et $\tilde{t}_p$. Ce graphe définit tous les chemins possibles entre i_p et $\tilde{t}_p$. L'algorithme 7 se déroule en deux étapes. La première étape consiste à générer le réseau de passages abstrait (la fonction `définirLeRéseauDePassagesAbstrait`). La deuxième étape consiste à développer, d'une manière incrémentale, le réseau abstrait pour l'affiner. Nous retenons de l'algorithme 7 la définition d'un réseau de passages pour chaque niveau hiérarchique l à partir du niveau hiérarchique précédent $l - 1$, la description de l'espace fournie par le triplet (Φ, R, I) ainsi que les contraintes cinématiques de l'agent $\mathcal{CC}(Ag)$. Cette définition est incrémentale et se fait du niveau hiérarchique le plus abstrait L_l au niveau hiérarchique le plus détaillé H_l .

Considérons l'environnement décrit par la figure 6.2. La figure 6.3 décrit l'utilisation de *SpaceOntology* (pour plus de lisibilité, nous ignorons les attributs). Nous avons trois instances du concept **Regions** à savoir O_{11}^{B1} , O_{12}^{B1} et H_{11}^{B1} et une instance du concept **HasRelation** : *link1*. Cette instance permet de représenter un lien spatial entre une région cible O_{12}^{B1} et une région de référence O_{11}^{B1} . Ce lien spatial concerne une relation topologique *insidebelow* instance du concept **TopologicalRelations** est définie par les concepts I_x et B : $O_{12}^{B1} \langle I_x, B \rangle O_{11}^{B1}$.

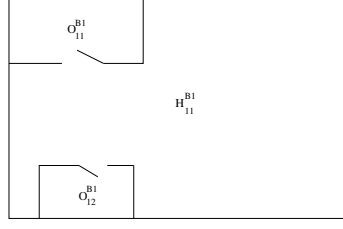


FIGURE 6.2 – Plan partiel du premier étage dans le bâtiment $B1$.

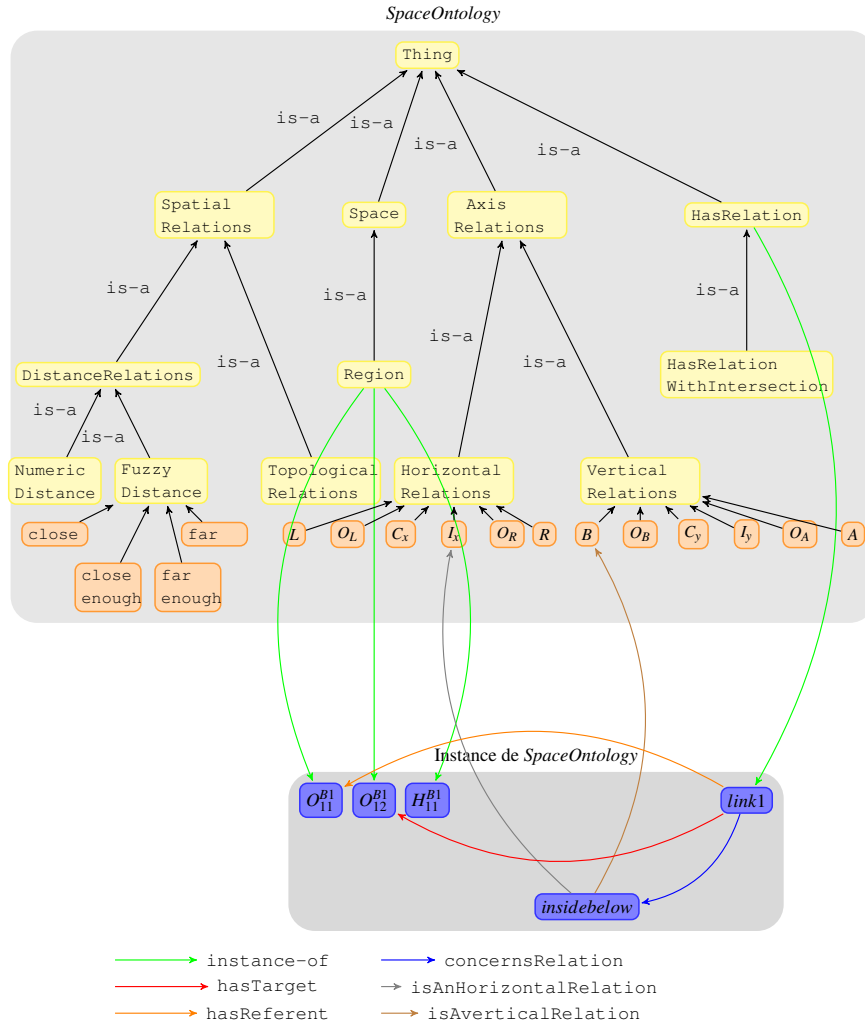


FIGURE 6.3 – Modélisation du plan décrit dans la figure 6.2 en utilisant les éléments de la TBox de *SpaceOntologie*. Les nœuds jaunes sont les concepts, les nœuds oranges sont des individus considérés comme des concepts et les nœuds bleus sont des instances décrivant des régions, des relations, etc. Les arcs noirs sont des liens hiérarchiques entre les concepts. Les arcs verts indiquent pour chaque instance le concept correspondant. Les autres arcs définissent certaines relations décrites dans *SpaceOntologie*.

Génération du réseau abstrait

La fonction décrite par l'algorithme 8 prend en entrée la description de l'environnement considéré définie par le triplet (Φ, R, I) , les positions initiale i_p et cible $\tilde{t}_p$ et les contraintes cinématiques de l'agent $\mathcal{KC}(Ag)$. Elle retourne le réseau de passages le plus abstrait Γ^{L_l} .

Algorithme 8 : définirLeRéseauDePassagesAbstrait

Données : $\Phi, R, I, i_p, \tilde{t}_p, \mathcal{KC}(Ag)$
Résultat : $\Gamma^{L_l} = (N^{L_l}, E^{L_l})$
début

```

  /** Définir le premier niveau hiérarchique abstrait et commun aux deux régions  $i_p, \tilde{t}_p$ 
  **/
   $L_l \leftarrow \text{définirLeNiveauAbstrait}(\Phi, R, I, i_p, \tilde{t}_p) + 1$ ;
  /** Initialisation de l'ensemble des sommets  $N^{L_l}$  et de l'ensemble des arcs  $E^{L_l}$  **/
   $N^{L_l} \leftarrow \emptyset, E^{L_l} \leftarrow \emptyset$ ;
  /**  $i_z$  correspond à la zone d'intérêt hiérarchique englobant  $i_p$  au niveau le plus
  abstrait  $L_l$  **/
   $i_z \leftarrow Z^{L_l}(i_p), N^{L_l} \leftarrow i_z$ ;
  /**  $t_z$  correspond à la zone d'intérêt hiérarchique englobant  $\tilde{t}_p$  au niveau le plus
  abstrait  $L_l$  **/
   $t_z \leftarrow Z^{L_l}(\tilde{t}_p), N^{L_l} \leftarrow N \cup t_z$ ;
  /**  $rg$  un compteur pour parcourir toutes les régions appartenant à niveau  $L_l$  **/
   $rg_i \leftarrow t_z$ ; /** Initialisation de  $rg$  avec  $t_z$  **/
  /**  $Rg^{L_l}$  l'ensemble de toutes les régions appartenant à niveau  $L_l$  **/
   $Rg^{L_l} \leftarrow \text{définir}(\Phi, I, L_l)$ ;
  tant que  $rg_i \neq i_z$  faire
    sélectionner  $rg$  de  $R^{L_l}$  tel que :
       $rg$  est concernée par la mission ;
       $rg$  n'est pas une impasse ;
       $rg_i$  et  $rg$  vérifient une relation d'adjacence directe ;
       $rg$  respecte  $\mathcal{KC}(Ag)$  ;
       $passerelles \leftarrow \text{définirPasserelles}(rg_i, rg)$  telle que chaque  $p_i \in passerelles$ 
      respecte  $\mathcal{KC}(Ag)$  ;
       $N^{L_l} \leftarrow N^{L_l} \cup rg$  ;
       $E^{L_l} \leftarrow E^{L_l} \cup \text{construireLesArcs}(rg_i, rg, passerelles)$  ;
       $rg_i \leftarrow rg$  ;
  fin
   $\Gamma^{L_l} \leftarrow (N^{L_l}, E^{L_l})$ ;

```

fin

Tout d'abord, nous considérons le niveau hiérarchique le plus abstrait L_l permettant de construire un réseau de passages entre les régions englobant les positions i_p et $\tilde{t}_p$ dans ce niveau définies respectivement par $Z^{L_l}(i_p)$ et $Z^{L_l}(\tilde{t}_p)$, telles que ces deux régions sont les descendants d'un même nœud (si $l \leftarrow L_l - 1$, on a $Z^l(i_p) = Z^l(\tilde{t}_p)$). Notons par i_z et t_z respectivement $Z^{L_l}(i_p)$ et $Z^{L_l}(\tilde{t}_p)$. Ensuite, l'idée consiste à trouver tous les chemins possibles entre i_z et t_z . Ces chemins doivent respecter les contraintes cinématiques de l'agent. La construction de ce réseau

se fait en se fixant la région cible t_z jusqu'à atteindre la région de départ i_z . La définition des régions i_z et t_z se base sur la définition hiérarchique de l'environnement et en particulier sur l'exploitation des relations définies dans *SpaceOntology* : *isPartOf* et *consistsOf*. Ces deux relations sont transitives (voir les règles $\mathcal{H}_{\mathcal{R}.1}$ et $\mathcal{H}_{\mathcal{R}.2}$, page 102). Pour définir les deux régions i_z et t_z , nous avons recours aux requêtes pour définir des zones d'intérêts hiérarchiques. La figure 6.4 est une instance de requêtes définissant la zone d'intérêt hiérarchique pour la position i_p au niveau hiérarchique L_l . La requête décrite par la figure 6.4 est adressée à *SpaceOntology* ; celle-ci fait appel aux mécanismes d'inférence pour exécuter les règles $\mathcal{H}_{\mathcal{R}.1}$ et $\mathcal{H}_{\mathcal{R}.2}$. Ce qui permet de répondre à cette requête.

```
PREFIX ns: <http://www.owl-ontologies.com/SpaceOntology_1.owl#>
SELECT ?father
WHERE  ?region rdf:type ns:Regions
        ?region ns:label "i_p"
        ?region ns:isPartOf ?father
        ?father ns:level ?level.
Filter ( ?level = "L_l" ).
```

FIGURE 6.4 – Requête pour définir la zone d'intérêt hiérarchique pour la position i_p au niveau hiérarchique L_l .

La fonction *définir*, décrite dans l'algorithme 8, prend en entrée l'ensemble des concepts Φ définis dans *SpaceOntology*, des individus décrivant l'environnement considéré et le niveau L_l considéré comme le niveau le plus abstrait dans notre hiérarchie. Cette fonction définit un ensemble d'instances Rg^{L_l} , tel que chaque instance est une région appartenant au niveau hiérarchique L_l . La fonction *définir* se base sur la requête décrite par la figure 6.5.

```
PREFIX ns: <http://www.owl-ontologies.com/SpaceOntology_1.owl#>
SELECT ?region
WHERE  ?region rdf:type ns:Regions
        ?region ns:level ?level
Filter (?level = "L_l")
```

FIGURE 6.5 – Requête utilisée dans la fonction *définir*. Cette requête retourne tout individu de type région et appartenant au niveau hiérarchique L_l .

Chaque région rg_i sélectionnée dans l'ensemble Rg^{L_l} doit respecter l'ensemble des critères suivants :

1. elle est concernée par la mission. Par exemple, dans notre cas d'étude, l'immeuble $B4$ n'est pas concernée par la mission ;
2. elle respecte les contraintes cinématiques de l'agent qui va la traverser ;
3. elle ne doit pas être une impasse.

Considérons l'exemple 6.1. La première région abstraite commune aux deux régions O_{11}^{B1} et E_2^{B6} est l'espace globale c'est à dire le complexe de bâtiments (figure 6.1, page 142). Le niveau hiérarchique est de 0 selon l'ontologie décrite dans l'annexe D. Nous considérons les régions englobant O_{11}^{B1} et E_2^{B6} au niveau 1. Selon la description (annexe D), ce sont les régions $B1$ et $B6$. Ensuite, nous cherchons les relations d'adjacence entre $B1$ et $B6$.

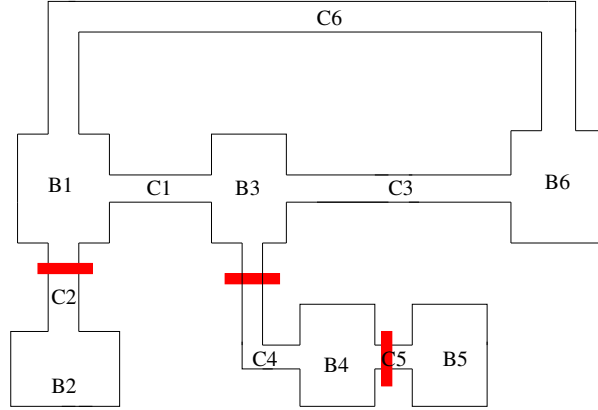
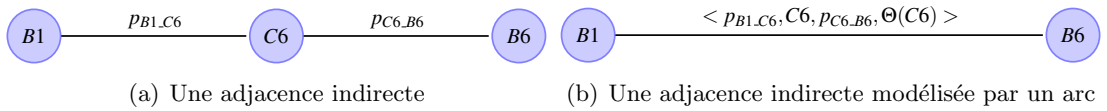


FIGURE 6.6 – Suppression des impasses.

À partir de l'ontologie nous pouvons considérer plusieurs chemins possibles entre les bâtiments $B1$ et $B6$ en considérant $B2$, $B3$, $B4$ et $B5$. Cependant passer par les bâtiments $B2$, $B4$ et $B5$ n'est pas nécessaire. Ces régions ne sont pas concernées par la mission. Les régions $B2$ et $B5$ sont des impasses (une passerelle pour entrer et sortir). La région $B4$ n'est pas incluse dans la mission et en plus elle est connectée à la région $B5$ qui est une impasse. Ainsi, $B4$ est une impasse (figure 6.6).

FIGURE 6.7 – Définition d'un arc dans le réseau de passages abstrait Γ^{L_i} .

Au niveau le plus abstrait L_i , pour construire le réseau de passages le plus abstrait Γ^{L_i} entre i_z et t_z , nous modélisons les relations d'adjacence directes et indirectes. Par exemple, $B1$ est adjacente à $C6$ par la passerelle $p_{B1.C6}$ (figure 6.7(a)) et $C6$ est adjacente à $B6$ par la passerelle $p_{C6.B6}$ (figure 6.7(a)). La région $C6$ est un couloir connectant les deux bâtiments $B1$ et $B6$. Au niveau hiérarchique suivant $C6$ n'a pas de définition hiérarchique (voir annexe D). Ainsi, la représentation de la relation d'adjacence dans la figure se traduit par la figure 6.7(b). Si la région $C6$ admettait une définition hiérarchique au niveau suivant alors nous aurions conservé la représentation de la figure 6.7(a). Deux régions adjacentes sont reliées par un arc dans le réseau de passages le plus abstrait Γ^{L_i} . Un arc spécifie une relation d'adjacence en définissant les deux

passerelles relatives à chaque région. S'il existe une région rg entre ces deux passerelles, celle-ci est décrite par l'arc ainsi que sa mesure $\Theta(rg)$ (définition 37). Une fois les régions sélectionnées, il faut définir les passerelles. Celles-ci doivent respecter les contraintes cinématiques de l'agent.

Définition 37 La mesure Θ est un couple $\Theta = (d, c)$ définissant une distance d et une mesure d'encombrement c , tel que :

- d est la longueur d'une région rg ou la distance séparant deux positions ;
- c est la mesure d'encombrement (congestion mesure [Belouaer et al., 2011]) d'une région rg ou de la portion séparant deux positions .

Développement hiérarchique

Algorithme 9 : développer

Données : $\Gamma^l = (N^l, V^l), i_p, \tilde{t}_p, \Phi, R, I, \mathcal{KC}(Ag)$
Résultat : $\Gamma^{l+1} = (N^{l+1}, V^{l+1})$
début
 /** Le niveau hiérarchique courant**/
 $l \leftarrow \text{graph.getLevel}();$
 $nodes \leftarrow \emptyset, edges \leftarrow \emptyset;$
 /** Définir la zone d'intérêt hiérarchique pour la zone i_p **/
pour chaque $n \in N^l$ **faire**
 si $(n == Z^{l+1}(i_p))$ **alors**
 $i_z \leftarrow Z^{l+1}(i_p);$
 $gr \leftarrow \text{graph}(i_z, gates, \mathcal{KC}(Ag), \Phi, R, I);$
 $nodes \leftarrow gr.nodes;$
 $edges \leftarrow gr.edges;$
 fin
 /** Définir la zone d'intérêt hiérarchique pour la zone $\tilde{t}_p$ **/
 sinon si $(n == Z^{l+1}(\tilde{t}_p))$ **alors**
 $t_z \leftarrow Z^{l+1}(\tilde{t}_p);$
 $gr \leftarrow \text{graph}(t_z, gates, \mathcal{KC}(Ag), \Phi, R, I);$
 $nodes \leftarrow nodes \cup gr.nodes;$
 $edges \leftarrow edges \cup gr.edges;$
 fin
 /** Sélection des nœuds et des arcs selon les passerelles**/
 sinon
 $g \leftarrow gates;$
 $gr \leftarrow \text{graph}(g, \mathcal{KC}(Ag), \Phi, R, I);$
 $nodes \leftarrow nodes \cup gr.nodes;$
 $edges \leftarrow edges \cup gr.edges;$
 fin
fin
 $graph_{l+1} \leftarrow (nodes, edges);$
fin

La construction du graphe Γ_G se fait d'une manière itérative. La première étape consiste à développer le réseau de passages le plus abstrait. Il sera enrichi à chaque étape par un réseau de passages de niveau hiérarchique l jusqu'à atteindre le niveau hiérarchique voulu. L'intuition est que à chaque niveau hiérarchique l , chaque nœud peut être relié par des arcs traduisant un lien hiérarchique (`consistsOf`) avec un ensemble de nœuds au niveau hiérarchique $l + 1$.

L'algorithme 9 décrit la fonction `développer`. Cette fonction prend en entrée un réseau de passages au niveau hiérarchique l pour retourner un réseau de passages au niveau hiérarchique suivant $l + 1$. Dans un niveau hiérarchique donné, la génération d'un réseau de passages est basée principalement sur les relations topologiques [Belouaer *et al.*, 2010a]. Pour cela, nous nous intéressons aux relations d'adjacence directes et indirectes. La relation d'adjacence directe est utilisée lorsque la passerelle sert de sortie pour une région et d'une entrée pour la région adjacente et que c'est deux régions appartiennent au même nœud au niveau plus abstrait. Par exemple, O_{11}^{B1} est adjacente à H_{11}^{B1} par la passerelle d_{11} . À chaque niveau hiérarchique, la définition des zones d'intérêt hiérarchiques pour les positions initiale et cible permet de gérer la connaissance spatiale floue.

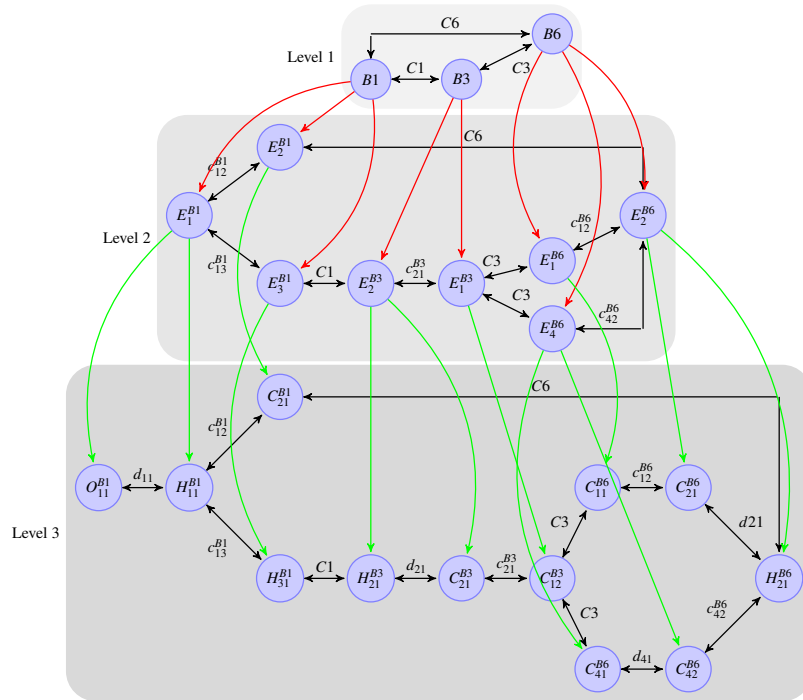


FIGURE 6.8 – Exemple de graphe de réseaux de passages

En appliquant, les algorithmes 7, 8 et 9, nous avons le graphe de réseaux de passages entre O_{11}^{B1} et la région où se trouve la machine à café dans la région E_2^{B6} comme décrit dans la figure 6.8.

6.5.2 Définition du chemin vers une cible floue

Nous nous intéressons maintenant à la recherche de chemins dans le graphe de réseaux de passages pour trouver un chemin entre i_p et $\tilde{t}_p$. La recherche de chemins se fait en deux étapes : trouver un chemin et définir une zone floue.

Trouver un chemin

La recherche de chemins se fait d'une manière hiérarchique en essayant de minimiser la distance entre deux positions et de faciliter l'accès à une région, afin d'assurer une exécution des actions d'une manière simple et rapide. Pour cela, nous utilisons la mesure Θ (définition 37) qui combine la distance et la mesure d'encombrement, tel que $\Theta(p_1, p_2) = (d(p_1, p_2), c(p_1, p_2))$ où $d(p_1, p_2)$ est une distance entre p_1 et p_2 et $c(p_1, p_2)$ est une mesure d'encombrement des régions entre les régions p_1 et p_2 . En effet, passer par certains couloirs de grande distance peut être plus rapide que passer par des couloirs de petites distances avec plusieurs obstacles.

Algorithme 10 : calculerChemin

Données : Graphe de réseaux de passages Γ_G

Résultat : le chemin solution à chaque niveau hiérarchique pth_{map}

début

pth_{map} est une liste (l, p) tel que l est le niveau hiérarchique et p un chemin au niveau l ;

pour $(i = \Gamma_G.firstLevel(); \Gamma_G.endLevel(); i++)$ **faire**

si i est le niveau hiérarchique le plus abstrait **alors**

$\Gamma^{L_i} \leftarrow \Gamma_G.getGraph(i);$

$p \leftarrow \text{DijkstraMC}(\Gamma^{L_i});$

ajouter (i, p) à pth_{map} ;

sinon

$levelbefore \leftarrow i - 1;$

$p \leftarrow pth_{map}.get(levelbefore);$

$pth_{list} \leftarrow \text{chercher}(p, levelbefore, \Gamma_G);$

$pth_{res} \leftarrow Pth(pth_{list});$

ajouter (i, pth_{res}) à pth_{map} ;

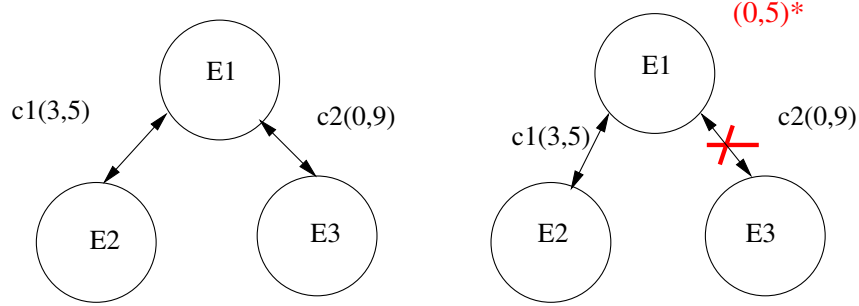
fin

fin

fin

L'algorithme 10 décrit la fonction `calculerChemin`. Cette fonction prend en entrée un graphe de réseaux de passages et retourne le chemin solution entre i_p et $\tilde{t}_p$. Dans le cas où i est le niveau le plus abstrait, la recherche de chemins se fait dans le réseau de passages le plus abstrait Γ^{L_i} . La fonction `DijkstraMC` utilise l'algorithme de Dijkstra multi-critère avec une seule source $Z^i(i_p)$. Cette fonction retourne le chemin le plus court entre le nœud source $Z^i(i_p)$ et le nœud cible $Z^i(\tilde{t}_p)$ (i correspond au niveau hiérarchique le plus abstrait L_i). Pour chaque niveau hiérarchique i , le plus court chemin est le chemin ayant la plus petite mesure $\Theta^i = (d, c)^i$, tel que d est la distance

et c est une mesure d'encombrement. Soient deux arcs e_1 et e_2 ayant respectivement les mesures Θ_{e_1} et Θ_{e_2} . Pour choisir quel arc, nous procédons comme suit. Tout d'abord, nous calculons une mesure optimale de Θ que nous notons $\Theta^* : \Theta^* = (d,c)^* = [Min(d_{e_1}, d_{e_2}), Min(c_{e_1}, c_{e_2})]$. Ensuite, nous sélectionnons l'arc qui nous permet de minimiser la distance avec Θ^* .



$$Min(Dist) = \begin{cases} Dist(\Theta_{c_{12}^{B6}}, \Theta^*) = Dist((3,5), (0,5)) = 3 \\ Dist(\Theta_{c_{42}^{B6}}, \Theta^*) = Dist((0,9), (0,5)) = 4 \end{cases}$$

$$\Theta^* = (d,c)^* = [Min(3,0), Min(5,9)]$$

FIGURE 6.9 – Calcul pour la sélection d'un nœud.

Pour chaque arc (figure 6.9), nous calculons la distance entre Θ^* et $\Theta_{c_{12}^{B6}}$ et Θ^* et $\Theta_{c_{42}^{B6}}$. Nous sélectionnons l'arc qui nous permet de minimiser la distance avec $\Theta^* : c_{12}^{B6}$ (figure 6.9).

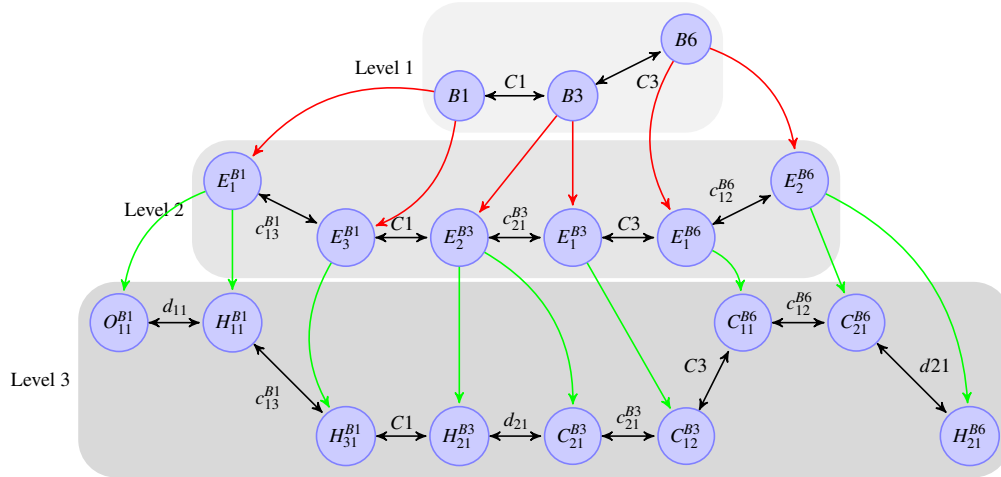


FIGURE 6.10 – Chemin solution hiérarchique.

Pour chaque niveau hiérarchique i la recherche de chemin respecte les critères d'optimalité en utilisant le principe décrit dans la figure 6.9. La recherche de chemins est réalisée en chaînage arrière à partir du nœud cible $Z^i(\tilde{t}_p)$, où i est un niveau hiérarchique intermédiaire. Aussi, nous ne nous intéressons pas dans un niveau hiérarchique au réseau de passages dans sa globalité.

En effet, nous considérons seulement les nœuds et les arcs appartenant au chemin solution du niveau $i - 1$ (*levelbefore*). La fonction **chercher** peut générer le chemin pour chaque nœud au niveau suivant, exploite l'organisation hiérarchique de l'espace et permet de réduire le temps de calcul. En effet, seulement les nœuds du chemin du niveau i sont explorés dans le niveau $i + 1$ et ainsi certains arcs dans le graphe ne seront pas explorés. Cette fonction utilise le même principe de la fonction **DijkstraMC**. La recherche de chemins dans un graphe de réseaux de passages consiste à rechercher un chemin dans un réseau de passages de chaque niveau hiérarchique. Un chemin solution est une liste de régions que l'agent doit traverser pour atteindre sa position cible. C'est une séquence de chemins $\{pth_0, \dots, pth_k\}$, telle que chaque pth_i contient des nœuds du i^e niveau hiérarchique et c'est une spécialisation des nœuds appartenant à pth_{i-1} . Par exemple (figure 6.10), le chemin solution au 1^{er} niveau est $B1 - C6 - B6$, la recherche de chemin au 2^e niveau ne va pas considérer les sous nœuds de $B3$.

Définir une zone floue

Pour définir une zone décrite d'une manière précise, une requête de proximité est posée à *SpaceOntology*. Considérons notre scénario de départ décrit dans l'exemple 5. La machine à café se trouve dans le hall H_{21}^{B6} (figure 6.11). Il existe un chemin entre les régions O_{11}^{B1} et H_{21}^{B6} (figure 6.10, page 155). Cependant, le bureau proche de la machine à café n'est pas encore défini. Pour le faire, il faut poser la requête de proximité suivante à *SpaceOntology* : $< close, CM >$. Cette requête fait appel à la règle $\mathcal{C}_{\mathcal{R}.1}$.

```
{translate(?link,?t_link) ∧ translate(?relation,?t_relation)
 ∧ concerns(?link,?relation) ∧ hasTarget(?link,?tgt) ∧ hasReferent(?link,?ref)
  ∧ isAFuzzyDistance(?relation) ∧ is(?t_relation,"close")
  ∧ isPartOf(?ref,?region) ∧ isPartOf(?tgt,?region)}
↓
{concerns(?t_link,?t_relation) ∧ hasTarget(?t_link,?tgt)
 ∧ hasReferent(?t_link,?ref) ∧ isANumericDistance(?t_relation)
 ∧ swrlb:greaterThan(distanceValue(?t_relation),0)
 ∧ swrlb:lessThanOrEqual(distanceValue(?t_relation),alpha(?region))}
```

($\mathcal{C}_{\mathcal{R}.1}$)

La règle $\mathcal{C}_{\mathcal{R}.1}$ permet d'interpréter une relation de distance floue en une relation de distance quantitative permettant de délimiter la zone cible d'une manière numérique. Cette règle stipule que toute distance floue ($isAFuzzyDistance(?relation)$) entre deux régions ($?ref$ et $?tgt$) dans une région courante ($?region$) définie comme proche ($is(?relation,"close")$) est comprise entre 0 et α (avec $\alpha = alpha(?region)$). En d'autres termes, si un bureau appartenant à la région H_{21}^{B6} est à une distance $d \in [0, \alpha]$, alors ce bureau est considéré comme proche à la

machine à café. Formellement, la requête de proximité suivante à *SpaceOntology* : $\langle close, CM \rangle$ considère chaque bureau O_i de cette zone vérifie le degré $N_{\alpha,\beta}(CM, O_i) = 1$.

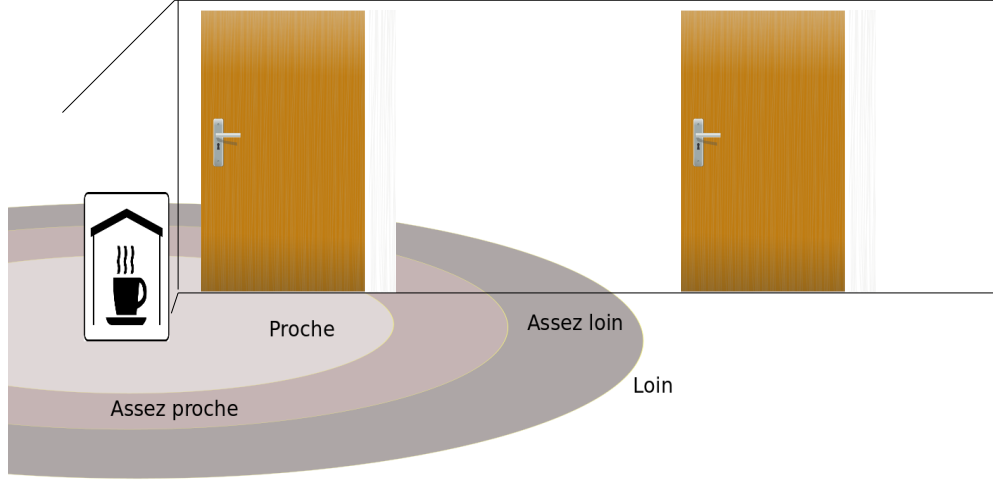


FIGURE 6.11 – Application de la règle $\mathcal{C}_R.1$.

Conclusion

Dans ce chapitre, nous avons montré comment on peut développer un processus de planification dans SPOON. Nous avons dans un premier temps montré comment définir un problème avec Spatial-PDDL. Ensuite, nous avons explicité toutes les étapes du processus de planification nécessitant à la fois un raisonnement symbolique et un raisonnement spatial.

Conclusion de la partie II

Nous avons défini un planificateur appelé SPOON. Celui-ci est un planificateur hybride combinant un raisonnement symbolique et un raisonnement spatial. Le raisonnement symbolique est réalisé par le planificateur de tâches. Le raisonnement spatial est réalisé quand à lui par le planificateur de chemins et *SpaceOntology*. Pour le raisonnement spatial, nous pouvons représenter les entités spatiales d'une manière géométrique et hiérarchique et représenter les relations topologiques, les relations de distance numérique et les relations de distance floue. Nous pouvons également raisonner en exploitant les règles d'inférence définies par *SpaceOntology*. Nous avons proposé Spatial-PDDL comme langage permettant d'exprimer un problème de planification tenant compte de la dimension spatiale. Pour le processus de planification dans SPOON, nous avons défini des requêtes et proposé une définition d'un opérateur de planification spatiale. Lors de ce processus, certaines actions nécessitent un calcul de chemins entre deux positions floues. Nous avons proposé une méthode pour cela. Cette méthode se base sur la génération d'un graphe de recherche de chemins en utilisant les liens hiérarchiques et topologiques à partir de la description de l'environnement selon le modèle de *SpaceOntology*. Pour délimiter les régions floues, nous nous basons sur l'évaluation des relations de distance floues comme décrit par les règles de *SpaceOntology*. La partie suivante étudie les performances de SPOON.

Troisième partie

Expérimentations

Chapitre 7

Performances de SPOON

Sommaire

7.1	Inférence de l'adjacence	164
7.1.1	Principe	164
7.1.2	Résultats expérimentaux	165
7.1.3	Étude de la complexité	165
7.2	Cohérence de la connaissance spatiale : justification du choix AC-3	165
7.3	Performances du planificateur de chemins	167
7.3.1	Performances du principe de la fonction DijkstraMC	167
7.3.2	Intérêt de la hiérarchisation de l'environnement	168
7.3.3	Recherche de chemins dans un graphe de réseaux de passages	168
7.4	Planification de tâches : étude avec HTN	170
7.5	Performances de SPOON	171
7.5.1	Rôle de la connaissance spatiale dans la planification	171
7.5.2	Comparaison avec d'autres planificateurs	172
7.6	Système complet de la planification	173

Ce chapitre présente une étude des performances du planificateur SPOON. Cette étude présente une expérimentation concernant la déduction des relations d'adjacence et justifie le choix d'une version de l'algorithme AC-3 pour maintenir la cohérence de la connaissance spatiale (sections 7.1 et 7.2). Ensuite, nous présentons les performances du planificateur de chemins (section 7.3). Nous présentons aussi, dans le cas de la technique HTN, comment se fait la génération d'un plan solution (section 7.4). Par la suite, nous présentons l'impact de la connaissance spatiale sur la planification ainsi qu'une comparaison avec d'autres planificateurs (section 7.5). Enfin, nous donnons un aperçu du système complet que nous avons mis en place (section 7.6).



7.1 Inférence de l'adjacence

Cette section présente les performances des mécanismes d'inférence de *SpaceOntology*.

7.1.1 Principe

Pour déduire une relation d'adjacence, nous exploitons les relations spatiales dans *SpaceOntology* de deux manières. La première utilise les relations hiérarchiques (figure 7.1(a)) et la deuxième exploite les relations topologiques (figure 7.1(b)). Considérons l'environnement décrit par la figure 6.1 (page 142).

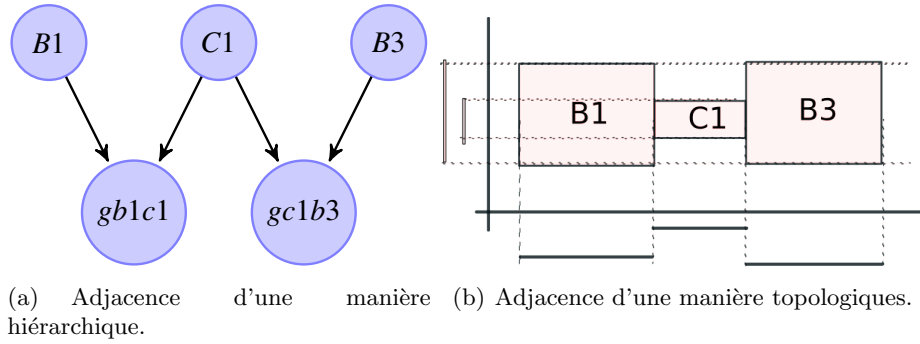


FIGURE 7.1 – Relations d'adjacence.

Considérons la figure 7.1(a). L'équation 7.1, présente la relation d'adjacence déduite entre B1 et B3 via C1. D'un point de vue topologique, la même relation d'adjacence entre B1 et B3 via C1 est déduite d'une manière plus complexe. L'équation 7.2 décrit la figure 7.1(b).

$$Adj\langle B1, B3, \{C1\} \rangle \leftarrow \begin{cases} \{gb1c1\} \in (B1 \wedge C1) \\ \{gc1b3\} \in (C1 \wedge B3) \end{cases} \quad (7.1)$$

$$\begin{cases} \langle B1, C1 \rangle = \langle L, I_y \rangle \wedge ((B1 \cap C1) = \{gb1c1\}) \\ \langle C1, B3 \rangle = \langle R, C_y \rangle \wedge ((C1 \cap B3) = \{gc1b3\}) \end{cases} \quad (7.2)$$

Pour avoir la relation topologique entre B1 et B3, il faut faire appel au mécanisme de composition suivant :

$$\langle B1, C1 \rangle \otimes \langle C1, B3 \rangle = \langle L, I_y \rangle \otimes \langle R, C_y \rangle \quad (7.3)$$

$$\begin{aligned} &= \langle L \otimes R, I_y \otimes C_y \rangle \\ &= \langle \forall r, I_y \vee C_y \rangle \end{aligned} \quad (7.4)$$

De plus, il faut appliquer les règles décrites par l'équation 7.5.

$$\begin{cases} (B1 \cap C1) = \{gb1c1\} \longrightarrow \{gb1c1\} \in (B1 \wedge C1) \\ (C1 \cap B3) = \{gc1b3\} \longrightarrow \{gc1b3\} \in (C1 \wedge B3) \end{cases} \quad (7.5)$$

Intuitivement, nous pouvons dire que la déduction d'une relation d'adjacence à partir des relations hiérarchiques est plus rapide que la déduction à partir des relations topologiques. Cette intuition est vérifiée d'une manière expérimentale et validée par une étude de complexité.

7.1.2 Résultats expérimentaux

Nous considérons 9 instanciations²⁰ de *SpaceOntology* comme décrit dans le tableau 7.1. chaque instance a un nombre précis d'individus²¹. Par exemple, l'instance numéro 1 décrit 15 individus, l'instance numéro 6 décrit 65 individus.

Instance	1	2	3	4	5	6	7	8	9
nombre d'individus	15	25	35	45	55	65	75	85	95

TABLE 7.1 – Différentes instanciations de *SpaceOntology*.

La déduction des liens d'adjacence n'est pas une opération coûteuse en temps de calcul. En effet, il reste de l'ordre de millisecondes. Toutefois, la déduction de l'adjacence à partir des liens hiérarchiques est plus rapide que la déduction de l'adjacence à partir des relations topologiques (figure 7.2).

7.1.3 Étude de la complexité

D'un point de vue théorique, nous confirmons les résultats expérimentaux. En effet, la génération d'un graphe de réseaux de passages dépend du nombre des passerelles considérées que nous notons $|G|$. Dans le cas de la déduction à partir des relations hiérarchiques, la complexité est de $O(|G|)$. Dans le cas de la déduction à partir des relations topologiques la complexité est de $O(|G|^2)$. Par conséquent, dès que nous avons la possibilité d'utiliser les liens hiérarchiques nous le faisons.

7.2 Cohérence de la connaissance spatiale : justification du choix AC-3

Le chapitre 4 a présenté comment maintenir la cohérence de la connaissance spatiale. L'idée est de formaliser toute la connaissance spatiale sous la forme d'un réseau de contraintes binaires qui doit être arc-consistant (définition 16, page 115). Une seule application de la fonction `reviserArc` n'est pas suffisante (algorithme 3, page 115). En effet, la suppression d'une valeur

20. Les exemples décrits dans l'annexe D sont des instances de *SpaceOntology*.

21. Un individu est une instance d'un concept. Par exemple, les régions *B3* et *B4* sont des individus, la relation *insidebelow* est un individu instanciant le concept *TopologicalRelations*.

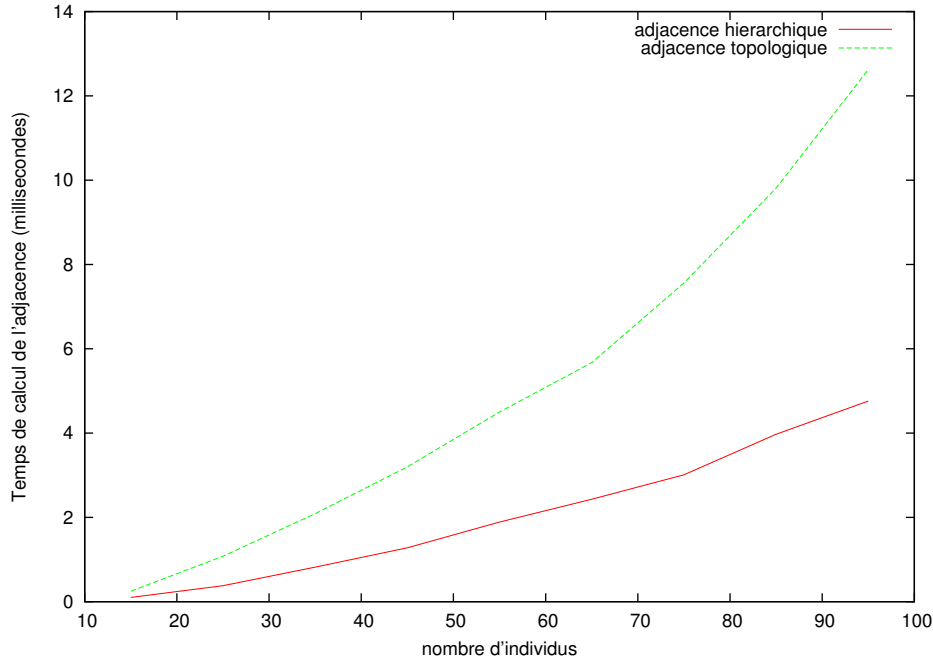


FIGURE 7.2 – Comparaison entre les méthodes de déduction à partir des relations topologiques et de déduction à partir des relations hiérarchiques.

d'un domaine pour une contrainte donnée peut entraîner la suppression d'autres valeurs dans d'autres domaines. D'où la nécessité de réviser à nouveau des arcs déjà révisés.

Dans la littérature, il existe plusieurs algorithmes permettant de rendre un réseau de contraintes binaires arc-consistant. Le plus utilisé est AC-3. Celui-ci est une extension de AC-1 [Mackworth, 1977]. Ce dernier tente de réviser systématiquement toutes les contraintes en dépit du fait qu'elles, ne soient pas affectées par une révision antérieure. La complexité de AC-1 est $O(nmd^3)$, avec n est le nombre d'appels de la fonction `reviserArc` et md^3 est le coût de la la fonction `reviserArc`, où m est le nombre de contraintes et d est la taille du plus grand domaine dans le problème [Mackworth, 1977].

Algorithme	Complexité temporelle	Complexité spatiale
AC-3 [Mackworth et Freuder, 1985]	$O(md^3)$	$O(m + nd)$
AC-4 [Mohr et Henderson, 1986]	$O(md^2)$	$O(md^2)$
AC-5 [Van Hentenryck Yves, 1992]	$O(md^2)$	$O(md)$
AC-6 [Bessière, 1994]	$O(md^2)$	$O(md^2)$
AC-7 [Bessière <i>et al.</i> , 1999]	$O(md^2)$	$O(md)$
AC-2000 [Bessière et Régis, 2001]	$O(md^3)$	$O(md)$
AC-2001 ou AC-3.1 [Bessière <i>et al.</i> , 2005]	$O(md^2)$	$O(md)$

TABLE 7.2 – Complexité des algorithmes d'arc-consistance.

L'amélioration apportée par AC-3 consiste à ne pas réappliquer `reviserArc` à tous les arcs, mais uniquement à ceux susceptibles d'être affectés par la suppression de la valeur d'un domaine. Les arcs devant être traités sont mis en attente dans une liste, et bien évidemment, un arc n'y est pas réinséré s'il y figure déjà. Depuis la proposition de AC-3, plusieurs algorithmes établissant l'arc-consistance ont été proposés, ces algorithmes visent à améliorer la complexité théorique et le temps de calcul effectif.

Soient n le nombre de variables, m le nombre de contraintes et d la taille du plus grand domaine dans le problème. L'AC-3 représente la complexité spatiale optimale (tableau 7.2). L'AC-4 une amélioration de AC-3 qui permet de réduire la complexité temporelle, mais augmente la complexité spatiale. AC-6 mixe les principes issus de AC-3 et AC4. Il élimine quelques vérifications redondantes par rapport à AC-4 et réduit sa complexité temporelle. AC-2000 a la même complexité temporelle qu'AC-3 mais il fait moins de tests de vérification de consistance. AC-2001 ou AC-3.1 est une amélioration d'AC-3 et possède la complexité temporelle optimale. Par conséquent, nous avons choisi d'implémenter AC-2001 ou AC-3.1. De plus, il est facile à implémenter. Pour plus de détails sur les algorithmes de filtrage, nous vous invitons à lire l'article [Dib *et al.*, 2010].

7.3 Performances du planificateur de chemins

Le graphe de réseaux de passages considère seulement les régions qui vérifient les critères suivants : une région ne doit pas être une impasse, doit être concernée par la mission et respecte les contraintes cinématiques de l'agent (algorithme 8, page 149). Ceci permet de réduire le temps de calcul. Cependant, un tel graphe peut être d'une grande taille. Afin de mesurer les performances du planificateur des chemins, nous avons mené des tests de recherche d'un chemin solution sur un ensemble de 81 graphes de réseaux de passages de différentes tailles. Ces expériences ont pour objectif de démontrer d'une part la performance de la fonction `DijkstraMC` (algorithme 10, page 154) et d'autre part l'intérêt de la hiérarchisation de l'espace afin de réduire la taille de l'espace de recherche.

7.3.1 Performances du principe de la fonction `DijkstraMC`

Nous avons effectué des expérimentations sur un ensemble de réseaux de passages générés d'une manière aléatoire. Nos tests concernent des réseaux de passages ayant entre 10 et 38000 nœuds. Considérons un réseau de passages $\Gamma^l = (V^l, E^l)$ au niveau hiérarchique l . Pour calculer un chemin minimisant le facteur Θ , nous avons utilisé l'algorithme Dijkstra multi-critères. L'implémentation de la file d'attente prioritaire avec un tas de Fibonacci rend la complexité $O(E^l + V^l \log V^l)$.

7.3.2 Intérêt de la hiérarchisation de l'environnement

Notre objectif est de démontrer l'intérêt de l'organisation hiérarchique du graphe de réseaux de passages et du fait qu'au niveau l , nous ne considérons que les nœuds descendants des nœuds retenus comme solution au niveau $l - 1$.

Nous avons effectué des expérimentations sur un ensemble de graphes de réseaux de passages générés d'une manière aléatoire. Nos tests concernent des graphes de réseaux de passages ayant entre 10 et 26000 nœuds (figure 7.3).

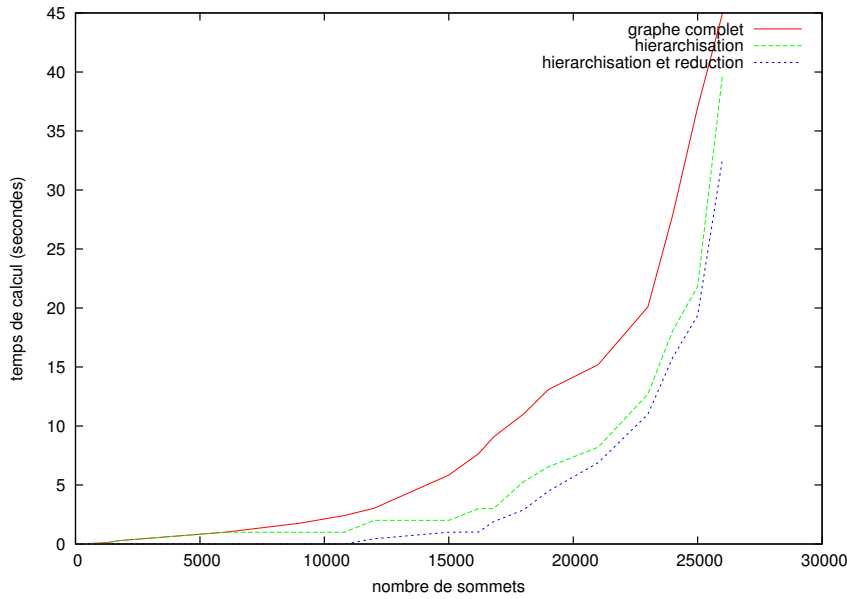


FIGURE 7.3 – Intérêt de l'organisation hiérarchique et de l'élégage.

Nous avons considéré une organisation hiérarchique lors de la génération du graphe de réseaux de passages. Ainsi, la recherche de chemins entre deux positions est décomposée en sous-problèmes en définissant des zones d'intérêts hiérarchiques. De plus, à chaque niveau hiérarchique l , la recherche de chemins ne considère pas le réseau courant dans sa globalité mais ne considère que les nœuds descendants des nœuds retenus en solution au niveau $l - 1$. Par conséquent, le temps de calcul d'un chemins est réduit d'une manière considérable. Par exemple pour 16800 nœuds, le temps de calcul en considérant le graphe comme un tout est réduit de 71% lorsqu'on considère la hiérarchisation et le principe d'élégage décrit ci-dessus.

7.3.3 Recherche de chemins dans un graphe de réseaux de passages

Résultats expérimentaux

Le tableau 7.3 présente les résultats expérimentaux pour la recherche de chemins réalisée par l'algorithme 10 (page 154). Le temps de calcul dépend de la taille du graphe de réseaux de passages. Celui-ci est défini en fonction du facteur de branchement moyen b et le nombre des

niveaux hiérarchiques. Le facteur de branchement d'un nœud n correspond au nombre de sous-nœuds du nœud n . Dans notre expérimentation, nous considérons que pour un niveau donné l on a un facteur de branchement moyen b . Ce facteur varie entre 2 et 10. Le nombre de niveau hiérarchique varie entre 2 et 10. Le tableau 7.3 présente le temps de calcul en fonction du facteur de branchement moyen et le nombre de niveau hiérarchique. Dans notre contexte, c'est-à-dire dans des environnements tels que des immeubles, des étages, etc. Les niveaux hiérarchiques ne dépassent pas 6 et le facteur de branchement moyen ne dépasse pas 5. Selon le tableau 7.3, notre méthode de recherche de chemins retourne un chemin solution en un temps acceptable. En effet, le temps de calcul appartient à l'intervalle $[0, 100]$ secondes pour un nombre de niveau appartenant à $[2, 5]$ et un facteur de branchement appartenant à $[2, 5]$.

l, b	$b = 2$	$b = 3$	$b = 4$	$b = 5$	$b = 6$	$b = 7$	$b = 8$	$b = 9$	$b = 10$
$l = 2$	< 1	< 1	< 1	< 1	< 1	< 1	1	2	40
$l = 3$	< 1	< 1	< 1	< 1	1.05	10	35	62.5	70
$l = 4$	< 1	< 1	< 1	< 1	85	120	342.1	427.2	600
$l = 5$	9	35	50	62.1	160	210.2	300.5	400.2	581.2
$l = 6$	13.2	64.2	80.03	100.2	195.2	281.3	450.5	907.2	990.5
$l = 7$	320	480.2	600.1	900.2	1050.4	1700	1970.01	-	-
$l = 8$	340 s	650.8	801.5	957	995.9	2000.01	-	-	-
$l = 9$	1200.1	2400.4	5000.7	9007	2400.4	-	-	-	-
$l = 10$	8000.4	14000	-	-	-	-	-	-	-

TABLE 7.3 – Temps de calcul pour la recherche de chemins (en secondes). Le symbole (-) représente un grand temps de calcul (> 30000 seconds).

Étude de la complexité

Soit N le nombre des nœuds dans le graphe de réseaux de passages et soit b_n le facteur de branchement qui correspond au nombre de sous-nœuds d'un nœud abstrait n . Nous définissons par b_{n_l} le nombre de sous-nœuds du nœud n au niveau hiérarchique l . Par conséquent, nous avons :

$$N = \sum_{l=\min_l}^{\max_l} \sum_{i=1}^{n_{\min_l-1}} b_{i_l}$$

tel que :

1. $\min_l$ est le niveau hiérarchique le plus abstrait dans le graphe de réseaux de passages,
2. $\max_l$ est le niveau hiérarchique le plus détaillé dans le graphe de réseaux de passages,
3. $n_{\min_l-1}$ l'ensemble de nœuds dans le niveau le plus abstrait dans le graphe de réseaux de passages,
4. b_{i_l} est le nombre de sous-nœuds du nœud i au niveau hiérarchique l ,
5. $\sum_{i=1}^{n_{\min_l-1}} b_{i_{\min_l-1}}$ est le nombre de nœuds dans le niveau $\min_l$.

Pour chaque niveau hiérarchique l , notons par E_l , le nombre d'arcs et $\sum_{i=1}^{n_{l-1}} b_{i_{l-1}}$ le nombre de nœuds, la complexité est de $O(E_l + \sum_{i=1}^{n_{l-1}} b_{i_{l-1}} \log(\sum_{i=1}^{n_{l-1}} b_{i_{l-1}}))$. La complexité totale est :

$$O\left(\sum_{l=\min_l}^{\max_l} E_l + \sum_{l=\min_l}^{\max_l} \left(\sum_{i=1}^{n_{l-1}} b_{i_{l-1}} * \log\left(\sum_{i=1}^{n_{l-1}} b_{i_{l-1}}\right)\right)\right)$$

7.4 Planification de tâches : étude avec HTN

Pour générer un plan solution, nous avons utilisé la technique GraphPlan et HTN. Le temps de génération d'un plan solution est presque le même dans les deux cas. La planification avec la technique HTN, dans nos travaux se fait comme suit. Nous considérons deux types d'hiérarchisation des actions : la simplification et la décomposition spatiale. La simplification consiste à simplifier une action complexe en des actions moins complexes ou primitives. Par exemple, l'action « ranger la cuisine » peut être simplifié en « laver la vaisselle », « ranger la vaisselle », etc. La décomposition spatiale consiste à décomposer une action qui se fait sur différents niveaux hiérarchiques en sous actions, où chaque action est concernée par un seul niveau hiérarchique. La hiérarchisation des actions utilise la simplification et la décomposition spatiale en même temps.

Exemple 6 Considérons la mission décrite par le scénario de l'exemple 5 (page 142). Pour réaliser cette mission, il faut que le robot se déplace entre O_{11}^{B1} et H_{21}^{B6} (figure 6.10, page 155). En d'autres termes, il faut appliquer l'action $go(O_{11}^{B1}, H_{21}^{B6})$.

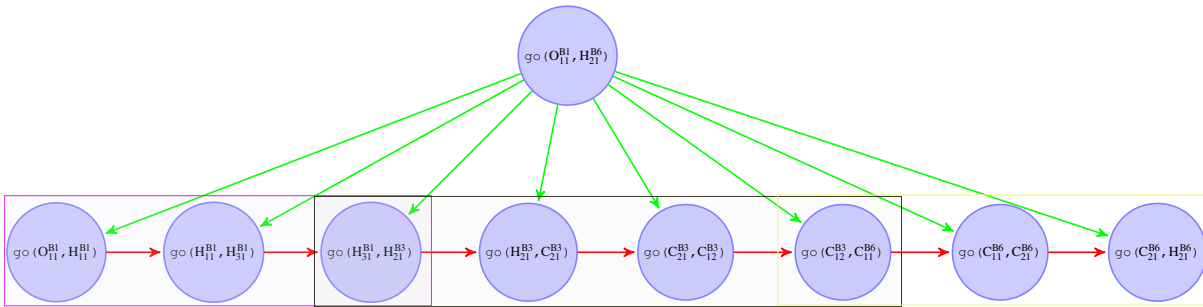


FIGURE 7.4 – Décomposition de l'action go .

La méthode $go(O_{11}^{B1}, H_{21}^{B6})$ est décomposée spatialement comme décrit dans la figure 7.4. Formellement, une méthode spatiale est définie en introduisant les éléments spatiaux d'un opérateur de planification spatiale (préconditions, effets et requêtes). De plus cette méthode est associée à un ensemble d'entiers représentant les niveaux hiérarchiques où elle peut être exécutée. L'idée est de décomposer l'action en fonction de chaque niveau hiérarchiques, où les conditions d'une action en niveau de l sont les effets de l'action au niveau précédent $l - 1$.

7.5 Performances de SPOON

Nous avons proposé le planificateur SPOON qui utilise un langage formel Spatial-PDDL, pour tenir compte de la composante spatiale. Dans ce paragraphe, nous montrons dans un premier temps l'impact de la connaissance spatiale (paragraphe 7.5.1), ensuite, nous comparons SPOON avec les planificateurs SGPlan et celui présenté dans [Guitton, 2008] (paragraphe 7.5.2).

7.5.1 Rôle de la connaissance spatiale dans la planification

La connaissance spatiale permet d'améliorer les performances du planificateur SPOON et la qualité de la solution. Nous considérons les différentes instanciations de *SpaceOntology* définies dans le tableau 7.1 (page 165) pour les expérimentations.

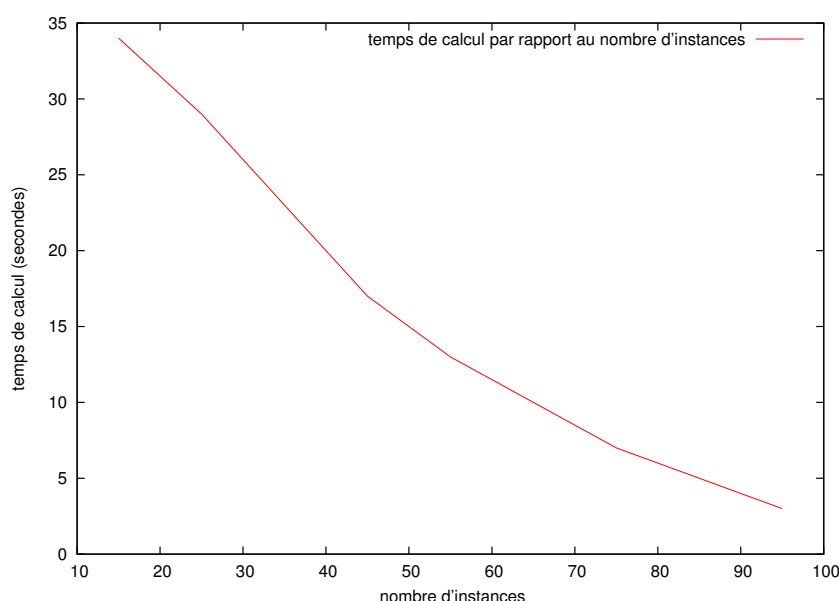


FIGURE 7.5 – Temps de calcul d'un plan par rapport aux nombres d'instances.

La figure 7.5 montre l'impact de la connaissance spatiale dans la génération d'un plan. Nous remarquons que plus la connaissance spatiale est disponible (ou nombre d'instance important), plus le temps de calcul d'un plan est rapide. Ceci s'explique par le fait qu'il y a moins d'inférence à faire pour déduire de la connaissance et de vérifier la cohérence et par conséquent un temps plus rapide pour la réponse aux requêtes.

Le temps de réponse aux requêtes a un impact sur le temps de résolution d'un problème. Pour une instanciation de 35 individus et nécessitant l'exécution de 6 règles, on a le tableau 7.4 qui donne le temps de calcul pour la résolution d'un problème en fonction de nombre de requêtes et le temps de réponse à celles-ci.

Requêtes		Résolution du problème
Nombre de requêtes	Temps de réponse	
[2-8]	< 1	< 1
[9-13]	[1-2.2]	[1 - 3.08]
[14-20]	[3.01-4.8]	[4.01 - 7]
[21-30]	[5.05-10.09]	[8.07 - 15.01]

TABLE 7.4 – Temps de calcul d’un plan solution en fonction des requêtes (en secondes).

7.5.2 Comparaison avec d’autres planificateurs

Pour comparer les planificateurs SGPlan et celui présenté dans [Guitton, 2008] avec SPOON, nous considérons les trois problèmes suivants.

Le premier est un benchmark de la compétition IPC²² : le problème de stockage²³ (*Storage*). Ce problème de planification implique le raisonnement spatial. L’idée du domaine est de déplacer un certain nombre de caisses de certains conteneurs à des dépôts par des palans. À l’intérieur d’un entrepôt, chaque palan peut se déplacer selon une carte spatiale reliant les différentes zones de l’entrepôt. Les problèmes tests pour ce domaine impliquent un nombre différent de dépôts, de palans, de caisses, des conteneurs et des zones de dépôt. Au début de chaque problème, une carte avec les différentes zones de dépôt est définie.

Le deuxième problème est décrit dans [Guitton, 2008], nous l’appelons *Rover*. La mission est la suivante : un robot mobile doit collecter un ensemble d’échantillons de roche dans l’environnement. Les échantillons de roche sont situés à des endroits précis de l’environnement, représentés par des objets. Le robot mobile n’a pas la connaissance sur la quantité d’échantillons disponible pour chaque objet. Il doit, préalablement à une collecte, effectuer une analyse du sol afin de connaître cette quantité. Ce robot dispose d’un espace de stockage limité. Lorsque l’espace de stockage devient insuffisant, il doit rentrer au dépôt central afin de décharger son stock. Si à la fin d’une collecte, il dispose de suffisamment de stockage libre, il peut effectuer un nouveau prélèvement avant de rentrer au dépôt central. L’objectif de la mission est que la quantité de roches prélevées et déchargées au dépôt central soit d’au moins 30 kilos.

Le troisième problème est le scénario que nous avons décrit dans l’exemple 4 (page 118). Nous nommons ce problème la mission du livre : *Book Mission*.

	<i>Storage</i>	<i>Rover</i>	<i>Book Mission</i>
SGPlan	< 1	-	-
planificateur [Guitton, 2008]	< 1	7.3	-
SPOON	< 1	6.09	7

TABLE 7.5 – Comparaison du temps de calcul (secondes).

22. <http://ipc.informatik.uni-freiburg.de/HomePage>

23. <http://zeus.ing.unibs.it/ipc-5/domain-descriptions/storage.txt>

Spatial-PDDL peut exprimer ces trois problèmes. À partir du tableau 7.5, nous remarquons que SPOON peut résoudre le problème *Storage* avec un temps équivalent aux planificateurs SGPLAN et planificateur [Guitton, 2008]. Cependant, seul SPOON peut résoudre le *Book Mission*. Le temps de calcul reste très acceptable 6.09 secondes, malgré le fait que l'introduction de la connaissance spatiale augmente le temps de calcul de la solution.

7.6 Système complet de la planification

Pour résoudre des problèmes de planification dans le cadre d'interaction homme-robot en tenant compte de la composante spatiale nous avons implémenté un simulateur avec des interfaces utilisateurs permettant de spécifier les données nécessaires pour la résolution.

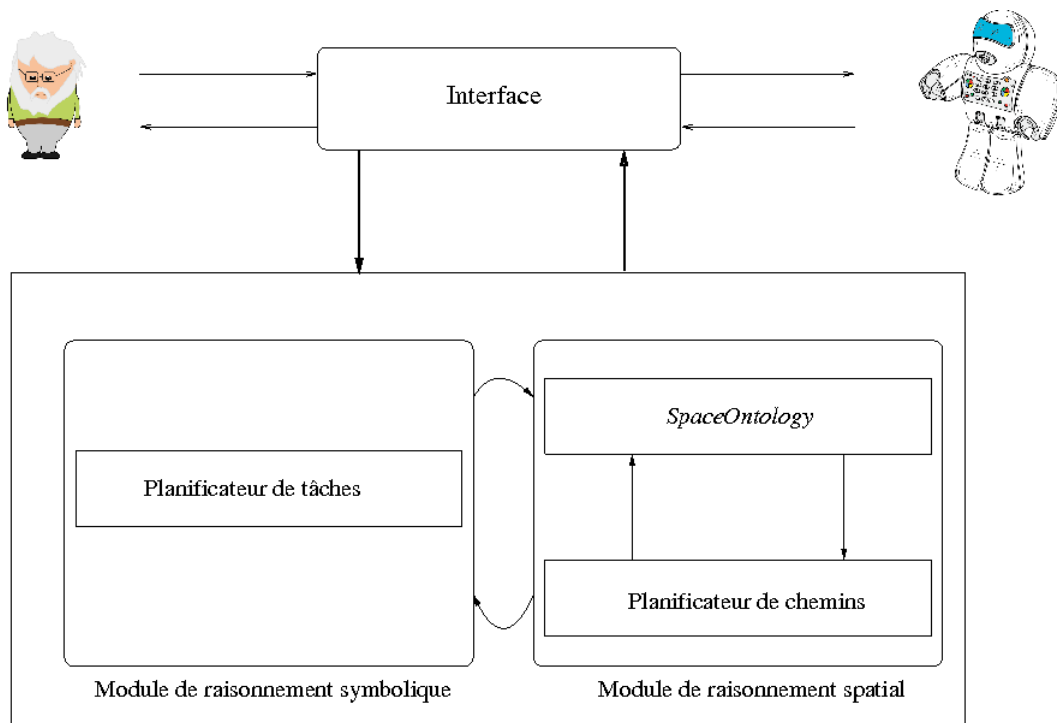
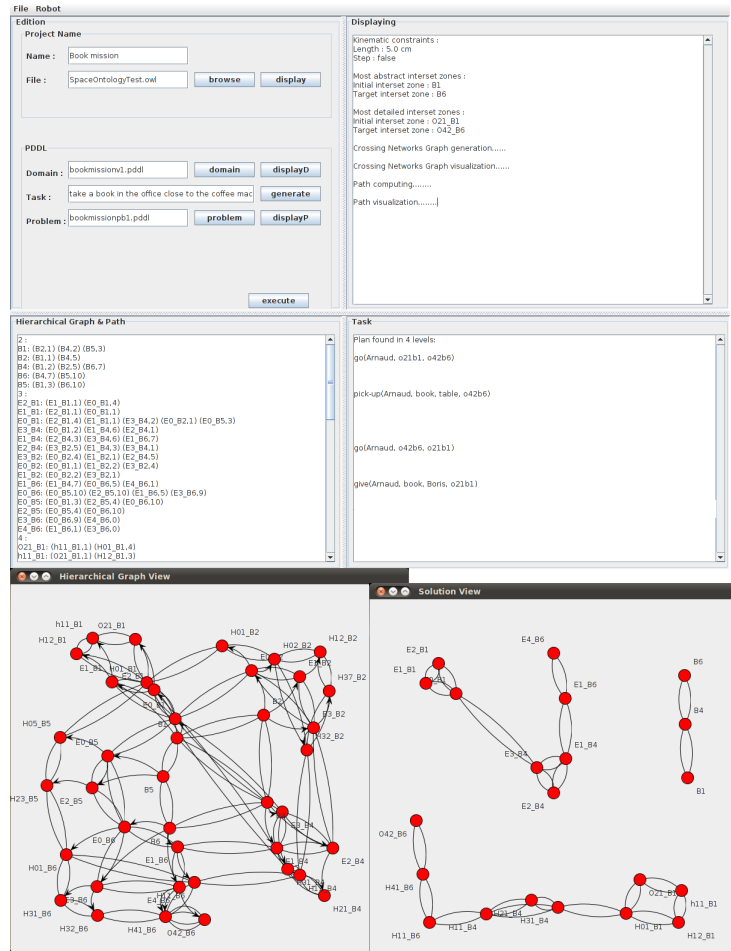


FIGURE 7.6 – Représentation graphique du système complet de la résolution de problèmes d'interaction homme-robot d'un point de vue spatial.

La figure 7.6 présente une schématisation de celui-ci. Il se compose de deux éléments : une interface principale et notre planificateur SPOON. Il procède en plusieurs étapes que nous résumons en :

1. formuler la communication entre les deux agents (par exemple, définition de la mission dans un langage formel compris par SPOON, description de l'environnement commun de manière à ce que les deux agents comprennent, communication du plan à exécuter, etc.) ;
2. activer le processus de planification (par exemple, définir la séquence d'actions, planifier une trajectoire, etc.).

L'interface n'intervient pas directement dans le processus de la planification. Elle l'initialise d'une manière automatique en formulant le problème de planification dans le langage Spatial-PDDL à partir de la demande définie par l'humain. Pour le module de raisonnement symbolique, l'interface fournit la description du domaine de planification décrit dans Spatial-PDDL. En effet, dans ce travail nous avons automatisé la génération d'un problème de planification spatiale à partir des demandes des agents et la connaissance déjà acquise. Le processus de planification terminé, l'interface fournit la solution finale sous forme d'un plan à l'agent qui doit l'exécuter. Ce plan englobe les actions à exécuter défini par le planificateur de tâches ainsi que les déplacements fournis par le module de raisonnement spatial pour la planification de chemins. Dans le cas où aucun plan n'est trouvé, l'interface communique aux agents qu'il n'est pas possible de réaliser cette mission.



via une interface. Cette interface permet la visualisation de la solution (le plan d'actions et les différents déplacements) permettant de résoudre le problème donnée en entrée.

Conclusion

Nous avons montré la capacité de SPOON à traiter des problèmes de planification classiques et des problèmes de planification tenant compte de la composante spatiale. Les performances de SPOON dépendent également du temps de réponse aux requêtes formulées au module du raisonnement spatial. Les performances de ce module ont un impact sur la performance globale de SPOON. Nous avons présenté une étude de complexité et quelques expérimentations pour décrire les performances du module de raisonnement spatial. Cependant, malgré l'introduction de la connaissance spatiale SPOON a des performances équivalentes aux planificateurs existants mais il peut traiter des problèmes que ces planificateurs ne sont pas capables de résoudre.

Conclusion et perspectives

Conclusion et perspectives

Le travail présenté dans cette thèse a eu pour premier objectif de contribuer à la mise en place de techniques permettant de faciliter l'interaction homme-robot. Nous nous sommes particulièrement intéressés à la prise en compte de l'information spatiale dans la planification d'une mission. Dans un premier temps, nous avons proposé une modélisation de l'information spatiale fondée sur les ontologies. Nous avons proposé *SpaceOntology* qui :

- gère les représentations qualitatives, quantitatives et hiérarchiques de la connaissance spatiale ;
- permet une description spatiale complexe, une gestion facile des différents aspects spatiaux, une structure facilement extensible et d'en déduire des informations spatiales ;
- fournit une connaissance structurée sur l'environnement à explorer dans le processus de planification ;
- définit un ensemble de règles permettant de faire un raisonnement spatial ;
- déduit de nouvelles informations pour compléter la description de l'environnement.

Ensuite, nous avons défini un planificateur hybride, appelé SPOON, qui combine un planificateur de tâches et un planificateur de chemins qui intègre *SpaceOntology*. Ces composants sont organisés en deux modules ; le module symbolique et le module spatial. Le module symbolique planifie une séquence d'actions. Le module spatial planifie un chemin, vérifie la satisfaction de toutes requêtes spatiales et s'assure que la connaissance spatiale reste cohérente.

Le processus de planification de SPOON basé sur un mécanisme de requêtes/réponses a nécessité la redéfinition de l'opérateur de planification classique. Nous avons proposé une nouvelle définition d'un opérateur de planification spatiale qui permet de modéliser différents types de connaissance symbolique et spatiale. La connaissance spatiale est considérée sous différents aspects, à savoir : qualitatif, quantitatif et hiérarchique. La prise en compte de la connaissance qualitative et en particulier floue, nous a conduit à proposer une méthode de planification de chemins entre positions floues. Cette méthode exploite l'organisation hiérarchique, les informations spatiales quantitatives et les informations spatiales qualitatives.

De plus, nous avons proposé une extension du langage PDDL, appelé Spatial-PDDL, pour décrire d'une manière formelle tout problème de planification intégrant la composante spatiale. Cette extension consiste à intégrer tous les éléments de la TBox dans PDDL. Les expérimentations présentées dans ce document ont permis de montrer qu'à l'aide de Spatial-PDDL, nous pouvons

exprimer des problèmes spatialement expressifs. De plus, l'intégration de *SpaceOntology* dans un planificateur permet d'améliorer l'efficacité du processus de planification.

Une première partie des perspectives de ce travail a pour objectif d'augmenter les performances de *SPOON*. En effet, de nombreux points restent encore à traiter à savoir ;

- intégrer des modules d'optimisation, de contrôle et de réparation de plans. Ceci implique une grande modification des algorithmes de planification classique. Des recherches sur ce thème seraient une continuation naturelle à cette thèse ;
- gérer les cas où la réponse à une requête floue reste ambiguë ;
- intégrer une classification des régions selon une sémantique, par exemple, cuisine, salle à manger, etc.

Par ailleurs, nous pensons que notre travail ouvre d'autres voies de recherche, plusieurs extensions peuvent être proposées. Nous envisageons plus particulièrement de nous intéresser à la définition d'un vocabulaire spatial plus ambiguë, à des problèmes faisant intervenir plusieurs agents, à l'intégration de certains prédicats symboliques ayant une signification spatiale dans *SpaceOntology* et pourquoi pas de combiner cette ontologie avec un modèle de planification probabiliste.

Actuellement, nous collaborons avec Frédéric Maris de l'IRIT autour de notre travail. Cette collaboration consiste à combiner la dimension spatiale et la dimension temporelle. En effet, un des défis majeurs pour la planification est de prendre en compte à la fois la dimension spatiale et la dimension temporelle. L'idée est de permettre la synchronisation d'actions non-instantanées dans l'espace. Pour cela, nous reprenons les règles de codage de TLP-GP [Maris, 2009] en intégrant de nouvelles règles qui utilisent *SpaceOntology*, et qui permettent de coder la dimension spatiale du problème de planification. Ce travail collaboratif a donné lieu à un article soumis à RFIA 2012. Dans ce travail nous avons montré comment intégrer deux approches originales de résolution de problèmes de planification que nous avons développées, l'une dans le cadre spatial et l'autre dans le cadre temporel. L'ontologie spatiale *SpaceOntology* nous permet d'obtenir une représentation spatiale optimale et simplifiée de l'état initial ainsi que de manipuler des distances floues entre entités spatiales. Par ailleurs, le principe de TLP-GP nous permet d'obtenir un système de planification temporelle capable de résoudre des problèmes qui nécessitent la concurrence des actions. Ce système qui utilise un solveur SMT et permet de bénéficier directement des améliorations apportées à ce type de solveur en termes de performances. Ainsi, nous avons proposé un nouveau codage SMT qui permet de compiler et résoudre des problèmes de planification spatio-temporelle pour lesquels toutes les solutions nécessitent la simultanéité des actions dans un espace multidimensionnel.

Annexes

Le détails de ces annexes est disponible à l'adresse : <https://belouaer.users.greyc.fr/>.

A - Planification de chemins

Cette annexe évoque différentes approches pour la planification de chemins :

- Graphe de visibilité ;
- Diagramme de Voronoï ;
- les méthodes probabilistes ;
- la planification par logique floue ;
- la méthode de la fenêtre dynamique ;
- la DKP : Deterministic Kinodynamic Planning.

B - Représentation et raisonnement spatial

Cette annexe présente les 169 relations de l'algèbre de rectangles et les tables de composition des relations d'Allen et celle des relations de RCC-8.

C - Langages et outils pour les ontologies

Nous passons en revue les principaux éléments du langage OWL. Nous présentons aussi un tableau comparatif des principaux moteurs d'inférence [Fournier-Viger, 2005] : Fact [Horrocks, 1998], Racer [Haarslev et Möller, 2002], Pellet [Sirin *et al.*, 2007], Fact++ [Tsarkov et Horrocks, 2004] et F-OWL [Zou *et al.*, 2005], et nous détaillons Pellet et Racer.

D - Raisonnement spatial

Dans cette annexe, nous présentons tout d'abord les tables de composition pour l'algèbre ABLR.

Nous présentons la structure de *SpaceOntology*. Nous présentons aussi deux extraits instantiant *SpaceOntology* :

- le premier extrait décrit l'environnement présenté dans l'exemple 3 (page 90) ;
- le deuxième extrait décrit l'environnement présenté dans l'exemple 5 (page 142). Nous ne donnons pas toute la définition, mais seulement les différentes régions et quelques relations.

E - Syntaxe de Spatial-PDDL

Cette annexe présente la syntaxe de Spatial-PDDL en insistant sur les extensions.

Bibliographie

- [Allen, 1983] ALLEN, J. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.
- [Angeli et al., 2008] ANGELI, A., DONCIEUX, S., MEYER, J. et FILLIAT, D. (2008). Incremental vision-based topological slam. pages 1031–1036.
- [Arras, 2003] ARRAS, K. (2003). *Feature-based robot navigation in known and unknown environments*. Thèse de doctorat, Citeseer.
- [Baader, 2003] BAADER, F. (2003). *The description logic handbook : theory, implementation, and applications*. Cambridge Univ Pr.
- [Baba, 2007] BABA, A. (2007). Cartographie de l’environnement et suivi simultané de cibles dynamiques par un robot mobile.
- [Bachimont, 2000] BACHIMONT, B. (2000). Engagement sémantique et engagement ontologique : conception et réalisation d’ontologies en ingénierie des connaissances. *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, pages 305–323.
- [Balbiani et al., 1998] BALBIANI, P., CONDOTTA, J. et del CERRO, L. (1998). A model for reasoning about bidimensional temporal relations. *way*, 10:1.
- [Baltié et al., 2008] BALTIÉ, J., BENSANA, E., FABIANI, P., FARGES, J., MILLET, S., MORIGNOT, P., PATIN, B., PETITJEAN, G., PITOIS, G. et PONCET, J. (2008). Multi-vehicle missions : architecture and algorithms for distributed online planning.
- [Barraquand et Latombe, 1991] BARRAQUAND, J. et LATOMBE, J. (1991). Robot motion planning : A distributed representation approach. *The International Journal of Robotics Research*, 10(6):628.
- [Bateman et Farrar, 2004] BATEMAN, J. et FARRAR, S. (2004). Towards a generic foundation for spatial ontology. In *Formal ontology in information systems : proceedings of the third conference (FOIS-2004)*, page 237. Ios Pr Inc.
- [Bateman et Farrar, 2005] BATEMAN, J. et FARRAR, S. (2005). Modelling models of robot navigation using formal spatial ontology. *Spatial Cognition IV. Reasoning, Action, and Interaction*, pages 366–389.
- [Belouaer et al., 2010a] BELOUAER, L., BOUZID, M. et MOUADDIB, A.-I. (2010a). Ontology based spatial planning for human-robot interaction. pages 103–110.

- [Belouaer *et al.*, 2010b] BELOUAER, L., BOUZID, M. et MOUADDIB, A.-I. (2010b). A spatial ontology for human-robot interaction. pages 154–159.
- [Belouaer *et al.*, 2011] BELOUAER, L., BOUZID, M. et MOUADDIB, A.-I. (2011). Spatial knowledge in planning language. *International Conference on Knowledge Engineering and Ontology Development*.
- [Bessière, 1994] BESSIÈRE, C. (1994). Arc-consistency and arc-consistency again* 1. *Artificial intelligence*, 65(1):179–190.
- [Bessière *et al.*, 1999] BESSIÈRE, C., FREUDER, E. et RÉGIN, J. (1999). Using constraint metaknowledge to reduce arc consistency computation. *Artificial Intelligence*, 107(1):125–148.
- [Bessière et Régin, 2001] BESSIÈRE, C. et RÉGIN, J. (2001). Refining the basic constraint propagation algorithm. 17(1):309–315.
- [Bessière *et al.*, 2005] BESSIÈRE, C., RÉGIN, J., YAP, R. et ZHANG, Y. (2005). An optimal coarse-grained arc consistency algorithm. *Artificial Intelligence*, 165(2):165–185.
- [Bloch, 1996] BLOCH, I. (1996). Fuzzy relative position between objects in images : a morphological approach. 1:987–990.
- [Bloch, 1999] BLOCH, I. (1999). Fuzzy relative position between objects in image processing : a morphological approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(7):657–664.
- [Bloch, 2002] BLOCH, I. (2002). Fuzzy relative position between objects in image processing : a morphological approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(7):657–664.
- [Blum et Furst, 1997] BLUM, A. et FURST, M. (1997). Fast planning through planning graph analysis* 1. *Artificial intelligence*, 90(1-2):281–300.
- [Bobillo et Straccia, 2008] BOBILLO, F. et STRACCIA, U. (2008). fuzzydl : An expressive fuzzy description logic reasoner. pages 923–930.
- [Bonet et Geffner, 1998] BONET, B. et GEFNER, H. (1998). Hsp : Heuristic search planner.
- [Bonet et Geffner, 2000] BONET, B. et GEFNER, H. (2000). Planning as heuristic search : New results. *Recent Advances in AI Planning*, pages 360–372.
- [Bonet et Geffner, 2001] BONET, B. et GEFNER, H. (2001). Planning as heuristic search. *Artificial Intelligence*, 129(1-2):5–33.
- [Bonet *et al.*, 1997] BONET, B., LOERINCS, G. et GEFNER, H. (1997). A robust and fast action selection mechanism for planning. pages 714–719.
- [Borgida, 1996] BORGIDA, A. (1996). On the relative expressiveness of description logics and predicate logics* 1. *Artificial intelligence*, 82(1-2):353–367.
- [Bouchon-Meunier *et al.*, 1996] BOUCHON-MEUNIER, B., RIFQI, M. et BOTHOREL, S. (1996). Towards general measures of comparison of objects. *Fuzzy sets and systems*, 84(2):143–153.

-
- [Bouguerra *et al.*, 2008] BOUGUERRA, A., KARLSSON, L. et SAFFIOTTI, A. (2008). Monitoring the execution of robot plans using semantic knowledge. *Robotics and Autonomous Systems*.
- [Bourigault *et al.*, 2004] BOURIGAULT, D., AUSSENAC-GILLES, N. et CHARLET, J. (2004). Construction de ressources terminologiques ou ontologiques à partir de textes : un cadre unificateur pour trois études de cas. *Revue d'Intelligence Artificielle*, 18(4):24.
- [Burggraf, 2006] BURGGRAF, D. (2006). Geography markup language. *Data Science Journal*, 5(0):178–204.
- [Car et Frank, 1994] CAR, A. et FRANK, A. (1994). General principles of hierarchical spatial reasoning : The case of wayfinding. 2:646–664.
- [Casati *et al.*, 1998] CASATI, R., SMITH, B. et VARZI, A. (1998). Ontological tools for geographic representation. In *Formal ontology in information systems*, pages 77–85. Citeseer.
- [Casati et Varzi, 1999] CASATI, R. et VARZI, A. (1999). *Parts and places : the structures of spatial representation*. The MIT Press.
- [Chaignaud et Lévy, 2001] CHAIGNAUD, N. et LÉVY, F. (2001). Planification et révision. *Révision, cognition et interaction*, page 61.
- [Chanthery, 2005] CHANTHERY, E. (2005). Planification de mission pour un véhicule aérien autonome.
- [Charlet *et al.*, 2003] CHARLET, J., BACHIMONT, B. et TRONCY, R. (2003). Ontologies pour le web sémantique. *Action spécifique*, 32:43–63.
- [Chmeiss, 1996] CHMEISS, A. (1996). *Réseaux de contraintes : algorithmes de propagation et de décomposition*. Thèse de doctorat.
- [Choi et Amir, 2009] CHOI, J. et AMIR, E. (2009). Combining planning and motion planning. pages 238–244.
- [Clementini et Di Felice, 1995a] CLEMENTINI, E. et DI FELICE, P. (1995a). A comparison of methods for representing topological relationships. *Information Sciences-Applications*, 3(3): 149–178.
- [Clementini et Di Felice, 1995b] CLEMENTINI, E. et DI FELICE, P. (1995b). A comparison of methods for representing topological relationships. *Information Sciences-Applications*, 3(3): 149–178.
- [Clementini *et al.*, 1993] CLEMENTINI, E., DI FELICE, P. et van OOSTEROM, P. (1993). A small set of formal topological relationships suitable for end-user interaction. In *Advances in Spatial Databases*, pages 277–295. Springer.
- [Cohn *et al.*, 1997] COHN, A., BENNETT, B., GOODAY, J. et GOTTS, N. (1997). RCC : a calculus for region based qualitative spatial reasoning. *GeoInformatica*, 1(3):275–316.
- [Craik, 1967] CRAIK, K. (1967). The nature of explanation.
- [Cullot *et al.*,] CULLOT, N., PARENT, C. et VANGENOT, C. On spatial ontologies.

- [da Costa *et al.*, 2008] da COSTA, P., LASKEY, K. et LASKEY, K. (2008). Pr-owl : A bayesian ontology language for the semantic web. 5327:88.
- [Dasiopoulou *et al.*, 2005] DASIOPOULOU, S., MEZARIS, V., KOMPATSIARIS, I., PAPASTATHIS, V. et STRINTZIS, M. (2005). Knowledge-assisted semantic video object detection. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(10):1210–1224.
- [Davis, 1986] DAVIS, E. (1986). *Representing and acquiring geographic knowledge*.
- [Dehak, 2002] DEHAK, S. (2002). Inférence quantitative des relations spatiales directionnelles.
- [Dib *et al.*, 2010] DIB, M., ABDALLAH, R. et CAMINADA, A. (2010). Arc-consistency in constraint satisfaction problems : A survey. pages 28–30.
- [Dijkstra, 1959] DIJKSTRA, E. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.
- [Dominey *et al.*, 2007] DOMINEY, P., MALLET, A. et YOSHIDA, E. (2007). Progress in programming the hrp-2 humanoid using spoken language.
- [Donnelly *et al.*, 2006] DONNELLY, M., BITTNER, T. et ROSSE, C. (2006). A formal theory for spatial representation and reasoning in biomedical ontologies. *Artificial Intelligence in Medicine*, 36(1):1–27.
- [Dou, 2008] DOU, D. (2008). The Formal Syntax and Semantics of Web-PDDL.
- [Duckett et Saffiotti, 2000] DUCKETT, T. et SAFFIOTTI, A. (2000). Building globally consistent gridmaps from topologies.
- [Dufourd, 2005] DUFOURD, D. (2005). Des cartes combinatoires pour la construction automatique de modèles d’environnement par un robot mobile.
- [Dylla *et al.*, 2006] DYLLA, F., FROMMBERGER, L., WALLGRUN, J. et WOLTER, D. (2006). Sparq : A toolbox for qualitative spatial representation and reasoning. In *Workshop at KI 2006*, page 79. Citeseer.
- [Egenhofer et Franzosa, 1991] EGENHOFER, M. et FRANZOSA, R. (1991). Point-set topological spatial relations. *International Journal of Geographical Information Science*, 5(2):161–174.
- [Egenhofer et Herring, 1994] EGENHOFER, M. et HERRING, J. (1994). Categorizing binary topological relations between regions, lines, and points in geographic databases. *The*, 9:94–1.
- [Egenhofer *et al.*, 1995] EGENHOFER, M., MARK, D., FRANK, A. et KUHN, W. (1995). Spatial information theory : A theoretical basis for GIS. In *International Conference COSIT*, volume 95, pages 1–15.
- [Egenhofer *et al.*, 1993] EGENHOFER, M., SHARMA, J. et MARK, D. (1993). A critical comparison of the 4-intersection and 9-intersection models for spatial relations : formal analysis. In *AUTOCARTO-CONFERENCE-*, pages 1–1. ASPRS AMERICAN SOCIETY FOR PHOTOGRAMMETRY AND.
- [Erol, 1996] EROL, K. (1996). Hierarchical task network planning : formalization, analysis, and implementation.

-
- [Erol *et al.*, 1994] EROL, K., HENDLER, J. et NAU, D. (1994). Umcp : A sound and complete procedure for hierarchical task-network planning. *In Proc. 2nd Intl. Conf. on AI Planning Systems*, pages 249–254. Citeseer.
- [Fabrizi et Saffiotti, 2002] FABRIZI, E. et SAFFIOTTI, A. (2002). Augmenting topology-based maps with geometric information. *Robotics and Autonomous Systems*, 40(2-3):91–97.
- [Falasconi *et al.*, 1997] FALASCONI, S., LANZOLA, G. et STEFANELLI, M. (1997). Ontology and terminology servers in agent-based health-care information systems. *Methods of information in medicine*, 36(1):30–43.
- [Fernandez et Gonzalez, 2001] FERNANDEZ, J. et GONZALEZ, J. (2001). *Multi-Hierarchical Representation of Large-Scale Space : Applications to Mobile Robots*. Kluwer Academic Publishers.
- [Fernandez-Madrigal, 2004] FERNANDEZ-MADRIGAL, J. (2004). Assistive navigation of a robotic wheelchair using a multihierarchical model of the environment. *Integrated Computer-Aided Engineering*, 11(4):309–322.
- [Fikes et Nilsson, 1971] FIKES, R. et NILSSON, N. (1971). STRIPS : A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208.
- [Fournier-Viger, 2005] FOURNIER-VIGER, P. (2005). *Un modèle de représentation des connaissances à trois niveaux de sémantique pour les systèmes tutoriels intelligents*. Mémoire de maîtrise (M. Sc.), Université de Sherbrooke, Sherbrooke, Canada.
- [Fox et Long, 2003] FOX, M. et LONG, D. (2003). PDDL2. 1 : An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, 20(1):61–124.
- [Frank, 1992] FRANK, A. (1992). Qualitative spatial reasoning about distances and directions in geographic space. *Journal of Visual Languages & Computing*, 3(4):343–371.
- [Frommberger et Wolter, 2008] FROMMBERGER, L. et WOLTER, D. (2008). Spatial abstraction : Aspectualization, coarsening, and conceptual classification. *Spatial Cognition VI. Learning, Reasoning, and Talking about Space*, pages 311–327.
- [Galindo *et al.*, 2008] GALINDO, C., FERNÁNDEZ-MADRIGAL, J., GONZÁLEZ, J. et SAFFIOTTI, A. (2008). Robot task planning using semantic maps. *Robotics and Autonomous Systems*, 56(11):955–966.
- [Galindo *et al.*, 2005] GALINDO, C., SAFFIOTTI, A., CORADESCHI, S., BUSCHKA, P., FERNANDEZ-MADRIGAL, J. et GONZALEZ, J. (2005). Multi-hierarchical semantic maps for mobile robotics. *In Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2278–2283.
- [Gandon, 2002] GANDON, F. (2002). Ontology engineering : A survey and a return on experience.
- [Ghallab et Laruelle, 1994] GHALLAB, M. et LARUELLE, H. (1994). Representation and control in ixtet, a temporal planner. 94:61–67.

- [Ghallab *et al.*, 2004] GHALLAB, M., NAU, D. et TRAVERSO, P. (2004). *Automated Planning : theory and practice*. Morgan Kaufmann Publishers.
- [Gilbert, 2007] GILBERT, L. (2007). Utilisation d'un sig et de methodes quantitatives pour l'analyse regionale de schémes d'établissement. *BAR INTERNATIONAL SERIES*, 1700:39.
- [Gisler *et al.*, 2011] GISLER, S., LINIGER, H. et PRASUHN, V. (2011). Carte à haute résolution du risque d'érosion au raster 2×2 m (cre2). *Recherche agronomique suisse*, 2(4):148–155.
- [Goodwin *et al.*, 2008] GOODWIN, J., DOLBEAR, C. et HART, G. (2008). Geographical linked data : The administrative geography of great britain on the semantic web. *Transactions in GIS*, 12:19–30.
- [Goulette, 1999] GOULETTE, J. (1999). Sémantique formelle de l'espace. Une application au raisonnement spatial qualitatif en architecture1. *Intellectica*, 2(29):9–34.
- [Gravot *et al.*, 2005] GRAVOT, F., CAMBON, S. et ALAMI, R. (2005). asymov : a planner that deals with intricate symbolic and geometric problems. *Robotics Research*, pages 100–110.
- [Green *et al.*, 1969] GREEN, C., SCIENCE, S. I. A. I. C. C. et DIVISION, T. (1969). *Application of theorem proving to problem solving*. Citeseer.
- [Green, 1987] GREEN, S. (1987). Spaces-a system for the representation of common sense knowledge about space for design. pages 195–206.
- [Grenona et Smitha, 2004] GRENONA, P. et SMITHA, B. (2004). Snap and span : Towards dynamic spatial ontology. *Event-Oriented Approaches in Geographic Information Science : A Special Issue of Spatial Cognition and Computation*, 4(1):69–104.
- [Gribble *et al.*, 1998] GRIBBLE, W., BROWNING, R., HEWETT, M., REMOLINA, E. et KUIPERS, B. (1998). Integrating vision and spatial reasoning for assistive navigation. *Assistive Technology and artificial intelligence*, pages 179–193.
- [Guere et Alami, 2001] GUERE, E. et ALAMI, R. (2001). Let's reduce the gap between task planning and motion planning. 1:15–20.
- [Guitton, 2008] GUITTON, J. (2008). PLANIFICATION D'ACTION ET DE DÉPLACEMENT POUR LA ROBOTIQUE MOBILE.
- [Güting, 1994] GÜTING, R. (1994). An introduction to spatial database systems. *The VLDB Journal*, 3(4):357–399.
- [Haarslev et Möller, 2002] HAARSLEV, V. et MÖLLER, R. (2002). Racer user's guide and reference manual : Version 1.7. 6.
- [Hall, 1966] HALL, E. (1966). The hidden dimension .
- [Hart *et al.*, 1968] HART, P., NILSSON, N. et RAPHAEL, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100–107.
- [Helmert, 2006] HELMERT, M. (2006). The fast downward planning system. *Journal of Artificial Intelligence Research*, 26(1):191–246.

-
- [Hernández, 1994] HERNÁNDEZ, D. (1994). *Qualitative Representation of Spatial Knowledge*. Springer.
- [Hioual et Boufaïda, 2010] HIOUAL, O. et BOUFAÏDA, Z. (2010). Vers une architecture à base d’agents pour une composition sémantique et dynamique des services web dans un contexte d’ebxml. pages 10–12.
- [Hoffmann *et al.*, 2006] HOFFMANN, J., EDELKAMP, S., THIÉBAUX, S., ENGLERT, R., dos SANTOS LIPORACE, F. et TRUG, S. (2006). Engineering benchmarks for planning : the domains used in the deterministic part of ipc-4. *Journal of Artificial Intelligence Research*, 26(1):453–541.
- [Hoffmann et Nebel, 2001] HOFFMANN, J. et NEBEL, B. (2001). The ff planning system : Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14(1): 253–302.
- [Horrocks, 1998] HORROCKS, I. (1998). The fact system. *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 307–312.
- [Horrocks *et al.*, 2004] HORROCKS, I., PATEL-SCHNEIDER, P., BOLEY, H., TABET, S., GROSOFF, B. et DEAN, M. (2004). Swrl : A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21:79.
- [Horrocks *et al.*, 2003] HORROCKS, I., PATEL-SCHNEIDER, P. et VAN HARMELEN, F. (2003). From shiq and rdf to owl : The making of a web ontology language. *Web semantics : science, services and agents on the World Wide Web*, 1(1):7–26.
- [Hudelot *et al.*, 2008] HUDELLOT, C., ATIF, J. et BLOCH, I. (2008). Fuzzy spatial relation ontology for image interpretation. *Fuzzy Sets and Systems*, 159(15):1929–1951.
- [Ishikawa et Montello, 2006] ISHIKAWA, T. et MONTELLO, D. (2006). Spatial knowledge acquisition from direct experience in the environment : Individual differences in the development of metric knowledge and the integration of separately learned places. *Cognitive Psychology*, 52(2):93–129.
- [Jacas et Recasens, 2000] JACAS, J. et RECASENS, J. (2000). One-dimensional indistinguishability operators. *Fuzzy sets and systems*, 109(3):447–451.
- [Kambhampati *et al.*, 1993] KAMBHAMPATI, S., CUTKOSKY, M., TENENBAUM, J. et LEE, S. (1993). Integrating general purpose planners and specialized reasoners : case study of a hybrid planning architecture. *Systems, Man and Cybernetics, IEEE Transactions on*, 23(6):1503–1518.
- [Kambhampati *et al.*, 1995] KAMBHAMPATI, S., KNOBLOCK, C. et YANG, Q. (1995). Planning as refinement search : A unified framework for evaluating design tradeoffs in partial-order planning. *Artificial Intelligence*, 76(1-2):167–238.
- [Kautz et Selman, 1996] KAUTZ, H. et SELMAN, B. (1996). Pushing the envelope : Planning, propositional logic, and stochastic search. *In Proceedings of the National Conference on Artificial Intelligence*, pages 1194–1201.

- [Keller et Wang, 1996] KELLER, J. et WANG, X. (1996). Learning spatial relationships in computer vision. *In Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, volume 1, pages 118–124. IEEE.
- [Kennedy *et al.*, 2007] KENNEDY, W., BUGAJSKA, M., MARGE, M., ADAMS, W., FRANSEN, B., PERZANOWSKI, D., SCHULTZ, A. et TRAFTON, J. (2007). Spatial Representation and Reasoning for Human-Robot Collaboration. *In PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 22, page 1554. Menlo Park, CA ; Cambridge, MA ; London ; AAAI Press ; MIT Press ; 1999.
- [Khatib, 1986] KHATIB, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90.
- [Klien et Lutz, 2005] KLIEN, E. et LUTZ, M. (2005). The role of spatial relations in automating the semantic annotation of geodata. *Spatial Information Theory*, pages 133–148.
- [Knoblock, 1992] KNOBLOCK, C. (1992). An analysis of abstrips. *In Artificial intelligence planning systems : proceedings of the first international conference, June 15-17, 1992, College Park, Maryland*, page 126. Morgan Kaufmann Pub.
- [Koczy, 1988] KOCZY, L. (1988). On the description of relative position of fuzzy patterns. *Pattern Recognition Letters*, 8(1):21–28.
- [Kortenkamp, 1993] KORTENKAMP, D. (1993). Cognitive maps for mobile robots : a representation for mapping and navigation.
- [Krishnapuram *et al.*, 1993] KRISHNAPURAM, R., KELLER, J. et MA, Y. (1993). Quantitative analysis of properties and spatial relations of fuzzy image regions. *Fuzzy Systems, IEEE Transactions on*, 1(3):222–233.
- [Kruijff *et al.*, 2007] KRUIJFF, G., ZENDER, H., JENSFELT, P. et CHRISTENSEN, H. (2007). Situated dialogue and spatial organization : What, where... and why. *International Journal of Advanced Robotic Systems*, 4(1):125–138.
- [Kuipers, 2000] KUIPERS, B. (2000). The Spatial Semantic Hierarchy* 1. *Artificial Intelligence*, 119(1-2):191–233.
- [Kuipers et Levitt, 1988] KUIPERS, B. et LEVITT, T. (1988). Navigation and mapping in large scale space. *AI magazine*, 9(2):25.
- [Kurata, 2009] KURATA, Y. (2009). 9-Intersection Calculi for Spatial Reasoning on the Topological Relations between Multi-Domain Objects.
- [Laborie *et al.*, 2006] LABORIE, S., EUZENAT, J. et LAYAÏDA, N. (2006). Adaptation spatiale efficace de documents SMIL. *In PATRICK BOUTHEMY, Y. D., éditeur : 15e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA)*.
- [Lamare et Ghallab, 1998] LAMARE, B. et GHALLAB, M. (1998). Integrating a temporal planner with a path planner for a mobile robot. pages 144–151.

-
- [Lautenschütz *et al.*, 2007] LAUTENSCHÜTZ, A., DAVIES, C., RAUBAL, M., SCHWERING, A. et PEDERSON, E. (2007). The influence of scale, context and spatial preposition in linguistic topology. *Spatial Cognition V Reasoning, Action, Interaction*, pages 439–452.
- [LaValle, 2006] LAVALLE, S. (2006). *Planning algorithms*. Cambridge Univ Pr.
- [Lesbegueries et Loustau, 2006] LESBEGUERIES, J. et LOUSTAU, P. (2006). Structuration d’information spatiale qualitative pour la recherche d’information. *Relation*, 1(1).
- [Levelt, 1982] LEVELT, W. (1982). Cognitive styles in the use of spatial direction terms. *RJ Jarvella; W. Klein (eds.), Speech, place, and action : studies of deixis and related topics*, pages 251–268.
- [Levinson, 1996] LEVINSON, S. (1996). *Frames of reference and Molyneux’s question : Crosslinguistic evidence*.
- [Ligozat, 1998] LIGOZAT, G. (1998). ESPACE ET TEMPS : Un nouveau domaine en Sciences Cognitives ?
- [Likitvivatanavong *et al.*, 2004] LIKITVIVATANAVONG, C., ZHANG, Y., BOWEN, J. et FREUDER, E. (2004). Arc consistency in mac : a new perspective. *Proceedings of CPAI*, 4:93–107.
- [Lindley *et al.*, 1994] LINDLEY, C., KUMAR, V., IRRGANG, V. R. et ROBERTSON, J. (1994). An evaluation of information retrieval methods and semantic network processing for automatic link generation in hypermedia systems. *Proceedings of the Second International Interactive Multimedia Symposium*, page [http ://www.aset.org.au/confs/iims/1994/km/lindley.html](http://www.aset.org.au/confs/iims/1994/km/lindley.html).
- [Liu *et al.*, 2010] LIU, C., WALKER, J. et CHAI, J. (2010). Ambiguities in Spatial Language Understanding in Situated Human Robot Dialogue. *In 2010 AAAI Fall Symposium Series*.
- [Longley, 2005] LONGLEY, P. (2005). Geographic information systems and science.
- [Lydia Kraus et Mijovi, 2011] LYDIA KRAUS, Mladen Stanojevic, N. T. et MIJOVI, V. (2011). A decision support system for building evacuation based on the emili site environment. *WE-TICE*.
- [Macario et Medeiros, 2009] MACARIO, C. et MEDEIROS, C. (2009). A framework for semantic annotation of geospatial data for agriculture. *International Journal of Metadata, Semantics and Ontologies*, 4(1):118–132.
- [MacGregor, 1999] MACGREGOR, R. (1999). Retrospective on loom. *Information Sciences Institute, University of Southern California, Tech. Rep*.
- [Mackworth, 1977] MACKWORTH, A. (1977). Consistency in networks of relations. *Artificial intelligence*, 8(1):99–118.
- [Mackworth et Freuder, 1985] MACKWORTH, A. et FREUDER, E. (1985). The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *Artificial intelligence*, 25(1):65–74.
- [Manola *et al.*, 2004] MANOLA, F., MILLER, E. et MCBRIDE, B. (2004). RDF primer. W3C recommendation. *World Wide Web Consortium*.

- [Maris, 2009] MARIS, F. (2009). Planification SAT et Planification Temporellement Expressive. Les Systèmes TSP et TLP-GP.
- [Mark et Egenhofer, 1994] MARK, D. et EGENHOFER, M. (1994). Modeling spatial relations between lines and regions : combining formal mathematical models and human subjects testing. *Cartography and Geographic Information Systems*, 21(4):195–212.
- [Martin, 2007] MARTIN, H. (2007). Relations spatiales qualitatives dans les ontologies géographiques avec ONTOAST.
- [Masolo *et al.*,] MASOLO, C., BORGIO, S., GANGEMI, A., GUARINO, N., OLTRAMARI, A. et SCHNEIDER, L. Dolce : a descriptive ontology for linguistic and cognitive engineering. *WonderWeb Project, Deliverable D*, 17.
- [McDermott *et al.*, 1998] MCDERMOTT, D., GHALLAB, M., HOWE, A., KNOBLOCK, C., RAM, A., VELOSO, M., WELD, D. et WILKINS, D. (1998). PDDL-the planning domain definition language. *The AIPS-98 Planning Competition Comitee*.
- [McGuinness *et al.*, 2004] MCGUINNESS, D., VAN HARMELEN, F. *et al.* (2004). OWL web ontology language overview. *W3C recommendation*, 10:2004–03.
- [Meger *et al.*, 2008] MEGER, D., FORSSÉN, P., LAI, K., HELMER, S., MCCANN, S., SOUTHEY, T., BAUMANN, M., LITTLE, J., LOWE, D. et GEORGE, C. (2008). An attentive semantic robot. *Robotics and Autonomous Systems*, 56(6):503–511.
- [Minton *et al.*, 1994] MINTON, S., BRESINA, J. et DRUMMOND, M. (1994). Total-order and partial-order planning : A comparative analysis. *Arxiv preprint cs/9412103*.
- [MIRON, 2009] MIRON, A. D. (2009). *Decouverte d’associations sémantiques pour le Web Sémantique Geospatial : le framework ONTOAST*. Thèse de doctorat.
- [Mohr et Henderson, 1986] MOHR, R. et HENDERSON, T. (1986). Arc and path consistency revisited* 1. *Artificial Intelligence*, 28(2):225–233.
- [Moratz *et al.*, 2003] MORATZ, R., TENBRINK, T., BATEMAN, J. et FISCHER, K. (2003). Spatial knowledge representation for human-robot interaction. *Spatial cognition III*.
- [Motik *et al.*, 2009] MOTIK, B., PATEL-SCHNEIDER, P., PARSIA, B., BOCK, C., FOKOUE, A., HAASE, P., HOEKSTRA, R., HORROCKS, I., RUTTENBERG, A., SATTLER, U. *et al.* (2009). OWL 2 web ontology language : Structural specification and functional-style syntax. *W3C Recommendation*, 27.
- [Nardi *et al.*, 2003] NARDI, D., BRACHMAN, R. *et al.* (2003). An introduction to description logics. *The description logic handbook : theory, implementation, and applications*, pages 1–40.
- [Nau *et al.*, 2003] NAU, D., AU, T., ILGHAMI, O., KUTER, U., MURDOCK, J., WU, D. et YAMAN, F. (2003). Shop2 : An htn planning system. *Journal of Artificial Intelligence Research*, 20(1):379–404.
- [Nau *et al.*, 1999] NAU, D., CAO, Y., LOTEM, A. et MUÑOZ-AVILA, H. (1999). Shop : Simple hierarchical ordered planner. *In Proceedings of the 16th international joint conference on Artificial intelligence-Volume 2*, pages 968–973. Morgan Kaufmann Publishers Inc.

-
- [Nebel, 1997] NEBEL, B. (1997). *Solving hard qualitative temporal reasoning problems : Evaluating the efficiency of using the ORD-Horn class*, volume 1. Springer.
- [Nieto et al., 2004] NIETO, J., GUIVANT, J. et NEBOT, E. (2004). The hybrid metric maps (hymms) : A novel map representation for denseslam. 1:391–396.
- [Nüchter et Hertzberg, 2008] NÜCHTER, A. et HERTZBERG, J. (2008). Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926.
- [Park et al., 2009] PARK, S., PARK, M. et PARK, S. (2009). Object-spatial layout-route based hybrid map and global localization for mobile robots. *International Journal of Control, Automation and Systems*, 7(4):598–614.
- [Pease et al., 2002] PEASE, A., NILES, I. et LI, J. (2002). The suggested upper merged ontology : A large ontology for the semantic web and its applications. In *Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web*, volume 28. Ed2 monton, Canada.
- [Pednault, 1994] PEDNAULT, E. (1994). ADL and the state-transition model of action. *Journal of Logic and Computation*, 4(5):467.
- [Pigot, 1991] PIGOT, S. (1991). Topological models for 3d spatial information systems. In *AUTOCARTO-CONFERENCE-*, volume 6, pages 368–368. ASPRS AMERICAN SOCIETY FOR PHOTOGRAMMETRY AND.
- [Prud’Hommeaux et al., 2006] PRUD’HOMMEAUX, E., SEABORNE, A. et al. (2006). Sparql query language for rdf. *W3C working draft*, 4.
- [Przytula-Machrouh et al., 2004] PRZYTULA-MACHROUH, E., LIGOZAT, G. et DENIS, M. (2004). Vers des ontologies transmodales pour la description d’itinéraires. *Revue Internationale de Géomatique : n Spécial sur les ontologies spatiales*, 14(2):285–302.
- [Pullar et Egenhofer, 1988] PULLAR, D. et EGENHOFER, M. (1988). Towards formal definitions of spatial relationships among spatial objects. In *Proceedings of the 3 rd International Symposium on Spatial Data Handling*, pages 225–242.
- [Pumain et al., 1997] PUMAIN, D., SAINT-JULIEN, T. et MATHIAN, H. (1997). L’analyse spatiale : Localisations dans l’espace.
- [Randell et al., 1992] RANDELL, D., CUI, Z., COHN, A., NEBEL, B., RICH, C. et SWARTOUT, W. (1992). A spatial logic based on regions and connection. In *KR’92. Principles of Knowledge Representation and Reasoning : Proceedings of the Third International Conference*, pages 165–176. Morgan Kaufmann.
- [Régis, 2004] RÉGIN, J. (2004). Modélisation et contraintes globales en programmation par contraintes.
- [Régis et Vidal, 2004] RÉGNIER, P. et VIDAL, V. (2004). *Algorithmique de la planification en IA*. Cépadués-Éditions.
- [Reitsma et Hiramatsu, 2006] REITSMA, F. et HIRAMATSU, K. (2006). Exploring geomarkup on the semantic web. pages 20–22.

- [Russell et Norvig, 2006] RUSSELL, S. et NORVIG, P. (2006). *Intelligence artificielle : [avec près de 400 exercices]*, volume 1184. Pearson Education.
- [Saval et al., 2009] SAVAL, A., BOUZID, M. et BRUNESSAUX, S. (2009). Vers une modélisation spatiale et sémantique pour l'interprétation des risques.
- [Schmidt-Schauß et Smolka, 1991] SCHMIDT-SCHAUSS, M. et SMOLKA, G. (1991). Attributive concept descriptions with complements. *Artificial intelligence*, 48(1):1–26.
- [Schockaert, 2008] SCHOCKAERT, S. (2008). *Reasoning about fuzzy temporal and spatial information from the web*. Thèse de doctorat, Ghent University.
- [Simon et Newell, 1972] SIMON, H. et NEWELL, A. (1972). Human problem solving.
- [Sirin et al., 2007] SIRIN, E., PARSIA, B., GRAU, B., KALYANPUR, A. et KATZ, Y. (2007). Pellet : A practical owl-dl reasoner. *Web Semantics : science, services and agents on the World Wide Web*, 5(2):51–53.
- [Smith et Grenon, 2002] SMITH, B. et GRENON, P. (2002). Basic formal ontology. *Draft. Downloadable at <http://ontology.buffalo.edu/bfo>*.
- [Straccia, 2011] STRACCIA, U. (2011). Reasoning within fuzzy description logics. *Arxiv preprint arXiv:1106.0667*.
- [Studer et al., 1998] STUDER, R., BENJAMINS, V. et FENSEL, D. (1998). Knowledge engineering : principles and methods. *Data & knowledge engineering*, 25(1-2):161–197.
- [Tenbrink et al., 2007] TENBRINK, T., MAISEYENKA, V. et MORATZ, R. (2007). Spatial reference in simulated human-robot interaction involving intrinsically oriented objects. *In Symposium Spatial Reasoning and Communication at AISB*, volume 7. Citeseer.
- [Thrun, 2002] THRUN, S. (2002). Robotic mapping : A survey.
- [Tomatis, 2001] TOMATIS, N. (2001). Hybrid, metric-topological, mobile robot navigation.
- [Tournassoud, 1992] TOURNASSOUD, P. (1992). Planification et contrôle en robotique : application aux robots mobiles et manipulateurs.
- [Trafton et al., 2005] TRAFTON, J., CASSIMATIS, N., BUGAJSKA, M., BROCK, D., MINTZ, F. et SCHULTZ, A. (2005). Enabling effective human-robot interaction using perspective-taking in robots. *Systems, Man and Cybernetics, Part A : Systems and Humans, IEEE Transactions on*, 35(4):460–470.
- [Tsarkov et Horrocks, 2003] TSARKOV, D. et HORROCKS, I. (2003). Dl reasoner vs. first-order prover. *In Proc. of the 2003 Description Logic Workshop (DL 2003*, volume 81, pages 152–159. Citeseer.
- [Tsarkov et Horrocks, 2004] TSARKOV, D. et HORROCKS, I. (2004). Efficient reasoning with range and domain constraints. *In 2004 International Workshop on Description Logics*, page 41. Citeseer.
- [Van Beek, 1992] VAN BEEK, P. (1992). Reasoning about qualitative temporal information. *Artificial intelligence*, 58(1-3):297–326.

-
- [Van Hentenryck Yves, 1992] VAN HENTENRYCK YVES, P. (1992). A generic arc-consistency algorithm and its specializations* 1. *Artificial Intelligence*, 57(2-3):291–321.
- [Vasudevan *et al.*, 2007] VASUDEVAN, S., GACHTER, S., NGUYEN, V. et SIEGWART, R. (2007). Cognitive maps for mobile robots—an object based approach. *Robotics and Autonomous Systems*, 55(5):359–371.
- [Vieu, 1997] VIEU, L. (1997). Spatial representation and reasoning in artificial intelligence. *Spatial and temporal reasoning*, pages 5–41.
- [Vilain et Kautz, 1986] VILAIN, M. et KAUTZ, H. (1986). Constraint propagation algorithms for temporal reasoning. pages 377–382.
- [Zacharias *et al.*, 2006] ZACHARIAS, F., BORST, C. et HIRZINGER, G. (2006). Bridging the gap between task planning and path planning. pages 4490–4495.
- [Zender *et al.*, 2007] ZENDER, H., JENSFELT, P., MOZOS, Ó., KRUIJFF, G. et BURGARD, W. (2007). An integrated robotic system for spatial understanding and situated interaction in indoor environments. 22(2):1584.
- [Zou *et al.*, 2005] ZOU, Y., FININ, T. et CHEN, H. (2005). F-owl : An inference engine for semantic web. *Formal Approaches to Agent-Based Systems*, pages 238–248.

Représentation de la connaissance spatiale pour la planification

Résumé

Nous nous intéressons à la prise en compte de l'information spatiale d'un point de vue de la représentation et du raisonnement afin de planifier une mission dans le cadre de l'interaction homme-robot. Une première partie de la thèse concerne la représentation et le raisonnement spatial. Nous proposons une ontologie spatiale : *SpaceOntology* permettant de représenter l'information spatiale qualitative, quantitative et de mener un raisonnement sur celle-ci. Sur la base de cette modélisation spatiale nous avons mis en place un planificateur avec deux modules : un module de raisonnement symbolique supporté par un planificateur de tâches et un module de raisonnement spatial supporté par le planificateur de chemins et *SpaceOntology*. Enfin, la troisième partie de nos contributions concerne l'extension du langage de planification PDDL à l'information spatiale. Cette extension permet de donner à un problème de planification une sémantique spatiale.

MOTS-CLÉS :

- **indexation Rameau : Intelligence artificielle ; Planification ; Représentation des connaissances ; Ontologies (informatique)**
- **indexation libre : Raisonnement spatial, Ontologie spatiale**

Spatial knowledge representation in planning

Abstract

We are interested in the integration of spatial information for representation and reasoning to plan a mission in the human-robot interaction case. The first part of our work is for representation and spatial reasoning. We propose a spatial ontology : *SpaceOntology* qualitative and quantitative representation and reasoning. Based on this spatial modeling we have implemented a planner with two modules : a module for symbolic reasoning supported by a task planner and a module for spatial reasoning supported by the path planning and *SpaceOntology*. The third part of our contributions is the extension of the PDDL planning language to spatial information. This extension permits to express spatial planning problems.

Discipline : Informatique et applications

Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen
CNRS UMR 6072, Université de Caen Basse-Normandie BP 5186, 14032 Caen Cedex, FRANCE