

NABIL SAHLI

**LA GÉOSIMULATION ORIENTÉE AGENT : UN
SUPPORT POUR LA PLANIFICATION DANS LE
MONDE RÉEL**

Thèse présentée
à la Faculté des études supérieures de l'Université Laval
dans le cadre du programme de doctorat en Informatique
pour l'obtention du grade de Philosophiæ Doctor (Ph.D.)

DÉPARTEMENT D'INFORMATIQUE
FACULTÉ DES SCIENCES ET DE GÉNIE
UNIVERSITÉ LAVAL
QUÉBEC

FEVRIER, 2006

Résumé

La planification devient complexe quand il s'agit de gérer des situations incertaines. Prédire de façon précise est une tâche fastidieuse pour les planificateurs humains. L'approche *Simulation-Based Planning* consiste à associer la planification à la simulation. Chaque plan généré est simulé afin d'être testé et évalué. Le plan le plus approprié est alors retenu. Cependant, le problème est encore plus complexe lorsque viennent s'ajouter des contraintes spatiales. Par exemple, lors d'un feu de forêt, des bulldozers doivent construire une ligne d'arrêt pour arrêter la propagation des feux. Ils doivent alors tenir compte non seulement de l'avancée des feux mais aussi des caractéristiques du terrain afin de pouvoir avancer plus facilement. Nous proposons une approche de géosimulation basée sur les agents et qui a pour but d'assister la planification dans un espace réel, à large échelle géographique et surtout à forte composante spatiale.

Un feu de forêt est un problème typique nécessitant une planification dans un monde réel incertain et soumis à de fortes contraintes spatiales. Nous illustrons donc notre approche (nommée ENCASMA) sur le problème des feux de forêts.

L'approche consiste à établir un parallélisme entre l'Environnement Réel ER (p.ex. une forêt incendiée) et un Environnement de Simulation ES (p.ex. une reproduction virtuelle de la forêt incendiée). Pour garantir un niveau acceptable de réalisme, les données spatiales utilisées dans l'ES doivent absolument provenir d'un SIG (Système d'information Géographique). Les planificateurs réels comme les pompiers ou les bulldozers sont simulés par des agents logiciels qui raisonnent sur l'espace modélisé par l'ES. Pour une meilleure sensibilité spatiale (pour tenir compte de toutes les contraintes du terrain), les agents logiciels sont dotés de capacités avancées telles que la perception.

En utilisant une approche par géosimulation multiagent, nous pouvons générer une simulation réaliste du plan à exécuter. Les décideurs humains peuvent visualiser les conséquences probables de l'exécution de ce plan. Ils peuvent ainsi évaluer le plan et éventuellement l'ajuster avant son exécution effective (sur le terrain).

Quand le plan est en cours d'exécution, et afin de garantir la cohérence des données entre l'ER et l'ES, nous gardons trace sur l'ES des positions (sur l'ER) des planificateurs réels (en utilisant les technologies du positionnement géoréférencé). Nous relançons la planification du reste du plan à partir de la position courante de planificateur réel, et ce de façon périodique. Ceci est fait dans le but d'anticiper tout problème qui pourrait survenir à cause de l'aspect dynamique de l'ER. Nous améliorons ainsi le processus classique de l'approche DCP (*Distributed Continual Planning*). Enfin, les agents de l'ES doivent replanifier aussitôt qu'un événement imprévu est rapporté.

Étant donné que les plans générés dans le cas étudié (feux de forêts) sont essentiellement des chemins, nous proposons également une approche basée sur la géosimulation orientée agent pour résoudre des problèmes particuliers de Pathfinding (recherche de chemin).

De plus, notre approche souligne les avantages qu'apporte la géosimulation orientée agent à la collaboration entre agents humains et agents logiciels. Plus précisément, elle démontre :

- Comment la cognition spatiale des agents logiciels sensibles à l'espace peut être complémentaire avec la cognition spatiale des planificateurs humains.
- Comment la géosimulation orientée agent peut compléter les capacités humaines de planification lors de la résolution de problèmes complexes.

Finalement, pour appliquer notre approche au cas des feux de forêts, nous avons utilisé MAGS comme plate-forme de géosimulation et Prometheus comme simulateur du feu.

Les principales contributions de cette thèse sont :

1. Une architecture (ENCASMA) originale pour la conception et l'implémentation d'applications (typiquement des applications de lutte contre les désastres naturels) dans un espace géographique réel à grande échelle et dynamique.
2. Une approche basée sur les agents logiciels pour des problèmes de Pathfinding (recherche de chemin) particuliers (dans un environnement réel et à forte composante spatiale, soumis à des contraintes qualitatives).

3. Une amélioration de l'approche de planification DCP (plus particulièrement le processus de continuité) afin de remédier à certaines limites de la DCP classique.
4. Une solution pratique pour un problème réel et complexe : la lutte contre les feux de forêts. Cette nouvelle solution permet aux experts du domaine de mieux planifier d'avance les actions de lutte et aussi de surveiller l'exécution du plan en temps réel.

Abstract

Planning becomes complex when addressing uncertain situations. Accurate predictions remain a hard task for human planners. The *Simulation-Based Planning* approach consists in associating planning and simulation. Each generated plan is simulated in order to be tested and evaluated. The most appropriate plan is kept. The problem is even more complex when considering spatial constraints. For example, when fighting a wildfire, dozers build a firebreak to stop fire propagation. They have to take into account not only the fire spread but also the terrain characteristics in order to move easily. We propose an agent-based geosimulation approach to assist such planners with planning under strong spatial constraints in a real large-scale space.

Forest fire fighting is a typical problem involving planning within an uncertain real world under strong spatial constraints. We use this case to illustrate our approach (ENCASM).

The approach consists in drawing a parallel between the Real Environment RE (i.e. a forest in fire) and the Simulated Environment SE (i.e. a virtual reproduction of the forest). Spatial data within the SE should absolutely come from a GIS (Geographic Information System) for more realism. Real planners such as firefighters or dozers are simulated using software agents which reason about the space of the SE. To achieve a sufficient spatial awareness (taking into account all terrain's features), agents have advanced capabilities such as perception.

Using a multiagent geosimulation approach, we can generate a realistic simulation of the plan so that human decision makers can visualize the probable consequences of its execution. They can thus evaluate the plan and adjust it before it can effectively be executed.

When the plan is in progress and in order to maintain coherence between RE and SE, we keep track in the SE of the real planners' positions in the RE (using georeferencing technologies). We periodically replan the rest of the plan starting from the current position of the real planner. This is done in order to anticipate any problem which could occur due

to the dynamism of the RE. We thus enhance the process of the classical *Distributed Continual Planning* DCP. Finally, the agents must replan as soon as an unexpected event is reported by planners within the RE.

Since plans in the studied case (forest fires) are mainly paths, we propose a new approach based on agent geosimulation to solve particular Pathfinding problems.

Besides, our approach highlights the benefits of the agent-based geo-simulation to the collaboration of both humans and agents. It thus shows:

- How spatial cognitions of both spatially aware agents and human planners can be complementary.
- How agent-based geo-simulation can complement human planning skills when addressing complex problems.

Finally, when applying our approach on firefighting, we use MAGS as a simulation platform and Prometheus as a fire simulator.

The main contributions of this thesis are:

1. An original architecture (ENCASMA) for the design and the implementation of applications (typically, natural disasters applications) in real, dynamic and large-scale geographic spaces.
2. An agent-based approach for particular Pathfinding problems (within real and spatially constrained environments and under qualitative constraints).
3. An enhancement of the DCP (particularly, the continual process) approach in order to overcome some limits of the classical DCP.
4. A practical solution for a real and complex problem: wildfires fighting. This new solution aims to assist experts when planning firefighting actions and monitoring the execution of these plans.

Avant-Propos

Tout d'abord je tiens à remercier sincèrement Dr. Alexis Drougoul pour avoir accepté d'être le prélecteur de cette thèse et de se déplacer au Québec pour évaluer ma soutenance. Je remercie également Dr. Jean-Pierre Muller et Dr. Driss Kettani d'avoir accepté de faire partie de mon jury et de venir assister à la défense de ma thèse.

Tout au long de mon périple, j'ai été très bien entouré par des personnes qui m'ont guidé, inspiré, encouragé, critiqué et surtout, qui m'ont permis d'avancer. Je prends ici le temps de sincèrement les remercier et je leur dois à tous, une partie importante de cette recherche.

Principalement, j'aimerais remercier mon directeur de recherche, Dr. Bernard Moulin, qui m'a fait confiance en m'accordant la chance d'apporter ma contribution à la recherche. Il a su me guider dans cette aventure, tout en me laissant la liberté suffisante pour accomplir mon propre parcours. Je le remercie également pour m'avoir permis de présenter mes travaux dans plusieurs conférences internationales. Je le remercie enfin pour ses efforts inlassables pour que j'accomplisse ce travail. Bernard, je te dois beaucoup!

Je remercie également GEOIDE et RDDC Valcartier pour avoir financé une partie du projet. Ma sincère reconnaissance va aussi à Gaétan Lemaire et François Lefebvre de la Société de protection des forêts contre le feu SOPFEU qui m'ont généreusement transmis une grande partie de leur expertise dans le domaine des feux de forêts.

Je voudrais également exprimer ma profonde gratitude à M. Cordy Tymstra de la Division de Protection des Forêts en Alberta, pour toute l'aide qu'il m'a fournie. Je le remercie aussi profondément pour m'avoir invité à Edmonton pour mieux comprendre le domaine des feux de forêts.

Merci à mes amis. Le doctorat est un processus à la fois scientifique et créatif. Il lui faut donc un peu de bohème pour voir le jour. À vous, qui êtes la mienne. À mon ami depuis déjà une décennie Nafâa Jabeur, à Lois Porath pour les moments durs et joyeux que nous avons passés ensemble ces dernières années, à Walid Ali pour toutes les balades que nous avons pu faire ensemble, à Jamal Bentahar pour toutes les gâteries que tu nous emmenais

au laboratoire, à Tarek Sboui, Mondher Bouden, Hédi Haddad, Boubaker Boulekrouche, Walid Chaker pour tous les bons moments que nous avons passés ensemble, à Dani Nassour pour tous les bons repas libanais auxquels tu m'invitais, à mes amis de jujutsu brésilien pour les moments d'échappement que je passe avec vous sur le dojo, à Anita Yovanivic pour ta chère amitié, à Alaeddine et Solène et à tous mes amis que j'ai connus sur le campus pendant ces 6 dernières années (Léo, Mag, Nicolas, Doug...), à tous mes amis qui me soutiennent depuis l'autre côté de l'Atlantique.

Un gros merci à tous ceux qui ont partagé mon quotidien au département, à tout le personnel du département. Que ce soit dans le laboratoire ou dans un corridor, lors des pauses-café ou autour de quelques gâteries, ce fut toujours un plaisir. Un merci particulier pour Lynda Goulet.

Merci à ma grande famille, tantes et oncles, sœurs et frères, cousins et cousines, pour tout l'amour que vous portez pour moi et pour m'avoir supporté toutes ces années.

Merci à ma future femme (?nom, ?prénom) et à mon (mes) futur(s) enfant(s), si j'en aurais inchallah.

Enfin, merci à mes parents, Salah et Saida. Ça doit faire déjà plus de 11 ans que j'étudie loin de vous, mais c'est grâce à vous et pour vous que j'ai fait ma thèse. Aucun mot sur cette page ne saurait exprimer ce que je vous dois, ni combien je vous aime.

*à mes parents car c'est d'eux que je tiens la
détermination et la persévérance qui m'ont
permis de réaliser ma thèse. Que sa
réalisation leur soit un témoignage de
reconnaissance et d'amour, eux qui, dans la
simplicité et l'humilité ont su transmettre à
leurs enfants et à moi-même les plus grandes
et les plus nobles des valeurs morales.*

Table des matières

INTRODUCTION	1
1.1. CONTEXTE	1
1.2. PROBLÉMATIQUE	3
1.2.1 DCP.....	4
1.2.2 Technologie Agent.....	5
1.2.3 SIG.....	9
1.2.4. Les techniques du raisonnement spatial.....	11
1.3. ÉNONCÉ DE LA PROBLÉMATIQUE DE RECHERCHE	12
1.4. OBJECTIFS DE RECHERCHE	13
1.5. PISTES DE SOLUTIONS ET MÉTHODOLOGIE DE RECHERCHE	15
1.6. PLAN DE LA THÈSE	17
ÉTAT DE L'ART.....	19
2.1. COGNITION HUMAINE.....	19
2.1.1. <i>La planification spatiale chez l'être humain</i>	20
2.1.1.1. La planification cognitive	20
2.1.1.2. Cas d'un environnement spatial.....	22
2.1.2. <i>Les limites de la planification cognitive dans un espace géographique</i>	24
2.1.2.1. Les limites de la planification cognitive	24
2.1.2.2. Les limites de la cognition spatiale.....	25
2.2. PLANIFICATION BASÉE SUR LES AGENTS DANS UN ENVIRONNEMENT DYNAMIQUE	29
2.2.1. <i>La planification par les agents</i>	30
2.2.1.1. Agents logiciels et Systèmes multiagent : Généralités	30
2.2.1.1.1. <i>Définition d'agent</i>	30
2.2.1.1.2. <i>Systèmes multiagents</i>	31
2.2.1.1.3. <i>Utilisation des agents</i>	34
2.2.1.2. Planification par agents : Historique.....	35
2.2.1.3. Planification dans un environnement multiagent.....	37
2.2.2. <i>Planification dans un environnement dynamique</i>	38
2.2.3. <i>Distributed continual planning (DCP)</i>	40
2.2.3.1. DCP : Principe	41
2.2.3.2. DCP: Évolution.....	42
2.2.4. <i>Planification dans un espace géographique</i>	43
2.2.4.1. Le raisonnement spatial qualitatif.....	43
2.2.4.2. Les agents au service du raisonnement spatial qualitatif.....	44
2.3. SIMULATION BASÉE SUR LES AGENTS : ÉTAT DE L'ART	45
2.3.1. <i>Simulation : Généralités</i>	46
2.3.1.1 Définition.....	46
2.3.1.2. Les phases de la simulation	46
2.3.1.3. Différents domaines d'application.....	47
2.3.2. <i>Simulation Orientée Agent (SOA)</i>	47
2.3.2.1. Introduction.....	47
2.3.2.2. Branches de la SOA	48
2.3.2.2.1. <i>Simulation Agent</i>	48

2.3.2.2.2. <i>Simulation basée sur les agents</i>	49
2.3.2.2.3. <i>Simulation supportée par les agents</i>	49
2.3.2.3. Implémentation de la SOA : réalité et défis	50
2.3.2.3.1. <i>Réalité</i>	50
2.3.2.3.2. <i>Défis et avenir</i>	50
2.3.2.4. Conclusion	52
2.4. PLANIFICATION BASÉE SUR LA SIMULATION	52
2.4.1. <i>Principe de l'intégration de la planification et de la simulation</i>	53
2.4.2. <i>Quelques travaux sur la SBP</i>	54
2.4.2.1. La planification par anticipation (<i>Anticipatory Planning</i>)	54
2.4.2.2. Framework SBP	55
2.4.2.3. Planificateur d'assauts pour un bataillon de chars	56
2.4.2.4. « Capture the Flag »	57
2.4.2.5. Le projet Phœnix	58
2.4.3. Conclusion	65
2.5. LES SIGS : UN MOYEN DE LUTTE CONTRE LES DÉSASTRES	66
2.5.1 <i>Motivation</i>	66
2.5.2. <i>Importance des SIG lors des désastres naturels</i>	67
2.5.3. <i>Les SIGs au support de la simulation et de la modélisation</i>	67
2.5.4. <i>Les SIG dans les systèmes d'aide à la décision</i>	68
2.5.5. Conclusion	70
2.6. GÉOSIMULATION	70
2.6.1. <i>Généralités</i>	70
2.6.2. <i>Techniques de réalisation</i>	72
2.6.2.1. Automates Cellulaires : Généralités	72
2.6.2.2. Géosimulation par SMA	73
2.6.2.3. Agents ou Automates Cellulaires?	73
2.6.3. <i>Géosimulation basée sur les agents dans un monde réel</i>	75
2.6.3.1. Applications	75
2.6.3.2. Plateformes de géosimulation basée sur les agents	75
2.6.3.2.1. <i>Swarm</i>	75
2.6.3.2.2. <i>MAGS</i>	76
CONCLUSION DU CHAPITRE	76
DOMAINE D'APPLICATION	77
3.1. GÉNÉRALITÉS SUR LES FEUX DE FORÊTS	77
3.2. FACTEURS INFLUENÇANT LA PROPAGATION DES FEUX	77
3.2.1. <i>Le triangle du feu</i>	77
3.2.2. <i>Autres facteurs favorisant la propagation des feux de forêts</i>	80
3.2.2.1 La topographie	80
3.2.2.2 Barrières naturelles et artificielles	80
3.2.2.3 Routes d'accès	80
3.2.3. <i>Propagation des incendies</i>	80
3.2.4. Conclusion	81
3.3. LUTTE CONTRE LE FEU	81
3.3.1. <i>Introduction</i>	81

3.3.2. <i>Lutte contre le feu : procédure</i>	82
3.3.3. <i>Différentes stratégies</i>	84
3.3.3.1. <i>Attaque directe</i>	86
3.3.3.2. <i>Attaque indirecte</i>	87
3.3.3.3. <i>Facteurs de décision</i>	88
3.3.3.3.1. <i>La sécurité des pompiers</i>	89
3.3.3.3.2. <i>Le comportement courant et futur du feu</i>	89
3.3.3.3.3. <i>La disponibilité des ressources</i>	91
3.3.3.3.4. <i>L'accès au feu</i>	91
3.3.3.4. <i>Évaluation des deux stratégies</i>	92
3.3.3.5. <i>Construction des lignes d'arrêt des incendies</i>	93
3.3.3.6. <i>Les tactiques de suppression à impact minimal</i>	95
3.4. LES SYSTÈMES DE LUTTE CONTRE LE FEU	96
3.4.1. <i>Introduction</i>	96
3.4.2. <i>Systèmes informatiques actuels de gestion des feux</i>	98
3.4.2.1. <i>Introduction : les types de systèmes utilisés</i>	98
3.4.2.2. <i>Systèmes de simulation de la propagation</i>	99
3.4.2.3. <i>Systèmes de planification dans les feux de forêt : outils</i>	100
CONCLUSION	100
APPROCHE ENCASMA : PRINCIPES ET ARCHITECTURE	102
4.1. PROBLÈME FORMEL	103
4.1.1. <i>Vue générale</i>	103
4.1.2. <i>Caractéristiques de l'environnement</i>	104
4.1.3. <i>But des acteurs</i>	105
4.2. PRINCIPES DE LA SOLUTION THÉORIQUE	106
4.2.1. <i>Introduction</i>	106
4.2.2. <i>La connaissance spatiale augmentée</i>	107
4.2.3. <i>Les représentants du monde réel dans la simulation</i>	109
4.2.3.1. <i>Les agents : représentants des ressources et des acteurs</i>	109
4.2.3.2. <i>Les modèles physiques : une simulation fiable des situations dangereuses</i>	110
4.2.4. <i>Le SIG : source de la connaissance spatiale augmentée</i>	110
4.2.5. <i>De nouvelles technologies au support de la communication entre le monde réel et de simulation</i>	111
4.3. ENCASMA : UN FRAMEWORK À QUATRE COUCHES POUR LUTTER CONTRE LES SITUATIONS DANGEREUSES	112
4.3.1. <i>Plate-forme de simulation : MAGS</i>	113
4.3.2. <i>Le modèle de simulation et ses liens avec la situation réelle</i>	114
4.3.3. <i>Le système multiagent et son lien avec le monde réel</i>	116
4.3.3.1. <i>Introduction</i>	116
4.3.3.2. <i>Les agents réels et les agents de simulation</i>	116
4.3.3.3. <i>Capacités des agents de simulation dans MAGS</i>	117
4.3.3.4. <i>Lien entre Agent Réel et Agent de Simulation</i>	117
4.3.4. <i>La couche application</i>	118
4.4. LA PLATE-FORME MAGS	119

4.4.1. Introduction	119
4.4.2. Les capacités des agents MAGS	119
4.4.2.1. La perception	119
4.4.2.2. La navigation	120
4.4.2.3. La connaissance	120
4.4.2.4. Le comportement	121
4.4.3. Les avantages de MAGS	122
4.5. LES APPLICATIONS D'ENCASMA SUR LES FEUX DE FORÊTS	122
4.6. TRAVAUX CONNEXES	124
4.7. CONTRIBUTIONS DU CHAPITRE	125
LA SIMULATION PAR AGENTS POUR LA RECHERCHE DE CHEMIN (PATHFINDING) DANS UN MONDE RÉEL	127
5.1. CARACTÉRISATION DU PROBLÈME DE PATHFINDING	129
5.1.1. Facteurs liés à l'environnement	129
5.1.2. Facteurs liés au Pathfinder.....	130
5.1.3. Facteurs liés à la procédure de recherche	131
5.2. PATHFINDING PAR SIMULATION BASÉE SUR LES AGENTS : APPROCHE PROPOSÉE	134
5.2.1. Principes	134
5.2.2. Architecture interne de l'agent de Pathfinding	137
5.3. ILLUSTRATION DE L'APPROCHE SUR LA CONSTRUCTION DE LA LIGNE D'ARRÊT.....	141
5.3.1. Scénario	141
5.3.2. Architecture	143
5.3.2.1. Simulateur Prometheus	144
5.3.2.2. Environnement.....	144
5.3.3. Agent Pathfinder	145
5.3.4. Éléments d'optimisation	149
5.3.4.1. Parallélisme par clonage	149
5.3.4.2. Pré-Pathfinding	149
5.4. ÉVITEMENT DES OBSTACLES	150
5.4.1. État de l'art.....	150
5.4.2. Solutions possibles pour l'Agent Pathfinder	153
5.4.3. Illustration sur la construction de la ligne d'arrêt.....	154
5.4.3.1. Les contraintes du problème	154
5.4.3.2. Solution proposée	155
5.5. VERS UN PATHFINDING EN TEMPS RÉEL	156
5.5.1. Motivations	156
5.5.2. Scénario	158
5.5.3. Extension de l'approche au temps réel.....	159
5.5.3.1. Principes.....	159
5.5.3.2. Solution proposée pour la construction de la ligne d'arrêt pour les feux ..	162
5.6. DISCUSSION	163
5.7. TRAVAUX CONNEXES	166
5.8. CONTRIBUTIONS DU CHAPITRE	170
ENCASMA : SUPPORT POUR LA DCP	171

6.1. DCP: ÉTAT DE L'ART	171
6.1.1. <i>Planification Distribuée en Continu : synthèse</i>	172
6.1.1.1. Planification distribuée	172
6.1.1.2. Planification en Continu : synthèse	172
6.1.1.5. Récapitulation et orientations du chapitre	176
6.1.2. <i>Entrelacement de la planification et de l'exécution dans un environnement dynamique et incertain</i>	177
6.1.3. <i>Limites de la DCP actuelle</i>	180
6.2. IMPLÉMENTATION DE LA DCP PAR UN SMA DANS UN ENVIRONNEMENT GÉOSPATIAL DE SIMULATION.....	181
6.2.1. <i>Problème à résoudre</i>	181
6.2.1.1. Scénario	181
6.2.1.2. Hypothèses.....	183
6.2.2. <i>Solution sur scénario : Planification de la construction de la ligne d'arrêt</i>	184
6.2.2.1.1. Recherche d'un plan global	185
6.2.2.1.2. Test et validation des plans.....	186
6.2.2.1.3. Exécution (réelle) du plan et replanification périodique anticipée	187
6.2.2.1.4. Replanification en cas d'événement imprévu.....	190
6.2.3. <i>Généralisation de la solution : Une DCP spatiale basée sur des agents et à exécution anticipée en mode simulation</i>	191
6.2.3.1. DCP spatiale à base d'agents logiciels dans un environnement simulé	191
6.2.3.2. Complémentarité entre les capacités humaines et logiciels pour une meilleure planification	194
6.2.3.3. Infrastructure de rencontre spatiale : un support pour la planification	195
6.3. CONTRIBUTIONS DU CHAPITRE	198
RÉALISATION	200
7.1. SCÉNARIOS.....	200
7.1.1. <i>Stratégie directe</i>	201
7.1.2. <i>Stratégie indirecte</i>	201
7.1.2.1. Scénario « avant la construction de ligne d'arrêt ».....	202
7.1.2.2. Scénario « pendant la construction de la ligne d'arrêt ».....	203
7.2. MAGS	204
7.2.1. <i>Architecture</i>	204
7.2.2. <i>Module Environnement</i>	206
7.2.3. <i>Module Agents</i>	208
7.2.4. <i>Module États</i>	210
7.3. IMPLÉMENTATION	213
7.3.1. <i>Architecture globale</i>	213
7.3.2. <i>Environnement</i>	214
7.3.3. <i>État TRoutePlanningState</i>	217
7.3.3.1. <i>Agent_Attaque_Directe</i>	220
7.3.3.2. <i>Agent_Meilleur_Chemin</i>	223
7.3.3.3. <i>Agent_Déjà_Fait</i>	223
7.3.3.4. <i>Agent_Vérificateur_Position</i>	224
7.3.3.5. <i>Agent_Vérificateur_PlanVsObstacle</i>	225

7.3.3.6. <i>Agent_Pathfinder</i>	228
7.3.4. <i>Module Fire</i>	235
7.3.4.1. ‘Chef d’orchestre des agents’	235
7.3.4.2. Intermédiaire entre Prometheus et MAGS.....	236
7.4. DISCUSSION ET CONCLUSION	239
RÉSULTATS ET VALIDATION	242
8.1. CRITÈRES DE VALIDITÉ.....	242
8.1.1. <i>Fidélité de la simulation</i>	242
8.1.2. <i>Vérification de la simulation</i>	243
8.1.3. <i>Validation de la simulation</i>	247
8.2. SCÉNARIOS.....	249
8.3. IMPLÉMENTATION ET RÉSULTATS	252
8.4. ANALYSE DES RÉSULTATS ET CONCLUSION	260
CONCLUSION	263
9.1. DISCUSSION GÉNÉRALE	263
9.2. CONTRIBUTIONS.....	267
9.3. TRAVAUX FUTURS.....	268
BIBLIOGRAPHIE.....	272
ANNEXES	304
ANNEXE A : LES FEUX DE FORÊTS	304
ANNEXE B : PROMETHEUS.....	326
ANNEXE C : APPLICATION 2 : « ASSISTANCE SUR LE TERRAIN »	333

Liste des tableaux

TABLEAU 2.1. TYPOLOGIE DES INTERACTIONS DANS UN SMA	33
TABLEAU 3.2. LES STRATÉGIES À ADOPTER SELON LE LFM [EIIFC 03]	90
TABLEAU 3.3. AVANTAGES ET INCONVÉNIENTS D'UNE ATTAQUE DIRECTE [BLM 03]	92
TABLEAU 3.4. AVANTAGES ET INCONVÉNIENTS D'UNE ATTAQUE INDIRECTE [BLM 03]	93
TABLEAU 4.1. CORRESPONDANCES ENTRE ENCASMA ET FEU DE FORET	123

Liste des figures

FIGURE 1.1. DIFFÉRENTS ASPECTS DE LA RECHERCHE EN « PLANIFICATION DANS UN ENVIRONNEMENT DYNAMIQUE À FORTE COMPOSANTE SPATIALE ».....	14
FIGURE 2.1. RELATIONS ENTRE SMA ET MODÉLISATION BASÉE SUR LES AUTOMATES CELLULAIRES.	74
FIGURE 3.1. TRIANGLE DU FEU.....	78
FIGURE 3.2. ATTAQUE DIRECTE AVEC DES OUTILS À MAIN ET DES POMPES À DOS [NICOLAS ET BEEBE 99].	87
FIGURE 3.3. CONSTRUCTION D'UNE LIGNE D'ARRÊT LORS D'UNE ATTAQUE INDIRECTE [NICOLAS ET BEEBE 99].	88
FIGURE 3.4. LORSQUE LA LONGUEUR DES FLAMMES DÉPASSE 11 PIEDS, L'ATTAQUE INDIRECTE EST INUTILE. LES ROUTES, LES RIVIÈRES ET D'AUTRES BARRIÈRES NATURELLES SONT UTILISÉES.	91
FIGURE 3.5. CONSTRUCTION D'UNE LIGNE D'ARRÊT À LA MAIN [CARR 02]	94
FIGURE 4.1. PROBLÈME À RÉSOUDRE	103
FIGURE 4.2. PRINCIPE DE LA CONNAISSANCE AUGMENTÉE DANS UN MONDE DE SIMULATION	109
FIGURE 4.3. ARCHITECTURE GÉNÉRALE D'ENCASMA.....	113
FIGURE 5.1. HIÉRARCHIE DES DIFFÉRENTS FACTEURS INFLUENÇANT LES TECHNIQUES DU PATHFINDING.....	133
FIGURE 5.2. ENCASMA POUR LES PROBLÈMES DU PATHFINDING	136
FIGURE 5.3. ARCHITECTURE INTERNE DE L'AGENT <i>PATHFINDER</i> : MODÈLE PPM.	141
FIGURE 5.4. DIAGRAMME D'ÉTAT DE L'AGENT <i>PATHFINDER</i> SELON LE MODÈLE PPM.....	141
FIGURE 5.5. L'ARCHITECTURE DE LA SOLUTION PROPOSÉE.	143
FIGURE 5.6. PROCÉDURE SUIVIE PAR L'AGENT <i>PATHFINDER</i> POUR SUIVRE UNE ESQUISSE (SNAPSHOT)	147
FIGURE 5.7. DÉFINITION DE LA <i>DIRECTION LIMITE</i> (DANS CE CAS LES DIRECTIONS CHOISIES DANS L'ÉTAPE (3) DOIVENT ÊTRE À DROITE DE LA <i>DIRECTION LIMITE</i>).....	148
FIGURE 5.8.A. ESQUISSE INITIALE FIGURE 5.8.B. ESQUISSE APRÈS TRANSLATION	150
FIGURE 5.9. CORRESPONDANCE ENTRE LE MONDE RÉEL ET LE MONDE DE SIMULATION POUR DES APPLICATIONS TEMPS RÉEL	159
FIGURE 5.10. DIAGRAMME D'ÉTAT DU FONCTIONNEMENT NORMAL DE L'AGENT <i>PATHFINDER</i> DANS UN CONTEXTE DE TEMPS RÉEL.	161
FIGURE 5.11. MODÈLE AGENT PPMC DANS UN CONTEXTE TEMPS RÉEL	161
FIGURE 5.12. RELATION ENTRE L'AGENT <i>PATHFINDER</i> ET LE BULLDOZER.....	162
FIGURE 5.13. APPLICATION DE LA PLANIFICATION ANTICIPÉE EN CONTINUE POUR LA RECHERCHE DE CHEMIN	163
FIGURE 6.1. L'EXPERT HUMAIN PROPOSE UNE ESQUISSE DE LA LIGNE D'ARRÊT DANS LE MONDE SIMULÉ	185
FIGURE 6.2. L'AGENT <i>AAS-BULLDOZER</i> (VIA SES AGENTS <i>PATHFINDER</i>) TROUVE UN MEILLEUR PLAN GLOBAL.....	186
FIGURE 6.3. ILLUSTRATION DU PRINCIPE DE LA REPLANIFICATION PÉRIODIQUE ANTICIPÉE.	190
FIGURE 6.4. PRINCIPE DE L'INFRASTRUCTURE DE RENCONTRE PAR MIGRATION DES AGENTS.	196
FIGURE 6.5. PRINCIPE DE L'INFRASTRUCTURE DE RENCONTRE PAR DÉLÉGATION AGENTS.	197

FIGURE 7.1. ARCHITECTURE GÉNÉRALE DE MAGS (VERSION ORIGINALE)	206
FIGURE 7.2. DIAGRAMME DE CLASSE SIMPLIFIÉ DU MODULE <i>ENVIRONNEMENT</i>	208
FIGURE 7.3. DIAGRAMME DE CLASSES DU MODULE <i>AGENTS</i>	209
FIGURE 7.4. DIAGRAMME DE CLASSES DU MODULE DES <i>ÉTATS</i>	212
FIGURE 7.5. ARCHITECTURE DE MAGS APRÈS INTÉGRATION DES MODULES SPÉCIFIQUES À L'APPLICATION DE CONSTRUCTION DES LIGNES D'ARRÊT.....	214
FIGURE 7.6. EXEMPLE DE CODE DE CRÉATION D'UN AGENT DANS MAGS	217
FIGURE 7.7. INTÉGRATION DE L'ÉTAT <i>TROUTEPLANNINGSTATE</i> DANS LE MODULE ÉTATS DES AGENTS MAGS.....	218
FIGURE 7.8. EXEMPLE DE CODE DE CRÉATION D'UN AGENT DANS MAGS AYANT L'ÉTAT A <i>ROUTEPLANNING_STATE</i>	219
FIGURE 7.9. DIAGRAMME D'ÉTAT DU FONCTIONNEMENT DE L' <i>AGENT_ATTAQUE_DIRECTE</i>	221
FIGURE 7.10. DIAGRAMME DE SÉQUENCE DÉCRIVANT LE FONCTIONNEMENT DE L' <i>AGENT_ATTAQUE_DIRECTE</i> LORSQU'UN NOUVEAU PÉRIMÈTRE S'AJOUTE À L'ENVIRONNEMENT.....	222
FIGURE 7.11. DIAGRAMME D'ACTIVITÉ DÉCRIVANT LE FONCTIONNEMENT DE L' <i>AGENT_VÉRIFICATEUR_POSITION</i>	225
FIGURE 7.12. ALGORITHME DE RECHERCHE DES EXTRÉMITÉS D'UN OBSTACLE PAR L' <i>AGENT_VÉRIFICATEUR_PLANVSOBSTACLE</i>	227
FIGURE 7.13. DIAGRAMME DE SÉQUENCE DÉCRIVANT LE DÉCLENCHEMENT DE L' <i>AGENT_VÉRIFICATEUR_PLANVSOBSTACLE</i>	228
FIGURE 7.14. DIAGRAMME DE SÉQUENCE DÉCRIVANT LE FONCTIONNEMENT DE L' <i>AGENT_PATHFINDER</i> EN CAS DE PLANIFICATION <i>RAFFINÉE</i>	229
FIGURE 7.15. DIAGRAMME DE SÉQUENCE DÉCRIVANT LE CAS OÙ L' <i>AGENT_VÉRIFICATEUR_POSITION</i> DÉCLENCHE UNE REPLANIFICATION.....	230
FIGURE 7.16. UTILISATION DU <i>TSTATICSTATE</i> POUR CONSERVER ET CONSULTER LES PARAMÈTRES DES AGENTS	233
FIGURE 7.17. ALGORITHME DE <i>MOVEAGENTTHROUGHARIANE</i> SUIVI PAR LES <i>AGENTS_PATHFINDER</i>	235
FIGURE 7.18. IMPORTATION DE LA LIBRAIRE DE PROMETHEUS EN CODE C++.....	236
FIGURE 7.19. EXTRAIT DU CODE C++ DE LA FONCTION <i>CREATESCENARIOTFIREPROMETHEUS</i> , LORS DE L'APPEL DES FONCTIONS DE L'INTERFACE COM DE PROMETHEUS	237
FIGURE 7.20. EXTRAIT DU CODE C++ DE LA FONCTION <i>CREATEPERIMETERAGENT</i> LORS DE L'EXTRACTION DES RÉSULTATS DE PROMETHEUS ET LA CRÉATION DES <i>AGENTS_FIRE</i>	238
FIGURE 8.1. L' <i>AGENT_ATTAQUE_DIRECTE</i> QUI SUIT UN PÉRIMÈTRE DE FEU AVEC SUCCÈS (SNAPSHOT)	244
FIGURE 8.2. EXEMPLE DE RECHERCHE DE CHEMIN SANS CLONAGE.....	245
FIGURE 8.3. EXEMPLE DE RECHERCHE DE CHEMIN AVEC CLONAGE (ICI CAS D'UNE REPLANIFICATION RAFFINÉE)	245
FIGURE 8.4. ÉVITEMENT D'OBSTACLE DES <i>AGENTS_PATHFINDER</i>	246
FIGURE 8.5. REPRÉSENTATION DES DONNÉES DU TERRAIN SUR L'ENVIRONNEMENT VIRTUEL	253
FIGURE 8.6. ESQUISSE (1) (À GAUCHE) ET ESQUISSE (2) PROPOSÉE PAR L'UTILISATEUR.	254
FIGURE 8.7. CAS OÙ LE TRACÉ PROPOSÉ PAR L'UTILISATEUR EST JUGÉ FAISABLE PAR L' <i>AGENT_ESTIMERLIGNEEXPERT</i>	255

FIGURE 8.8. LES PÉRIMÈTRES DE FEU GÉNÉRÉS PAR PROMETHEUS EN L'ABSENCE DE LIGNE D'ARRÊT.	256
FIGURE 8.9. CAS OÙ LE TRACÉ PROPOSÉ PAR L'UTILISATEUR EST JUGÉ INFASIBLE PAR L'AGENT <i>ESTIMERLIGNEEXPERT</i>	257
FIGURE 8.10. CAS OÙ LES AGENTS REPENNENT LA RECHERCHE DU PLAN À ZÉRO.....	259
FIGURE 8.11. CAS OÙ LES AGENTS CORRIGENT LE PLAN DE L'EXPERT HUMAIN.....	260

CHAPITRE 1

Introduction

Lors des désastres naturels (feu de forêt majeur, marée noire, inondations, etc.) ou lors des situations de crise, des équipes d'intervention ont souvent la difficile tâche de planifier des actions (de secours, de lutte, d'évacuation, etc.) sur un terrain incertain et soumis à des changements continus. La complexité du problème, le dynamisme de l'environnement, et la diversité des facteurs influençant la prise de décision, font que la planification est difficile dans ces conditions. La planification classique de l'IA s'est même avérée inefficace dans ces conditions.

Dans ce chapitre, nous décrivons avec plus de détails les caractéristiques de ces environnements et les problèmes qui en découlent. Nous examinons également les outils informatiques disponibles et nous proposons des pistes de solutions.

1.1. Contexte

Lors d'un désastre naturel par exemple, l'espace considéré est souvent de grande échelle (forêt, océan, ville, etc.), le plan à élaborer est complexe et enfin l'environnement est dynamique puisque plusieurs paramètres externes (la météo, les courants marins, ...) ou internes (échec d'un plan, problème imprévu...) peuvent changer la situation.

Nous nous situons donc dans le contexte d'une planification :

1. Complexe.
2. Dans un environnement très dynamique : plusieurs imprévus peuvent survenir durant la planification ou pendant l'exécution du plan.
3. Dans un espace géographique à grande échelle : nous nous intéressons plutôt à des espaces tels que définis par [Egenhofer et Mark 95] dans leur théorie de

Géographie naïve¹. L'espace géographique est un espace à grande échelle qui va au-delà du corps humain et qui peut être représenté par différentes géométries et à différentes échelles.

Nous examinons dans ce qui suit, les difficultés que rencontrent les planificateurs humains dans de telles circonstances et sans assistance informatique.

Bien que la planification chez l'être humain soit une activité cognitive très raffinée basée sur l'anticipation et la schématisation [Hoc 87] (voir 2.1 du Chapitre 2), elle reste assez limitée. Notons tout d'abord que la planification devient une tâche très complexe pour n'importe quel problème réel, lorsqu'il s'agit d'opérer dans un environnement sur lequel le planificateur n'a aucun contrôle ou lorsqu'il est face à une incertitude sur les informations disponibles ou sur les actions des autres agents internes ou externes à l'environnement. Dans de telles circonstances, la planification est difficile à faire et les résultats de l'exécution sont aussi difficiles à prévoir [Lee et Fishwick 94].

Toutefois, ceci ne constitue pas la seule difficulté que rencontrent les planificateurs humains lors des désastres naturels. Le fait que l'environnement soit très dynamique, oblige le planificateur humain à essayer d'envisager le déroulement des futurs plans et à les ajuster au fur et à mesure que le plan s'exécute. Ceci implique de grandes capacités d'anticipation dont le planificateur humain ne dispose pas (voir 2.1 du Chapitre 2).

Enfin, il est à noter que la planification dans un espace géographique à grande échelle nécessite aussi une connaissance parfois assez détaillée et immédiate de l'espace géographique en question. Dans ce cadre, l'être humain semble avoir des capacités cognitives exceptionnelles. Par exemple, il est capable grâce à son intelligence et à son sens de l'anticipation, de compléter les informations géographiques (Le raisonnement géographique est typiquement basé sur des informations incomplètes, voir 2.1 du chapitre 2).

¹ La géographie naïve reflète la façon dont les personnes raisonnent sur l'espace géographique et le temps, que ce soit consciemment ou inconsciemment. La Géographie naïve est probablement la forme d'intelligence humaine la plus commune et la plus élémentaire. En effet, les gens emploient tous les jours des méthodes de raisonnement spatial pour déduire des informations à propos de leur environnement et la manière dont il évolue dans le temps.

Toutefois, et encore une fois, ces capacités sont limitées. En effet, ces planificateurs humains ont des capacités physiques et mentales limitées (p.ex. de perception) qui ne leur permettent pas plus qu'une vision partielle de la situation courante (généralement dans leur voisinage immédiat). Ensuite, la carte cognitive humaine² [Kuipers 82] semble être erronée et biaisée (voir 2.1 du Chapitre 2).

Dans la première section du chapitre 2 nous discutons en détail des caractéristiques et des limites de la cognition humaine. Toutefois, ce que nous pouvons indiquer pour le moment, c'est que le planificateur humain désirant planifier dans un environnement complexe, dynamique et à forte composante spatiale, a des points forts mais aussi des points faibles.

Pour faire face aux difficultés évoquées ci-dessus, les planificateurs humains ont certainement besoin d'assistance informatique. Nous présentons dans la prochaine section (1.2) les solutions informatiques candidates pour résoudre ce problème ainsi que leurs limitations. À la lumière de ces limitations, nous formulons respectivement dans les sections 1.3 et 1.4 notre question de recherche et nos objectifs de recherche.

1.2. Problématique

Afin de pouvoir mieux planifier dans les circonstances énoncées dans la section 1.1, il était nécessaire de trouver le moyen d'aider le planificateur humain à accomplir au mieux cette tâche. Avec l'expansion de l'informatique, les chercheurs ont essayé d'implémenter des solutions informatiques pour le problème de planification dans un environnement dynamique. L'approche DCP (*Distributed Continual Planning*) est peut-être la solution la plus récente et la plus appropriée puisqu'elle se base sur la continuité, ce qui rejoint l'idée de l'anticipation chez l'être humain. Nous examinons donc en premier lieu cette approche et nous relevons ses limites.

Dans un environnement dynamique, la planification classique monolithique de l'intelligence artificielle (IA) s'est avérée inefficace [Golden 97, Olawsky et Gini 90]. La

² Dans un sens général, une carte cognitive est une construction mentale qu'on utilise pour comprendre et connaître l'environnement.

planification à base d'agents logiciels est alors apparue comme une alternative sérieuse [Georgeff 83], [Conry et al., 89], [Von Martial 92], [Barruffi et al., 98]. La DCP est souvent basée sur l'utilisation d'un système multiagent. Pour tenir compte de la dispersion géographique des planificateurs, la DCP emploie des agents qui communiquent entre eux par échanges de messages distants ou en se déplaçant (agents mobiles). Nous examinons donc la technologie agent pour voir si dans son état actuel elle peut assurer une planification selon le contexte discuté en 1.1.

Vu que les données sur l'espace sont essentielles pour mieux planifier et pour assurer un meilleur réalisme aux plans élaborés, il est important de disposer de données spatiales plus fiables et plus précises. Nous faisons ainsi appel aux Systèmes d'Informations Géographiques (SIG) et nous discutons dans le point 1.2.3 les caractéristiques et les limites de cette technologie.

Enfin, et puisque nous nous intéressons à un problème de planification spatiale, recourir au raisonnement spatial semble inévitable. Ce raisonnement peut être aussi bien quantitatif que qualitatif. Nous présentons dans le point 1.2.4 un aperçu sur l'utilisation actuelle du raisonnement spatial qualitatif afin de voir si oui ou non nous pouvons utiliser les techniques actuelles pour planifier (en tenant compte des contraintes spatiales aussi bien quantitatives que qualitatives) des opérations de lutte lors des désastres naturels.

1.2.1 DCP

L'une des approches les plus appropriées aux environnements dynamiques et géographiquement distribués, est la Planification Distribuée en Continu (*Distributed Continual Planning DCP*) [Desjardins et al., 99]. Cette approche prend en compte la dispersion des planificateurs dans l'espace (et donc en particulier dans un espace géographique) ainsi que l'aspect dynamique de l'environnement. Pour gérer les changements qui peuvent survenir dans l'environnement, l'approche de l'entrelacement du processus de planification et du processus d'exécution est souvent employée [Ambros-Ingerson et Steel 88]. C'est ce processus de continuité qui est supposé assurer l'ajustement des plans.

La planification en continu semble inévitable dans un environnement réel, complexe et dynamique (Section 2.2.3 du chapitre 2). Toutefois, dans son état actuel, elle présente quelques inconvénients. L'un des problèmes majeurs de cette planification en continu classique, est que les plans sont construits étape par étape sans aucune garantie quant à l'aboutissement du plan à une solution finale acceptable. Cette approche devrait être évitée le plus possible surtout dans des situations critiques telles que les champs de bataille ou les feux de forêts. En effet, supposons que nous adoptons cette approche pour planifier la construction d'une ligne d'arrêt lors d'un feu de forêt. Si le plan (qui est ici le chemin que le bulldozer doit suivre) est construit tranche par tranche, il se pourrait très bien que le bulldozer arrive à un point (un cul de sac ou un obstacle naturel quelconque) à partir duquel il n'est plus possible d'avancer. Le plan est alors condamné à l'échec alors qu'on en a peut-être déjà exécuté une bonne partie.

Une deuxième limite aussi importante que la première est la communication. En effet, à cause de l'aspect distribué (même si dans cette thèse nous nous intéressons pas à l'aspect distribué de l'approche) de la planification, le flux de communications distantes est souvent important. Or, l'environnement, qui rappelons-le est dynamique et incertain, ne peut toujours assurer une communication fiable et à grande bande passante. Le problème est encore plus accentué dans le domaine militaire. En effet, un grand flux de communications distantes, en plus des problèmes de la bande passante et de la fiabilité, pose des défis considérables surtout en termes de sécurité. L'approche DCP basée en général sur des communications distantes entre les différents planificateurs doit donc absolument trouver des moyens efficaces pour réduire le flux de ces communications.

Une autre limite moins critique de la DCP mais qui empêche cependant une meilleure continuité de la planification, est le fait qu'on soit obligé d'attendre l'exécution d'une étape du plan avant de pouvoir entamer la planification de l'étape suivante. Il nous semble que cet aspect pourrait très bien être amélioré afin d'offrir de meilleurs résultats.

1.2.2 Technologie Agent

La technologie agent a fait ses preuves dans plusieurs domaines comme le commerce électronique et dernièrement dans les problèmes de planification. De plus, l'utilisation du paradigme agent est très bénéfique pour le raisonnement spatial. Kray énumère dans ce

cadre plusieurs avantages de l'utilisation des agents et des systèmes multiagent pour un tel raisonnement [Kray 01]. Nous nous sommes donc demandés si cette technologie, dans son état actuel, pouvait assurer une planification efficace dans un espace géographique à grande échelle. Malheureusement, nos recherches indiquent que cette technologie a encore plusieurs obstacles à franchir avant d'y arriver. Nous présentons ci-dessous les principaux obstacles :

1. ***Peu d'anticipation sur l'exécution du plan*** : les agents n'ont pas la capacité cognitive humaine pour exécuter les plans alternatifs dans un « espace de modélisation » afin de choisir le meilleur plan à exécuter dans le monde réel, ni la vision cognitive humaine pour faire le choix adéquat ou anticiper les problèmes. Le choix du plan se base par contre sur la satisfaction des contraintes et sur des calculs métriques sur le coût de chaque plan alternatif. Appliquées toutes seules, ces méthodes d'évaluation, bien qu'elles soient formelles, ne sont pas très appropriées dans un problème de planification complexe. En effet, elles ne bénéficient pas des capacités cognitives de planification humaine (anticipation et jugement).
2. ***La planification automatique impose au lieu de proposer*** : Un plan proposé par la machine, bien que fondé sur des contraintes réelles et des calculs formels, est souvent refusé par la personne responsable ou par le décideur humain car il ne respecte pas certaines doctrines, ne semble pas très faisable par expérience, ne convainc pas le décideur ou tout simplement parce que le décideur y voit (anticipe) un problème. Nous pensons ainsi qu'ignorer la participation cognitive humaine lors de la planification d'un problème complexe peut mener le plan à l'échec. Dans certaines applications de planification à base d'agents telles que les applications militaires ou des situations d'urgence, les agents logiciels planificateurs ne devraient pas imposer un plan mais plutôt en suggérer un. L'adoption ou le rejet du plan devrait rester à la charge d'un décideur humain qui interagit avec le système.
3. ***Utilisation incomplète du paradigme agent dans le raisonnement spatial*** : Dans le raisonnement spatial, le terme « agent » est souvent utilisé pour décrire des

systemes qui exécutent des tâches spécifiques comme « *route advice* » dans [Rogers et al., 99] ou *incremental route instructions* dans [Maaß et al., 95]. Ceci rejoint la définition d'un agent logiciel, telle que formulée par [Wooldridge 99] : « *Des systèmes informatiques capables de mener des actions autonomes dans un certain environnement afin de réaliser les objectifs pour lesquels ils ont été conçus* ». Toutefois, ces systèmes de raisonnement spatial à base d'agents ne peuvent être considérés comme des systèmes multiagents [Kray 01]. En effet, les systèmes multiagents sont conçus et implémentés sous forme de plusieurs agents interagissant [Jennings et al., 98]. Dans le raisonnement spatial, par contre, les avantages des vrais systèmes multiagents ne sont pas encore complètement présents [Kray 01] puisque l'interaction entre agents est très faible sinon absente. Plus récemment, Drogoul et ses collègues [Drogoul et al., 02] ont démontré que d'après leurs études, les propriétés des agents (autonomie, proactivité et interaction) définies à un niveau métaphorique, ne sont pas traduites en propriétés computationnelles. Les agents computationnels tels que décrits par [Jennings 00a], ne sont pas utilisés par la Simulation Orientée Agent.

4. ***Faible sensibilité géospatiale***: La capacité cognitive spatiale de l'humain est limitée (voir Section 2.1 du Chapitre 2). Dans un contexte de planification où les contraintes géographiques sont importantes (comme dans l'exemple de la lutte contre les feux de forêts), les agents planificateurs devraient pouvoir combler en partie cette limitation humaine. Les agents logiciels actuels commencent à être sensibles à la localisation grâce à l'arrivée de la technologie du positionnement géoréférencé tel que le GPS. Ils sont désormais capables de déterminer leurs localisations géographiques exactes et d'en tenir compte lors de la prise de décision. Cependant, la sensibilité géospatiale des agents logiciels est encore loin de ce qu'on pourrait espérer.

De plus en plus de chercheurs s'intéressent à la sensibilité spatiale des agents. [Minsky 86] était l'un des premiers à parler des agents spatiaux. [Rodrigues et Raper 97] ont ensuite donné une définition plus « moderne » des agents spatiaux. [Bandini et al., 02] ont dernièrement proposé un modèle pour caractériser les

agents sensibles à l'espace. Quant à la communauté géomatique, elle commence petit à petit à voir l'intérêt des agents dans la résolution des problèmes spatiaux [Andrew et al., 01]. Cependant, la plupart de ces travaux n'ont pas vraiment donné de solutions pratiques et efficaces pour offrir aux agents une sensibilité géospatiale satisfaisante. Cette insatisfaction est due essentiellement à trois facteurs :

- a. Sensibilité spatiale « trop » simple : Beaucoup de travaux qui parlent de la sensibilité à l'espace prennent le terme « espace » comme référant aux données spatiales « classiques » telles que la distance, l'éloignement, le rapprochement, la taille, et non pas à toutes les données géographiques qu'on peut obtenir sur un terrain réel telles que le degré de pente, l'altitude, le type du sol, le type de la végétation, largeur du lac... bref, toutes les données qu'on retrouve dans un Système d'Information Géographique SIG.
- b. La taille de l'espace considéré : Les travaux actuels sur la sensibilité des agents logiciels à l'espace, ne considèrent que les espaces à petites échelles et non pas les espaces géographiques à grandes échelles comme définis par la géographie naïve. Il est clair que les problèmes ne sont pas les mêmes dans les deux types d'espaces. Dans une application nécessitant une planification contrainte par l'espace, la connaissance géospatiale détaillée est primordiale et l'espace est souvent à grande échelle. Malheureusement, les agents logiciels actuels ne permettent pas encore de tenir compte de la plupart des données géospatiales d'un espace à grande échelle et de les utiliser pour planifier.
- c. Manque de réalisme : les systèmes à base d'agents logiciels qui planifient (ou raisonnent de manière plus générale) dans un espace géographique, manquent souvent de réalisme. L'espace géographique, qui par nature est très complexe, est souvent réduit à un espace bidimensionnel avec deux ou trois propriétés caractéristiques. Les détails sur l'espace sont souvent négligés, ce qui entraîne un manque de réalisme. Les solutions proposées

sont ainsi souvent restées dans le cadre de l'expérimentation en laboratoire. En effet, plusieurs systèmes de planifications à base d'agents dans des espaces géographiques à grande échelle, n'implémentent que des scénarios très simples ou encore un jeu d'illustration. Le jeu « *Capture the flag* » en est un exemple caractéristique [Atkin et al., 98]. Dans [Hanks et al., 93], Pollack attire l'attention sur ce problème de manque de réalisme dans les recherches en IA en général.

Dans l'exemple de la lutte contre les incendies des forêts, il serait judicieux par exemple si les agents logiciels pouvaient « percevoir » les caractéristiques géographiques de la région qui entoure le feu afin de proposer les meilleurs endroits possibles pour construire les lignes d'arrêt³. Un endroit favorable à la construction d'une ligne d'arrêt doit avoir un point d'ancrage et doit surtout être composé de végétation à essence légère. D'autres contraintes peuvent aussi être considérées telles que le degré de la pente et la facilité d'accès. Cette perception de l'espace doit être précise afin d'offrir plus de réalisme à la planification. Dans [Moulin et Sahli 02] nous avons déjà attiré l'attention sur la nécessité d'inclure la « sensibilité géospatiale » à la liste des capacités connues des agents. En effet, les agents seront bientôt de plus en plus sollicités pour résoudre des problèmes où l'aspect spatial (ou géospatial) est très présent. Agarwal et Abrahart sont parmi les rares chercheurs qui se sont intéressés à ces aspects et se sont prononcés sur l'importance d'incorporer dans les agents des capacités de raisonnement ainsi qu'un framework cognitif pour la géométrie, la topologie et les relations spatio-temporelles [Agarwal et Abrahart 03].

1.2.3 SIG

Selon le comité fédéral de coordination inter-agences pour la cartographie numérique [FICCDC, 88], un système d'information géographique est un "*système informatique de matériels, de logiciels, et de processus conçus pour permettre la collecte, la gestion, la*

³ Technique utilisée lors d'un feu de forêt majeur. Elle peut être réalisée en aménageant une ligne d'arrêt loin de l'incendie mais dans son chemin, à l'aide d'outils manuels et des bulldozers, tout en utilisant des barrières naturelles comme des lacs, des rivières et des affleurements rocheux. Le combustible qui se trouve devant l'incendie peut alors être enlevé en le brûlant à partir de la ligne d'arrêt en direction de l'incendie.

manipulation, l'analyse, la modélisation, et l'affichage de données à référence spatiale afin de résoudre des problèmes complexes d'aménagement et de gestion".

À un niveau technique, les SIGs ont certaines limites surtout lorsque le temps d'exécution est important. En effet, dans le contexte du monde réel, quand un utilisateur interroge un SIG pour obtenir de l'information ou pour demander des instructions de navigation par exemple, il est souvent confronté à la complexité du calcul du raisonnement spatial [Cohen 97]. Bien que des méthodes de résolution en un temps polynomial existent, le grand nombre d'arguments (p.ex. le nombre d'objets dans un SIG) et le temps requis pour déduire de l'information qualitative à partir des données métriques (p.ex. exécuter des opérations sur des données bitmap), nécessitent l'application d'heuristiques ou à la limite des stratégies d'adaptation comme les algorithmes *anytime* [Zilberstein 93] afin de rendre des réponses dans un temps acceptable.

À un niveau plus cognitif, les SIGs ont également certaines limites. En effet, le besoin d'intégrer la connaissance et le raisonnement de la géographie naïve dans les SIG est devenu une nécessité. Les concepts et les méthodes qu'utilisent les humains pour inférer de l'information sur l'espace géographique et le temps deviennent de plus en plus importants pour une meilleure interaction entre les utilisateurs humains et les SIGs. Bien que les inférences spatiales semblent triviales pour une personne, elles sont cependant extrêmement difficiles à formaliser pour pouvoir être implémentées sur ordinateur. Les méthodes actuelles pour dériver l'information spatio-temporelle sur l'espace géographique sont encore limitées. Egenhofer et Mark [Egenhofer et Mark 95] voient un grand fossé entre ce que l'humain veut faire avec un SIG et ce que le SIG peut offrir en termes de concepts spatiaux. Il est ainsi nécessaire d'intégrer les concepts spatio-temporels humains dans les SIGs et d'imiter le raisonnement humain. Egenhofer et Mark affirment qu'il n'y a pas de modèles pour le traitement des différents concepts spatiaux qui sont cognitivement plausibles [Egenhofer et Mark 95]. Ils voient ainsi la nécessité de trouver des méthodes avancées pour extraire des résultats à partir des études cognitives comme le fait que la nature des erreurs dans les cartes cognitives sont souvent métriques et rarement topologiques [Lynch 60] ou encore comment les structures topologiques sont utilisées pour le raisonnement spatial humain [Stevens et Coupe 78]. D'autres chercheurs

ont identifié différents types d'espaces avec des méthodes d'inférence correspondantes [Piaget et Inhelder 67], [Golledge 78] et [Couclelis et Gale 86]. Egenhofer et Mark affirment enfin que si les SIGs arrivent à raisonner comme un expert humain, ils seront en mesure de fournir des outils plus efficaces pour accomplir des tâches faisant appel au raisonnement spatial.

Pour résumer, on dira que dans le monde réel, les décideurs humains ne sont pas forcément des experts en SIG. Ils utilisent souvent des termes linguistiques (ou graphiques) pour exprimer les contraintes qu'ils veulent imposer au problème. Ils ont besoin donc d'outils intuitifs et faciles à utiliser lors du processus de prise de décision ou de planification, sans avoir à interpréter par eux-mêmes de grandes quantités de données spatiales.

1.2.4. Les techniques du raisonnement spatial

L'un des problèmes les plus connus nécessitant un raisonnement spatial quantitatif, est la recherche du chemin ou le *Pathfinding*. Plusieurs algorithmes ont été proposés pour ce genre de problème (Chapitre 5). La difficulté majeure réside dans le fait que certains problèmes requièrent un raisonnement spatial qualitatif car dans ce cas, les solutions ne sont pas évidentes. Toutefois, plusieurs chercheurs affirment que certaines techniques de raisonnement spatial pourraient aider à surmonter cette difficulté [Forbus 95; Forbus et al., 91].

Plusieurs chercheurs en IA affirment que le raisonnement spatial quantitatif est insuffisant pour résoudre certains problèmes qui sont plus complexes. Le besoin est plus tangible pour les problèmes de recherche de chemin. Le problème classique de stratégie *Massed Fires*, est souvent cité en exemple pour démontrer le besoin de nouvelles approches autres que celles basées sur les algorithmes classiques de l'IA. Dans [Forbus et al., 01], les auteurs ont étudié le jeu Diablo II pour démontrer les faiblesses des approches classiques pour résoudre des problèmes plus complexes de Pathfinding (ou recherche de chemin). Ils expliquent ces lacunes par le fait que les algorithmes classiques de Pathfinding ne sont pas capables de considérer des contraintes qualitatives.

Selon Forbus et ses collègues [Forbus et al., 01], de tels problèmes que manifeste l'IA vis-à-vis du raisonnement spatial, peuvent être mieux abordés en utilisant des techniques basées sur le raisonnement spatial qualitatif [Hernandez 94].

Dans les dernières années, certains travaux se sont intéressés à cet aspect du raisonnement spatial comme dans [Forbus 95] et [Forbus et al., 01]. Toutefois, ces recherches sont pour la plupart orientées vers les jeux. A notre connaissance, très peu de travaux se sont intéressés au problème de raisonnement spatial qualitatif dans des environnements géographiques réels et à grande échelle et plus précisément au problème de Pathfinding sous de telles contraintes.

1.3. Énoncé de la problématique de recherche

Il est clair qu'en se situant dans le contexte cité dans la section 1.1, il est très difficile d'implémenter une planification à base d'agents logiciels dans un environnement réel, complexe, dynamique et à forte composante spatiale. En effet, d'une part, les agents logiciels semblent avoir du mal à planifier de façon analogue au processus humain : ils n'ont pas les capacités cognitives humaines de planification (surtout l'anticipation et la simulation). Ceci explique pourquoi plusieurs systèmes informatiques de planification, dédiés surtout aux situations de crises, ont été abandonnés par les utilisateurs au profit de méthodes manuelles de planification (à l'aide de cartes sur papier) [Sauvagnargues et al., 00]. D'autre part, bien que les agents logiciels soient de plus en plus utilisés dans des applications spatiales, le paradigme agent n'est pas encore exploité comme il faut. En effet, la plupart des recherches sur le raisonnement spatial à base d'agents ne font pas appel à la notion de système multiagent. Il semble aussi que les agents logiciels n'ont pas encore les moyens de percevoir, de manipuler et de raisonner sur les données géospatiales dans un monde réel et complexe. La plupart des recherches actuelles portent sur des espaces restreints, des scénarios trop simplifiés et des données spatiales très réduites. Ceci explique peut-être l'absence d'applications à composantes spatiales performantes dans des situations réelles telles que les situations de crise. L'approche DCP (plus précisément l'aspect continuité) actuelle a des défauts (p.ex. le succès du plan final n'est jamais garanti au départ) qui peuvent être critiques pour le genre de situations auxquelles les décideurs sont confrontés (situations réelles et critiques). Il est à noter ici que sur les

deux aspects que renferme l'approche DCP (la distribution⁴ et la continuité), nous ne nous intéressons qu'au problème de continuité du processus de planification. Les techniques basées sur le raisonnement spatial qualitatif ne sont pas encore assez évoluées pour permettre de résoudre des problèmes spatiaux dans un environnement géographique réel et à grande échelle. Toujours dans le contexte du 1.1, les données spatiales sont essentielles pour la planification et doivent donc absolument être prises en compte. Pratiquement, ces données sont souvent fournies par des SIGs qui malheureusement ne sont pas bien exploitées surtout à cause du temps d'accès important qu'ils nécessitent (ce qui n'est pas acceptable pour des applications critiques à temps réel). Nous disposons ainsi de données spatiales « précieuses » (dans les SIGs) dont nous ne pouvons pas profiter pour planifier !

La problématique de cette thèse peut alors s'énoncer comme suit :

Quelles technologies et procédures faut-il utiliser et comment, pour

- Permettre aux agents logiciels de planifier (avec des capacités de raisonnement spatial aussi bien quantitatives que qualitatives) dans un environnement réel, complexe, dynamique et à forte composante spatiale (dans un espace géographique à grande échelle);
- Faire le rapprochement entre les agents logiciels planificateurs et les capacités cognitives humaines (surtout d'anticipation et de raisonnement spatial);
- Profiter des données SIG afin de réaliser un certain degré de réalisme.

1.4. Objectifs de recherche

Dans le cadre du présent travail, nous nous intéressons à la planification dans un environnement dynamique et caractérisé par un espace géographique à grande échelle. Nous visons essentiellement à assister les planificateurs humains (qui interviennent dans de tels environnements) en se basant sur les agents logiciels. Nous comptons ainsi atteindre les sous-objectifs suivants :

- Offrir une certaine sensibilité spatiale aux agents logiciels planificateurs.

⁴ La considération de l'aspect distribué de l'approche suggère l'examen de plusieurs sous-domaines de recherches tels que la communication, la négociation et la coordination des agents logiciels, etc., ce qui ne

- Planifier de façon plus efficace dans un environnement dynamique et complexe.
- Rallier les capacités cognitives des planificateurs humains à celles des planificateurs logiciels pour construire de meilleurs plans (plus réalistes, plus efficaces et plus précis).
- Pouvoir assister le planificateur humain avec du raisonnement spatial qualitatif (et non seulement quantitatif).
- Surmonter les lacunes de l'approche DCP (l'aspect de continuité), au moins dans le contexte précis de notre travail.

Il est important ici de préciser les aspects de la planification sur lesquels nous allons travailler. Dans le cas d'une planification dans un environnement dynamique à forte composante spatiale, nous avons relevé plusieurs aspects pouvant être sujets de recherche. La figure suivante dresse une liste (non exhaustive) de ces éléments.

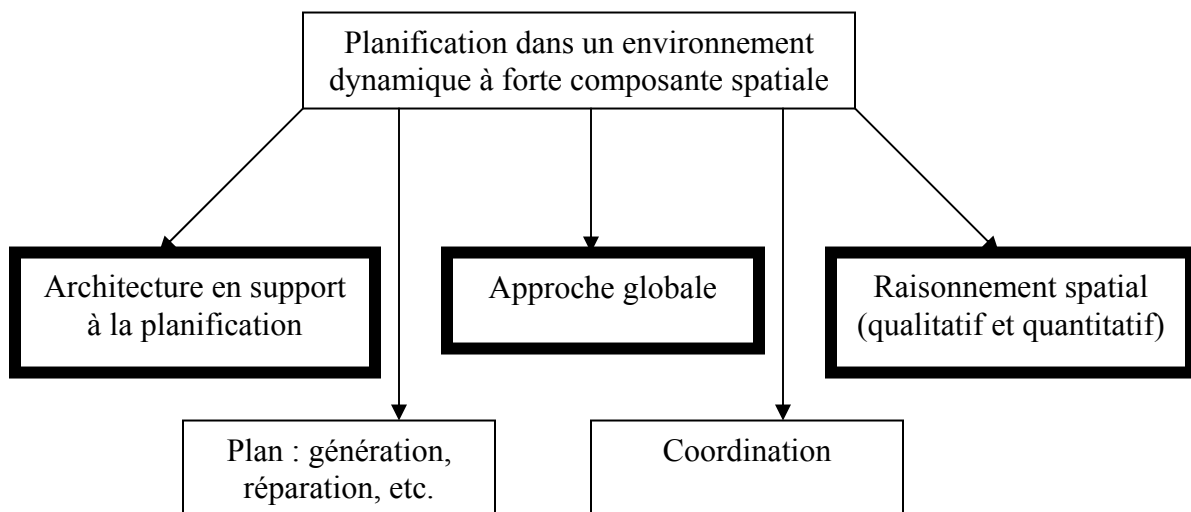


Figure 1.1. Différents aspects de la recherche en « Planification dans un environnement dynamique à forte composante spatiale »

- *Architecture en support à la planification* : C'est l'architecture qui va supporter l'approche globale, la coordination entre les planificateurs, bref, tout le processus de planification.
- *Approche globale* : C'est l'idée générale derrière le processus de planification. La DCP est l'approche qui nous concerne dans ce travail. Dans cette DCP, il n'y a que l'aspect en continu qui sera étudié dans cette thèse.
- *Raisonnement spatial (qualitatif et quantitatif)* : C'est l'étude du raisonnement sous contraintes spatiales. Comme mentionné précédemment (dans l'énoncé de la problématique), nous nous intéressons dans le cadre de cette thèse à quelques aspects du raisonnement spatial qualitatif.
- *Plan* : La recherche autour du « plan » concerne sa génération, sa réparation, etc.
- *Coordination* : Lorsqu'il y a plusieurs planificateurs en jeu, la coordination est une étape primordiale dans le processus de la planification.

Dans cette thèse, nous ne nous intéressons qu'aux trois aspects suivants (entourés en gras dans la Figure 1.1) : l'architecture, l'approche et le raisonnement spatial. Tout ce qui concerne le plan (sa génération, son contenu, sa validation, etc.) ou la coordination entre les différents planificateurs, ne fait pas l'objet de recherche dans cette thèse.

1.5. Pistes de solutions et méthodologie de recherche

Au début de cette thèse, nous nous sommes intéressé à la planification dans un environnement dynamique et plus particulièrement à l'utilisation de la planification en continu. Nous avons mis l'emphase sur les environnements réels, dynamiques et à forte composante spatiale. Vu les difficultés que les planificateurs humains rencontrent dans ce genre de situations, nous avons pensé à transposer le problème réel dans un monde virtuel. Cela revenait en fait à construire un environnement de simulation parallèle au monde réel. Pour assurer un certain réalisme à nos applications futures, nous avons eu besoin de gérer des données SIG. Cela revenait en fait à faire de la géosimulation.

Nous nous sommes ensuite demandé si l'utilisation des agents logiciels pourrait aider à surmonter les difficultés des planificateurs humains. C'est ainsi que nous avons commencé à étudier l'utilisation des agents logiciels pour planifier dans des environnements dynamiques à forte composante spatiale. Nous avons alors noté que très peu de travaux font appel au paradigme agent pour résoudre de tels problèmes de planification. La plupart des travaux qui ont essayé d'appliquer les agents pour des problèmes similaires aux nôtres présentaient plusieurs limites : mauvaise utilisation du paradigme agent, manque de réalisme, non fiabilité des données, application difficile dans le monde réel, négligence de l'aspect spatial de l'environnement, etc.

Nous avons mené aussi une étude sur la cognition spatiale humaine et nous avons alors constaté ses limites et ses points forts. Afin d'éviter les solutions « tout agent » (n'impliquant pas l'humain) qui ne marchent malheureusement pas et pour assurer l'ajustement des plans dû au dynamisme de l'environnement, nous avons opté pour l'implémentation d'une planification qui soit à la fois collaborative (agent/humain) et en continu. Nous avons repris alors les principes de l'approche DCP et nous avons essayé de l'implémenter par des agents logiciels, et ce de façon plus efficace. Nous avons alors utilisé l'environnement virtuel qui est similaire au monde réel et dans lequel les agents peuvent interagir et planifier des actions au profit des intervenants humains qui sont sur le terrain. Nous avons ainsi proposé l'architecture ENCASMA (Chapitre 4).

Ensuite, pour remédier au fait que la technologie agent actuelle traite mal l'espace, nous avons essayé de donner aux agents logiciels de nouvelles capacités spatiales.

Nous avons également remarqué que les applications qui ont le plus besoin de la planification en continu sont les applications militaires, les opérations de secours et de gestion des crises et les applications aérospatiales. Suite à l'examen de plusieurs applications possibles, nous avons trouvé en l'opération de lutte contre les feux de forêt un bon exemple d'illustration.

En essayant d'appliquer cette solution sur les cas des feux de forêts et en examinant de plus près les spécificités du domaine, nous avons pu proposer une nouvelle approche de Pathfinding (recherche de chemin) basée sur la solution décrite plus haut et qui traite des

cas complexes qui ne peuvent être résolus avec les approches classiques couramment appliquées au problème de Pathfinding.

Étant donné que le domaine d'application est critique et réel, il nous était difficile de valider notre approche sur le terrain lors d'un vrai feu. Nous avons quand même essayé de la valider autrement (Chapitre 8) à l'aide des experts du domaine.

1.6. Plan de la thèse

Cette thèse est composée de 8 chapitres (sans compter le présent chapitre) :

Le Chapitre 2 (« État de l'art ») donne un aperçu sur les différentes techniques que nous comptons utiliser pour résoudre le problème énoncé au Chapitre 1: la simulation par agent, la planification par agent, la planification par simulation, les SIG et la géosimulation. Les limites de ces techniques sont citées au fur et à mesure.

Le Chapitre 3 (« Domaine d'application : les feux de forêts ») présente le domaine d'application : le feu de forêt, qui est un cas typique d'un environnement réel, dynamique et à grande échelle et où la planification nécessite une bonne connaissance de l'espace.

Le Chapitre 4 (« Architecture ENCASMA ») propose une architecture basée sur la simulation par agent capable de résoudre le problème de planification qui nous intéresse. C'est en quelque sorte le cœur de la solution.

Le Chapitre 5 (« Pathfinding réel par simulation orientée agent ») traite un problème particulier de planification qui est la recherche de chemin ou le *Pathfinding*. Ce chapitre montre comment utiliser l'architecture ENCASMA et plus particulièrement la simulation par agent pour résoudre certains problèmes particuliers de Pathfinding.

Le Chapitre 6 (« ENCASMA au support de la DCP ») montre comment l'architecture ENCASMA peut supporter l'implémentation du processus de continuité de la DCP pour le genre de problème étudié.

Le Chapitre 7 (« Réalisation ») donne quelques détails sur l'implémentation de l'application de la construction de la ligne d'arrêt lors d'un feu de forêt.

Le Chapitre 8 (« Résultats et Validation ») expose les résultats de notre prototype et discute de la validation de l'approche.

Le Chapitre 9 conclut enfin cette thèse en résumant nos contributions et en identifiant les directions pour les travaux futurs.

CHAPITRE 2

État de l'art

Dans ce chapitre, nous faisons tout d'abord un retour plus détaillé (Section 2.1) sur les problèmes liés à la cognition humaine afin de montrer ses points forts et ses points faibles.

Comme énoncé au chapitre précédent, nous nous proposons dans cette thèse de résoudre un problème de planification dans un environnement dynamique et à grande échelle en utilisant des agents logiciels. La deuxième section de ce chapitre donne un bref aperçu du domaine.

Nous avons suggéré comme piste de recherche (Section 1.5 du Chapitre 1), l'utilisation de la simulation comme moyen de planification. C'est la raison pour laquelle, nous consacrons la Section 2.3 à la définition des principes de base liés à la simulation et plus particulièrement à la simulation orientée agent. Nous faisons un survol des principaux travaux qui ont utilisé la technique de planification par simulation dans la Section 2.4. Nous avons déjà dit que l'utilisation des données SIG est incontournable pour donner à la résolution d'un problème tel que la lutte contre les feux de forêt un plus grand degré de réalisme. Nous présentons alors dans la Section 2.5 un petit aperçu sur l'utilisation des SIG dans la lutte contre les catastrophes et plus particulièrement dans la lutte contre les feux de forêts. Lorsqu'on fait une simulation basée sur des données SIG, on parle alors de géosimulation. La Section 2.6 fait le point sur cette nouvelle technique de simulation avec un intérêt particulier porté à l'utilisation des agents logiciels dans la géosimulation.

2.1. Cognition humaine

Nous nous intéressons donc aux problèmes de planification complexe dans un environnement dynamique et à fortes contraintes géospatiales. Il est nécessaire de comprendre comment les êtres humains planifient dans de telles circonstances. La sous-

section 2.1.1 montre comment les humains planifient en général, ensuite la sous-section 2.1.2 fait une synthèse du même comportement humain mais dans un contexte spatial.

2.1.1. La planification spatiale chez l'être humain

2.1.1.1. La planification cognitive

Craik, un psychologue, a dit en 1943 : *“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 43]

Dès 1943 Craik annonce ainsi le fonctionnement humain lors de la planification : la planification est une activité cognitive très raffinée impliquant une formulation mentale des futurs états du monde. Il conjecture que 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. Planifier c'est essayer des choses dans sa tête en utilisant un modèle interne du monde [Craik 43], [Dennett 78] et [Sutton et Barto 81]. [Norman et Rumelhart 75] affirment que toute la force du modèle mental s'appuie sur des capacités de simulation chez l'humain.

Selon plusieurs chercheurs tel que [Goel 02], la planification exige de l'agent humain de tracer un chemin entre A et B dans un « espace de modélisation » (*space modeling*) (représentation mentale dans la tête de l'être humain) sans avoir à « se heurter » au monde réel. Le sens même de la planification est de prévoir les conséquences d'une action avant son exécution. La seule façon de faire est donc d'exécuter l'action dans un « espace de modélisation » et d'observer les conséquences. Si le résultat est satisfaisant, le plan est adopté et exécuté dans le monde réel, sinon le plan est révisé.

En des termes plus précis, [Hoc 87] affirme que la planification humaine est basée sur l'anticipation et la schématisation. Ces deux aspects ont été longtemps étudiés par les psychologues et ont été par la suite l'objet de formalisations par les chercheurs en IA.

Quand un individu humain est face à une nouvelle situation, il est capable de détecter des similarités avec des situations connues et d'anticiper des événements ou des propriétés qui ne sont pas encore complètement identifiés [Hoc 87]. Grâce à l'anticipation, l'être humain est capable de tenir compte des événements futurs lors de sa prise de décision. Il anticipe de façon schématique dans la majorité des cas. Ceci est possible en faisant appel aux connaissances acquises dans le passé. L'anticipation est fortement liée à la schématisation surtout dans les cas de résolution complexes. La notion de schéma a été introduite par [Bartlett 32] comme une « organisation active des réactions passées ou des expériences antérieures ». Les schémas sont un moyen d'inférence pour anticiper à partir des cas précédents. Selon [Piaget 47], agir dans un monde présuppose l'assimilation d'un schéma formulé au préalable. L'interprétation de la réalité est basée sur des attentes relativement à ce que la réalité sera. La formalisation des schémas a donné naissance à la notion de *frame* [Minsky 75] (dans le domaine de la reconnaissance des patterns) et de script [Shank et Abelson 79].

Prenons un exemple concret de planification par des agents humains (des pompiers) qui essayent d'encercler un feu de forêt en construisant des lignes d'arrêt tout autour de la zone sinistrée (Section 3.3.3.5 du Chapitre 3). Le but est de trouver les meilleurs endroits pour construire ces lignes (à l'aide de bulldozers) tout en tenant compte de la progression des feux (le feu ne doit pas traverser la ligne avant sa construction), des conditions météorologiques et des types d'essences d'arbres présentes, et de construire par la suite ces lignes

Afin de trouver un plan d'action, chaque pompier a besoin de comprendre le problème. Pour ce faire, il doit se baser sur plusieurs mécanismes [Hoc 87] dont :

- Les mécanismes de perception qui permettent la reconnaissance des patterns. Le pompier perçoit de son entourage ce qui constitue sa vision locale de la situation. Cette vision va influencer sa compréhension de la situation.
- La mémoire qui fournit les structures nécessaires pour la reconnaissance. (On pense que les grands joueurs d'échec ont un répertoire de plusieurs dizaines de milliers de positions [Chase et Simon 73]).

Le pompier construit ainsi un plan plausible. Il l'exécute ensuite mentalement dans son « espace de modélisation » afin de vérifier son efficacité en se basant sur son expérience et sur des cas antérieurs semblables qu'il connaît, ceci correspond à la faculté d'anticipation de l'être humain. Le pompier vérifie ainsi plusieurs plans alternatifs et choisit celui qui a le meilleur résultat. S'il s'agit d'une planification complexe, le pompier utilise un mécanisme cognitif de schématisation afin d'anticiper les éventuels échecs.

Le plan choisi est ensuite exécuté dans le monde réel. Nous supposons pour des raisons d'illustration que le plan adopté par le pompier P1 est de construire une ligne entre le point A et B dans un temps T. Après un temps $T' < T$ le pompier P1 se rend compte qu'il ne pourra jamais arriver au point B dans le délai fixé. Cette estimation de la part du pompier est le fruit de sa capacité d'anticipation. Le pompier est devant un cas où le plan ne s'est pas exécuté dans le monde réel comme prévu dans son espace de modélisation. Il doit alors replanifier.

Nous remarquons donc que le pompier n'attend pas toujours de constater l'échec de son plan au dernier moment, mais qu'il est capable de reconnaître que le plan va échouer si aucune mesure corrective n'est prise. Il s'agit ici d'une planification continue : l'agent (ici humain) est capable de replanifier lors du processus d'exécution réelle si jamais le plan échoue ou se dirige vers un échec. Ce comportement humain découle du fait que l'anticipation humaine peut très bien porter sur des événements ou des propriétés qui ne sont pas encore complètement identifiés [Hoc 87].

2.1.1.2. Cas d'un environnement spatial

Dans le cadre de cette thèse, nous nous intéressons plutôt à des espaces tels que définis par [Egenhofer et Mark 95] dans leur théorie de Géographie naïve⁵. L'espace géographique est un espace à grande échelle qui va au-delà du corps humain et qui peut être représenté par différentes géométries et à différentes échelles. Cette définition rejoint

⁵ La géographie naïve reflète la façon dont les personnes raisonnent sur l'espace géographique et le temps, que ce soit consciemment ou inconsciemment. La Géographie naïve est probablement la forme d'intelligence humaine la plus commune et la plus élémentaire. En effet, les gens emploient tous les jours

les points de vue de certains auteurs [Kuipers 78] et [Kuipers et Levitt 88] qui affirment qu'un tel espace ne peut pas être observé d'un seul point de vue. L'espace géographique comprend donc plus d'éléments qu'une personne peut voir. Il peut être un hôtel avec toutes ses chambres, une ville avec toutes ses routes et constructions, l'Europe avec toutes ses montagnes, lacs et rivières, etc. On explore donc l'espace géographique en y naviguant et on le modélise selon divers points de vue qu'on rassemble (mentalement) comme un puzzle. Ceci fait de l'espace géographique un espace différent de l'espace à petite échelle dans lequel les objets sont considérés comme manipulables et où l'observateur peut déplacer ces objets, les toucher et les mesurer pour avoir toute l'information nécessaire.

Pour planifier dans de tels espaces (à grandes échelles), l'être humain s'appuie sur une représentation mentale spatiale. La représentation mentale dont parle Craik [Craik 43] est dans ce cas plus complexe (que dans le cas d'un espace restreint) et fait appel à d'autres notions. En effet, la capacité de planifier et d'exécuter des mouvements dans un environnement spatial semble requérir qu'on possède une carte cognitive⁶ [Kuipers 82] de cet environnement en plus de tenir compte des stimuli directement disponibles par le système de perception. Il est en général admis que la carte cognitive est une construction mentale qui influence les comportements et qu'en examinant la représentation externe des connaissances de notre carte cognitive, nous pouvons comprendre les prises de décisions immédiates ainsi que les comportements futurs [Kitchin 94].

Le raisonnement géographique est typiquement basé sur des informations incomplètes. Cependant, les humains sont capables d'en tirer des conclusions assez précises en complétant l'information intelligemment ou en appliquant certaines règles par défaut basées souvent sur le bon sens. Des études cognitives ont démontré que l'être humain emploie des schémas organisés de façon hiérarchique pour raisonner sur l'espace

des méthodes de raisonnement spatial pour déduire des informations à propos de leur environnement et la manière dont il évolue dans le temps.

⁶ Il s'agit d'un processus composé d'une série de transformations psychologiques par lesquelles un individu acquiert, stocke et décode de l'information sur l'emplacement et les attributs d'un phénomène, dans son environnement spatial de tous les jours. Dans un sens général, une carte cognitive est une construction mentale qu'on utilise pour comprendre et connaître l'environnement. Cela suppose que les gens stockent de l'information sur leur environnement, qu'ils utilisent ensuite pour prendre des décisions spatiales.

géographique et compenser le manque d'information [Hirtle et Jonides 85]; [McNamara et al., 89].

2.1.2. Les limites de la planification cognitive dans un espace géographique

Dans cette section nous présentons les différentes limites cognitives humaines citées dans la littérature et qui démontrent par conséquent le besoin du recours à l'informatique.

2.1.2.1. Les limites de la planification cognitive

L'être humain se base sur des mécanismes cognitifs sophistiqués tels que l'anticipation et la schématisation pour planifier.

Cependant, ces capacités cognitives humaines sont limitées. Reprenons l'exemple des pompiers luttant contre un feu de forêt. Un pompier qui essaye de planifier une ligne d'arrêt, est confronté à plusieurs limitations :

- Limitation dans la capacité de calcul : Pour trouver un bon plan, il faut connaître les estimations de l'avancement du feu avec précision, les prévisions météorologiques, etc. Si l'espace de recherche d'un plan est grand, il est quasi impossible pour un humain de vérifier toutes les alternatives possibles. Il est alors contraint à se limiter à quelques plans possibles. Il pourrait ainsi passer à côté de la meilleure alternative.
- Limitation dans la communication : Lorsqu'il s'agit de négocier un plan entre plusieurs pompiers, le flux de communication est limité.
- Limitations physiques : Les pompiers ne peuvent pas surveiller tous les paramètres externes qui influencent la progression des feux et l'exécution du plan de façon permanente.
- Probabilité d'erreur : Le pompier peut commettre des erreurs de calcul ou d'inattention.

Ces limites sont dues à la nature humaine qui fait qu'on n'a pas les mêmes capacités computationnelles que la machine. En plus, lorsqu'il s'agit de planifier, une autre capacité essentielle fait défaut chez l'être humain : c'est la capacité de simuler des phénomènes.

La simulation chez l'être humain

[Norman et Rumelhart 75] disaient que toute la force du modèle mental prenait naissance des capacités de simulation chez l'humain. Cependant, [Kahman et Tversky 82] ont affirmé plus tard que la simulation est difficile pour les humains. Les expériences de [Forbus 81] en étaient une preuve. [Forbus 81] et [Kahman et Tversky 82] ont insinué que la simulation n'est probablement faite par les humains que dans les cas triviaux.

La planification devient une tâche très complexe pour n'importe quel problème réel lorsqu'il s'agit d'opérer dans un environnement sur lequel le planificateur n'a aucun contrôle ou lorsqu'il est face à une incertitude sur les informations disponibles ou sur les actions des autres agents internes ou externes à l'environnement. Dans de telles circonstances, la planification est difficile à faire et les résultats de l'exécution sont aussi difficiles à prévoir [Lee et Fishwick 94].

En général, un planificateur humain, malgré ses capacités cognitives « exceptionnelles », a des limitations lorsqu'il s'agit d'assurer des activités de planification dans le monde réel telles que la planification dans un environnement complexe (plusieurs paramètres sont en jeu), dynamique et à forte composante spatiale. Ses capacités de simulation, qui se trouvent être à l'origine de son raisonnement, semblent bien être limitées.

2.1.2.2. Les limites de la cognition spatiale⁷

Étant donné que nous nous intéressons à la planification dans un espace géographique à grande échelle, il est nécessaire de mentionner que dans de tels espaces, l'être humain, malgré une cognition spatiale sophistiquée, reste très limité surtout au niveau de la perception.

La topologie est considérée comme la première classe d'information dans un espace géographique tandis que les propriétés métriques telles que les distances et les formes, sont utilisées pour des raisons de raffinement et sont souvent capturées de façon moins précise par l'humain. [Lynch 60] a démontré que les erreurs dans les cartes cognitives humaines sont souvent de nature métrique et rarement topologique. C'est la raison pour laquelle la plupart des limitations que nous allons énumérer sont de nature métrique:

Les distorsions

Quand un être humain forme sa carte cognitive, le cerveau utilise certaines heuristiques pour simplifier la formation et la sauvegarde de la carte cognitive [Tversky 81]. Ceci entraîne des distorsions dues au fait que l'humain a toujours tendance à trouver des alignements selon des points de repère. [Tversky 81] identifie deux types de distorsions :

- Des erreurs d'alignement : les places dans la carte cognitive ont plus tendance à s'aligner avec des lignes verticales et horizontales. Par exemple, les nord américains croient à tort que des villes européennes comme Rome (42° de latitude) ou Madrid (40° de latitude) sont situées au Sud des villes américaines telles que Chicago (42° de latitude) ou Philadelphie (40° de latitude).
- Des erreurs de rotation : les structures subissent souvent des rotations pour que leurs axes naturels soient alignés verticalement ou horizontalement. L'Écosse est ainsi représentée au-dessus de l'Angleterre au lieu d'être légèrement à l'Ouest.

Les expériences de [Stevens et Coupe 78] ont démontré que dans les cartes cognitives les informations ne forment pas une carte continue mais plutôt des structures hiérarchiques (comme un atlas). La ville de Reno (du Nevada) est ainsi vue à tort comme à l'Est de San Diego (de la Californie) parce que le Nevada est à l'Est de la Californie.

[Lloyd et Helvly 87] suggèrent trois facteurs qui peuvent causer la distorsion :

⁷ La cognition spatiale est définie comme la connaissance et la représentation interne ou cognitive de la structure, des entités et des relations de l'espace. Reflet interne et reconstruction de l'espace dans l'esprit [Kitchin 94].

- Les facteurs topographiques comme les densités et les proéminences relatives des zones spatiales.
- Les facteurs humains comme le type d'activité des personnes, leur familiarité avec l'environnement...
- Les facteurs de traitement comme les heuristiques utilisées pour simplifier et structurer l'information spatiale.

Les distances

Plusieurs chercheurs comme [Denis 89] et [Gahegan 95] croient que l'être humain construit mentalement une zone d'influence autour des objets spatiaux qu'il perçoit dans son environnement. Cette zone est proportionnelle à l'importance de l'objet. [Kettani 99] donne l'exemple suivant : on dira de deux montagnes de l'Himalaya distantes de 10km qu'elles sont proches, par contre on dira de deux voitures distantes de 10 km qu'elles sont loin l'une de l'autre. En effet, même si les deux distances sont quantitativement les mêmes, l'humain les perçoit différemment à cause de la taille des objets.

Un pompier pourrait ainsi croire que le feu qu'il perçoit est proche d'un lac alors qu'il est assez loin en réalité. La dynamique des objets peut également être en cause de certaines erreurs de jugement. Deux voitures roulant à grande vitesse l'une derrière l'autre peuvent nous sembler très proches, alors que deux vélos séparés par la même distance, nous sembleraient beaucoup plus éloignés que les voitures. Toutes ces difficultés à estimer les distances et les objets spatiaux limitent donc la capacité humaine à planifier dans un contexte spatial.

[Rothkegel et al., 98] distingue entre deux types de distances :

- Distance égocentrique : distance ayant pour origine l'observateur (l'observateur perçoit la distance entre lui et un point de l'environnement)
- Distance exocentrique : Distance entre deux objets autre que l'observateur.

Le jugement de la distance dépend en fait du type de la distance. Pour les distances égocentriques, les recherches montrent que :

- L'espace visuel n'est pas une transformation linéaire de l'espace physique [Luneburg 47]
- Les courtes distances physiques sont surestimées tandis que les plus grandes distances sont sous-estimées [Kerst et al., 87] et [Wender et al., 97].

Concernant les distances Exocentriques, [Rothkegel et al., 98] affirme que les distances correspondant à une ligne dans un plan horizontal ou vertical ne sont pas jugées de la même façon que les distances correspondant à des plans obliques. En effet, les lignes sur des plans obliques sont beaucoup plus difficiles à juger et le jugement induit souvent en erreur.

Dépendamment de la perception, deux types de distances sont à distinguer :

- Les distances perceptibles : les distances que l'observateur peut percevoir sans changer de position [Baird 70]. L'observateur peut commettre des erreurs en les jugeant [Rothkegel et al., 98], [Luneburg 47], [Kerst et al., 87] et [Wender et al., 97]
- Les distances environnementales : les distances qui ne peuvent pas être perçues sans que l'observateur ne se déplace autour [Montello 88]. Pour évaluer ces distances, la mémoire joue un grand rôle. Dans le cas d'un feu de forêt, les pompiers sont souvent pris dans la fumée et sont par conséquent incapables de percevoir l'environnement par simple observation. Ils doivent ainsi juger des distances environnementales. [Rothkegel et al., 98] affirme qu'estimer ce genre de distances dépend fortement de la méthode utilisée.

Les distances sont aussi asymétriques !

Si la géométrie Euclidienne met en évidence que la distance entre A et B est la même que celle entre B et A, cette règle est souvent violée dans le raisonnement spatial humain. Même si le chemin suivi est le même dans les deux sens, les humains perçoivent les deux distances différemment [Golledge et al 69] dépendamment des circonstances (le trafic, l'état du terrain...)

Les directions

Les humains ont souvent des problèmes pour déterminer les bonnes directions (Nord, Est, Ouest et Sud) sur le terrain. Les humains ont incontestablement des biais sur les directions.

L'espace de la géographie naïve est bidimensionnel

La réalité est tridimensionnelle : Tout objet aussi petit qu'il soit, a trois dimensions. A l'opposé, l'espace dans la géographie naïve est essentiellement bidimensionnel. A titre d'exemple, l'humain surestime souvent et de façon exagérée la raideur des pentes et les profondeurs des canyons. Au lieu de percevoir l'espace sous trois axes indépendants, les humains l'interprètent plutôt comme un espace horizontal à deux dimensions. La troisième dimension est souvent réduite à un attribut de position [Egenhofer et Mark 95].

Les cartes sont plus réelles que l'expérience humaine

[Egenhofer et Mark 95] affirme que les cartes sont souvent plus fidèles pour décrire l'espace géographique que l'expérience directe sur l'espace même. Il paraît alors que les cartes représentent mieux la réalité spatiale que la mémoire et l'expérience humaines.

2.2. Planification basée sur les agents dans un environnement dynamique

Notre objectif est d'essayer de résoudre un problème de planification en utilisant les agents logiciels. C'est pourquoi la Sous-Section 2.2.1 fait une petite revue de littérature sur l'évolution de la planification par agents ainsi que les principaux travaux de recherche utilisant cette technique. Nous évoquons ensuite brièvement les contraintes de la planification dans un environnement dynamique (Sous-Section 2.2.2) ainsi que les principaux travaux dans le domaine. Nous mettons ensuite l'emphase sur l'approche que nous adoptons (Sous-Section 2.2.3) pour la résolution de notre problème : la DCP, qui, bien entendu, semble être appropriée (en théorie au moins) au contexte de planification dans un environnement dynamique. Étant donné l'importance de l'aspect spatial dans le problème que nous voulons résoudre, un bref survol des recherches sur l'utilisation des

agents dans le raisonnement spatial, et plus particulièrement dans le raisonnement spatial qualitatif, est présenté dans la Sous-Section 2.2.4.

2.2.1. La planification par les agents

2.2.1.1. Agents logiciels et Systèmes multiagent : Généralités

2.2.1.1.1. Définition d'agent

Le concept d'agent a été longtemps utilisé en intelligence artificielle pour résoudre des problèmes de systèmes à base de connaissances, de robotique ou de langage naturel. Avec l'avènement de nouvelles technologies et l'expansion de l'Internet, l'agent est maintenant associé à plusieurs types d'applications comme *agent courtier*, *assistant personnel*, *agent interface*, etc.

Dans la littérature, les définitions de l'agent se ressemblent. Les différences relèvent du domaine d'application et du degré de complexité de l'agent. Une définition de la notion d'agent a été proposée par Ferber [Ferber 95]

« Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multiagent, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents. »

Récemment, Jennings [Jennings 98] a proposé la définition suivante pour un agent :

« Un agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu. »

A partir de cette dernière définition, un agent peut être caractérisé par son :

- *autonomie* : l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne.
- *flexibilité* : l'agent doit être :

capable de répondre à temps : L'agent doit être capable de percevoir son environnement et de répondre au bon moment à un éventuel changement de cet environnement.

proactif : l'agent doit être capable de prendre l'initiative au "bon" moment. On parle ici d'opportunisme.

social : l'agent doit être capable d'interagir avec les autres agents (logiciels et humains) en cas de besoin tout en s'adaptant au contexte (de concurrence, de collaboration, de confiance, etc.)

Les propriétés citées ci-dessus, rendent légitime la distinction entre le paradigme agent et les autres systèmes conventionnels. Elles soulignent en particulier les différences entre un objet (tel que défini par les modèles orientés objets) et un agent.

D'autres propriétés [OMG 99], comme la réactivité, la rationalité, l'engagement, l'intention, la mobilité, l'intelligence, la transparence, la capacité d'apprendre et la capacité de coopérer, peuvent s'ajouter pour mieux distinguer cette fois entre les différents types d'agents. Il est à souligner que l'importance de chacune des propriétés de l'agent varie selon le champ d'application.

2.2.1.1.2. Systèmes multiagents

Un *système multiagent (SMA)* est un système distribué composé d'un ensemble d'agents. Ferber [Ferber 95] définit un système multiagent comme étant un système composé des éléments suivants :

- Un environnement (c'est un espace disposant généralement d'une métrique).
- Un ensemble d'objets situés (un objet situé en est un pour lequel il est possible d'associer une position dans l'environnement à un moment donné). Ces objets sont passifs, c'est à dire qu'ils peuvent être perçus, créés, modifiés et détruits par les agents.
- Un ensemble d'agents, qui sont des objets particuliers et qui représentent les entités actives du système.

- Un ensemble de relations qui unissent des objets (et donc des agents) entre eux.
- Un ensemble d'opérations permettant aux agents de percevoir, produire, consommer, transformer et manipuler les objets.
- Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

Un SMA est généralement caractérisé par :

- Un point de vue partiel pour chaque agent : l'agent possède des informations ou des capacités limitées de résolution de problèmes.
- Contrôle décentralisé : il n'y a aucun contrôle global du système multiagent.
- les données sont décentralisées.
- le traitement (ou le raisonnement) est asynchrone.

Les agents au sein d'un SMA interagissent, le plus souvent, dans un cadre de *coopération*, de *concurrence* ou de *coexistence* [Chaib 94], [Chaib 96] et [Moulin 96]. Le type de l'interaction se définit en fait selon trois paramètres principaux : la nature des buts (objectifs) des agents, les compétences de ces agents et la disponibilité des ressources. Ferber dans [Ferber 95], dresse une typologie⁸ des interactions des agents selon ces trois paramètres (Tableau 2.1.).

⁸ L'antagonisme correspond à la concurrence et l'indifférence à la coexistence

Tableau 2.1. Typologie des interactions dans un SMA

Buts	Ressources	Compétences	Types de situation	Catégorie
Compatibles	Suffisantes	Suffisantes	Interdépendance	Indifférence
Compatibles	Suffisantes	Insuffisantes	Collaboration simple	Coopération
Compatibles	Insuffisantes	Suffisantes	Encombrement	
Compatibles	Insuffisantes	Insuffisantes	Collaboration coordonnée	
Incompatibles	Suffisantes	Suffisantes	Compétition individuelle pure	Antagonisme
Incompatibles	Suffisantes	Insuffisantes	Compétition collective pure	
Incompatibles	Insuffisantes	Suffisantes	Conflits individuels pour des ressources	
Incompatibles	Insuffisantes	Insuffisantes	Conflits collectifs pour des ressources	

Les SMA sont souvent efficaces pour résoudre des problèmes possédant de multiples méthodes de résolution. En effet, ils héritent, d'une part, des avantages traditionnels de la résolution distribuée et concurrente de problèmes comme la modularité, la vitesse (avec le parallélisme), et la fiabilité (due à la redondance) et d'autre part des bénéfices de l'Intelligence Artificielle comme le traitement symbolique (au niveau des connaissances), la facilité de maintenance, la réutilisation et la portabilité. Les SMA ont ensuite l'avantage de faire intervenir des schémas d'interaction performants : la coopération, la coordination (organiser la résolution d'un problème de telle sorte que les interactions nuisibles soient évitées ou que les interactions bénéfiques soient exploitées) et la négociation (parvenir à un accord acceptable pour toutes les parties concernées).

Les agents autonomes et les SMA ont donné naissance à une nouvelle approche pour l'analyse, la conception et l'implantation des systèmes informatiques complexes. Les techniques et les outils qui accompagnent la notion d'agent ont le potentiel d'améliorer considérablement les systèmes logiciels.

En accord avec [Jennings 00b] et [Chaib et al., 01], nous pensons que « la technologie agent va représenter durant les prochaines années, un nouveau paradigme de programmation pour le génie logiciel. Un nouveau paradigme “similaire” à la *programmation orientée-objet* et qui d’ailleurs pourrait s’intituler *programmation orientée-agent*. »

2.2.1.1.3. Utilisation des agents

Bien que la technologie agent soit assez récente, elle a pu en l’espace de quelques années s’imposer comme outil de conception dans plusieurs domaines d’application.

Plusieurs classes d’agents sont donc apparues selon leur usage. Notre but n’étant pas de dresser une liste exhaustive de toutes les classes d’agents, nous citons, à titre indicatif, dans ce qui suit les plus connues [OMG 99] :

- Les agents de gestion des systèmes et des réseaux. Les compagnies de télécommunication sont les plus actives dans ce domaine. Par exemple, Appleby et Steward [Appleby et Steward 94] ont conçu un prototype pour contrôler les réseaux de télécommunication via un système basé sur les agents mobiles.
- Les agents d’aide à la décision et de support à la logistique : Ces agents sont surtout utilisés par les militaires pour la synthèse des informations et l’aide à la décision [Mulvehill et al., 04; Petrov et Stoen 00] mais aussi dans d’autres domaines tels que le *e-supply chain* [Sadeh-Konieczpol et al. 03]. Ces systèmes peuvent prévenir un opérateur d’un éventuel problème, fournir de l’information nécessaire pour prendre des décisions complexes, etc.
- Les agents *Interest matching* : Ce sont les agents les plus utilisés bien que la plupart de ceux qui les utilisent ne soient pas au courant. En fait, ces agents sont déployés par les sites Web commerciaux pour offrir des recommandations. Ils (les agents) se basent sur le travail de Patti Maes de MIT Media Labs pour observer les patterns d’intérêt et d’usage afin de faire des recommandations. Le site amazon.com à titre d’exemple, utilise cette technique.

- Les agents d'assistance aux utilisateurs : Ces agents opèrent au niveau de l'interface utilisateur en fournissant des informations et des conseils aux utilisateurs. Ils sont parfois représentés visuellement par des personnages animés par les grandes compagnies telles que Microsoft, Lotus ou Apple. Ces agents « Interface » ont plusieurs rôles qui ont suscité l'intérêt de plusieurs chercheurs : Les guides pour les utilisateurs [Kozierok et Maes 93], l'aide à la mémoire [Liebermann 95], vente et achat au profit de l'utilisateur [Chavez et Maes 96], divertissement [Maes 95], filtrage et critique [Sheth et Maes 93], etc.

2.2.1.2. Planification par agents : Historique

La recherche en intelligence artificielle (IA) s'est essentiellement concentrée avant les années 80 sur les diverses composantes d'un agent. En fait, la majeure partie de ces composantes étaient le fruit des résultats de recherche en IA. L'IA semblait pouvoir ainsi transmettre certains de ses principes à la conception des agents. Durant cette période, le champ d'intérêt qui a le plus en commun avec les agents autonomes est celui de la planification.

La planification en IA :

En IA, la planification est la réponse à la question : Que faire ? Et plus précisément : quelles actions poser et dans quel ordre ?

L'intérêt de la planification pour les agents autonomes découle de cette définition. En effet, l'agent n'est autre qu'une entité qui pose des actions et par conséquent a besoin de planifier ses actions.

Jusqu'au début des années 80, la recherche sur la planification s'est concentrée sur les trois aspects suivants :

- Un modèle symbolique de l'environnement de l'agent.
- Une spécification symbolique des actions que l'agent doit exécuter.

- Un algorithme de planification qui manipule les symboles de la spécification pour générer un plan d'actions que l'agent doit exécuter pour atteindre ses buts.

Le système STRIPS (*Stanford Research Institute Problem Solver*) [Fikes et Nielsson 71] est l'un des premiers systèmes de planification basés sur les aspects mentionnés ci-dessus. Ce système avait pour objectif de concevoir des algorithmes capables de générer des plans que l'agent pourrait suivre pour atteindre son but final. Par contre, même si ces algorithmes donnaient des résultats satisfaisants pour les problèmes simples, ils se sont avérés inefficaces face à des problèmes beaucoup plus complexes (*i.e.* contrôle du trafic aérien). En effet, les tâches du monde réel, étant donné leur complexité, nécessitent un espace de recherche énorme pour les algorithmes de planification. La taille de cet espace croît d'une façon exponentielle avec la complexité du problème à résoudre [Chaib et al., 01].

Face à l'échec de l'IA à développer des algorithmes de planification capables de satisfaire les exigences du monde réel, plusieurs chercheurs ont remis en cause les bases de l'IA, surtout en ce qui concerne le raisonnement symbolique.

D'autres part, les suppositions sur lesquelles se sont basés les chercheurs comme Filks et Nielsson [Filks et Nielsson 71] pour concevoir des algorithmes de planification ne sont pas vraies dans un problème réel. Parmi ces suppositions nous citons :

- L'état du monde réel peut être observé et défini de façon formelle et correcte.
- L'agent est le seul capable de modifier l'environnement.

A la fin des années 80, de fortes critiques se sont portées sur ces suppositions.

Les agents réactifs

Une alternative au raisonnement symbolique fut proposée par Brooks qu'il appela IA réactive (SMA réactifs) [Brooks 86], [Brooks 91a] et [Brooks 91b]. Brooks précise que le comportement intelligent devrait émerger de l'interaction entre plusieurs comportements plus simples. Ainsi, Brooks a conçu ses agents comme étant un ensemble de comportements responsables de certaines tâches. Les comportements sont modélisés par

des machines à états finis et l'ensemble des comportements est représenté sous forme d'une hiérarchie de couches. Les couches inférieures représentent les comportements les plus concrets alors que les couches supérieures représentent les comportements les plus abstraits.

Malgré la simplicité de cette approche et les bons résultats obtenus pour certaines applications [Chapman et Agre 86] et [Brooks 91a], plusieurs points faibles sont apparus [Bousquet et Lepage 01] comme :

- En l'absence d'un modèle de l'environnement, l'agent doit disposer de beaucoup d'informations qui lui permettront d'agir.
- Il est difficile pour un tel agent (qui se base sur les informations locales) de tenir compte des informations non locales.
- Il est difficile de voir comment un agent réactif peut apprendre et améliorer ses performances.

2.2.1.3. Planification dans un environnement multiagent

Un plan multiagent est un plan élaboré pour être exécuté par plusieurs agents. La planification peut être faite par un seul agent. On parle alors de planification multiagent centralisée. Un agent est responsable de la création du plan qui spécifie les sous-plans de chaque agent. Georgeff a proposé une autre façon d'implémenter la planification centralisée [Georgeff 83]; chaque agent élabore son plan individuellement; ensuite un agent centralisateur rassemble les différents plans afin d'identifier les conflits et de les résoudre s'il y a lieu.

La planification peut également être faite par plusieurs agents, on parle alors de planification distribuée. Cette approche est utilisée quand un agent ne peut pas avoir une vue d'ensemble du groupe d'agents. Von Martial distingue deux classes de problèmes de planification distribuée [Von Martial 90]:

- Planification dirigée par les tâches : il y a un but initial qui est décomposé en sous-buts et qui est réparti entre les différents agents du groupe.

- Coordination des plans : elle vise à réconcilier les différents sous-plans avant de les exécuter dans un environnement commun. Les problèmes de conflits peuvent être résolus par un agent particulier (médiateur) ou par négociation.

Par la suite plusieurs modèles ont été proposés pour résoudre le problème de coordination de plans dont celui de Von Martial [Von Martial 92] (basé sur les relations positives et négatives entre buts).

Conry [Conry et al., 89, Conry et al., 91] a proposé quant à lui un protocole de négociation « *multi-stage negotiation* » pour résoudre de manière coopérative les conflits de l'allocation des ressources dans la planification.

2.2.2. Planification dans un environnement dynamique

La planification traditionnelle est basée sur l'hypothèse du monde fermé (*Closed World Assumption* (CWA)) [Weld 94] qui peut être résumée comme suit :

- Toute information sur le monde est connue.
- Tout ce qui est inconnu est considéré comme faux.

Ces hypothèses ne permettent pas de résoudre des problèmes réels de planification lorsqu'on manque de connaissances sur l'environnement. En effet, la planification doit être basée sur l'hypothèse du monde ouvert (*Open World Assumption* (OWA)) [Golden 97, Olawsky et Gini 90] qui considère comme inconnu tout ce qui n'est pas indiqué comme « connu » dans la liste des faits. Dans un monde ouvert on distingue entre deux grands types de planification:

- Planification avec une acquisition off-line de la connaissance i.e. durant le processus d'exécution.
- Planification avec une acquisition on-line de la connaissance i.e. durant le processus de planification.

Parmi les systèmes qui planifient avec une acquisition *off-line*, le système de [Golden 97] prend les décisions en se basant sur les hypothèses fondées sur les faits inconnus. Ces

hypothèses sont vérifiées durant le processus d'exécution via les *sensing actions* introduites d'avance dans le plan. Si ces hypothèses se révèlent fausses, l'agent planificateur doit replanifier.

Peot et Smith [Peot et Smith 92] ont conçu des plans conditionnels. Ces plans ont des branches correspondant à différentes alternatives. Selon les réponses obtenues pour les *sensing actions* correspondantes, des plans appropriés seront choisis parmi l'ensemble des plans durant le processus d'exécution.

Les approches à acquisition *off-line* ne sont appropriées que si le nombre de possibilités à considérer est limité. Quant à la planification à acquisition on-line, l'information est collectée dynamiquement durant le processus de planification. Une telle approche nécessite l'entrelacement de la planification et des *sensing actions* [Golden et Weld 96]

Barruffi et ses collègues [Barruffi et al., 98] définissent la planification dans un environnement à information incomplète et incertaine comme un problème de satisfaction des contraintes (*Constraint Satisfaction Problem (CSP)*) où des variables *range on*⁹ sont partiellement ou complètement inconnues. L'approche hybride proposée comme une solution (basée sur les Contraintes Interactives) –dérivée du CSP- est utilisée afin de planifier et replanifier dans un environnement dynamique à information incomplète.

Pour certains domaines complexes tels que le Commandement et le Contrôle (militaire) ou la gestion des situations de crise, l'environnement est tellement dynamique qu'il devient imprévisible [Al Timimi 98]. Les techniques conventionnelles de planification sont alors incapables de supporter les changements dans un environnement aussi dynamique qu'un champ de bataille ou une région affectée par un désastre naturel.

Les systèmes de planification doivent supporter tous les changements possibles en fournissant une génération rapide de plans alternatifs et en détectant les événements qui nécessitent une révision du plan. La détection des événements peut être supportée par des sentinelles introduites dans le plan. Les agents sont responsables de la surveillance des hypothèses du plan, de ses dépendances et des contraintes qui doivent être détectées quand un plan nécessite une modification (par un humain ou par un planificateur

automatique). Le modèle de sentinelles proposé dans [Kettler et Pemberton 98] est composé de trois grandes parties :

- L'acquisition et la collecte des données (à partir d'une variété de ressources : bases de données, humains, etc.)
- Le traitement : il évalue les données reçues afin de déterminer si un événement est survenu.
- Notification : Il notifie d'autres sentinelles, applications et opérateurs humains.

Une sentinelle dans un plan complexe doit satisfaire les trois exigences suivantes :

- L'habilité de mener des raisonnements complexes : la capacité de la sentinelle à inférer (à partir des données disponibles) un événement complexe ou un changement de situation.
- *Agency* : la capacité d'une sentinelle complexe à s'établir de façon proactive à travers la spécialisation, à chercher et traiter des données et à passer les informations pertinentes aux autres agents (humains ou logiciels).
- Extensibilité : Comment les fonctionnalités d'une sentinelle peuvent-elles être étendues afin de supporter plus d'environnements complexes et d'événements ?

Parmi les approches récentes destinées à planifier dans un environnement dynamique, la DCP est certainement l'une des plus appropriées. Dans la sous-section suivante nous donnons plus de détails sur ce nouveau paradigme.

2.2.3. Distributed continual planning (DCP)

Dans cette sous-section nous nous contentons de présenter les principes de base de la DCP ainsi que son évolution. Plus de détails sur l'approche seront présentés dans le chapitre 4.

⁹ Qui varient dans un ensemble de valeurs données.

2.2.3.1. DCP : Principe

Le processus de la planification classique développé au début de la recherche sur Intelligence Artificielle (IA), s'est avéré inefficace pour les applications du monde réel. Les chercheurs de l'IA se sont alors concentrés sur de nouvelles solutions (de nouveaux algorithmes et de nouveaux systèmes) dans lesquelles la planification est souvent distribuée et la génération du plan est entrelacée avec le processus d'exécution. Un nouveau paradigme a ainsi émergé au sein de la communauté de recherche en IA, appelé Planification Distribuée en Continu (*Distributed Continual Planning* DCP). Le terme « distribuée » réfère à un environnement dans lequel la planification est distribuée sur plusieurs agents ou processus. Quant au terme « en Continu », il désigne un processus continu et dynamique dans lequel la planification et l'exécution sont entrelacées.

Plusieurs chercheurs affirment que l'utilisation de la DCP dépend du domaine d'application. Durfee [Durfee 99] relève trois caractéristiques principales (de l'application) dont le paradigme DCP a besoin :

- L'application doit être telle que le succès d'un agent dépend du choix immédiat des actions qui mènent à de bons choix lors des décisions futures.
- La connaissance de l'agent à propos du domaine de l'application et/ou les objectifs de l'agent, peuvent changer durant le temps. L'information peut être révélée de façon incrémentale comme elle peut changer dynamiquement hors du contrôle de l'agent. Cet agent doit ainsi prendre en compte les éventuels changements de façon continue afin de pouvoir réévaluer et réviser ses plans.
- La formulation du plan peut nécessiter la participation de plusieurs agents qui combinent leurs connaissances et leurs expertises. En effet, d'une part, certains changements (perçus par l'agent) au sein de l'environnement sont dus à l'activité d'autres agents. D'autres part, les capacités d'un agent dépendent parfois de l'intervention d'autres agents.

2.2.3.2. DCP: Évolution

Dans cette section, nous présentons un framework proposé dans [Desjardins et al., 99] afin de catégoriser les systèmes à base de planification et d'exécution, selon le contexte supporté par ces systèmes. Les quatre types de systèmes qui suivent sont présentés selon l'ordre chronologique du plus ancien au plus récent.

Contexte qui ignore la planification et l'exécution

A cause de la difficulté à planifier, les chercheurs en IA ont introduit des hypothèses et des simplifications afin de rendre la planification plus faisable. L'agent planificateur était alors supposé avoir le contrôle de son monde : les changements qui peuvent survenir ne sont alors dus qu'aux actions délibérées de l'agent. Ces hypothèses et ces simplifications irréelles ont mené à la sérialisation du problème de planification : « planifier ensuite exécuter ».

Contexte qui tolère la planification et l'exécution

Dans un environnement dynamique et incertain, un agent est incapable de faire des prédictions précises sur les actions futures. Afin de résoudre ce problème, plusieurs approches ont été proposées

Une première approche supporte l'incertitude en énumérant les états possibles (contingences) et en construisant par la suite un plan pour chaque état possible.

Si jamais l'agent n'a pas la connaissance suffisante, une meilleure approche serait de surveiller l'exécution du plan et de le réparer si nécessaire [Ambros-Ingerson et Steel 88]. L'agent construit un plan initial. Durant l'exécution du plan, sa progression est surveillée. Si l'agent détecte une déviation, il arrête cet épisode (le sous-plan exécuté) et revoit ses décisions de planification.

Dans un environnement complexe et quand les buts de l'agent ainsi que le monde évoluent continuellement au lieu de rester stables durant un épisode de planification, l'agent doit évaluer et réviser ses plans. La révision du plan doit être un processus continu et non pas un processus qui n'est appelé qu'en cas de problème.

Contexte qui exploite la planification et l'exécution

L'approche par planification en continu évoquée ci-dessus, suggère un environnement dans lequel un agent vise à atteindre ses objectifs malgré les changements au sein de l'environnement. Cependant, un agent doit tolérer la présence d'autres agents et même en profiter en collaborant. En particulier, un agent doit interagir avec des agents humains afin d'améliorer la planification. Il doit aussi collaborer avec d'autres agents logiciels.

Même si les agents ne coopèrent pas explicitement en partageant leurs expertises et leurs capacités, un agent peut réduire son incertitude et améliorer la qualité de ses plans en exploitant le fait que les changements au sein de l'environnement sont souvent des actions entreprises par les autres agents.

Contexte qui établit la planification et l'exécution

Les différentes approches discutées précédemment ne sont que des versions distribuées de l'approche classique « planifier et agir ». Pour réaliser un vrai système de planification distribué en continu, chaque agent doit planifier en continu et élaborer des décisions compatibles (de préférence mutuellement supportées). L'agent doit avoir un mécanisme qui lui permet d'évaluer en continu les relations entre des décisions de raffinement et de rediriger les décisions de façon coopérative. Nous établissons ainsi un contexte plus large pour planifier (travailler ensemble pour planifier) et pour exécuter (agir en vue de buts communs) afin d'améliorer ce qu'un agent peut accomplir [Tambe 97, Shoham et Tennenholtz 92, Stone et Veloso 99].

2.2.4. Planification dans un espace géographique

2.2.4.1. Le raisonnement spatial qualitatif

Le raisonnement spatial, dans nos interactions de tous les jours avec le monde réel, est dans la plupart des cas le fruit d'abstractions qualitatives et non pas celui des connaissances quantitatives complètes. C'est la raison pour laquelle le raisonnement qualitatif est considéré comme un support pour les théories sur le raisonnement spatial. Ceci justifie l'intérêt croissant que portent les chercheurs à étudier les concepts spatiaux

d'un point de vue cognitif. C'est ce qui a donné naissance en fait au raisonnement spatial qualitatif en IA et en SIG [Hernandez 94]. L'un des défis majeurs du raisonnement spatial qualitatif est de fournir des calculs qui permettent à la machine de représenter et de raisonner sur les entités spatiales sans avoir recours aux techniques quantitatives traditionnelles.

La représentation spatiale qualitative concerne plusieurs aspects de l'espace tels que la topologie, l'orientation, la forme, la grandeur et la distance.

Applications du raisonnement spatial qualitatif

La recherche en raisonnement spatial qualitatif est motivée par une grande variété d'applications possibles : SIG, navigation des robots, vision haut niveau, sémantique propositionnelle spatiale pour les langages naturels, conception ingénieur, etc.

Même si les SIG sont devenus très courants, l'interaction est toujours considérée comme un problème majeur. En effet, les actuels SIG ne supportent pas assez les interactions intuitives ou de bon sens entre l'homme et la machine, et ce à cause de l'énorme quantité d'information (sauvegardée sous forme vectorielle ou raster). Les utilisateurs aimeraient en fait pouvoir en extraire (en spécifiant des requêtes) de l'information plus globale et qualitative. Les conclusions de Cohen et Hazarika [Cohen et Hazarika 02] laissent entendre que les prochaines générations de SIG seraient conçues en se basant sur les concepts de la géographie naïve [Egenhofer et Mark 95].

2.2.4.2. Les agents au service du raisonnement spatial qualitatif

La planification spatiale dans un espace à grande ou moyenne échelle est très complexe [Ferrand 96]. Plusieurs aspects relatifs à ce domaine ne sont pas supportés par les SIG actuels : l'échange et la co-évolution des représentations spatiales entre les acteurs distants, le support à la négociation et la simulation, l'aide à la décision pour la planification spatiale en continu, etc. Plusieurs chercheurs comme Ferrand [Ferrand 96] ont alors proposé d'utiliser les SMA comme moyens d'améliorer ou de développer ces fonctionnalités.

Très peu de travaux ont utilisé les agents logiciels pour faire du raisonnement spatial qualitatif. Par exemple, Stolzenburg et ses collègues ont proposé une architecture multi-couches pour des agents spatio-temporels [Stolzenburg et al., 99]. Ces agents (conçus pour répondre aux besoins de RoboCup) qui simulent un match de football, utilisent des classes de prédicats nommées QUALITIES. À titre indicatif, et concernant la distance de l'agent par rapport à la balle, les agents peuvent raisonner sur certaines distances : *close*, *near*, *short*, *far away*, *remote* (hors portée).

La prise de décision par collaboration spatiale qualitative est un domaine très important puisque souvent les gens communiquent leur connaissance spatiale sous forme de relations spatiales et rarement en termes de coordonnées absolues. Karacapilidis et ses collègues [Karacapilidis et al., 95] ont proposé un modèle computationnel capable de supporter une collaboration spatiale qualitative basée sur les agents logiciels. Ils ont appliqué le modèle sur un problème d'aménagement spatial. Les travaux similaires se font cependant rares dans la littérature.

Il en est de même pour les travaux concernant la recherche de chemins (sous contraintes spatiales qualitatives) par agents logiciels. Lewis et ses collègues [Lewis et al., 01] figurent parmi les rares chercheurs qui ont essayé de résoudre des problèmes de *Pathfinding* par des agents qui tiennent en compte des contraintes non tangibles. C'est en fait ce domaine particulier qui nous intéresse le plus dans cette thèse et sur lequel nous donnons plus de détails dans le Chapitre 5.

2.3. Simulation basée sur les agents : État de l'art

Dans cette section nous présentons tout d'abord des généralités sur le domaine de la simulation (Section 2.3.1). Nous mettons ensuite l'emphase sur le sous domaine de la simulation qui nous intéresse dans ce travail : la Simulation Orientée Agent (SOA) (Section 2.3.2).

2.3.1. Simulation : Généralités

2.3.1.1 Définition

Selon Shannon [Shannon 75], la simulation est « le processus qui construit un modèle pour un système réel et qui mène des expériences sur ce modèle afin de comprendre le comportement du système ou d'évaluer différentes stratégies (en respectant les limites imposées par certains critères) pour le fonctionnement du système. »

Le terme « modèle » évoqué dans cette définition réfère à « quelque chose que nous utilisons au lieu d'une chose réelle afin de comprendre quelque chose à propos de la chose réelle. » [Fishwick 94]

2.3.1.2. Les phases de la simulation

La simulation est un processus à trois principales composantes itératives et inter-reliées :

- Conception du modèle : il s'agit de rassembler des données sur le système à simuler. Les données peuvent être sous forme symbolique ou numérique. Les données numériques sont typiquement obtenues en utilisant des senseurs physiques ou humains. Les autres données sont obtenues en interviewant des humains ou en appliquant des techniques d'acquisition de la connaissance, utilisées pour obtenir de la connaissance qualitative. A partir des données et les connaissances acquises à partir des expériences antérieures sur des systèmes similaires, un modèle est formulé.
- Exécution du modèle
- Analyse de l'exécution : Des tests sont faits, sur les données générées à partir du modèle, en faisant appel à des analyses spécifiques telles que l'analyse de Fourier ou des analyses statistiques. L'analyse la plus basique consisterait tout simplement à « regarder » les données et à en déduire des conclusions.

2.3.1.3. Différents domaines d'application

La simulation joue un grand rôle dans plusieurs domaines [Fishwick 94] dont nous citons les suivants :

- Calcul informatique: La simulation est la *lingua franca* du calcul informatique [Fishwick 94]. Il s'agit d'utiliser des ordinateurs (ou des super ordinateurs) pour simuler des phénomènes complexes et à grande échelle.
- Systèmes complexes et chaotiques : La plupart des modèles à comportement chaotique ne peuvent pas être traités par une analyse statique. La simulation est alors nécessaire pour fournir une vue plus détaillée du système en question.
- Réalité virtuelle : La réalité virtuelle RV pousse de plus en plus l'analyste à utiliser la simulation. Afin d'être efficace, la technologie RV doit incorporer des méthodes pour construire un monde digital (ou virtuel). La construction d'un monde dynamique est justement ce que la simulation informatique permet de faire.
- Vie artificielle : Une expérience de la vie artificielle est une expérience dans laquelle un programme informatique simule des formes de la vie artificielle basées sur des métaphores telles que la reproduction génétique et la mutation.
- Animation informatique et modélisation basée sur le matériel (*Physically Based Modeling*) : La modélisation basée sur le matériel a connu une nette progression due à l'avancée du graphisme informatique. Les animations actuelles peuvent porter sur des modèles physiques très complexes et offrent grâce à la simulation des résultats intéressants.

2.3.2. Simulation Orientée Agent (SOA)

2.3.2.1. Introduction

La simulation orientée agent est de plus en plus utilisée dans des différents domaines. Les techniques de simulation orientée objets [Troitzsch 97], de micro-simulation [Orcutt 57]

et centrées sur l'individu [Harding 99], sont en train de céder leur place à cette nouvelle technique : la SOA, ou en anglais : *Agent-Directed Simulation* (ADS). La SOA est maintenant présente dans différents secteurs : biologie [Resnick et Turtles 95], physique [Schweitzer et Zimmermann 01], chimie [Resnick et Turtles 95], écologie [Huberman et Glance 93], économie [Ben Said et al., 02], sociologie [Pietrula et al., 98; Axtell 00], etc. Drogoul et ses collègues [Drogoul et al., 02] justifient le fait que la SOA est devenue le support de choix pour la simulation de systèmes complexes par les deux principales raisons suivantes :

- La capacité de la SOA de s'adapter à des modèles très différents des entités à simuler. Elle utiliserait ainsi des agents réactifs [Drogoul 95] pour simuler des entités simples et déploierait des agents cognitifs [Jennings 00b] pour simuler des entités plus complexes.
- La SOA offre au concepteur la possibilité de manipuler différents niveaux de représentations (p.ex. « individu », « groupe » et « cluster »), contrairement à une approche basée sur les automates cellulaires [Vanbergue et al., 00].

2.3.2.2. Branches de la SOA

Selon Ören et ses collègues [Ören et al., 00], les agents peuvent être combinés de trois façons possibles à la simulation, ce qui a donné naissance aux trois sous branches suivantes :

- Simulation agent.
- Simulation basée sur les agents.
- Simulation supportée par les agents.

2.3.2.2.1. Simulation Agent

La simulation agent (ou *Agent simulation*) est la simulation d'entités qui peuvent être représentées par des agents [Ören et al., 00]. Elle est la façon naturelle de modéliser et de simuler des entités intelligentes. Ces entités peuvent être des humains ou encore des

entités artificielles telles que les plateformes intelligentes, les équipements, etc. Dans le domaine militaire par exemple, la simulation agent serait donc de modéliser des entités intelligentes et quasi autonomes. Les agents pourraient représenter des individus amis ou ennemis, des troupes, des plateformes, des armes ou tout autre équipement.

2.3.2.2.2. Simulation basée sur les agents

La simulation basée sur les agents (ou *Agent-based simulation*) est le fait d'utiliser des agents dans la simulation afin de générer un modèle de comportement [Ören et al., 00]. Elle offre également des possibilités additionnelles pour la simulation numérique, tout comme dans la simulation des systèmes experts ou les systèmes de simulation qualitative. A titre indicatif, dans le domaine militaire, la simulation basée sur les agents est utilisée pour modéliser le comportement humain lors d'un conflit, d'une évacuation ou une situation de crise.

2.3.2.2.3. Simulation supportée par les agents

La simulation supportée par les agents (ou *Agent-supported simulation*) est le fait d'utiliser des agents dans la simulation des opérations de soutien qui peuvent être des opérations d'interface *front-end* ou *back-end* [Ören et al., 00].

Ce type de simulation est très utilisé par les militaires pour décrire des scénarios, représenter ou analyser des menaces, représenter le champ de bataille ou de façon plus générale l'environnement, évaluer les capacités à comprendre et à apprendre à fournir des alternatives sous le stress, visualiser l'influence de l'information sur les décisions en situation de combat, etc.

Bien que la SOA couvre maintenant plusieurs domaines et semble, du moins théoriquement, être d'une grande utilité pour la simulation des systèmes complexes, il est toujours légitime de se demander si en pratique les projets qui utilisent la SOA bénéficient vraiment des avantages de cette technique.

2.3.2.3. Implémentation de la SOA : réalité et défis

2.3.2.3.1. Réalité

Il est vrai que les projets de SOA existants portent sur plusieurs secteurs et donnent des résultats encourageants en général. Cependant, bien que ces projets se réclament du domaine des SMA, ils n'utilisent pas le paradigme agent comme il faut. Dans ce cadre, les études de Drogoul et ses collègues [Drogoul et al., 02] montrent que la plupart des réalisations en SOA n'utilisent les « vraies » propriétés de l'agent que dans le modèle conceptuel. Le modèle opérationnel, quant à lui, ne se base pas sur des agents computationnels, ce qui prive ces réalisations des avantages que pourrait apporter l'application des principes des SMA.

2.3.2.3.2. Défis et avenir

En dépit des handicaps cités ci-haut, les chercheurs restent optimistes et croient qu'il est très possible de ramener les agents computationnels, considérés comme le futur de l'informatique [Servat et Drogoul 02], dans les implémentations de la SOA. Certains même, ont pu spécifier les applications pour lesquelles les agents computationnels pourraient être d'une grande utilité à la SOA. Ces applications, selon [Drogoul et al., 02], sont : la conception participative, l'interprétation assistée par agents, l'adaptation au niveau du système, et le calibrage assisté par agents.

Afin de réaliser de tels projets basés sur la SOA et profitant vraiment des avantages du paradigme agent, les chercheurs ont plusieurs défis à relever. Nous en citons ici quelques-uns ainsi que les pistes probables pour résoudre ces problèmes, telles que présentées par [Drogoul et al., 02].

Formalisation des comportements des agents

Lors de la conception de SOA, l'une des tâches les plus difficiles est de formaliser les comportements des agents à partir de données théoriques ou empiriques à disposition. Pour automatiser le processus d'extraction de ces connaissances, il serait judicieux d'utiliser des agents dotés de capacités d'apprentissage et d'une autonomie décisionnelle. Plusieurs recherches actuelles en IA, et surtout en robotique, visent dans ce cadre, le

développement des procédures d'apprentissage en ligne telles que l'autonomie ajustable [Chalupsky et al., 01], l'apprentissage par imitation [Dautenhahn 95] et l'apprentissage interactif situé [Hugues et Drogoul 00]. L'agent apprend ainsi son comportement progressivement tout en impliquant l'expert humain. Selon [Drogoul et al., 02], cette approche replace les agents computationnels au centre de la SOA et fera sans doute partie de la modélisation future des systèmes à base de SOA. Toutefois, beaucoup de travail reste à faire dans les années à venir et particulièrement dans la recherche de techniques génériques indépendantes du domaine [Drogoul et al., 02].

Complexité

Pour des systèmes à base de SOA impliquant un grand nombre d'agents (des milliers), la complexité générée par l'interaction des agents est parfois similaire à celle d'un système naturel [Servat et al., 98].

Un grand travail attend les chercheurs pour les prochaines années afin de remédier à ce problème. Certains même, ont déjà proposé des solutions prometteuses. Servat et ses collègues [Servat et al., 98] proposent dans ce cadre de doter les agents de capacités d'interprétation locales. Les macro-connaissances des thématiciens seraient ainsi distribuées parmi les agents.

Distribution de la simulation

La simulation de systèmes complexes par des milliers d'agents nécessite également une grande capacité de calcul. La distribution de la simulation sur un réseau de machines est devenue souvent une nécessité lors de la conception de tels systèmes. Cependant, cette distribution génère à son tour de nouveaux défis pour la SOA tels que le maintien de la cohérence globale du système, la répartition de la charge sur les différents nœuds de la distribution et la tolérance aux pannes (p.ex. le système doit assurer un bon fonctionnement si un nœud tombe en panne).

Les SMA semblent en tout cas porter des solutions sérieuses. En effet, grâce à leur nature distribuée, les SMA peuvent jouer un rôle de régulateur. Dans [Michel et al., 02] les auteurs proposent par exemple une régulation à base d'agents mobiles. Leurs capacités

(migration ou clonage) permettraient de distribuer automatiquement la simulation et même d'adapter cette distribution aux particularités de la simulation.

2.3.2.4. Conclusion

Cette brève étude bibliographique montre que la SOA est une nouvelle technique de simulation très prometteuse. Cependant, plusieurs problèmes doivent être résolus pour que la SOA puisse donner les résultats escomptés. Sur les projets actuels, nous avons constaté que les agents utilisés dans la SOA ne sont pas de vrais agents computationnels, ce qui prive la SOA de démontrer ses vrais apports. Plusieurs chercheurs se sont alors penchés sur ce problème afin d'identifier les applications qui pourraient mettre en évidence le vrai apport des agents logiciels dans la SOA. La plupart de ces applications ont pour objectifs de simuler. Étant donné que la SOA s'applique à des systèmes assez complexes et semble avoir des avantages prometteurs par rapport aux autres approches, nous nous sommes demandés si elle pouvait être utilisée pour résoudre des problèmes de planification qui sont souvent complexes surtout dans un environnement géographique, dynamique et réel. La section suivante donne un aperçu sur les approches existantes de planification par simulation.

2.4. Planification basée sur la simulation

Dans notre hypothèse de recherche, nous avons parlé de notre intention d'utiliser la simulation comme moyen de planification. Nous avons également noté que certains travaux ont déjà utilisé ce principe pour planifier. Dans cette section, nous présentons d'abord rapidement les principes sous-jacents à cette approche. Nous donnons ensuite un bref aperçu sur différents systèmes qui ont essayé de combiner la simulation à la planification. La plupart des projets de recherche de ces dernières années sont surtout militaires. Toutefois, de rares exemples existent dans d'autres domaines. Le projet Phoenix [Cohen et al., 89] est certainement le plus connu. Nous mettons plus l'accent sur ce dernier, surtout que ses auteurs ont illustré leur théorie sur le même domaine d'application que le nôtre : les feux de forêt.

2.4.1. Principe de l'intégration de la planification et de la simulation

La planification devient très complexe pour n'importe quel problème de planification dans un monde réel où le planificateur n'a aucun contrôle sur ce qui se passe ou lorsque les informations disponibles ou les futures actions des autres agents sont incertaines. Dans de telles circonstances, des prédictions précises des états résultant de l'exécution du plan sont difficiles. Plusieurs approches ont été introduites pour résoudre ce problème [Dean et Kanazawa 87, Hammond 89, Schoppers 87].

Certaines approches utilisent des systèmes experts à base de règles comme SOAR [Braudaway 93, Salisbury et Tallis 93]. Puisque des milliers de règles sont impliquées juste pour modéliser un objet, la tâche de maintenir et de raisonner sur des plans devient très complexe, surtout que les systèmes à base de règles sont souvent centralisés.

Une autre alternative consiste en la Planification basée sur la Simulation (*Simulation-Based Planning* SBP). Lorsqu'un ensemble de plans est généré, des simulations sont utilisées pour tester et évaluer les plans afin de choisir le plus approprié pour la situation courante. Cette approche peut ainsi résoudre le problème de précision des prédictions dans un environnement incertain en utilisant des modèles de simulation individuels pour prédire le comportement des objets individuels dans le monde. Elle peut également produire des plans à différents niveaux d'abstraction. La multi-modélisation [Fishwick et Zeigler 92, Fishwick 92, Fishwick 95] est alors utilisée pour modéliser les processus et les agents à différents niveaux d'abstraction.

La planification basée sur la simulation réfère à l'utilisation de la simulation par ordinateur en vue d'aider la prise de décision. Elle utilise la même approche itérative que les arbres de jeu adoptent pour déterminer la meilleure suite d'actions à faire dans le jeu. SBP se base ainsi sur les points suivants:

- Un modèle d'action est exécuté pour déterminer l'efficacité d'une donnée en entrée ou une décision de contrôle. Un arbre de jeu implémente généralement une

fonction d'évaluation statique, alors que dans SBP, un modèle est exécuté pour déterminer les effets d'une action.

- Différents modèles sont employés à différents niveaux d'abstraction dépendamment du temps qui reste au planificateur pour prendre sa décision. SBP peut ainsi exécuter des modèles de phénomènes physiques de façon plus détaillée lorsque le temps de planification le permet ou lorsque les machines utilisées sont suffisamment puissantes.

Les militaires, à titre d'exemple, utilisent cette approche depuis des décennies sous forme de simulation à *modèle constructif*. Un modèle constructif est basé sur des équations d'attrition et sur des cartes (carrelées par des carrés ou des hexagones) utilisant des déplacements discrets dans l'espace et dans le temps. Pour décider si une action doit être acceptée ou non, un modèle constructif évalue plusieurs alternatives. Dans ce cas, ce modèle est plutôt appelé un *wargame*.

2.4.2. Quelques travaux sur la SBP

2.4.2.1. La planification par anticipation (*Anticipatory Planning*)

Le Général Wass de Czege [Wass de Czege 99] a radicalement proposé une nouvelle approche pour la planification et l'exécution dans le domaine militaire, qu'il a appelée : *Anticipatory Planning* (planification par anticipation). Selon lui, il est futile d'essayer de prédire le futur le plus probable et de construire un plan juste pour ce cas. De tels plans ont de faibles chances de survivre à cause de l'aspect très dynamique et incertain de l'environnement. Wass de Czege affirme par contre qu'il faut planifier le plus possible de plans (autant que les différentes actions possibles de l'ennemi dans le cas d'un conflit militaire, par exemple) et créer ensuite un plan pour les situations les plus probables et les plus dangereuses. Le plan doit fournir autant de branches que le temps le permet. L'habilité à développer et considérer plusieurs branches dans un plan, nécessite un processus de *Anticipatory Planning*. Au lieu de choisir une seule action et de la suivre jusqu'à la conclusion, l'*Anticipatory Planning* appelle à maintenir autant d'actions que possible. Le plan est alors vu comme un arbre : les nœuds sont les états et les branches

sont les actions à prendre pour passer d'un état à un autre. Lorsque de nouvelles branches sont construites, le processus d'*Anticipatory Planning* continue de planifier le long de ces branches. Contrairement à la planification réactive qui planifie lorsqu'une branche est concernée, l'*Anticipatory Planning* prépare à l'avance les plans pour plusieurs branches possibles et les maintient le plus longtemps possible.

Surdu et Hill [Surdu et Hill 00] ont proposé une méthodologie pour implémenter cette approche (*Anticipatory Planning*) basée sur la simulation, nommée *Anticipatory Planning Support System* (APSS). Les différentes branches sont évaluées par simulation. Les auteurs ne spécifient pas le type de simulation qu'il faut faire. Ils affirment que la simulation adoptée dans la méthodologie doit considérer les caractéristiques du terrain. Ils recommandent alors une représentation minimale du terrain comprenant l'élévation et les portions faciles et difficiles du terrain. Cependant, les auteurs ne proposent aucun moyen concret de faire ce genre de simulation des caractéristiques du terrain.

2.4.2.2. Framework SBP

Dans un système multiagent, on pourrait avoir besoin de simuler les agents individuels ainsi que la coordination entre ces agents et les méthodes (interaction, coopération, etc.) utilisées pour atteindre leurs objectifs. Le but du framework SBP [Lee et Fishwick 94; Lee et Fishwick 97] est de créer des modèles pour les SMA qui tiennent compte des contraintes de l'environnement. Parmi ces contraintes :

- Les chemins géométriques à suivre
- Une mission à accomplir
- Un ensemble de conditions que tous les agents doivent respecter pendant une opération

La méthodologie SBP consiste à imbriquer une simulation dans une autre. La simulation imbriquée simule les actions des agents avant de s'engager sur un plan. Les actions peuvent ainsi être évaluées. SBP comprend trois composantes :

- Le bloc de simulation (*Trial*) : il est représenté par un algorithme qui simule plusieurs exécutions possibles. Le principe est d'évaluer les situations qui ont le plus de probabilité de survenir.
- Le bloc de conception expérimentale (*Executive*) : Exécute les simulations en essayant au maximum de réduire la complexité du problème et le temps d'exécution de la simulation.
- Les blocs d'analyse des résultats (*Replicator, Evaluator et Analyser*) : implémentent une stratégie pour extraire les résultats les plus appropriés et les analyser.

Le framework a été appliqué sur le scénario d'interdiction aérienne [Lee et Fishwick 97]. Bien que les auteurs disent avoir appliqué la SBP dans un environnement multiagent, notre étude du projet révèle en fait que les agents utilisés ne sont que des processus. Le framework se base sur des données simplifiées de l'espace.

2.4.2.3. Planificateur d'assauts pour un bataillon de chars

Horn et Baxter [Horn et Baxter 00] utilisent une hiérarchie de commandement des agents pour planifier et exécuter en continu des petites unités de tâches (pour planifier un assaut d'un bataillon de chars). Le système développé est capable d'analyser des positions potentielles des chars sur la carte en produisant un nombre de cartes de coût (*cost maps*) reliées à différents critères que l'expert utilise pour prendre des décisions. Ces critères sont très variables : visibilité de l'ennemi, profil du terrain, etc. Ce projet permet une sensibilité spatiale assez intéressante. Une planification en continu est également adoptée pour corriger les plans construits (par replanification). Cependant, le système développé reste un outil d'entraînement pour les commandants novices. En effet, ce travail n'indique en aucun cas comment le système pourrait être utilisé pour des planifications dans le monde réel. Les agents utilisés sont également des processus et non des agents autonomes et proactifs qui peuvent planifier dans un cadre multiagent.

2.4.2.4. « Capture the Flag »

Atkin et ses collègues [Atkin et al., 98] ont proposé une architecture *Hierarchical Agent Control* (HAC) pour décrire le comportement des agents. HAC renferme des mécanismes pour contrôler les agents, passer des messages, coordonner les actions des agents, résoudre les conflits entre agents, etc. Les systèmes créés sous la base de HAC sont ensuite simulés par une infrastructure de simulation appelée *Abstract Force Simulator* (AFS). AFS permet de simuler une variété de domaines qui peuvent être caractérisés par des agents en mouvement ayant une certaine force à appliquer. AFS représente les agents de façon abstraite comme des particules qui ont un petit ensemble de caractéristiques physiques (masse, vitesse, friction, force d'attaque, etc.). Une particule peut être une armée, un soldat ou une entité politique. Elle est capable d'accomplir des actions primitives : le déplacement, le changement de forme et l'application de force. Tous les autres comportements dérivent de ces trois actions. AFS est ainsi un simulateur de processus physique qui agit dans un espace 2D qui est un terrain représenté par une grille de cellules rectangulaires.

HAC contient également un planificateur qui utilise des solutions stockées (plans) qui ne sont pas totalement élaborées. Ce type de planification est en fait le *Partial Hierarchical Planning* de [Georgeff et Lansky 86]. La flexibilité de ces plans aide à gérer l'aspect dynamique de l'environnement. Les plans sont évalués par une simulation en avant. AFS simule en fait les conséquences des actions planifiées avant leur exécution.

Le scénario repris souvent par les auteurs pour tester HAC et son simulateur est celui du jeu *Capture the Flag*. Ce travail de recherche ne met pas l'accent sur l'aspect spatial qui est primordial pour le type d'applications qu'il vise (militaires sur un espace étendu). Le scénario *Capture the Flag* montre qu'il reste beaucoup de travail à faire pour pouvoir implémenter des applications réelles. En effet, l'espace géographique est très simplifié, l'approche de planification choisie pour un environnement aussi dynamique n'est peut-être pas la meilleure alternative et enfin aucune mention n'est faite par les auteurs sur la possibilité de faire le lien entre le monde de simulation et le monde réel.

2.4.2.5. Le projet Phœnix

Toutes les informations (sauf avis contraire) dans cette section ont été prises de [Cohen et al., 89]

2.4.2.5.1 Le problème

La tâche de Phœnix est de contrôler une simulation de feux de forêt impliquant des bulldozers, des équipes, des avions, tous simulés.

L'environnement Phœnix simule plus particulièrement le parc national de Yellowstone pour lequel une carte numérique a été conçue et utilisée comme support à la simulation.

Selon ses auteurs, Phœnix est un simulateur précis pour les feux de forêt.

2.4.2.5.2 Caractéristiques de Phœnix

L'environnement Phœnix est :

- Dynamique parce que tout peut changer : vitesse du vent, sa direction, l'humidité, type d'essence, intensité du feu, la disponibilité des ressources, etc.
- Continu dans le sens qu'il n'y a pas un seul problème bien défini à résoudre mais plutôt un flux continu de problèmes dont la plupart ne peuvent pas être anticipés.
- En temps réel dans le sens que c'est le feu qui établit l'allure et la progression auxquelles les agents doivent s'adapter. L'environnement est en perpétuel changement même durant les actions (y compris la planification) des agents. Les agents doivent raisonner sur les effets probables de leurs actions et particulièrement sur le temps requis pour ces actions.
- Imprévisible parce que le feu peut éclater à n'importe quel moment et à n'importe quel endroit, et aussi parce que les agents peuvent rencontrer des imprévus sur le terrain. Chaque agent doit alors être capable de répondre aux événements imprévus qui entravent ses actions. Ceci nécessite un entrelacement de la planification, de l'exécution et de la surveillance.

- Distribué dans l'espace et les agents ont seulement une connaissance limitée et locale de l'environnement.

2.4.2.5.3 Les couches de Phœnix

Phœnix a été conçu en quatre couches, les deux premières constituent l'environnement Phœnix. La couche du plus bas niveau est le coordinateur de tâches qui maintient l'illusion de la simultanéité entre plusieurs processus (réflexifs, percepteurs, cognitifs et environnementaux). La deuxième couche implémente l'environnement proprement dit : les cartes du parc et la simulation des feux. La troisième couche contient les définitions des agents alors que la quatrième prend en charge l'organisation de ces agents, leurs communications ainsi que les relations d'autorité qui les relient.

a- Les deux premières couches ; l'environnement Phœnix

Les deux premières couches de Phœnix constituent le simulateur d'événements.

Le coordinateur de tâches est responsable de simuler la simultanéité des trois événements suivants :

- Les feux : plusieurs feux peuvent se déclencher. Ces feux sont essentiellement des automates cellulaires qui se propagent selon les conditions environnementales (direction et vitesse du vent, type d'essence, humidité et inclinaison du terrain).
- Les actions physiques des agents : Les agents se déplacent d'un endroit à un autre, rapportent ce qu'ils perçoivent et construisent des lignes d'arrêt.
- Les actions internes des agents incluant la perception, la planification et les réactions réflexives face aux conditions immédiates de l'environnement.

Quant à la deuxième couche (la carte), elle représente le parc à simuler et est composée de plusieurs structures bidimensionnelles et d'informations sauvegardées pour chaque coordonnée concernant la couverture du terrain, l'élévation, les routes, les maisons, les rivières, etc.

Le simulateur d'événements Phoenix (les deux premières couches) est générique. Il peut être appliqué à n'importe quelle simulation qui implique une carte et des processus. Par exemple, l'environnement des feux de forêt peut très bien être remplacé par un environnement de marée noire. Les processus de feu peuvent être remplacés par des processus de propagation de pétrole. Les définitions des bulldozers et des avions d'arrosage seront aussi remplacées par des définitions de bateaux et des barrages.

b- Troisième couche : le modèle agent

Dans Phoenix, un agent a deux mécanismes indépendants pour la génération des actions :

- Un premier mécanisme génère rapidement des actions réflexes (de l'ordre de quelques secondes en temps de simulation). Les réflexes qui peuvent surgir dans le comportement d'un agent sont provoqués par les notifications des senseurs associés à l'agent. Ces réflexes permettent à l'agent d'agir de façon rapide afin d'éviter d'éventuelles catastrophes par analogie aux réflexes humains. Par exemple, un bulldozer (l'agent) peut être amené à s'arrêter sur le coup par un réflexe si jamais il fonçait sur le feu.
- Un deuxième mécanisme génère des plans qui peuvent prendre des heures (en temps de simulation) pour s'exécuter. Ceci est à la charge de la composante cognitive de chaque agent. Au lieu de générer toujours de nouveaux plans comme cela est fait par les planificateurs hiérarchiques classiques, la composante cognitive de Phoenix instancie et exécute des squelettes de plans déjà sauvegardés dans une librairie de plans. Les auteurs de Phoenix pensent que quelques squelettes sont suffisants pour satisfaire la demande de planification dans le problème des feux. Ceci éviterait de les générer en temps réel. Le fonctionnement de cette composante cognitive est illustré par les auteurs via l'exemple suivant : lorsqu'un nouveau feu est détecté, un plan approprié est sélectionné dans la librairie des plans. L'agent instancie ce plan (qui est en fait un squelette) avec les données courantes sur la météo, les ressources, etc. Par exemple, si le feu n'est pas large et le temps est calme, un plan impliquant un seul bulldozer est sélectionné et ensuite instancié selon les données de la situation en cours.

Mais qu'est ce qui se passe si tout ne va pas comme prévu ? Les agents de Phœnix ont trois capacités rudimentaires pour gérer les événements imprévus :

- Les réflexes (expliqués ci-dessus) qui ont, entre autres, pour tâche d'assurer le fonctionnement de l'agent jusqu'à ce que sa composante cognitive arrive à un plan.
- Le recouvrement des erreurs et la replanification : Lorsqu'une erreur est rapportée, un mécanisme renvoie le type de l'erreur et sélectionne un plan d'action associé à l'erreur.
- Les *enveloppes* : les agents Phœnix peuvent surveiller leur propre progression en générant des états futurs et en les comparant à la progression actuelle afin de prédire les échecs avant qu'ils ne surviennent. L'état futur attendu est représenté par des fonctions. Par exemple, si un agent doit être à un rendez-vous à une heure précise, une projection est faite pour déterminer si l'agent est capable de s'y rendre dans les délais en se basant sur sa vitesse et la distance à parcourir.

c- Quatrième couche : l'organisation des agents

Les agents dans Phœnix ont une organisation hiérarchique et centralisée. Les auteurs de Phœnix avouent que leur modèle centralisé n'est ni robuste (p.ex. si l'agent chef-pompier devient invalide) ni particulièrement efficace.

Un agent chef-pompier coordonne les activités de tous les agents bulldozer en leur envoyant des directives et en recevant des rapports sur les observations du feu, les mises à jour des positions et les actions accomplies. Le chef-pompier maintient ainsi une vue globale de la situation et se base donc sur ces rapports pour choisir des plans globaux à partir de sa librairie de plans. Les actions de ces plans sélectionnés sont transmises aux agents bulldozers, qui à leur tour sélectionnent des plans de leurs propres librairies afin de mettre en œuvre les directives du plan global. Toutes les communications dans ce modèle centralisé se passent entre le chef-pompier et les agents individuels ; aucune communication n'est prévue entre ces agents individuels (les bulldozers).

Le chef-pompier maintient la cohérence globale, coordonne les ressources disponibles afin de contrôler le feu. Il est également responsable de coordonner les agents en calculant leurs points de rendez-vous, en décidant comment déployer les ressources, en notant quand le feu est sous contrôle, etc.

Quant aux autres agents (bulldozers), chacun d'entre eux a une vision locale de l'environnement basée sur ses propres capacités sensorielles. Il a aussi un accès aux cartes décrivant les caractéristiques statiques du parc. Il n'a cependant de connaissance sur le processus dynamique (progression des feux) que de ce qu'il perçoit par ses propres moyens ou ce qu'il reçoit dans les directives de son chef.

Les senseurs associés aux agents ont un champ de vision limité. Un agent bulldozer par exemple a une vision de 512 mètres. Il a également un autre senseur qui lui permet de relever le contour du feu (dans son champ de vision). Quand le bulldozer construit une ligne d'arrêt sur le contour du feu, il utilise un senseur qui suit le contour feu, en combinaison avec une action réflexive qui aide à garder la ligne parallèle au contour du feu. Au fur et à mesure que le contour change, le senseur qui suit le contour enregistre les changements, ce qui déclenche un réflexe pour ajuster le mouvement du bulldozer.

2.4.2.5.4 Résumé

Phoenix est un framework pour implémenter et tester plusieurs agents autonomes dans un environnement complexe. Il est nécessaire cependant de distinguer entre le simulateur Phoenix de l'environnement Phoenix et les agents Phoenix [Hanks et al., 93].

Le simulateur a essentiellement trois fonctions : maintenir et mettre à jour la carte, synchroniser les activités de l'environnement et des agents et enfin rassembler les données. L'environnement Phoenix inclut une représentation de la carte (ici le parc *Yellow Stone*) ainsi que les tâches qui implémentent le feu. Les agents Phoenix quant à eux, génèrent des tâches pour simuler le chef-pompier, les bulldozers, les hélicoptères, etc. Les tâches agents incluent le déplacement des agents sur la carte, la construction des lignes d'arrêt, la prédiction de la propagation des feux, la planification de comment les bulldozers vont attaquer le feu, la surveillance de la progression, la détection des éventuels échecs de plan, etc.

L'environnement Phœnix ne contient que deux types d'objets : les agents et les feux [Hanks et al., 93]. Cependant, chaque cellule de la carte de l'environnement contient des informations que les agents et les feux utilisent pour déterminer leurs comportements. Par exemple, un bulldozer se déplacerait plus facilement dans les cellules désignant une route et les feux brûlent plus rapidement dans les cellules à pente élevée.

2.4.2.5.5 Critique

Nous énumérons ici les points qui nous semblent problématiques dans Phœnix:

- Phœnix utilise une librairie de plans afin de permettre aux agents de planifier. Nous pensons qu'il est plutôt recommandé qu'un spécialiste humain (gestionnaire des feux) prenne l'initiative de la planification, ne serait ce que pour choisir un point d'ancrage (barrière naturelle) à partir duquel les agents peuvent bâtir leurs plans. Cette suggestion est issue de notre étude du comportement réel lors de la lutte contre les feux : la décision (au moins stratégique) relève des tâches de l'humain (Section 3.2.4 du Chapitre 3).
- L'environnement Phœnix n'est pas tout à fait agentifié : seuls les feux et les acteurs (bulldozers, avions, etc.) sont représentés par des agents. Il serait intéressant d'agentifier les ressources et les structures. Par exemple, un agent représentant une infrastructure de la région pourrait influencer et même participer à la planification en se manifestant si les feux s'approchent dangereusement de lui.
- Grâce aux nouvelles technologies (GPS, télédétection et senseurs web), une vision plus précise de la situation en cours peut être établie et former ainsi une meilleure base pour une planification plus effective. Dans Phœnix, la planification se base sur une approximation de la situation qui est souvent assez loin de la réalité.
- Phœnix, bien que riche en idées, reste un outil d'entraînement car il ne fait pas le lien entre le monde réel et le monde de simulation. Pour arriver à un tel but, un pont doit être construit entre les deux mondes; Ce pont est possible encore une fois grâce aux nouvelles technologies de communication.

- Phoenix ne permet pas une sensibilité spatiale fiable et réaliste aux agents. Les mécanismes de perception, de navigation et de raisonnement dans l'espace de simulation ne sont pas très développés.
- Dans la stratégie indirecte de lutte contre le feu, un point essentiel lors de la construction des lignes d'arrêt – et qui semble être négligé un peu dans Phoenix – est le choix de l'endroit où passe la ligne d'arrêt. Ce choix dépend énormément des propriétés du terrain. Le choix doit aussi respecter certaines règles assez contraignantes : minimiser l'espace sacrifié, éviter les courbes serrées et profiter au maximum des barrières naturelles (voir Chapitre 3 : Domaine d'application). Le gestionnaire peut également contribuer de façon très significative en choisissant le point d'ancrage. Nous pensons qu'actuellement une approche à base d'agents sensibles à l'espace géographique peut être mise en place afin de résoudre de tels problèmes et planifier avec précision l'itinéraire de la ligne à tracer. De tels plans précis pour construire des lignes d'arrêt sont faisables et crédibles car les bulldozers peuvent suivre de façon précise le tracé choisi en se guidant par GPS. La technologie nécessaire existe et est actuellement utilisée dans le domaine de l'agriculture de précision [O'Connor 96; Marchant et al., 97; Zhu et al., 98].
- Dans Phoenix, le mécanisme de planification et de replanification est assez simple. Le fait que les auteurs utilisent un système multiagent laisse croire que Phoenix profite des avantages de la planification multiagent. Ce qui n'est malheureusement pas le cas. En effet, dans Phoenix, la seule communication inter-agent se fait entre le chef-pompier et les autres agents. Cette organisation centralisée risque de ne pas marcher dans un monde réel faute de robustesse et d'efficacité. D'ailleurs les auteurs de Phoenix avouent eux-mêmes cette faiblesse. Il est alors plus judicieux de construire un vrai système multiagent dans un environnement distribué. Les agents doivent pouvoir communiquer entre eux afin de résoudre les conflits, trouver de meilleures alternatives lors des échecs, mieux se coordonner et surtout se débarrasser d'une centralisation très coûteuse et peu fiable. En effet, l'un des buts de l'environnement de simulation est de permettre

aux différents planificateurs et acteurs d'interagir afin de trouver de meilleures solutions en combinant les efforts des différents participants.

- Phœnix introduit la notion d'*enveloppe* pour anticiper les échecs, ce qui est une bonne initiative. Toutefois, ces enveloppes sont des fonctions qui ne profitent pas de l'environnement de simulation. Il serait plus judicieux de simuler les actions futures au lieu de se contenter de les calculer. En effet, la simulation visuelle est un moyen de validation visuelle qui sollicite les capacités cognitives humaines. Ces capacités cognitives humaines sont essentielles pour aboutir au réalisme qui fait défaut dans la plupart des systèmes de décision en temps réel.
- Enfin, Phœnix a été développé avant l'apparition de l'approche DCP. Il serait donc intéressant de tester cette approche sur le problème des feux de forêt.

2.4.3. Conclusion

Notre étude des différents travaux actuels qui utilisent les agents logiciels dans la planification basée sur la simulation confirme les limites de la technologie agent que nous avons citées dans le chapitre précédent (Section 1.2 du Chapitre 1) : utilisation incomplète du paradigme agent dans le raisonnement spatial, faible sensibilité géospatiale, manque de réalisme, peu d'anticipation sur les plans, etc. Ces limites peuvent en fait être résumées en deux grands handicaps : faible implémentation de l'approche DCP par les agents logiciels et faible considération des contraintes spatiales de l'environnement par les agents.

Nous avons toutefois noté quelques idées intéressantes comme la notion d'*enveloppe* dans Phœnix, l'approche *Anticipatory Planning* implémentée par [Surdu et Hill 00] et la notion de *cost maps* adoptée par [Horn et Baxter 00].

2.5. Les SIGs : un moyen de lutte contre les désastres

2.5.1 Motivation

Lors d'un incendie de forêt, les pompiers donnent beaucoup d'importance à la connaissance du terrain [Sauvagnargues et al., 00]. Par exemple, il est très déconseillé d'envoyer des véhicules dans la forêt en feu sans vérifier si le chemin pris est sans cul-de-sac ou s'il y a des réservoirs d'eau sur place. Ces informations peuvent être fournies par des cartes. Toutefois, la nouvelle tendance veut qu'on délaisse ces cartes au profit des SIGs. Pourquoi ne pas utiliser des cartes ? Nous allons nous référer à l'expérience de la France en la matière.

En 1970, une réforme française a indiqué qu'il fallait fournir aux pompiers de meilleures cartes qui ont un système de coordonnées spécifique et qui contiennent toutes les informations pertinentes pour les pompiers et qui n'existent pas dans les cartes conventionnelles, par exemple : les chemins forestiers (pente, largeur, etc.), les ressources d'eau (types, capacités, etc.), etc.

Cependant, ce système de carte avait deux grands inconvénients [Sauvagnargues et al., 00] :

- Le coût élevé de ces cartes : il fallait pour chaque département fournir entre 15 et 20 cartes à 15\$ la carte et ce dans tous les véhicules, ce qui revenait à plusieurs centaines de milliers de dollars pour chaque département.
- La mise à jour des informations : les cartes deviennent obsolètes un an après leur production (changement des chemins, des ressources d'eau...), ce qui demande un renouvellement de toutes les cartes, déjà très coûteuses.

Il s'est avéré ainsi que les cartes papier ne sont pas très efficaces et rentables. Les départements du Sud de la France ont alors implémenté une application SIG [Sauvagnargues et al., 00] pour répondre à leurs besoins.

2.5.2. Importance des SIG lors des désastres naturels

Les désastres naturels sont le résultat de plusieurs caractéristiques géophysiques complexes ainsi que des circonstances sociales liées [Herath 01]. Le danger peut être météorologique à l'origine tels que les cyclones ou les grandes tempêtes, un processus de la terre tels que les séismes, les volcans, ou encore une combinaison des deux comme dans le cas d'une inondation. Tous ces événements dépendent de l'espace puisque le danger est aggravé par les caractéristiques géologiques, topographiques et géographiques du terrain. Ils peuvent aussi tourner au désastre quand ils affectent la société ou l'environnement. Afin de mieux contrôler l'impact de ces désastres, il est nécessaire de comprendre et de connaître les différentes caractéristiques du terrain ainsi que de la société (population, infrastructures...) [Herath 01]. Dans ce sens, les SIG jouent un rôle très important puisqu'ils ont la capacité de représenter la situation du monde réel de façon assez précise et selon plusieurs couches d'informations (cartes) qui peuvent être combinées de manière bien précise pour déterminer l'impact du désastre naturel en introduisant la dimension du risque (ou le danger). Dans le cas d'une inondation, les informations de cette « dimension risque » sont la hauteur des eaux, la vitesse et la distribution de l'inondation sur la région. Dans le cas, d'un feu de forêt, il s'agirait surtout de la propagation des feux et de la fumée. En combinant ces informations avec celles relatives au terrain (les chemins, les pentes...) et à la société (les infrastructures, la population...), plusieurs opérations telles que l'évacuation de la population peuvent être menées de façon plus efficace et précise.

2.5.3. Les SIGs au support de la simulation et de la modélisation

La simulation mathématique et la modélisation des processus du désastre constituent la principale procédure pour les prévisions de tels événements [Herath 01]. Malheureusement, les SIGs actuels ne peuvent pas supporter, lors de la modélisation des désastres, des informations qui varient dans le temps et des mises à jours dynamiques de ces informations [Herath 01]. Toutefois, les SIGs sont très utilisés pour préparer les informations en entrée des modèles mathématiques.

Le SIG est devenu un outil important pour la planification des infrastructures urbaines en général. En cas de désastre, le SIG est bien utilisé pour aménager les terrains, reconstruire et même parfois pour réagir face au désastre [Herath 01]. Le SIG est ainsi un outil utilisé pour sécuriser les lignes électriques, l'eau et le gaz en identifiant les portions qui deviennent non fonctionnelles après un désastre. Le SIG est également sollicité pour le processus de rétablissement de la situation après le désastre en identifiant les endroits sûrs pour l'évacuation.

Dans le passé, les systèmes de réaction (*response systems*) basés sur le SIG ont connu une réussite surtout dans l'industrie de la lutte contre les feux, où la propagation des feux et les conditions des feux ont été généralisées et le système utilisé en continu. Cependant, pour les gros désastres, l'aspect chaotique et changeant de la situation a empêché l'utilisation effective de tels systèmes [Herath 01].

Toutefois, le récent progrès dans le domaine de la communication est en train de remédier à ce problème. En effet, l'accès facile au GPS et à la communication mobile rend possible la conception de système de réaction en temps réel. Herath [Herath 01] mentionne l'exemple d'un accident chimique dans un complexe industriel. La propagation du gaz est modélisée en utilisant les données les plus récentes sur le vent ainsi que les informations sur les bâtiments et les infrastructures des alentours. Les gestionnaires peuvent par la suite utiliser les résultats de la simulation (prévisions de la progression du gaz) pour avertir les bâtiments en danger. Pour ce faire, le recours au GPS et aux réseaux de diffusion, est imminent. Le couplage du SIG avec la communication et les outils de modélisation, est nécessaire pour pouvoir utiliser un SIG en vue de répondre à un désastre [Herath 01].

2.5.4. Les SIG dans les systèmes d'aide à la décision

La plupart des initiatives intégrant le SIG dans un système d'aide à la décision pour les désastres, sont apparues dans cette dernière décennie. Nous allons mettre surtout l'emphase sur les réalisations dédiées à l'analyse des scénarios temps réel.

Contrairement aux applications relativement simplistes du SIG qui utilisent leur puissance cartographique [Johanson 94, Mather 93, Simon et Greenaway 94], Lanza et Siccardi [Lanza et Siccardi 95] ont proposé une méthodologie pour évaluer les dangers d'inondation en intégrant l'utilisation du SIG et la modélisation du phénomène. Un SIG est utilisé pour déduire des entrées pour le modèle, alors que ce modèle est exécuté à l'extérieur du SIG, vu que ce dernier a des capacités de modélisation numérique et de simulation limitées [NewKirk 93].

La modélisation de l'évacuation basée sur un SIG est bien traitée par la recherche sur les désastres naturels. Les données physiques du danger sont aussi importantes que celles concernant la population (et les infrastructures). Ces recherches ont surtout été incitées par des désastres tels que celui du réacteur nucléaire de Three Mile Island en 1979 et celui de Chernobyl en 1986. Dans le cadre de la modélisation de l'évacuation basée sur le SIG, Cova et Church [Cova et Church 97] ont proposé le concept de zone de planification d'urgence (Emergency Planning Zone EPZ). Dans [De Silva et al., 93], les auteurs proposent un exemple de système d'aide à la décision spatial (*Spatial Decision Support System* SDSS) pour une planification en cas d'urgence et qui communique avec des modèles de simulation et des logiciels SIG commerciaux, et ce afin de modéliser les routes d'évacuation lors d'un désastre radiologique. SDSS utilise des structures de données spatiales et des algorithmes de modélisation SIG du réseau routier, combinés avec des modèles programmés de simulation afin de prédire le trafic dans le réseau routier sous différents scénarios (pannes de voitures, fermetures de routes, etc.). Le SDSS est conçu comme un outil de planification de contingences afin d'évaluer différentes stratégies possibles face à un événement imprévu. Cependant, les simulations en temps réel ne sont pas possibles car le système modélise le comportement individuel de chaque véhicule, ce qui nécessite un temps d'exécution assez considérable. De semblables travaux ont été réalisés tels que le modèle *DYNamic Evacuation* DYNEV [Newsom et Mitrani 93]. D'autres projets ont également été menés pour d'autres types de désastre : Johanson et Smith [Johanson et Smith 94] ont appliqué un SIG pour gérer les risques des cyclones; Vermeire et Watson [Vermeire et Watson 94] ont proposé un système basé sur la modélisation numérique et le SIG pour gérer les risques des tempêtes.

2.5.5. Conclusion

Nous constatons que la technologie SIG est désormais nécessaire pour une meilleure utilisation de l'espace géographique par les applications d'aide à la décision. En effet, le SIG offre des données spatiales fiables exploitables par des programmes informatiques ou par des agents logiciels, ce qui laisse croire que le SIG est un outil incontournable pour la planification dans des situations de crise telles que les désastres naturels et plus particulièrement les feux de forêt.

Cependant, la plupart des études qui intègrent le SIG à la modélisation et à la simulation, ne s'intéressent malheureusement pas au temps réel, ce qui limite beaucoup leur efficacité. Beaucoup de chercheurs affirment que la génération actuelle des SIGs commerciaux est incapable de faciliter l'aide à la décision en temps réel pour gérer des désastres naturels sans apporter des modifications significatives ou sans intégration de ces SIGs à des modèles externes [Zerger et Smith 03].

Comme nous avons mentionné dans la Section 2.4, la simulation peut être un bon support à la planification. Dans ce cas précis, impliquer les SIGs au processus de planification, revient à faire des simulations basées sur des données SIG. On parle alors de géosimulation. Une synthèse du domaine (géosimulation) est présentée dans la section suivante.

2.6. Géosimulation

2.6.1. Généralités

La géosimulation est un terme utilisé pour décrire une nouvelle tendance de la modélisation de la simulation spatiale. Elle dérive certainement de la modélisation urbaine et de la simulation classique, mais aussi des récents développements en informatique et en science d'information géographique.

La géosimulation est basée sur une combinaison de diverses techniques et de théories, offrant ainsi une perspective unique qui fait défaut chez la simulation traditionnelle. Plus précisément, elle offre une vue sur les phénomènes urbains, basée sur des objets en

interaction représentant le dynamisme d'individus et d'autres unités du monde réel, et ce, dans un temps approchant le temps réel.

La géosimulation est assez similaire aux approches traditionnelles de modélisation telle que la micro-simulation. Cependant, plusieurs facteurs [Benenson et Torrens 04] la distinguent de ces prédécesseurs :

- La représentation des unités spatiales : Les modèles urbains traditionnels se basent sur des unités spatiales modifiables alors que la géosimulation utilise des objets discrets et spatialement non modifiables, et ce à une échelle « microscopique ».
- Description des relations spatiales : La modélisation classique des interactions spatiales décrit une et une seule forme d'interaction, qui est le flux d'information entre les unités spatiales. À l'opposé, la géosimulation met l'accent sur le comportement interactif des objets géographiques élémentaires selon une large variété de moyens. Les interactions peuvent être aussi bien des flux que d'autres relations spatiales telles que la diffusion, l'agrégation, l'action à distance, etc. De plus, les relations qu'on peut observer à grande échelle (p.ex. à un niveau intra-urbain) peuvent être modélisées en assemblant ces unités élémentaires.
- Le temps : Les modèles urbains classiques ne considèrent pas le temps dans sa continuité, mais seulement certains instants précis. Les périodes intermédiaires sont ainsi pauvres en information. La géosimulation, par contre, se base sur un temps quasi continu approchant le temps réel. Il va sans dire que le nombre d'hypothèses qui peuvent être explorées dans la simulation est beaucoup plus important dans ce cas.
- L'objectif de la simulation : La géosimulation s'écarte de l'objectif par défaut de la simulation qui se limite à la prédiction. Elle tend, à travers les projets réalisés jusqu'à date, vers des simulations pour exploration de scénarios. La géosimulation est en train de devenir un outil d'aide à la décision.

2.6.2. Techniques de réalisation

La recherche en géosimulation s'appuie sur plusieurs techniques pour améliorer la technologie de la simulation spatiale : la dérivation de nouveaux algorithmes pour le calcul spatial, les nouvelles méthodologies pour concevoir des entités spatiales et les relations qui les relient, l'application des modèles de simulation aux problèmes du monde réel, les nouveaux logiciels pour expérimenter les systèmes géographiques. Plusieurs idées ont été inspirées à partir d'autres domaines tels que l'informatique, la physique, la chimie, les mathématiques, écologie, l'économie et la biologie. Parmi ces disciplines, l'informatique se distingue comme l'une des plus pertinentes pour la géosimulation [Benenson et Torrens 04]. Alors que plusieurs géographes reprochent aux méthodologies empruntées des autres domaines que la géographie le fait d'être développées selon des contextes non spatiaux, la géosimulation semble par contre trouver en l'informatique et plus précisément en l'IA et les systèmes multiagents, un bon potentiel qui peut tenir compte du contexte spatial. Les automates cellulaires et les SMA sont les deux techniques les plus utilisées jusqu' à date pour implémenter la géosimulation.

2.6.2.1. Automates Cellulaires : Généralités

Alan Turing, John von Neumann et Stanislaw Ulam sont les premiers à avoir utilisé les automates cellulaires comme *computing media*. Plus tard, l'idée a été reprise par des domaines autres que l'informatique et implémentée sur plusieurs applications. La géosimulation figure parmi ces applications. Les composantes clefs des automates cellulaires sont : les cellules, les états des cellules, le voisinage, les règles de transition et le temps. L'espace d'une cellule a une structure régulière, qui ressemble à une grille à carreaux (comme un échiquier), et une étendue infinie. Les cellules sont occupées à tout instant par un état discret (p.ex. plein ou vide, mort ou vivant, bleu ou jaune, 1 ou zéro, etc.). Dans un contexte urbain, l'état de la cellule peut représenter tout attribut de l'environnement urbain : utilisation (résidentielle ou commerciale), densité (haute ou basse), texture (forestière ou non), etc. Chaque cellule a un voisinage de cellules adjacentes qui l'entourent. Le voisinage peut être décrit comme « Moore » (la cellule en

question et les huit cellules qui l'entourent) ou « Von Neumann » (la cellule en question et les quatre cellules cardinales qui l'entourent).

Malgré tous les avantages qu'offre l'approche à base d'automates cellulaires, son application sur la géosimulation n'est pas encore mûre [Benenson et Torrens 04]. En effet, il paraît que l'approche fait face à plusieurs sérieux défis, surtout lorsqu'elle essaye de sortir des expérimentations académiques théoriques vers des applications plus réelles telles que la simulation des scénarios opérationnels, le support à la planification ou encore la gestion urbaine. Certaines difficultés, auxquelles fait face l'approche dans le contexte urbain, sont décrites dans [O'Sullivan et Torrens 00].

2.6.2.2. Géosimulation par SMA

Dans un contexte urbain par exemple, l'environnement artificiel (ou virtuel) est l'espace dans lequel les agents doivent « habiter ». Ces agents sont capables d'y naviguer, d'explorer et d'interagir. Ils peuvent être conçus pour simuler une variété d'entités urbaines (si on se place dans un contexte urbain) telles que les personnes, les véhicules, etc. Les agents ont un ensemble d'attributs qui décrivent leurs caractéristiques. Les concepteurs essaient toujours de choisir des attributs qui reflètent le plus possible les entités réelles simulées. Quant à l'échange d'information entre les agents, il est beaucoup plus explicite qu'avec les automates cellulaires (via le voisinage). Les agents peuvent communiquer avec n'importe quel autre agent et même avec l'environnement lui-même.

Le comportement des agents est souvent contrôlé par des règles de transition. Ces règles gèrent les activités des agents : les objectifs que les agents cherchent à satisfaire, leurs préférences, etc.

2.6.2.3. Agents ou Automates Cellulaires?

Les SMA ont des caractéristiques similaires à celles des automates cellulaires. Cependant, les agents sont beaucoup plus libres de naviguer et d'explorer leur environnement spatial que les machines d'état individuelles des automates cellulaires. Ceci est dû au fait que le comportement spatial des agents n'est pas contraint à un treillis et que les interactions peuvent avoir lieu au-delà d'un voisinage. Le potentiel

d'interaction est ainsi beaucoup plus grand avec les SMA qu'avec les automates cellulaires.

La figure suivante (extraite du livre de Benenson et Torrens [Benenson et Torrens 04]) illustre la relation entre les SMA et la modélisation par automates cellulaires.

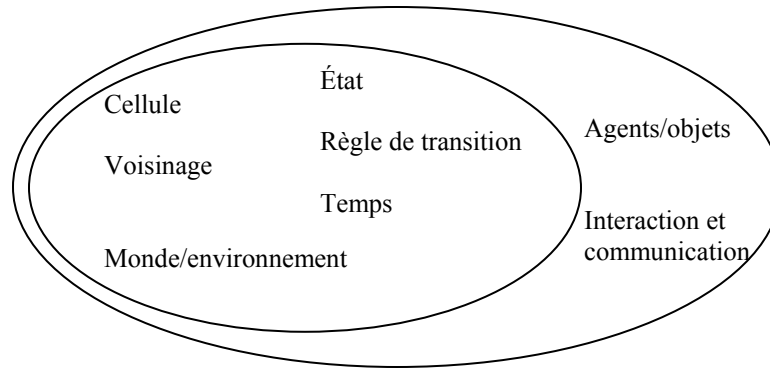


Figure 2.1. Relations entre SMA et modélisation basée sur les automates cellulaires.

De plus, les automates cellulaires, comparés aux SMA, semblent selon [Benenson et Torrens 04] manquer de :

- Mobilité
- Sens et intérêt
- Habilité sociale
- Adaptabilité
- Hétérogénéité des règles de transition.

C'est une combinaison de toutes ces raisons qui nous a poussé, dans notre travail, à opter pour le choix des SMA comme moyen de réalisation de la géosimulation.

2.6.3. Géosimulation basée sur les agents dans un monde réel

2.6.3.1. Applications

Bien que l'infrastructure nécessaire existe (en terme de SIG à haute définition) déjà pour appliquer la géosimulation à base d'agents sur des problèmes du monde réel, les réalisations restent très peu nombreuses [Benenson et Torrens 04]. Nous citons à titre indicatif les travaux de Turner et Penn [Turner et Penn 02] qui ont modélisé le comportement des piétons par des SMA, ou encore les recherches en modélisation du trafic routiers par des agents comme est le cas dans les systèmes PARAMICS [Paramics 05] ou TRANSIMS [Beckman 97].

2.6.3.2. Plateformes de géosimulation basée sur les agents

Plusieurs plateformes de géosimulation basées sur les agents existent déjà depuis des années. Swarm [Swarm 05] est probablement l'environnement de développement le plus connu et le plus utilisé dans la recherche en sciences sociales et en Vie Artificielle.

2.6.3.2.1. Swarm

La plateforme Swarm a été développée par Nelson Minar et Chris Langton au Santa Fe Institute. Swarm est un logiciel destiné à la simulation multiagent de systèmes complexes. Elle a pour objectif d'aider les chercheurs à étudier des modèles basés sur les agents dans divers domaines. L'architecture de base de Swarm est une simulation de plusieurs ensembles d'agents qui interagissent de façon concurrentielle. Avec une telle architecture, les chercheurs ont pu développer une large variété de modèles basés sur les agents.

Plusieurs extensions de Swarm ont été proposées telles que Kenge [Kenge 99] ou MAML [MAML 00].

2.6.3.2.2. MAGS

MAGS [Moulin et al., 03] est une plateforme de géosimulation basée sur des agents logiciels. Ces agents interagissent dans un environnement virtuel et y agissent en faisant appel à des capacités cognitives spatiales. La sensibilité spatiale de ces agents, ainsi que d'autres caractéristiques des agents (perception, mémoire, etc.) font que la plateforme MAGS est bien adaptée pour faire de la planification dans un contexte de géosimulation. En effet, grâce à des capacités cognitives évoluées, les agents de MAGS sont capables de mener des actions aussi complexes que la planification dans un environnement à forte contrainte spatiale. MAGS est la plate-forme choisie pour implémenter notre approche. Plus de détails sur MAGS seront présentés dans le Chapitre 4 et 7.

Conclusion du chapitre

Nous avons déjà conclu dans la Section 2.5 que le SIG est un outil incontournable pour la planification dans des situations de crise telles que les désastres naturels. Cependant, la géosimulation semble manquer de moyens pour faire de l'aide à la décision dans de telles circonstances. Le but du chapitre 4 est justement de proposer une architecture à base d'agents logiciels qui permet d'utiliser la géosimulation comme moyen de planifier dans des environnements dynamiques, réels et à forte composante spatiale. Un feu de forêt est justement un cas typique d'un tel environnement. Le chapitre suivant présente donc ce domaine d'application.

CHAPITRE 3

Domaine d'application

Ce chapitre est dédié au domaine d'application : les incendies de forêt. La première partie du chapitre donne un aperçu général sur les feux dans les forêts, ceci pour démontrer que le problème est très dynamique et à forte composante spatiale. La deuxième partie est consacrée à la procédure de lutte contre les feux ainsi qu'aux contraintes écologiques, stratégiques et logistiques qui entourent la procédure. Notre travail de recherche va consister en fait à construire un système de planification qui résout une partie de ces problèmes. Enfin, la dernière partie du chapitre fait une synthèse sur les différentes technologies et les différents systèmes informatiques qui sont actuellement utilisés dans la lutte contre les feux de forêts.

3.1. Généralités sur les feux de forêts

Chaque année des millions d'hectares de terres boisées brûlent, faisant dépenser des milliards de dollars pour leur extinction et occasionnant d'immenses pertes en bois, biens et espaces de récréation et, dans certains cas, vies humaines. Le phénomène est encore plus présent en Amérique du Nord. En 2002, le Canada a connu 7242 feux de forêt [RNC 02] qui ont ravagé plus de 2600000ha. Pour savoir plus sur les feux et leurs causes, voir Annexe A.

3.2. Facteurs influençant la propagation des feux

3.2.1. Le triangle du feu

D'après [ALCIF 02], la compréhension du comportement du feu implique la connaissance de la façon dont les combustibles prennent feu, dont les flammes se

développent et dont un feu se propage. Les gestionnaires des incendies ont besoin d'avoir des connaissances pratiques de certains principes scientifiques universels qui s'appliquent à tous les incendies. Le principe de « triangle du feu » est sûrement l'un des plus importants.

Pour se déclencher et progresser, le feu a besoin des trois conditions suivantes (formant le triangle du feu Figure 3.1.) :

- **une source de chaleur** (flamme, étincelle) : très souvent, l'homme est à l'origine des feux de forêts par imprudence (travaux agricoles et forestiers, cigarettes, barbecues, dépôts d'ordures,...) accident ou malveillance ;
- **un apport d'oxygène** : le vent active la combustion ;
- **un combustible** (végétation) : le risque de feu est davantage lié à l'état de la forêt (sécheresse, disposition des différentes strates, état d'entretien, densité, relief,...) qu'à l'essence forestière elle-même (chênes, conifères,...).

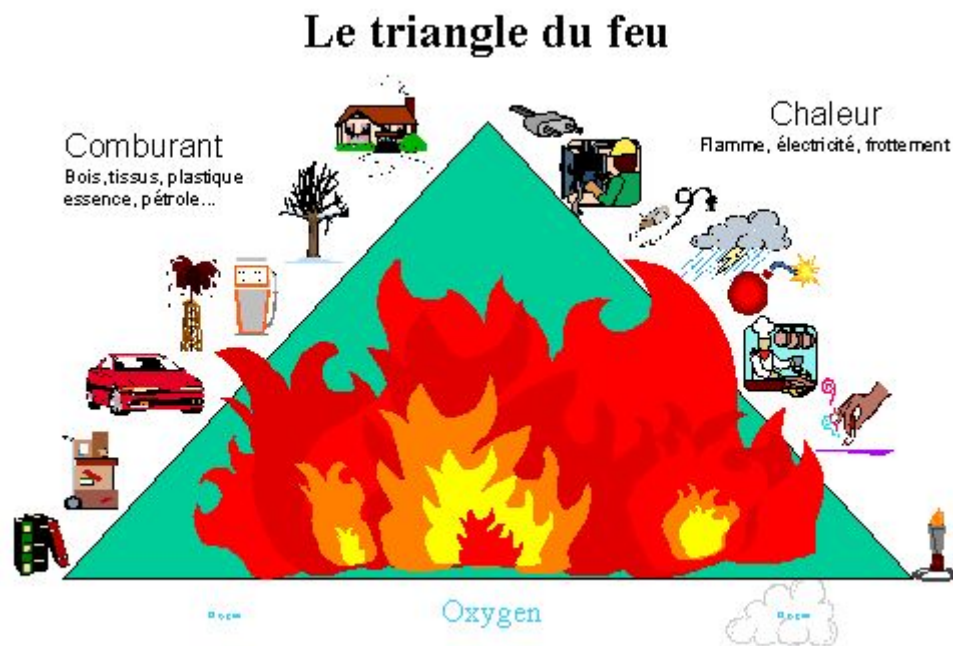


Figure 3.1. Triangle du feu

Si un de ces éléments est absent, un incendie ne peut pas se déclarer ou se propager. En enlevant ou en affaiblissant un des côtés du triangle du feu, l'intensité de l'incendie est réduite et la menace peut être éliminée.

Plus précisément, les incendies de forêt ont besoin de « combustibles légers » (aiguilles de conifères, des feuilles de feuillus, de l'herbe, du lichen, de la mousse ou de petits arbustes, etc.) continus et secs à la surface du sol pour se propager. Le comportement du feu est influencé par la teneur en humidité de ces combustibles. Une haute teneur en humidité ralentira l'incendie parce que la plus grande partie de l'énergie thermique sera utilisée pour éliminer l'humidité dans les combustibles. Les arbres en vie contiennent généralement beaucoup d'humidité tandis que les arbres morts en contiennent très peu. La grosseur et la disposition des combustibles sont également importantes : les gros combustibles ont besoin de plus d'énergie pour s'allumer et brûler que les combustibles légers.

Le deuxième élément nécessaire du triangle est l'oxygène. Il doit être présent en quantité suffisante pour assurer la combustion. Le vent peut fournir une nouvelle source d'oxygène à un incendie et orienter cet incendie vers une nouvelle source de combustible.

Le dernier élément requis pour compléter le triangle et par la suite enclencher et propager un feu, est une source de chaleur ou d'inflammation. Pour ce qui est des incendies de forêt, il peut s'agir d'une source naturelle, comme un coup de foudre, ou d'une cause artificielle, comme les êtres humains (feux de camp ou étincelles produites par de l'équipement). La température joue un rôle important car des températures élevées peuvent accélérer le séchage des combustibles, facilitant ainsi l'allumage et la propagation de l'incendie.

Lorsque les éléments du triangle du feu sont réunis, le processus de combustion peut débiter. Ce processus consiste en trois étapes plus ou moins distinctes qui se chevauchent (pour plus de détail consulter [ALCIF 02]).

3.2.2. Autres facteurs favorisant la propagation des feux de forêts

3.2.2.1 La topographie

La topographie joue un rôle déterminant pour la propagation du feu et son intensité. Le facteur pente est sûrement le plus influant. En effet, les pentes créent des flux naturels de vent qui accroissent donc la progression des feux. A titre d'exemple, une pente de 30 fait doubler (par rapport à un terrain plat) le taux de propagation du feu et réduit l'effet des efforts de lutte à la moitié [NFPA 89].

L'orientation des pentes peut aussi influencer le comportement du feu. Une pente qui donne sur le Sud ou sur le Sud-Ouest est plus touchée par le feu. Des drainages étroits et raides facilitent également la propagation des feux [PNWCG 02].

3.2.2.2 Barrières naturelles et artificielles

L'absence ou la présence de barrières naturelles (étendue d'eau, sol rocheux...) et artificielles (routes, lignes coupe-feu...) peut influencer énormément la propagation des feux et même leur contrôlabilité.

3.2.2.3 Routes d'accès

L'une des raisons pour lesquelles le feu peut devenir hors contrôle, est la difficulté d'accès. En effet, si l'acheminement des ressources (personnel et équipement) est retardé à cause de l'éloignement ou à la difficulté d'accès, le feu peut prendre de l'ampleur et devient ainsi incontrôlable.

3.2.3. Propagation des incendies

Les incendies de forêt peuvent se propager de trois façons différentes :

- Les incendies sous la surface brûlent dans la matière organique se trouvant sous la litière au sol et ils sont approvisionnés par une combustion couvante.

- Les incendies de surface se propagent avec un front de flammes qui brûlent la litière de feuilles, les branches tombées et d'autres combustibles se trouvant au sol.
- Les incendies de cimes brûlent au sommet des arbres, ce qui cause un feu très intense qui est difficile à contrôler. Les incendies de cimes nécessitent également un incendie de surface intense pour bien brûler.

Les spots de feu

Les spots de feu surviennent lorsque des braises sont emportées par le vent vers des zones non encore atteintes par les feux. Ces braises peuvent atterrir sur les toits des maisons ou sur une végétation inflammable ce qui entraîne de nouveaux feux. Ces nouveaux feux sont donc imprévisibles en raison de la nature même de leur déclenchement. Des pompiers peuvent très bien se faire entourer par les feux à cause de ce phénomène.

3.2.4. Conclusion

A l'issue de cette énumération des facteurs influençant l'allumage et la propagation du feu, nous pouvons conclure que les conditions météorologiques, l'essence et la topographie sont les principaux éléments favorisant la propagation des incendies de forêts. Quand un feu est allumé et puisque les gestionnaires des feux ne peuvent agir ni sur le climat ni sur la topographie, ils agissent généralement sur l'essence en essayant de raser la végétation dans certains endroits particuliers (lignes d'arrêt des incendies) afin d'empêcher la progression des feux. On parle alors d'attaque indirecte. Cette stratégie de lutte contre le feu ainsi que d'autres sont détaillées dans la section suivante.

3.3. Lutte contre le feu

3.3.1. Introduction

Le caractère destructif des incendies de forêts a forcé les pays à mettre en place des services d'incendie performants. Ces services mènent essentiellement des activités de prévention et de lutte

Si nous ne pouvons pas faire grand-chose pour empêcher les feux de causes naturelles, nous pouvons, jusqu'à un certain point, les prévoir et nous préparer à intervenir avec le personnel et l'équipement adéquat à l'endroit et au moment voulus. Par une bonne gestion du combustible, nous pouvons réduire la probabilité que les feux, qu'ils soient de cause humaine ou naturelle, deviennent hors de contrôle. Nous pouvons également réglementer diverses activités responsables d'incendies, notamment les pratiques de brûlage des déchets, et veiller à ce que l'aménagement du paysage autour de nos maisons et de nos quartiers d'habitation ne favorise pas les incendies et ne représente pas un danger pour nos biens et nos vies [PP 99].

La prévention des feux a pour objectif de minimiser l'incidence des feux. L'évaluation des risques de déclenchement des feux est alors nécessaire. Pour y parvenir, les causes des incendies précédentes sont enregistrées et analysées. En effet, si on ne connaît pas les causes exactes des incendies, on ne pourra pas les prévenir.

3.3.2. Lutte contre le feu : procédure

La lutte contre les incendies comporte toutes les activités liées à la maîtrise et à l'extinction du feu, de sa découverte jusqu'à son élimination totale. Une bonne planification de la lutte exige des informations détaillées sur les facteurs environnementaux influençant le comportement du feu, des ressources à affecter à son élimination et des modèles bien conçus de prédiction de son évolution. Cela signifie recueillir un grand nombre d'informations sur les conditions météorologiques, la topographie et le type de combustible ainsi que sur les données sur la détection et l'extinction collectées, emmagasinées et analysées [ALCIF 02].

A titre d'exemple, nous présentons dans ce qui suit la procédure de lutte adoptée en Ontario par [ALCIF 02].

Des séances de planification des mesures d'intervention ont lieu deux fois par jour pendant la saison des incendies. L'équipe de planification est dirigée par un agent de service et comprend des membres représentant diverses unités du programme (intervention en cas d'incendie, renseignements techniques et généraux sur les incendies,

services d'incendie, gestion des aéronefs, détection des incendies et prévention des incendies).

Une séance de planification en après-midi détermine le « *plan de fonctionnement stratégique* » pour la journée suivante. Une autre séance de planification a lieu le matin suivant pour obtenir une mise à jour sur la situation ou la température et modifier le plan au besoin. Une « *mise à jour du plan de fonctionnement stratégique* » est ensuite présentée.

La planification des mesures d'intervention se fait aussi dans les divers postes de gestion des incendies situés de façon stratégique dans la province. Les équipes d'intervention qui se trouvent dans les postes (dirigées par des agents d'intervention de secteur) fournissent des renseignements importants au cours des séances quotidiennes de planification des mesures d'intervention qui se déroulent dans les centres provinciaux et régionaux d'intervention. Les agents d'intervention de secteur sont chargés de fournir le plan d'intervention de la journée, selon les recommandations de l'agent de service et conformément au plan de fonctionnement stratégique et à sa mise à jour.

Le plan de fonctionnement stratégique et sa mise à jour traitent des sujets suivants :

- situation actuelle des incendies,
- mises à jour sur la température,
- prévisions concernant l'arrivée et le comportement des incendies,
- zones prioritaires et préoccupations,
- alertes et emplacement des équipes d'attaque initiale,
- exigences en matière de détection planifiée,
- problèmes et inquiétudes concernant les services, alertes et emplacement des hélicoptères et des avions-citernes.

Avant de se précipiter sur les lieux de l'incendie, certains facteurs déterminants sont à considérer [Firewise 05].

- Examiner l’historique des feux antérieurs qui ont affecté la région sinistrée afin de chercher certains patterns qui peuvent donc aider les gestionnaires à mieux comprendre la situation.
- Connaître les prévisions météorologiques de la journée (température, humidité, vitesse et direction des vents) et voir si ces conditions vont être stables ou si par contre des changements sont prévus.
- Connaître l’essence forestière de la région (herbes, arbres, tourbe...) et son âge (qui affecte sa combustibilité). Localisation des structures et leurs matériaux de construction car se sont également des essences.
- Examiner la topographie de la région. Par exemple, noter la présence des canyons et voir s’ils sont parallèles ou perpendiculaires à la direction prévue du vent (un vent parallèle au courant d’un étroit canyon peut engendrer un désastre). Être informé de la raideur des crêtes car le feu se propage sur les pentes de façon très rapide (plus vite que la vitesse d’un homme qui court). Noter la présence des barrières naturelles et artificielles car elles peuvent servir de point de contrôle pour le feu.
- Examiner les facteurs humains qui peuvent intervenir (des enfants qui sont de retour de l’école, des groupes de chasse...)
- Avoir une idée sur l’état du feu : sa vitesse de propagation, sa position, son comportement (il grimpe, il recule, il avance ou il fait des spots)
- Connaître tous ces facteurs dans une phase préliminaire permettrait de prendre de meilleures décisions. Cependant, il ne faut pas oublier qu’entre temps le feu est entrain de progresser et de gagner du terrain. Cette phase doit donc être assurée de la façon la plus rapide et efficace possible.

3.3.3. Différentes stratégies

Un incendie peut être arrêté en perturbant le triangle du feu par l’enlèvement d’au moins un des éléments du triangle : chaleur, oxygène ou combustible (Section 3.2.1). En

utilisant de l'eau ou du sol minéral pour lutter contre l'incendie, le feu peut être éteint en refroidissant le combustible sous le feu jusqu'à ce qu'il soit en dessous du point d'inflammation, ou en étouffant le feu (en enlevant l'oxygène présent). L'incendie peut également être arrêté en enlevant les combustibles disponibles, c'est-à-dire en les rendant non brûlable temporairement par l'application d'un mélange d'eau et de mousse [ALCIF 02].

Lorsque les équipes de lutte arrivent sur les lieux d'un incendie, des motopompes sont installées près de sources d'eau à proximité et des boyaux sont connectés. Selon la taille et l'intensité de l'incendie, ce dernier peut être éteint avec une attaque directe où des activités sont réalisées à la bordure du feu avec des outils manuels et de l'eau fournie par les motopompes, les avions-citernes et les réservoirs héliportés. Les équipes de lutte contre les incendies ont également le choix d'arroser seulement avec de l'eau ou avec de la mousse [ALCIF 02].

Si l'incendie est trop chaud pour être approché ou s'il se propage trop rapidement, les équipes de lutte peuvent faire une attaque indirecte. Ceci peut être réalisé en aménageant une ligne d'arrêt loin de l'incendie mais dans son chemin, à l'aide d'outils manuels et de machinerie lourde, tout en utilisant des barrières naturelles comme des lacs, des rivières et des affleurements rocheux. Le combustible qui se trouve devant l'incendie peut alors être enlevé en le brûlant à partir de la ligne d'arrêt en direction de l'incendie.

Ce brûlage peut se faire au sol en utilisant un lance-flammes portatif à action localisée ou à partir des airs avec une hélitorche attachée sous un hélicoptère [ALCIF 02].

Plusieurs mesures de contrôle peuvent être utilisées pour lutter contre un incendie. Un côté ou flanc peut être attaqué directement tandis qu'un autre est contrôlé à l'aide d'un brûlage des combustibles. Les équipes de lutte contre les incendies disposent souvent leurs boyaux de façon à profiter des barrières naturelles comme les chemins, les lacs, les rivières, les marécages, les affleurements rocheux et les peuplements de feuillus [ALCIF 02].

Pour la détermination de la stratégie d'attaque la plus appropriée à une situation donnée, plusieurs facteurs sont à considérer, comme le taux de propagation, le type d'essence, la

longueur des flammes, la disponibilité des ressources, les prévisions météorologiques, etc. [BLM 03]

Dans cette section, nous présentons des éléments de décision qui aident les gestionnaires des feux pour choisir la stratégie de lutte.

3.3.3.1. Attaque directe

Cette attaque peut consister à vaporiser les bords du feu avec un tuyau ou une pompe à dos, à écraser les flammes avec des pelles, à arroser les flammes (avec de l'eau ou du sable) par voie aérienne ou à construire une ligne d'arrêt le long des feux. Ce genre d'attaque est toujours plus sécuritaire pour les pompiers car ils ont toujours la possibilité de fuir les feux vers les aires déjà brûlées [Nicolas et Beebe 99]. L'équipe d'intervention (les pompiers) attaque généralement le feu par son flanc (le côté) et progresse vers le front du feu (*head*) jusqu'à encercler tout le feu. Lorsqu'il s'agit de construire une ligne pour arrêter le feu, les pompiers peuvent procéder au brûlage de l'essence qui sépare le feu à combattre et la ligne construite afin de minimiser le risque d'avoir des spots ou des couronnes de feu. Quant à la ligne en tant que telle, elle peut être construite par des outils manuels ou par les bulldozers [DPI 03]. Des agents de patrouille doivent régulièrement vérifier la ligne derrière l'équipe d'intervention afin de s'assurer que les feux ne traversent pas la ligne.



Figure 3.2. Attaque directe avec des outils à main et des pompes à dos [Nicolas et Beebe 99].

3.3.3.2. Attaque indirecte

Une attaque indirecte utilise des techniques de contrôle du feu tout en se tenant à une distance sécuritaire par rapport au feu. Les pompiers construisent des lignes assez profondes (d'environ 3m de large) qui entourent le périmètre du feu [Nicolas et Beebe 99]. L'attaque consiste à faire du *backfiring* à partir d'une ligne d'arrêt qui peut très bien être une barrière naturelle, une barrière construite sur-le-champ ou une barrière artificielle déjà existante (une route ou une ligne d'arrêt par exemple). L'objectif est de brûler et nettoyer une aire (aire rasée) d'environ 30m de large tout en essayant de sacrifier le moins de d'aire possible. Idéalement, l'aire à raser doit être entre deux lignes d'arrêt afin d'éviter les turbulences lorsque deux feux se rencontrent [Nicolas et Beebe 99].

Ce genre d'attaque nécessite aussi bien l'habilité et l'expérience que les ressources adéquates. Les intervenants peuvent utiliser des outils à main pour construire les lignes mais de préférence les bulldozers (si cette ressource est disponible).

Remarque: le *Backfiring* doit être mené seulement lorsque les conditions sont favorables : l'essence est uniforme, le vent est à moins de 15km.hr-1 et l'aire est éclairée [Nicolas et Beebe 99].



Figure 3.3. Construction d'une ligne d'arrêt lors d'une attaque indirecte [Nicolas et Beebe 99]

3.3.3.3. Facteurs de décision

Afin de choisir la meilleure stratégie à adopter pour combattre le feu, les gestionnaires (plus précisément les chefs des opérations) tiennent en compte plusieurs facteurs [DPI 03]:

- La sécurité des pompiers.
- Le comportement courant et futur du feu.
- La valeur du risque relative au feu.
- Les objectifs de la région (en termes d'aménagement).
- La disponibilité des ressources.
- L'accès au feu.

- L'économie
- La probabilité de succès des méthodes alternatives

Nous allons dans ce qui suit examiner les principaux facteurs avec plus de détails.

3.3.3.3.1. La sécurité des pompiers

La sécurité des pompiers doit rester l'une des premières priorités du gestionnaire lors de la prise de décision. La stratégie de lutte à adopter doit ainsi garantir la sécurité des pompiers avant toute chose. Il est à noter que l'attaque directe est plus sécuritaire en général pour les pompiers (car ils peuvent fuir vers les zones brûlées). Cependant, pour des grands feux, une telle stratégie devient fastidieuse pour les pompiers qui ne peuvent trop s'approcher des flammes à cause de la fumée et de la chaleur dégagée par le feu.

[Beaver 01] présente en détail un aperçu sur les règles à respecter lors du choix de la stratégie afin de garantir la sécurité des pompiers.

3.3.3.3.2. Le comportement courant et futur du feu

Afin de pouvoir choisir la meilleure stratégie, le comportement du feu doit être évalué. Plusieurs indices peuvent alors être calculés tels que la longueur des flammes ou encore l'humidité de l'essence vivante (Live Fuel Moisture LFM)

Longueur des flammes

Cet indice est donc calculé une fois le feu est en activité.

- Flammes de moins de 4 pieds de hauteur: Le feu peut généralement être attaqué sur le front ou le flanc par des pompiers utilisant des outils à main.
- Flammes de 4 à 8 pieds de hauteur: le feu est considéré comme assez intense pour engager une attaque directe sur le front avec des outils à main. Des bulldozers, des machines et des arrosages peuvent alors être plus utiles.
- Flammes de 8 à 11 pieds de hauteur: le feu présente dans ce cas de sérieux problèmes de contrôle. En effet, des couronnes ou des spots peuvent survenir et

aggraver ainsi la situation. Les efforts sur le front sont probablement inefficaces. Une attaque indirecte s'impose. Des arrosages par avion-tanker ou par hélicoptère sont nécessaires pour réduire la propagation des feux avant que les lignes d'arrêt construites par les bulldozers ou les pompiers ne soient effectives.

- **Flammes de plus de 11 pieds de hauteur:** Une attaque directe est inutile. Il faut se retirer vers les routes et les autres barrières naturelles ou artificielles (Figure 3.4.). Il faut également brûler de l'essence forestière entre les lignes d'arrêt et le feu.

Humidité de l'essence vivante (Live Fuel Moisture LFM)

Contrairement à la longueur des flammes, le LFM peut être calculé au préalable sur tout le territoire ce qui permet de prévenir les risques des grands feux et avoir une idée sur leurs ampleurs et sur leurs comportements futurs si jamais ils sont allumés (Tableau 3.2).

Tableau 3.2. Les stratégies à adopter selon le LFM [EIIFC 03]

LFM	Stratégie à adopter
>180%	Attaque directe sur le front, les lignes construites avec des outils à main doivent tenir sans problème.
151-180%	Une attaque directe est possible mais la construction des lignes doit être plus difficile. Des machines seront nécessaires sur le front.
126-150%	Une attaque directe est encore possible mais des machines sont nécessaires sur le front ainsi que des bulldozers pour construire les lignes. Si par contre le vent est assez fort, une attaque indirecte s'impose.
101-125%	Une attaque directe n'est possible que pour les petits feux. Sinon, une attaque indirecte est nécessaire.
75-100%	Une attaque indirecte est inévitable. Les bulldozers et les machines doivent être utilisées pour soutenir les

	opérations de brûlage et pour protéger les structures.
<74%	Le <i>backfiring</i> à partir des lignes indirectes et des barrières naturelles doit être considéré. Les avions d'arrosage doivent être prudents aux turbulences autour du feu.



Photograph courtesy of Ben Hennack.

Figure 3.4. Lorsque la longueur des flammes dépasse 11 pieds, l'attaque indirecte est inutile. Les routes, les rivières et d'autres barrières naturelles sont utilisées.

3.3.3.3. La disponibilité des ressources

Il est clair que le choix de la stratégie dépend fortement aussi des ressources dont dispose le décideur. Ces ressources incluent les équipes de pompiers, les véhicules lourds (bulldozers, tracteurs, camions de pompier), les avions-citernes et les hélicoptères.

3.3.3.4. L'accès au feu

Pour pouvoir appliquer une stratégie offensive (attaque directe), il faut être capable d'accéder au feu (au front). Le décideur doit donc être sûr que des routes peuvent mener les ressources dont il dispose à l'endroit voulu, à défaut de quoi une attaque indirecte est adoptée. Dans ce dernier cas, le décideur doit bien choisir l'emplacement de la ligne d'arrêt selon la possibilité d'accès. Il ne faut surtout pas oublier que certaines régions sinistrées ont un relief difficile et donc pas accessibles aux bulldozers.

3.3.3.4. Évaluation des deux stratégies

Dans ce paragraphe nous allons essayer de déduire quelques critères de décisions à partir des avantages et des inconvénients de chacune des deux stratégies : directe (Tableau 3.3) et indirecte (Tableau 3.4).

Tableau 3.3. Avantages et inconvénients d'une attaque directe [BLM 03]

Avantages	Inconvénients
La surface brûlée est minimale. Pas de terres additionnelles brûlées intentionnellement	Les pompiers peuvent être gênés par la chaleur, la fumée et les flammes.
Plus de sécurité pour les pompiers : ils peuvent s'enfuir vers les aires brûlées.	Les lignes d'arrêt peuvent être très longues et irrégulières car elles suivent l'allure des feux.
Possibilité de réduire le risque de propagation du feu vers les couronnes des arbres.	Ne prend pas avantage des barrières naturelles existantes pour stopper le feu.
Élimination des incertitudes de la stratégie indirecte.	Les pompiers peuvent accidentellement aider le feu à se propager (laisser échapper des matières inflammables)

Tableau 3.4. Avantages et inconvénients d'une attaque indirecte [BLM 03]

Avantages	Inconvénients
Profiter des barrières naturelles existantes pour stopper le feu.	Danger potentiel pour les pompiers car ils ne peuvent pas observer le feu.
Les pompiers travaillent sans être dérangés par la fumée ou par la chaleur.	Le feu peut traverser la ligne avant qu'elle soit prête.
Plus de temps à disposition pour construire la ligne.	Ne profite pas des lignes qui sont déjà brûlées.
Permet de choisir l'endroit de construction de la ligne (choisir des endroits à essence légère)	Risque du au <i>backfiring</i> .
Les lignes sont plus courtes et plus droites	

3.3.3.5. Construction des lignes d'arrêt des incendies

La ligne d'arrêt peut être construite par des pompiers avec des outils à main (Figure 3.5.) ou avec des bulldozers. Généralement, la méthode manuelle est utilisée dans les attaques directes car le feu est moins important et donc n'exige pas une ligne très large. A l'opposé, quand il s'agit d'une attaque indirecte, les bulldozers sont utilisés surtout pour avoir des lignes plus larges mais aussi plus profondes pour éviter que le feu passe par les racines des arbres. A titre indicatif, la largeur de la ligne doit être de même ordre de grandeur que la végétation à côté. Il faut ainsi une ligne d'au moins 1.5m de large si la végétation environnante est de 1.5m de hauteur [Nicolas et Beebe 99].



Figure 3.5. Construction d'une ligne d'arrêt à la main [Carr 02]

Pour construire une ligne d'arrêt, plusieurs facteurs sont à considérer pour décider quand et surtout où construire cette ligne [PFMT 03]. Ces facteurs sont essentiellement :

- Le type et la quantité de l'essence : il est recommandé de construire la ligne dans un endroit où l'essence n'est pas très inflammable et aussi facile à couper.
- La topographie : Il est clair que pour pouvoir construire la ligne, il faut éviter dans la mesure du possible les pentes car les bulldozers y sont moins performants et le feu y est plus dévastateur.
- Les conditions météorologiques : Il faut surtout tenir compte de la vitesse du vent afin de pouvoir estimer l'arrivée du feu à la ligne. La ligne doit donc être complétée bien avant et surtout doit être suffisamment large selon la vitesse du vent.

Pour construire des lignes sûres et sécuritaires, certains standards sont à suivre :

- Éviter le plus possible les endroits à essence lourde.
- Essayer de construire dans la mesure du possible des lignes droites et éviter les virages serrés.
- Utiliser les barrières existantes (naturelles et artificielles).
- Relier la ligne à un endroit sûr (inflammable).

Les points d'ancrages

Un point d'ancrage est une barrière pour le feu. Il peut être une route, un lac, une rivière ou même un large rocher. Ce point d'ancrage est utilisé pour empêcher que le feu n'entoure les pompiers quand ils construisent la ligne d'arrêt. Il est aussi utilisé comme base pour construire la ligne d'arrêt. En effet, le gestionnaire du feu essaye toujours de choisir l'emplacement de la ligne d'arrêt à partir d'un point d'ancrage. Ces points d'ancrage sont donc d'une très grande importance lors de la lutte contre le feu.

3.3.3.6. Les tactiques de suppression à impact minimal

Les tactiques de suppression à impact minimal ou *Minimum Impact Suppression Tactics (MIST)* [CPBWTF 99] ne forment pas une nouvelle classification de méthodes de suppression des feux mais plutôt une attitude à adopter et à respecter pour le bien de l'environnement.

Le concept de MIST consiste à utiliser le minimum possible d'équipements lourds pour éteindre le feu en respectant l'écosystème. Malheureusement, l'usage de tels équipements est parfois nécessaire pour sauver des vies et des propriétés surtout quand le feu est de grande ampleur. Cependant, des recommandations peuvent être déduites du concept MIST et qui peuvent donc être appliquées pour ce genre de situation. Il est évident que la sécurité des pompiers doit rester la première priorité. Voici quelques exemples de ces recommandations [CPBWTF 99]:

- Utiliser les barrières naturelles, les sentiers, etc. à disposition pour acheminer les bulldozers.
- Faire acheminer tous les véhicules (dans la mesure du possible) par le même chemin afin de limiter les dégâts sur la flore.

Utiliser des bulldozers pour construire des lignes d'arrêt quand c'est nécessaire dans le seul but de supprimer ou mitiger un feu et non pas pour sacrifier le sol.

3.4. Les systèmes de lutte contre le feu

3.4.1. Introduction

Dans le passé, et dans une large mesure aujourd'hui, la lutte contre le feu était pour l'essentiel un processus de réaction accompagné d'une faible capacité de traitement des données. Il n'y avait aucun centre de contrôle de l'information et des ressources, et en général ces dernières n'étaient allouées qu'après l'écllosion de l'incendie. Désormais, grâce au perfectionnement des techniques et à la grande diffusion des ordinateurs et de logiciels adaptés, on peut prévoir les incendies et leur évolution, prédisposer les ressources voulues et traiter le grand nombre de données nécessaires pour prendre des décisions résolutes en matière de lutte.

Un grand nombre de recherches ont été menées sur la mise au point de systèmes d'évaluation des risques d'incendie pour déterminer l'inflammabilité, la vitesse de propagation du feu, les obstacles aux opérations de lutte et l'impact du feu [Merrill et Alexander 87]. L'évaluation des risques d'incendie est désormais un outil de routine de la gestion des feux qu'utilisent de nombreux pays forestiers du monde mais qui a vu le jour et s'est développé principalement en Amérique du Nord (Canada et États-Unis), en Australie et dans l'ex-URSS. Plusieurs autres pays ont soit mis au point leur propre système d'évaluation des risques d'incendie soit adapté ceux existants à leurs propres besoins. Un système d'évaluation suédois connu sous le nom d'*Angstrom Index* a servi d'indicateur d'éclussions attendues d'incendies dans certaines parties de la Scandinavie. Le système australien a été testé et appliqué dans la région méditerranéenne de l'Espagne. Un système canadien modifié est en usage dans certaines parties de l'Espagne, au Mexique, au Venezuela et, dans une certaine mesure, en Argentine [Chandler et al., 91].

Grâce aux systèmes d'évaluation des risques d'incendie, les responsables de la gestion des feux sont maintenant en mesure de se fixer des objectifs, de formuler des politiques de gestion et d'établir des normes pour des situations forestières typiques. Cependant, la gestion des incendies et la prédiction de leur évolution sont des questions très complexes du fait du caractère incertain des feux de forêt. Dans la planification visant l'élimination

des incendies, il est indispensable de disposer en temps utile d'informations fiables sur les combustibles, les conditions météorologiques et la topographie, ainsi que de mettre au point des modèles bien conçus de prédiction du comportement du feu. La précision de cette prédiction dépend aussi, dans une large mesure, de la qualité de l'information sur les facteurs environnementaux. En outre, les responsables de la gestion des incendies devront acquérir une expérience spécialisée pour utiliser cette information aux fins d'une bonne planification du feu et de prises de décision appropriées.

Des techniques de pointe ont permis de mettre en place des systèmes informatisés avancés de gestion des incendies qui consentent de surmonter les faiblesses inhérentes à sa planification. La télédétection, l'intelligence artificielle, de nouveaux systèmes d'information électroniques et des systèmes de soutien de la prise de décision en matière d'incendies sont largement utilisés par les services d'incendie du monde entier. Les systèmes d'information géographique (SIG) se fondent sur l'emmagasiner, l'intégration, la manipulation, l'analyse et la présentation de données spatiales sur les terres qui permettent de planifier la gestion des feux de friche et sont directement utilisables par les services concernés [Fuenekes et Methven 88, Hamilton et al., 91].

Terral [Terral 96] a utilisé des SIG pour prévoir l'évolution du feu en évaluant sa propagation à différents moments, en estimant le degré d'urgence sur la base d'informations pertinentes sur les zones habitées et la végétation, en identifiant les ressources disponibles pour les opérations de lutte, en indiquant le matériel d'extinction le plus proche et la route d'accès à l'incendie, et en préparant une intervention et en l'introduisant dans une carte. Dans [Bilgili et Baskent 97], les auteurs ont donné une description détaillée du rôle des SIG dans la planification de la gestion des incendies. Xian Wen et ses collègues [Xian Wen et al., 97] ont utilisé des données par satellite pour évaluer la quantité de litière et estimer les pertes dues au feu.

Pour plus de détails sur les technologies utilisées, consulter Annexe A.

3.4.2. Systèmes informatiques actuels de gestion des feux

3.4.2.1. Introduction : les types de systèmes utilisés

Les systèmes d'information sur la gestion des incendies permettent de prendre des décisions judicieuses au moment nécessaire et de continuer à créer de nouveaux outils afin d'améliorer la gestion des feux. Toutefois, il est évident que le système informatique utilisé pour gérer les feux diffère d'un centre à un autre et d'un pays à un autre. A titre indicatif, nous allons parler du cas du Canada et plus particulièrement de l'Ontario.

Plusieurs types de systèmes sont utilisés : certains pour des buts logistiques (la gestion du matériel de lutte contre les incendies ainsi que la gestion des coûts et l'utilisation des aéronefs), certains pour enregistrer les informations en vue d'un usage ultérieur (faire des statistiques, aménager les terrains de façon plus sécuritaire, etc.) et enfin d'autres pour la simulation des opérations (et surtout la simulation des feux). Nous allons parler brièvement du deuxième type mais nous allons surtout nous concentrer sur les deux derniers (de simulation).

Système de gestion quotidienne de lutte contre les incendies

Il s'agit d'un système informatique utilisé pour enregistrer des détails sur les incidents et les incendies, la récolte de données sur la température et la foudre ainsi que l'affichage de données à jour et de prévisions sur une base quotidienne. Ces renseignements sont utilisés pour déterminer les besoins et l'emplacement des ressources (équipes, aéronefs, etc.).

À la fin de chaque saison des incendies, les renseignements récoltés avec ce système sont enregistrés dans une base de données appelée les « archives du feu ». Ces données historiques sur le feu sont utilisées pour analyser les incendies, établir des modèles et faire des exercices de planification à long terme.

Système d'analyse et de simulation : l'exemple de LEOPARDS

Le système LEOPARDS [ALCIF 02] est un modèle informatique qui simule les opérations sur une base quotidienne et horaire d'un organisme de gestion des incendies. L'utilisateur précise la composition de l'organisme de lutte (abondance, emplacement

initial et capacité des ressources ainsi que politiques de l'organisme, comme les règles de répartition).

Cet organisme virtuel est ensuite utilisé par le modèle informatique pour lutter contre des incendies. Les incendies pris en exemple par le modèle proviennent de données historiques et sont donc représentatifs de la charge combustible que doit affronter l'organisme de gestion des incendies. Pour chaque journée de la saison des incendies, le modèle évalue les besoins en matière de ressources et répartit les ressources disponibles. Au fur et à mesure que les incendies se produisent dans la journée, les ressources sont envoyées sur les lieux des incendies et chaque incendie est combattu sur une base individuelle. Les résultats de chaque incendie sont ensuite compilés et les totaux quotidiens sont surveillés. Ce processus se répète pendant chaque jour de la saison des incendies pour toutes les années où des données sont disponibles.

En utilisant le modèle de simulation plusieurs fois tout en faisant varier les ressources, les utilisateurs peuvent déterminer la meilleure utilisation des ressources. Par exemple, combien d'avions-citernes fourniraient les mesures d'intervention les plus efficaces et économiques? En plus de faire varier les ressources, les utilisateurs peuvent tester ce qui se passe lorsqu'ils modifient l'assise territoriale protégée ou ils peuvent évaluer différentes stratégies de répartition des ressources en tenant compte des coûts globaux et de l'efficacité des mesures de lutte. Le programme peut traiter une vaste gamme de questions.

Le programme de gestion des incendies continue de parfaire et de spécialiser le modèle et étudie la possibilité de modifier le modèle pour qu'il puisse être utilisé dans d'autres compétences territoriales.

3.4.2.2. Systèmes de simulation de la propagation

Les systèmes d'analyse et de simulation ont certainement besoin de modèles de propagation des feux. Ces modèles de propagation sont très difficiles à concevoir et à gérer en raison du grand nombre de variables qui sont impliquées. Les chercheurs essaient cependant de construire des modèles de plus en plus efficaces qui sont donc plus flexibles et se basent sur le plus de variables et d'informations possibles.

Plusieurs modèles de propagation existent. Nous allons encore une fois présenter ce qui est disponible au Canada (voir Annexe A). Le Modèle canadien de propagation des feux de végétation (MCPFV) [SCF 03], qui se base sur l'inventaire des complexes combustibles du Canada.

De tels modèles de propagation sont à la base des systèmes de simulation de la propagation des feux. Le modèle de propagation américain, basée surtout sur des modèles mathématiques, a donné naissance au système Farsite [Farsite 05]. Quant au modèle canadien, basé sur des données aussi bien théoriques qu'empiriques, il est utilisé pour un système canadien de simulation de la propagation, nommé Prometheus (voir Annexe B). C'est ce dernier système que nous avons adopté pour notre application.

3.4.2.3. Systèmes de planification dans les feux de forêt : outils

Plusieurs outils existent déjà pour traiter les problèmes de feux de forêt. La plupart de ces outils logiciels sont dédiés à la prévention des feux et à la simulation de la propagation des feux (voir [FRAMES 03] pour une liste complète de ces outils). Par contre, peu de travaux portent sur la planification de la lutte contre les feux de forêt en temps réel ou à la simulation des feux en temps réel. Parmi ces travaux, on peut citer Phoenix et Charade (voir Annexe A).

Conclusion

Les incendies de forêt représentent un scénario typique d'un problème de planification complexe dans un environnement dynamique et à forte composante spatiale. L'étude approfondie des incendies des forêts montre que le domaine est très complexe et justifie l'adoption d'une approche DCP pour planifier les opérations de lutte. Malgré tous les progrès technologiques (télédétection, senseurs web, GPS, modèles de propagation, etc.), la planification des opérations de lutte contre les feux, reste toujours une tâche fastidieuse et complexe. La plupart des outils informatiques actuels sont destinés à entraîner les équipes d'intervention et les planificateurs humains (les commandants) ou bien à fournir les données nécessaires à ces planificateurs pour prendre des décisions dans des situations réelles. Nous pensons aussi que le scénario des feux est idéal pour illustrer l'apport de

nos recherches. Nous visons donc la planification des opérations de lutte contre les feux en temps quasi réel et en tenant compte des caractéristiques du terrain et l'évolution continue de la situation.

CHAPITRE 4

Approche ENCASMA : Principes et Architecture

Nous nous proposons dans ce chapitre de construire une architecture capable d'aider des décideurs humains à planifier de façon plus efficace (cognitivement) en temps réel (ou quasi-réel) tout en tenant compte des contraintes géospatiales de l'environnement. Nous prenons comme environnement d'illustration un feu de forêt. En effet, tel que décrit dans le Chapitre 3, planifier des actions de lutte lors d'un feu de forêt n'est pas une tâche facile car l'environnement est très dynamique, à forte composante spatiale et à grande échelle géographique. Comme énoncé dans le Chapitre 1, nous comptons utiliser une approche de simulation basée sur les agents logiciels. Dans le Chapitre 2, nous avons mentionné le potentiel des agents logiciels ainsi que de la simulation pour résoudre des problèmes complexes de planification. Nous avons également énuméré les faiblesses des travaux précédents pour mettre ces éléments ensemble et surtout dans le cadre des applications réelles survenant dans un espace à grande échelle. Nous essayons alors dans ce chapitre de mieux utiliser la Simulation Orientée Agent pour résoudre des problèmes réels de planification dans un environnement géographique dynamique et à grande échelle.

Avant de parler de la solution en tant que telle, nous nous proposons de préciser le problème (évoqué dans le chapitre 1) que nous voulons résoudre, sous une forme plus formelle (Section 4.1). Ensuite, dans la Section 4.2, nous explicitons les principes sur lesquels se base la solution. L'architecture solution (ENCASMA) est présentée dans la Section 4.3 et la plate-forme de simulation utilisée dans la Section 4.4. La Section 4.5 donne un aperçu sur le genre d'applications que peut supporter l'architecture proposée. Les travaux connexes à notre architecture sont énumérés dans la Section 4.6.

4.1. Problème formel

4.1.1. Vue générale

Selon le contexte général de recherche énoncé dans la Section 1.1 du Chapitre 1, nous considérons un espace géographique (à grande échelle) réel dans lequel des acteurs disposant de ressources d'assistance diverses tentent de trouver un plan et de l'exécuter pour faire face à une situation dangereuse qui menace certaines ressources de l'espace considéré. Leur but est de contenir le danger avec le minimum de pertes possibles. Nous avons également souligné que les désastres naturels correspondent parfaitement à ce genre de situations. Le schéma de la Figure 4.1 illustre la situation.

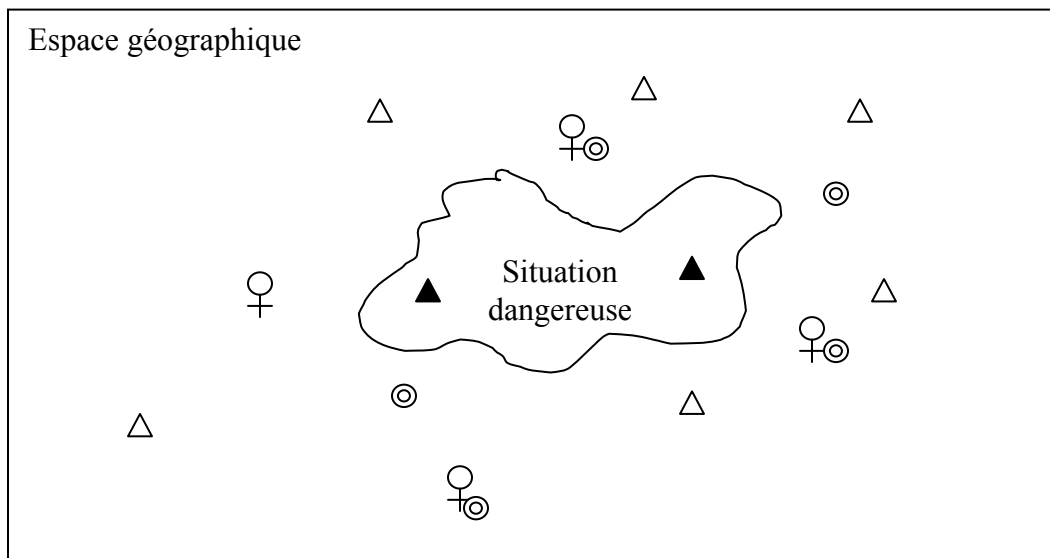


Figure 4.1. Problème à résoudre

Légende :

- △ Ressource à protéger
- ▲ Ressource déjà perdue
- ◎ Ressource d'assistance
- ♀ Acteur

4.1.2. Caractéristiques de l'environnement

Le genre de problème décrit ci-dessus est le cas typique d'un désastre naturel tel qu'un feu de forêt, une inondation ou une marée noire (Chapitre 2, Section 2.5 sur SIG et désastres). La planification dans de telles circonstances est soumise à un environnement qui complique davantage la tâche des acteurs. En effet, cet environnement est :

- *Réel* : Contrairement à beaucoup de travaux de recherche, il ne s'agirait pas de trouver des solutions pour un environnement d'essai ou un environnement de jeu. Il s'agit d'un espace réel avec tout ce que cela peut impliquer en termes de précision,
- *Dynamique* : tout peut changer. La situation dangereuse peut évoluer de façon imprévisible sous l'effet de facteurs externes ou par manque de connaissance de la part des acteurs (mauvaise estimation de la situation). Les imprévus peuvent également porter sur les ressources d'assistance (p.ex. ressource qui devient non fonctionnelle) et parfois même les acteurs (p.ex. obstacle imprévu sur le terrain).
- *Continu* dans le sens qu'il n'y a pas un seul problème bien défini à résoudre mais plutôt un flux continu de problèmes dont la plupart ne peuvent pas être anticipés. En effet, la situation dangereuse peut engendrer une succession de problèmes favorisés par l'étendue de l'espace et l'aspect dynamique de l'environnement.
- *En temps réel* dans le sens que c'est la situation dangereuse qui établit les données du problème auxquelles les acteurs doivent s'adapter. L'environnement est en perpétuel changement même durant les actions (y compris la planification) des acteurs.
- *Spatial* dans le sens que les données géographiques de l'espace considéré sont importantes et parfois indispensables pour que les acteurs puissent agir. Cet aspect a été souvent négligé ou peu abordé par la plupart des travaux sur la planification dans un environnement dynamique. Il est cependant clair que dans des situations de désastres naturels (Chapitre 2, Section 2.5 sur SIG et désastres) les contraintes

géographiques de l'espace sont plus que pertinentes et ne peuvent donc être négligées ou sous-évaluées.

- *Distribué* dans l'espace : les ressources (d'assistance et à protéger) sont réparties sur un grand espace géographique, ce qui rend encore la tâche des acteurs plus compliquée. Les acteurs sont par conséquent dispersés sur l'espace, ce qui demande donc un effort de communication et de coordination assez considérable. Dans cette thèse nous ne mettons pas l'emphase sur cet aspect (la distribution).

4.1.3. But des acteurs

Nous avons mentionné ci-dessus que le but des acteurs est de faire face à la situation dangereuse. Cette action peut être menée selon trois grandes stratégies :

- *Stratégie d'attaque* : les acteurs emploient leurs ressources d'assistance pour lutter directement contre la situation.
- *Stratégie de défense* : Il est bien sûr préférable de lutter directement pour épargner le plus possible de ressources. Toutefois, cette stratégie n'est pas toujours possible : les acteurs peuvent être contraints à défendre les ressources de l'espace sans pouvoir maîtriser la situation dangereuse. Il est à noter aussi, que dans certains cas, une stratégie de défense peut neutraliser la situation dangereuse. L'exemple le plus parlant est l'attaque indirecte des feux de forêt (Chapitre 3 sous-Section 3.3.3.2).
- *Stratégie hybride* : C'est une combinaison des deux premières. Les acteurs attaquent directement quand ils peuvent et défendent quand ce n'est plus possible.

Dans notre travail, nous nous intéressons à la stratégie défensive. Cette stratégie est malheureusement celle qui est la plus utilisée car souvent la situation dangereuse est très difficile à contenir. En effet, dans la plupart des désastres naturels, la situation est tellement chaotique et critique que les acteurs n'ont pas d'autres choix que de défendre au maximum les ressources en péril. Mais que veut-on dire par défendre des ressources ? En fait, il faut tout d'abord distinguer entre deux grands types de ressources :

- Les *ressources déplaçables* : ce sont des ressources que les acteurs peuvent très bien défendre en les déplaçant. Il s'agit souvent d'êtres humains, de matériels, animaux... L'action qui vise à déplacer ce genre de ressources dans des situations dangereuses, est souvent appelée « évacuation ». Nous ne mettons pas trop l'emphase sur cet aspect dans le présent travail par faute de temps.
- Les *ressources non déplaçables* : Contrairement aux précédentes, ces ressources ne peuvent pas bouger (p.ex. des bâtiments, des plantations, etc.). Les acteurs sont alors contraints à les défendre.
- Les *ressources difficilement déplaçables* : certaines ressources, bien qu'elles soient à l'origine de nature déplaçable, pourraient devenir non déplaçables ou du moins difficilement. En effet, lorsque la situation est chaotique, lorsque les ressources d'assistance font défaut, lorsque le nombre des ressources à protéger est très élevé et donc difficile à évacuer, les acteurs peuvent également être contraints à défendre les ressources au lieu de les déplacer.

Pour défendre ces ressources non déplaçables¹⁰ (ou difficilement déplaçables), plusieurs techniques peuvent être adoptées selon le type de la situation, les contraintes de l'environnement et la disponibilité des ressources d'assistance. Cependant, en général, il s'agit souvent de construire une barrière entre la situation dangereuse et la ressource.

Nous visons ainsi, dans ce travail, la création d'un système de planification basé sur la stratégie défensive¹¹ des ressources non déplaçables sous la menace d'une situation.

4.2. Principes de la solution théorique

4.2.1. Introduction

Dans le Chapitre 1 et surtout dans le Chapitre 2 (Section 2.1), nous avons mentionné certains problèmes qui empêcheraient des acteurs humains de faire face (p.ex. planifier

¹⁰ Dans tout le reste du travail, le terme « non déplaçable » fera référence aux ressources non déplaçables et difficilement déplaçables, pour des raisons de simplicité.

¹¹ La stratégie pourrait être parfois hybride, mais défensive en majeure partie.

des actions de lutte) à une situation dangereuse. En effet, ces acteurs ont une vision locale restreinte de l'espace et ne savent donc pas ce qui se passe ailleurs, ne possèdent pas toute l'information pertinente sur le terrain, estiment mal l'aspect géographique (limites de la cognition spatiale), ne peuvent pas communiquer de façon efficace et facile avec les autres acteurs, etc.

Même si les acteurs humains arrivent à élaborer des plans d'action qui tiennent compte du maximum de données disponibles, il leur est très difficile de valider l'efficacité de ces plans avant leur exécution. En fait, les humains ont des capacités d'anticipation remarquables qui leur permettent d'anticiper les problèmes et d'imaginer les plans futurs dans leur espace mental. Cependant, ces capacités sont encore une fois limitées lorsque les données sont nombreuses et le problème a beaucoup de paramètres et de contraintes.

En fait, pour récapituler, dans un monde réel, l'information est souvent manquante ou erronée, l'espace perçu est très limité vu l'étendue de l'espace global, la communication est souvent non fiable (communication distante), la planification est souvent difficile et sa validation l'est encore plus. Ces handicaps sont d'autant plus marqués lorsque les acteurs ont besoin, en plus, de connaître les caractéristiques précises géographiques et topologiques de l'espace dans lequel ils agissent.

Nous pensons alors qu'il est nécessaire de porter le problème dans un autre environnement qui permettrait de combler les manques de l'environnement réel.

Dans les sous-sections suivantes, nous présentons les principes fondamentaux de la solution que nous proposons. Les détails seront abordés dans la présentation de la solution pratique (Chapitre 5 et 6).

4.2.2. La connaissance spatiale augmentée

Comme nous l'avons avancé dans notre hypothèse de recherche, nous pensons qu'un environnement de simulation est capable de résoudre le problème. Toutefois, il ne s'agit pas de n'importe quel environnement de simulation. La simulation est souvent utilisée comme un outil d'aide à la décision. Peu de travaux de recherche utilisent la simulation pour planifier. La plupart des travaux intégrant la planification et la simulation, ne font

pas le lien entre l'environnement de simulation et le monde réel, et manquent donc de réalisme. En conséquence, la plupart des systèmes développés ne sont pas appliqués dans des situations réelles (comme lors des désastres naturels), sont restés dans les laboratoires de recherches ou dans les meilleurs des cas sont utilisés comme outils d'entraînement.

Nous estimons que le lien entre les deux environnements (réel et de simulation) est crucial. Un environnement de simulation sert en fait à apporter aux acteurs du monde réel la connaissance qui leur manque sur le terrain en termes de données spatiales, de données sur l'évolution de la situation et de l'état des autres acteurs et ressources. De plus, le monde de simulation doit faire le lien avec le monde réel de façon efficace, fidèle (précise) et en temps réel. Nous parlons ainsi de *connaissance augmentée*. En d'autres termes, nous disposons d'un environnement (réel) où la connaissance est manquante, et nous lui avons associé un autre environnement parallèle (de simulation) où la connaissance est plus disponible. Dans le cas particulier de la connaissance spatiale, les acteurs manquent (ou perçoivent mal) les données spatiales sur le terrain dans le monde réel. L'environnement de simulation doit alors combler cette faiblesse en mettant à disposition la connaissance spatiale requise. Nous parlons ainsi de *connaissance spatiale augmentée*. Le schéma de la Figure 4.2 illustre la notion de connaissance augmentée. L'acteur dans le monde réel dispose d'une connaissance spatiale réduite qu'il perçoit (soit directement par la vue, soit indirectement par des cartes). Il peut également avoir, dans les meilleurs des cas, une vision locale et imprécise de la situation dangereuse. Il est aussi capable de communiquer avec les autres acteurs (non représentés sur la Figure 4.2) sur le terrain. Cette communication est confrontée à plusieurs limitations : manque de fiabilité et de performance, limitation du flux, etc. Dans le monde de simulation, l'acteur a un représentant qui a accès à la connaissance spatiale augmentée, qui a une vue globale et précise de la situation et qui peut interagir avec les autres représentants présents de façon plus efficace, rapide et sûre.

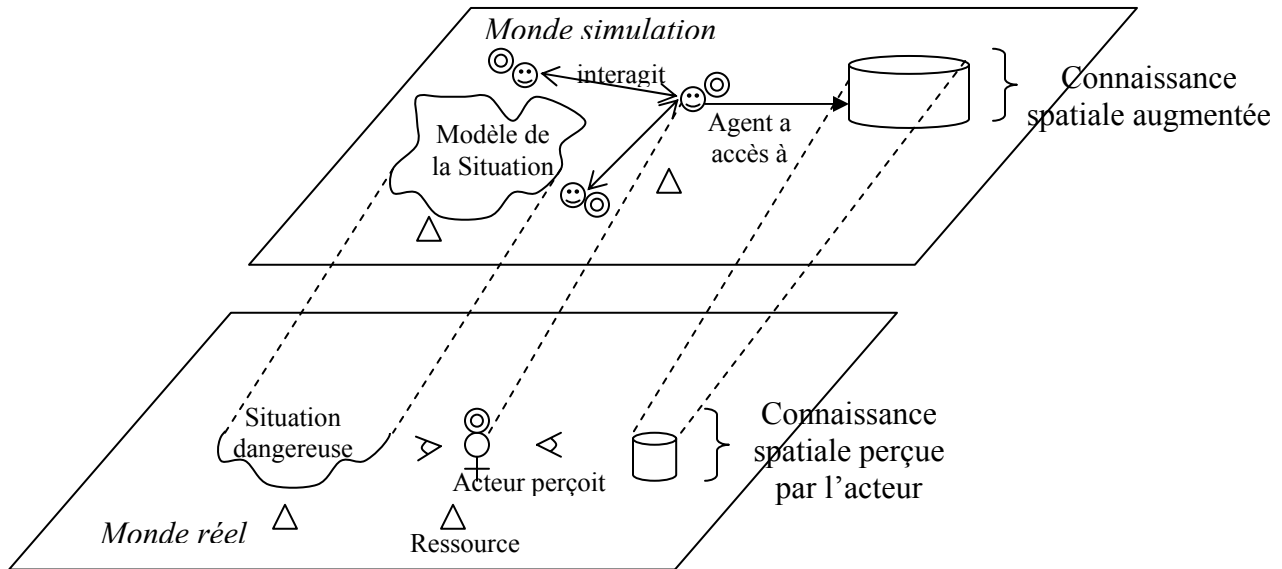


Figure 4.2. Principe de la connaissance augmentée dans un monde de simulation

Pour faire maintenant le lien entre les deux mondes, trois questions se posent :

1. Qui va représenter les acteurs, les ressources et la situation dans le monde de simulation ?
2. Comment pouvons nous obtenir (concrètement) une connaissance spatiale augmentée précise et fidèle à la réalité ?
3. Comment pouvons nous assurer la communication entre les deux mondes ?

Les trois sections qui suivent répondent respectivement à ces interrogations.

4.2.3. Les représentants du monde réel dans la simulation

4.2.3.1. Les agents : représentants des ressources et des acteurs

Les acteurs (et leurs ressources d'assistance) du monde réel ont certainement besoin de représentants dans le monde de simulation afin d'accomplir certaines tâches à leur place. Ces représentants doivent, entre autres, inspecter le terrain de façon plus étendue et précise, évaluer la situation et le danger en ayant à disposition l'information nécessaire, interagir avec les autres représentants afin de coopérer (voir Figure 4.2), etc. Nous

pensons alors que les agents logiciels seraient les meilleurs représentants qui peuvent accomplir ces tâches. A chaque acteur (réel) sera donc associé un agent logiciel représentant.

Quant aux ressources à protéger, certaines pourraient être représentées aussi par des agents : agents ressources (Figure 4.2). Ces agents devront donc interagir avec les agents représentants des acteurs afin de les aider à les protéger (nous donnons plus de détails plus loin).

4.2.3.2. Les modèles physiques : une simulation fiable des situations dangereuses

Afin de représenter la situation dangereuse dans le monde de simulation, nous avons besoin d'un modèle physique qui simule le phénomène en question (Figure 4.2). Dans certains systèmes comme Phoenix [Cohen et al., 89], la situation est représentée par un agent. Vu la complexité de modélisation d'une situation réelle, cet agent ne peut donner qu'une idée générale - qui n'est pas basée sur des calculs précis- sur la situation. Nous pensons que des modèles réalistes (généralement implémentés dans des systèmes indépendants) sont nécessaires pour représenter le monde réel de manière plus fidèle et atteindre ainsi un meilleur réalisme dans la résolution du problème (problème décrit dans la Section 4.1 de ce chapitre).

Heureusement, avec le progrès scientifique, il est maintenant possible d'avoir des modèles offrant une reproduction assez fidèle de la situation en cours ainsi que des prédictions relativement précises. Par exemple, il est désormais possible de trouver des modèles de propagation des feux ou des gaz comme dans [Prometheus 05] ou [Farsite 05]. Il est même possible de modéliser des inondations [Herath 01]. Cependant, certaines situations restent très difficiles à modéliser telles que les séismes.

4.2.4. Le SIG : source de la connaissance spatiale augmentée

La connaissance spatiale augmentée doit satisfaire aux deux contraintes essentielles suivantes:

- La complétude : Les données spatiales dans le monde de simulation doivent être les plus complètes possibles. Le problème de l'acteur dans le monde réel réside dans le fait qu'il n'a qu'une vision locale de la situation, du terrain et de la position des autres acteurs. Si nous arrivons à offrir une vision plus globale et donc plus complète de ces données, nous aurons alors doté l'environnement de simulation de connaissances spatiales augmentées.
- La fiabilité : La complétude cependant ne suffit pas. Il faut aussi que la connaissance spatiale soit fiable. En effet, devant des situations dangereuses, les détails ne peuvent pas être négligés car la survie de certaines ressources (surtout humaines) peuvent en dépendre.

Pour représenter de façon assez précise et complète l'espace géographique étendu dans un monde de simulation, le SIG est le meilleur candidat. En effet, un SIG contient une multitude de données spatiales, fiables et complètes, décrivant au mieux l'espace. Cependant l'un des gros problèmes des SIG est la mise à jour des données, Nous supposons dans notre travail que les SIG utilisés ont été mis à jour¹².

L'environnement de simulation doit par conséquent être basé sur des données SIG afin de garantir la meilleure reproduction de l'espace réel. Il est à noter que ces données doivent être accessibles de façon simple et efficace aux différents agents représentants. Nous décrirons plus tard la solution technique adoptée pour y arriver.

4.2.5. De nouvelles technologies au support de la communication entre le monde réel et de simulation

Nous avons mentionné plus haut qu'il ne suffit pas de construire un monde de simulation précis et complet mais qu'il faut aussi permettre une communication efficace et en temps réel entre le monde réel et celui de simulation. En d'autres termes, d'une part, les agents représentants, ayant accès à la connaissance augmentée doivent en faire profiter les

¹² Cette hypothèse est crédible car les situations dangereuses n'arrivent pas tous les jours. Nous supposons alors que des experts mettent à jour les données du SIG une fois par an. Dans le cas des feux de forêt cette hypothèse est réelle car la saison des feux ne dure que quelques mois (l'été) et pendant le reste de l'année les données sont donc mises à jour dans les SIG afin de se préparer pour la prochaine saison des feux.

acteurs réels. D'autres part, tout ce qui se passe dans le monde réel (au niveau des ressources, des acteurs ou de l'évolution de la situation) doit être rapporté au monde de simulation afin de garantir la cohérence entre les deux mondes.

Le progrès technologique nous offre plusieurs solutions de communication :

- La télédétection : Elle est surtout utilisée pour surveiller l'évolution de la situation dangereuse (surveillance à un niveau macro).
- Les senseurs web [Delin et al., 99]: Ils permettent de surveiller la situation de façon plus détaillée (surveillance à un niveau micro).
- Les réseaux mobiles et sans fil : Ils permettent aux acteurs (ou plutôt aux agents acteurs) de communiquer avec leurs représentants dans le monde de simulation.
- Le positionnement géoréférencé (p.ex. GPS) : Il permet de garder la cohérence entre les localisations des éléments du monde réel avec leurs correspondants dans le monde de simulation. Par exemple, un acteur réel communique sa position géoréférencée à son représentant afin que ce dernier ajuste sa position dans l'environnement de simulation.

Dans la section suivante nous allons voir comment utiliser les principes énumérés ci-dessus pour proposer un framework de lutte contre les situations dangereuses.

4.3. ENCASMA : un Framework à quatre couches pour lutter contre les situations dangereuses

Afin de résoudre les problèmes tels que spécifiés dans la Section 4.1.1, nous nous proposons de développer un Framework basé sur les notions décrites dans la Section 4.2. Nous appelons le Framework : Environnement à Connaissance Augmentée basé sur un Système MultiAgent, ENCASMA. ENCASMA s'applique bien entendu à des problèmes et à des environnements tels que spécifiés dans la sous-Section 4.1.1.

ENCASMA fait la correspondance entre le monde réel et le monde de simulation. Son architecture est décrite par la Figure 4.3.

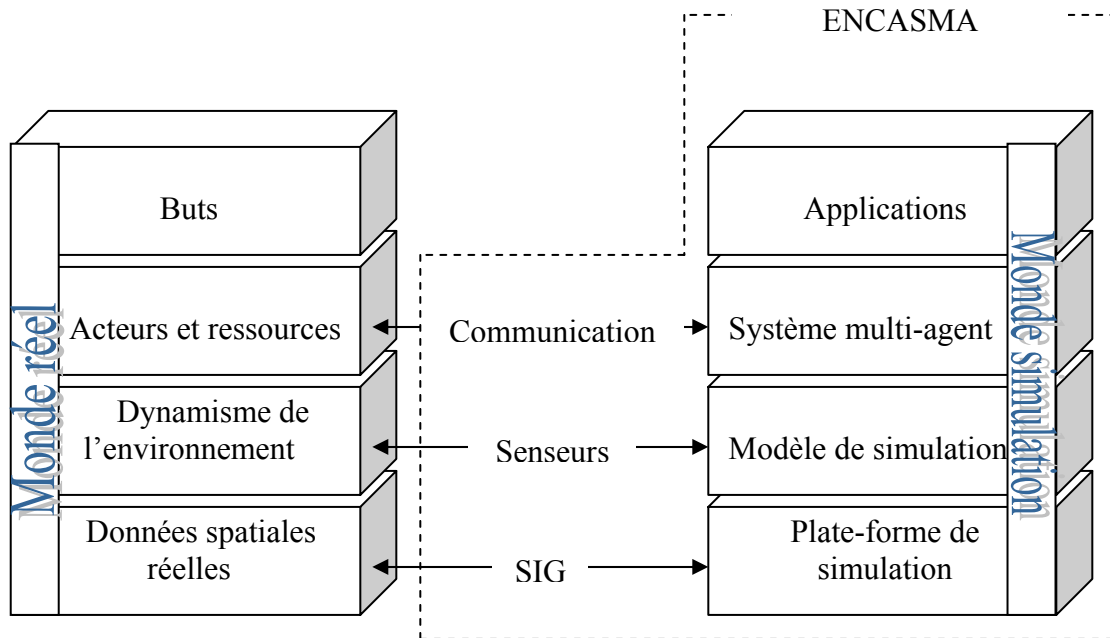


Figure 4.3. Architecture générale d'ENCASMA

ENCASMA –représentant le monde de simulation- est constituée de quatre couches en étroite lien avec le monde réel. Nous allons dans ce qui suit détailler une par une, les différentes couches ainsi que les différents liens qui les relient au monde réel.

4.3.1. Plate-forme de simulation : MAGS

Nous avons dit auparavant (sous-Section 4.1.2.4) qu'un SIG est nécessaire pour traduire de façon fidèle et précise les données spatiales réelles. Or, un SIG ne peut constituer un environnement de navigation pour les agents logiciels. Les utilisateurs humains ont également besoin d'un outil beaucoup plus visuel pour superviser le déroulement des actions. Il est donc nécessaire de traduire ces données SIG en une plate-forme de simulation qui soit visuelle pour l'utilisateur humain et navigable pour les agents logiciels.

Nous suggérons l'utilisation de MAGS [Moulin et al., 03], une plate-forme géospatiale générique qui est capable de simuler de façon fiable les données spatiales réelles et qui offre un environnement de navigation et de perception performant pour les agents logiciels.

Le projet MAGS sera décrit plus loin (Section 4.4).

4.3.2. Le modèle de simulation et ses liens avec la situation réelle

Un environnement réel est souvent dynamique et soumis à des phénomènes physiques complexes. Afin de pouvoir simuler au mieux la réalité, il est nécessaire de pouvoir reproduire ces phénomènes physiques dans le monde de simulation. Le modèle de simulation (généralement un modèle physique) simulant la situation dangereuse doit être fiable et réaliste. Pour ce faire, deux conditions sont à satisfaire absolument :

- Le modèle même doit être basé sur des études précises.
- Les données d'entrée du modèle doivent être continuellement mises à jour selon l'évolution de la situation réelle.

Pour le premier point, nous nous fions aux efforts des physiciens et météorologues pour avoir un modèle le plus réaliste possible.

Quant au deuxième point, il est clair que les prévisions futures fournies par le modèle ne peuvent être correctes à 100% en raison de la complexité du problème d'une part et les changements imprévus qui peuvent survenir d'autre part. Des senseurs doivent donc assurer la correction et la mise à jour des données fournies par le modèle. Nous distinguons plusieurs types de senseurs :

- Satellites : la télédétection (voir plus haut)
- Senseurs humains : Un acteur présent sur le terrain peut également rapporter ce qu'il perçoit sur l'évolution de la situation. Bien que ces données puissent ne pas être précises (la cognition spatiale humaine est parfois biaisée), elles sont parfois très utiles pour avoir des constats sûrs concernant l'évolution générale de la situation. Les rapports des acteurs peuvent être acheminés de plusieurs manières à l'environnement de simulation. Nous décrivons dans ce qui suit trois méthodes possibles :

- Méthode manuelle : Un acteur peut rapporter ses constats verbaux sur la situation au moyen d'un téléphone mobile par exemple. Ces données sont ensuite utilisées par l'utilisateur du système de simulation pour mettre à jour (si nécessaire) la situation et bâtir les prochaines prédictions en considérant ces nouveaux constats.
- Méthode automatique : Grâce au progrès de l'informatique omniprésente, il est possible que des agents logiciels intégrés dans des dispositifs électroniques (à la disposition des acteurs sur le terrain), puissent « observer » l'environnement et communiquer avec leurs correspondants du monde de simulation pour les informer de l'évolution de la situation sur le terrain. Les agents représentants tâchent par la suite de refléter ces données sur le modèle de simulation de la situation. Cette technologie est déjà utilisée dans la navigation autonomes des machines forestières [Hellstrom 02] afin d'éviter les obstacles.
- Méthode semi-automatique : C'est une combinaison des deux premières méthodes. Lorsque des agents logiciels sur le terrain envoient des notifications aux agents du monde de simulation, des opérateurs humains prennent le relais et s'occupent d'appliquer ces données sur le modèle de simulation. Cette approche hybride peut être adoptée si la mise à jour du modèle se révèle difficile techniquement par les agents représentant ou encore si l'opérateur humain préfère garder le contrôle afin de juger de la pertinence des notifications reçues depuis le terrain pour déterminer s'il faut les inclure ou non dans la simulation.
- Senseurs Web : Cette nouvelle technologie décrite précédemment (sous-Section 4.2.5) permettrait également de notifier l'environnement de simulation de l'évolution de la situation. Cette méthode, bien que moins sûre qu'une constatation humaine, présente plusieurs avantages. En effet, elle permet une meilleure couverture de l'espace, réduit les risques auxquels pourraient être confrontés les humains en allant chercher de telles informations et offre une mise à jour continue de la situation.

Tous ces moyens technologiques combinés permettent une meilleure couverture de la situation dangereuse, ce qui est nécessaire pour mieux cerner le problème à résoudre. Le modèle de simulation peut alors fournir une simulation fiable et surtout évolutive de la situation. Cette simulation constitue donc une couche qui vient se poser sur la première couche (la plate-forme de simulation). Nous formons ainsi un environnement de simulation qui reflète ce qui se passe dans la réalité. Il reste maintenant à rajouter les représentants des acteurs.

4.3.3. Le système multiagent et son lien avec le monde réel

4.3.3.1. Introduction

La troisième couche constituant ENCASMA est la couche multiagent. Elle représente les acteurs du monde réel ainsi que les ressources importantes que les acteurs cherchent à protéger. Afin de permettre un lien efficace entre ces acteurs et ressources réels, et leurs correspondants sur l'environnement de simulation, nous adoptons une stratégie complètement agentifiée : tout est agent. En d'autres termes, nous aurons des agents logiciels sur le terrain et d'autres dans le monde de simulation.

4.3.3.2. Les agents réels et les agents de simulation

Dans le monde réel, tout acteur actif a un agent représentant cette fois dans le monde réel. Concrètement, un acteur aura à sa disposition un dispositif électronique (p.ex. une station mobile) où un agent logiciel est chargé d'interagir avec l'acteur et faire par la suite le lien entre lui et le monde de simulation. Cet agent qu'on appelle *Agent Acteur Réel AAR* interagit avec l'acteur humain via une interface et communique avec l'environnement de simulation via des messages distants ou bien par migration.

Il serait aussi possible de représenter des ressources (acteurs passifs) du monde réel par des agents. Un agent intégré dans un capteur est placé au niveau de chaque ressource importante afin de rapporter les événements qui peuvent survenir à l'environnement de simulation. Cette idée est difficile à mettre en œuvre surtout si les ressources sont très dispersées et sur un espace très étendu. Nous nous contentons donc des *AAR* dans ENCASMA.

Dans le monde de simulation maintenant, nous avons également des agents logiciels. A chaque *AAR* nous associons un *Agent Acteur de Simulation AAS* pour le représenter. Nous associons également aux ressources à protéger au niveau de l'environnement de simulation des *Agents Ressources de Simulation ARS*. Un agent *ARS* n'a donc pas de correspondant sur le monde réel, mais sa présence est importante dans le monde de simulation afin d'interagir avec les *AAS* et former un vrai système multiagent. Par exemple, si un *ARS* se rend compte dans le monde de simulation que la situation dangereuse est à sa proximité mais qu'aucune action de la part des acteurs (représentés par des *AAS*) n'a été entreprise, il peut lancer un appel de détresse aux *AAS* qui l'entourent pour chercher leur support.

4.3.3.3. Capacités des agents de simulation dans MAGS

Afin que les agents de simulation puissent profiter d'un environnement à connaissance augmenter, il est nécessaire de les doter de capacités cognitives développées. Les agents de la plate-forme MAGS, disposent justement de telles capacités (Section 4.4).

4.3.3.4. Lien entre *Agent Réel* et *Agent de Simulation*

Avant que les agents de simulation puissent agir (naviguer, percevoir, etc.) dans l'environnement de simulation, il est très important d'établir un lien de cohérence entre eux et leurs correspondants dans le monde réel. Pour ce faire, la technologie moderne offre plusieurs moyens de communication entre les agents. Nous nous contentons dans ENCASMA de parler du principe.

L'*Agent Réel* doit informer son correspondant (*Agent de Simulation*) de sa localisation réelle. L'agent du monde de simulation se place ainsi à la position correspondante au niveau de la plate-forme de simulation. Les positions sont bidimensionnelles mais peuvent être tridimensionnelles si l'altitude est pertinente. Dans ce cas, l'environnement de simulation doit être en 3D. Si l'*Agent Réel* se déplace, il doit fournir périodiquement à son correspondant sa position géospatiale afin qu'il puisse reporter sa position dans le monde de simulation. Les modes d'interactions entre agents réels et agents de simulation ainsi que les technologies utilisées, seront présentés plus loin.

L'*Agent de Simulation* peut à son tour communiquer avec son correspondant dans le monde réel afin de lui faire parvenir des informations qu'il a pu recueillir à partir de son environnement qui, rappelons le, a une connaissance augmentée.

4.3.4. La couche application

Les trois couches précédentes sont complètement génériques et peuvent s'appliquer à plusieurs problèmes impliquant différents types de situations dangereuses. Cependant, ces trois couches ne constituent qu'un support pour des applications ayant pour objectif de lutter contre la situation dangereuse, assister les acteurs sur le terrain ou encore aider les utilisateurs à prendre les bonnes décisions stratégiques pour se défendre contre les problèmes engendrés par la situation dangereuse. Ces applications expriment donc les buts des utilisateurs. Bien qu'elles puissent être décrites de façon générique indépendamment du type de la situation, nous nous restreignons - pour des raisons de simplification et de clarté, à une situation bien particulière : un feu de forêt.

Dans la Section 4.5, nous étudions plus en détail la couche application dans le cas d'un feu de forêt. Cette couche englobe différentes applications qui peuvent être d'une grande utilité lors de la lutte contre les feux. Ces applications, bien qu'elles soient destinées aux feux de forêt, peuvent être adaptées pour n'importe quel problème analogue.

Une première application assez simple permet d'assister les acteurs sur le terrain en leur communiquant l'information requise à partir de l'environnement de simulation qui bénéficie d'une connaissance augmentée.

La deuxième application est un peu plus complexe et vise à implémenter l'approche DCP afin d'encercler le feu. Il s'agit ici d'un problème de planification complexe que nous comptons résoudre grâce à ENCASMA (Chapitre 6).

4.4. La plate-forme MAGS

4.4.1. Introduction

MAGS est plate-forme générique pour la création de géosimulations multiagent impliquant un grand nombre (des milliers) d'agents logiciels interagissant dans des environnements géographiques virtuels.

Les agents dans MAGS sont capables d'accomplir des activités spatiales et cognitives dans le système de géosimulation (ce que nous avons appelé dans ENCASMA la plate-forme de simulation). Pour ce faire, les agents sont capables de percevoir l'environnement spatial avec ce qu'il contient comme objets et agents logiciels. L'environnement spatial doit être généré à partir des données stockées dans des systèmes d'information géographique SIG.

4.4.2. Les capacités des agents MAGS

Les agents dans MAGS sont dotés de capacités spatiales et cognitives que nous discutons ci-dessous [Moulin et al., 03].

4.4.2.1. La perception

Les agents de *MAGS* sont capables de percevoir plusieurs objets de natures différentes :

- La perception des caractéristiques du terrain de l'environnement de simulation (p.ex. élévations et pentes).
- La perception des chemins, des routes et des régions.
- La perception du paysage entourant les agents (les bâtiments et les objets statiques).
- La perception d'autres agents logiciels. Les agents perçus par un agent doivent être situés dans son champ visuel.

- La perception des aires dynamiques qui ont des propriétés spécifiques (les aires de fumée et les zones d'odeur).
- La perception d'événements spéciaux (p.ex. explosions et détonations).
- La perception de messages communiqués par d'autres agents.

Afin de simuler la perception de milliers d'agents en temps réel, des algorithmes efficaces et optimisés ont été implémentés.

4.4.2.2. La navigation

Les agents se déplacent dans l'environnement de simulation de façon autonome [Moulin et al., 03]. La navigation peut se faire selon deux modes :

- Le mode *following-a-path* où les agents suivent des chemins matérialisés (p.ex. route, trottoirs) tout en évitant les autres agents.
- Le mode *obstacle-avoidance* où les agents naviguent à travers des espaces ouverts en évitant collisions avec les objets de l'environnement et les autres agents.

Dans les deux modes de navigation les agents utilisent leur mécanisme de perception pour identifier les chemins (mode *following-a-path*) ainsi que les autres objets et agents (mode *obstacle-avoidance*).

4.4.2.3. La connaissance

Afin d'assurer leur propre autonomie, les agents doivent être capables d'interagir avec l'environnement de simulation et prendre des décisions qui respectent leurs états internes et leurs préférences [Moulin et al., 03]. En fait, un agent dans MAGS est caractérisé par un certain nombre de variables dont les valeurs décrivent l'état de l'agent à chaque instant durant la simulation. On distingue deux types d'états : des *états stables ou statiques* et des *états dynamiques*.

- Un état stable ou statique : ne change pas durant la simulation et est représenté par une variable et sa valeur courante (p.ex. le nom de l'agent) [Moulin et al., 03].

- Un état dynamique : peut changer durant la simulation (p.ex. le degré de fatigue de l'agent). Il est représenté par une variable associée à une fonction qui est utilisée pour calculer comment la variable change sa valeur durant la simulation [Moulin et al., 03]. Cette variable est caractérisée par plusieurs paramètres : une valeur initiale, une valeur maximale, un taux d'augmentation, un taux de diminution, un seuil minimal et un seuil maximal. Ces paramètres sont utilisés par la fonction associée à la variable. En utilisant ces paramètres le système peut simuler l'évolution des états dynamiques des agents et déclencher le comportement approprié [Moulin et al., 03].

4.4.2.4. Le comportement

Un agent dans MAGS a des objectifs à satisfaire. Ces objectifs sont organisés de façon hiérarchique et sont associés à des actions que l'agent peut accomplir. Chaque agent possède un ensemble d'objectifs reliés à ses besoins. Un objectif est relié à des règles qui contiennent des contraintes sur les actions et sur l'activation de l'objectif. Les contraintes sont dépendantes du temps, des états internes de l'agent ainsi que des états de l'environnement [Moulin et al., 03].

Le comportement courant de l'agent est défini selon la priorité de ses objectifs. En fait, chaque besoin (de l'agent) est associé à une priorité qui varie selon le profil de l'agent. La priorité des objectifs peut varier en fonction des opportunités que l'agent peut percevoir et en fonction des contraintes temporelles sur ces objectifs. Le comportement de chaque agent dépend également de son profil. Ce profil renferme un ensemble de rôles qu'un agent peut jouer durant la simulation et peut être utilisé pour personnaliser les besoins de l'agent. Un rôle est représenté par un arbre d'objectifs. La structure de l'arbre permet de définir le comportement à plusieurs niveaux d'abstraction. La racine de l'arbre est constituée d'objectifs composés et les feuilles sont constituées d'objectifs élémentaires. Les actions sont associées à ces objectifs élémentaires.

4.4.3. Les avantages de MAGS

La plate-forme *MAGS* diffère des autres plates-formes et langages de simulation. Elle présente plusieurs avantages [Moulin et al., 03] dont nous citons les principaux :

- *MAGS* permet de créer des agents avec une capacité de perception visuelle (objets physiques et agents) et non visuelle (odeur et son). De plus, les agents peuvent utiliser leur mécanisme de perception de message pour communiquer avec d'autres agents.
- *MAGS* permet de créer des agents avec une capacité de navigation importante dans l'environnement. Les agents peuvent suivre des chemins et des routes, peuvent éviter des collisions, etc.
- *MAGS* permet la création d'agents logiciels complets avec des états internes statiques et dynamiques.
- Malgré la structure riche des agents en terme de connaissances et de comportements, *MAGS* permet la création de milliers d'agents qui peuvent interagir dans un environnement virtuel en *temps réel*. Pour ce faire, les processus de perception et de navigation ont été optimisés.
- *MAGS* offre une capacité d'affichage en mode 2D et 3D. En utilisant cette plate-forme nous pouvons afficher la simulation en mode *bird's eye* (2D) et en mode 3D qui offre à l'utilisateur de l'outil de simulation la possibilité d'explorer l'espace de simulation.

4.5. Les applications d'ENCASMA sur les feux de forêts

Avant de parler des applications, nous allons instancier le Framework selon les données du problème des feux de forêt.

Le Tableau 4.1 illustre les correspondances entre ENCASMA et le domaine d'application.

Tableau 4.1. Correspondances entre ENCASMA et FEU DE FORET

ENCASMA	FEU DE FORET
Espace	Forêt ou parc
Situation dangereuse	Feu qui se propage
Acteur	Pompier
Ressources d'assistance	Avions citerne, bulldozer, etc.
Ressources à protéger	Habitations, habitants, pompes à essence, la forêt même, etc.
But des acteurs	Encercler le feu pour l'éteindre et protéger les ressources
Ressources déplaçables	Les habitants, les véhicules, etc.
Ressources non déplaçables	Les habitations, les pompes à essences, etc.
AAR	Agent Pompier Réel APR
AAS	Agent Pompier de Simulation
ARS	Agent x de simulation ($x \in \{ \text{Habitations, habitants, pompes à essence, etc.} \}$)
Dynamisme de l'environnement	Feux
Modèle de simulation (des feux)	Modèle de propagation des feux (selon la librairie PROMETHEUS [Prometheus 05])
Plate-forme de simulation	MAGS
Senseurs	Satellite de détection des feux, avions de reconnaissance, <i>autonomous fire detectors</i> , etc.

Plusieurs applications liées au domaine des feux de forêt sont possibles et illustrent bien l'apport de l'architecture ENCASMA. Nous mettons par contre l'emphase sur une seule application dans cette thèse : « L'attaque indirecte des feux ». Cette application est complexe et fait appel à certains principes théoriques qui n'ont pas été encore abordés. L'idée est d'utiliser ENCASMA comme support pour la planification en continu. Nous consacrons les deux prochains chapitres à cette application et aux notions théoriques qui

en découlent. Une autre application intéressante est décrite à l'annexe C. Cette application « Assistance sur le terrain », n'a pas été implémentée par manque de temps et de moyens, mais donne une idée sur ce que peut offrir ENCASMA comme support pour assister les pompiers lors des opérations de lutte.

4.6. Travaux connexes

Plusieurs travaux ont proposé l'utilisation de la simulation basée sur les agents logiciels pour atteindre certains objectifs tels que la planification. Ces recherches ont été soulignées dans le Chapitre 2 Section 2.4.2 : [Lee et Fishwick 97], [Horn et Baxter 00], [Atkin et al., 98] et [Cohen et al., 89]. Cependant, la plupart de ces travaux ne font pas le lien avec le monde réel et ne peuvent donc s'appliquer sur des applications du monde réel et en temps réel. Récemment, quelques chercheurs ont commencé à s'intéresser à cet aspect. Dans [Weynes et al., 05], les auteurs utilisent des agents situés pour contrôler un système de transport automatisé qui utilise des véhicules guidés automatiquement (*Automatic Guided Vehicles* AGV) pour transporter des charges dans des entrepôts. Étant donné que l'environnement physique des AGV est très restreint, il offre peu de possibilités aux agents pour utiliser l'environnement. L'idée des auteurs était alors d'introduire un environnement virtuel pour que les agents y cohabitent. Cet environnement virtuel est pour ces agents un moyen pour échanger de l'information et coordonner leurs comportements. Il est distribué sur les différents AGV et la communication entre les AGV est assurée par un réseau sans fils.

Cet environnement virtuel offre des capacités de perception, d'action et de communication à l'agent AGV. Il assure ainsi à l'agent les fonctions primitives de haut niveau afin qu'il puisse agir, percevoir et interagir dans son monde réel. A titre d'exemple, la perception est assurée par un *Perception Manager*. Quand l'agent AGV a besoin de percevoir, par exemple la position courante des AGVs voisins, le *Perception Manager* lui cherche ces informations à partir de l'environnement virtuel local.

Les auteurs illustrent l'utilité de leur approche sur le problème de l'évitement des collisions. Les agents AGV se fient à leurs capacités de perception et de communication

afin d'éviter les collisions dans le monde virtuel, ce qui est par la suite exécuté dans le monde réel.

L'idée d'utiliser un environnement virtuel pour planifier des actions qui doivent s'exécuter par la suite dans un monde réel ressemble beaucoup à ce que nous avons proposé dans ENCASMA. Cependant, dans ENCASMA l'environnement virtuel n'est pas distribué, il est commun à tous les agents et joue ainsi le rôle d'une infrastructure de rencontre [Moulin et al., 01]. De plus, ENCASMA est destinée à des espaces géographiques réels à grande échelle contrairement à ce que Weynes et ses collègues proposent. Mais la différence la plus marquée réside dans la portée de l'architecture même. En effet, à notre connaissance, aucune autre architecture ne donne un appariement aussi global entre l'environnement réel et l'environnement de simulation pour des applications à forte composante spatiale. ENCASMA s'intéresse à toutes les entités pertinentes du monde réel (données sur l'espace et sur le dynamisme, sur les acteurs et leurs objectifs, sur les ressources, etc.), à leurs correspondants dans le monde virtuel et aux différents moyens de liaison entre les deux mondes.

L'une des composantes les plus importantes de l'architecture ENCASMA est le système multiagent correspondant aux acteurs et aux ressources du monde réel. La majorité des systèmes multiagents existants ont comme responsabilité de faciliter la communication entre les agents à travers un service de transport de messages ou une infrastructure *broker*. Parmi ces systèmes nous pouvons citer Jade [Bellifemine et al., 00], Jack [Howden et al., 01], JAFMAS [Chauhan 97], RETSINA [Sycara et al., 03], ZEUS [Nwana et al., 98], IMPACT [Rogers et al., 99], SIM_AGENT [Sloman et Poli 96], et Cougaar [Coogaar 05]. Pour modéliser l'environnement et le SMA dans ENCASMA, nous avons adopté la plate-forme MAGS car cette dernière ne permet pas juste aux agents de communiquer mais leur offre aussi une panoplie de capacités (perception, navigation, mémoire, etc.) nécessaires pour appréhender l'espace.

4.7. Contributions du Chapitre

Rappelons que le but initial de cette thèse est de trouver un moyen d'implémenter l'approche DCP pour que les agents logiciels puissent planifier de façon plus efficace

(cognitivement) en temps réel (ou quasi-réel) tout en tenant compte des contraintes géospatiales de l'environnement. Dans ce chapitre, nous avons proposé un framework (ENCASMA) pour la conception et l'implémentation d'applications (typiquement des applications de lutte contre les désastres naturels) dans un espace géographique réel à grande échelle et dynamique. ENCASMA peut supporter, entre autre, des applications de planification. En effet, l'architecture offre un monde virtuel similaire au monde réel, dans lequel des agents logiciels représentant les acteurs et les ressources du monde réel peuvent interagir entre eux et avec l'espace afin de planifier des actions, les tester et les évaluer avant de les exécuter sur le terrain. Dans le Chapitre 6 nous donnons plus de détails sur la façon dont ENCASMA supporte l'implémentation de l'approche DCP. Il est à noter par contre que l'architecture ENCASMA n'est pas dédiée exclusivement aux problèmes de planification. En effet, nous avons donné un exemple d'application (« Assistance sur le terrain ») qui démontre les éventuels avantages de l'adoption d'ENCASMA pour d'autres problèmes que la planification.

CHAPITRE 5

La simulation par agents pour la recherche de chemin (Pathfinding) dans un monde réel

Dans ce chapitre nous traitons une situation particulière impliquant un processus de recherche de chemin¹³ : Pathfinding. Mais tout d'abord, il est nécessaire de clarifier les termes utilisés dans la littérature et particulièrement la différence entre “*wayfinding*” et “Pathfinding”.

La recherche en *wayfinding* s'intéresse aux processus qui ont lieu lorsque des gens s'orientent eux-mêmes et naviguent dans l'espace [Raubal 01]. Cette recherche a pour but d'expliquer comment ces gens trouvent leurs chemins dans le monde réel, de quoi ils ont besoin pour y arriver, comment ils communiquent les directions, et enfin comment les capacités verbales et visuelles de l'homme influencent le processus de *wayfinding*.

Le Pathfinding, quant à lui, est un problème moins complexe puisqu'il n'inclut pas le facteur humain. C'est le processus de recherche d'un chemin entre une origine et une destination (tout en évitant les obstacles) sur une carte, ce qui implique souvent la recherche du chemin le moins coûteux. Le Pathfinding sur une carte est un problème fondamental pour plusieurs applications telles que la robotique et les jeux d'ordinateur. Ce problème de Pathfinding se manifeste dans la vie de tous les jours lorsque l'un de nous essaye d'aller quelque part avec un minimum d'effort possible et aussi vite que possible. Selon les caractéristiques de l'espace, les contraintes du problème, et les capacités de celui qui cherche le chemin (*Pathfinder*), le problème sera plus ou moins complexe. Durant les dernières décennies, plusieurs algorithmes ont été développés pour trouver le meilleur chemin. A* est certainement l'un des plus efficaces jusqu'à date. Il est largement appliqué dans la robotique et dans les jeux d'ordinateur. Cependant, certains

problèmes réels particuliers sous soumis à trop de contraintes pour pouvoir être résolus avec de tels algorithmes. Par exemple, face à un feu de forêt, les chefs d'opérations ont souvent besoin de construire une ligne d'arrêt pour les feux. Ils ont en fait suffisamment d'expérience pour évaluer la situation courante et proposer une esquisse de ce que pourrait être la ligne d'arrêt qui sera en général construite par des bulldozers. Étant donné que cette esquisse est approximative, il est souvent possible de trouver un meilleur chemin pour les bulldozers. Ce chemin doit respecter l'allure générale de l'esquisse proposée par l'expert humain ainsi que certaines règles et doctrines imposées par le domaine (p.ex. pas de virages étroits sur la ligne). De plus, les extrémités de la ligne à construire ne sont pas connues de façon précise, elles souvent sont « quelque part » près des zones d'ancrage (un lac ou une zone rocheuse). Chercher un chemin sous de telles contraintes n'est pas un problème classique de Pathfinding qui peut être résolu facilement avec les algorithmes connus. En effet, ces algorithmes sont conçus pour trouver le meilleur chemin sous des contraintes plus simples et plus relaxées. Malheureusement, certains problèmes réels ont des contraintes assez complexes et plus strictes et ne peuvent pas donc être rangés dans une des classes (voir section suivante) de problèmes classiques de Pathfinding. C'est la raison pour laquelle nous proposons dans la Section 5.1 une plus large caractérisation des problèmes de Pathfinding afin de pouvoir classer de telles situations particulières. Dans la Section 5.2 nous proposons une approche basée sur la simulation multiagent pour résoudre de tels problèmes de Pathfinding (dans un espace géographique réel à grande échelle et sous contraintes spatiales qualitatives). Cette approche est illustrée sur le cas des feux de forêt dans la Section 5.3 et plus particulièrement sur le cas de recherche de chemin pour les bulldozers lors de la construction des lignes d'arrêt pour les feux. Puisque le processus de Pathfinding doit également gérer les obstacles, nous présentons dans la Section 5.4 comment les agents peuvent contourner les obstacles lors de la recherche de chemin. La Section 5.5 propose une extension de l'approche proposée dans la Section 5.2 pour pouvoir résoudre des problèmes de Pathfinding similaires mais pour des applications en temps réel. Ce complément de l'approche est également illustré sur le cas des feux de forêt. Nous

¹³ Le processus de recherche de chemin sera appelé le long de ce chapitre selon le terme anglais : Pathfinding. Celui qui cherche le chemin sera appelé Pathfinder.

discutons ensuite l'approche proposée (et son extension) dans la Section 5.6 et nous faisons un bref aperçu sur les travaux connexes dans la Section 5.7. Enfin, une conclusion récapitule les principaux points abordés dans ce chapitre et résume les contributions de recherche apportées par ce chapitre.

5.1. Caractérisation du problème de Pathfinding

Il y a une variété de techniques pour le Pathfinding. Une technique peut exceller sous certaines contraintes, mais ne peut convenir sous d'autres [Reese et Stout 99]. Le succès d'une technique donnée dépend en fait de plusieurs facteurs qui peuvent influencer le processus de Pathfinding. Plus il y a des facteurs qui caractérisent un problème de Pathfinding, plus la solution est complexe. Dans les prochaines sous sections, nous essayons de classer les facteurs les plus importants. Ces facteurs peuvent être présentés sous trois grands volets : l'environnement, le *Pathfinder* et les contraintes de la procédure de recherche.

5.1.1. Facteurs liés à l'environnement

Dynamisme

Étant donné que les algorithmes de Pathfinding sont souvent utilisés pour les jeux ordinateur, les chercheurs comme Reese et Scout [Reese et Scout 99] décrivent le facteur *dynamisme* selon la nature du changement : (i) environnement statique (ii) joueurs mobiles et obstacles statiques (iii) obstacles mobiles (iv) obstacles qui apparaissent/disparaissent (v) manipulation. Dans un contexte plus général, nous pouvons distinguer entre un environnement statique et un environnement dynamique (ceci inclut toutes sortes d'obstacles comme par exemple les obstacles mobiles). Dans les environnements statiques, le Pathfinding est relativement facile. Il existe déjà un grand nombre de solutions pour ce genre de problème. Les algorithmes de recherche à base de graphes exhaustifs, tels que A*, se présentent comme les seuls algorithmes optimaux connus [Hu et al., 93]. Dans les environnements dynamiques, il est très coûteux de replanifier à chaque fois qu'une divergence est détectée. C'est la raison pour laquelle, des

algorithmes dynamiques tels que D* [Stentz 95] ou Kinetic Data Structures [Basch et al., 97] sont nécessaires.

Incertitude

L'information sur l'environnement peut être limitée – aussi bien de façon intentionnelle pour apporter le sens du réalisme aux jeux ordinateur, que non intentionnelle comme c'est le cas pour les systèmes distribués ou les applications réelles.

Évidemment, plus l'information est disponible, plus le chemin trouvé est exact. Cependant, puisque beaucoup de domaines d'application se basent sur des environnements incertains, plusieurs algorithmes ont été proposés. L'algorithme D* et ses dérivés (focused D* [Stentz 95], D* Lite [Koenig et Likhachev 02], etc.) sont justement utilisés dans ce sens. D* génère un plan initial en utilisant des données sur les coûts qui peuvent être connues, fausses et/ou estimées. D* commence ensuite l'exécution du plan et replanifie rapidement à chaque fois que de nouvelles informations sur les coûts lui parviennent.

5.1.2. Facteurs liés au *Pathfinder*

Tout d'abord, il est nécessaire de justifier pourquoi nous devons considérer ce facteur lors du choix de la technique appropriée à la réalisation du Pathfinding. Supposons que le *Pathfinder* est un véhicule, que l'environnement est représenté par une grille de cellules, et qu'une fonction de coût '*cost*' représente le coût de déplacement d'une cellule à une autre. En réalité, les différents véhicules ont plus ou moins de difficultés à monter ou descendre une pente. Prendre ceci en compte dans un environnement de simulation en altérant la fonction '*cost*' pourrait ne pas être approprié. En effet, ceci implique qu'il faut changer le coût de la grille représentant l'environnement pour chaque type de véhicule. Ceci est un exemple qui démontre la nécessité de considérer les caractéristiques du *Pathfinder* sans altérer l'environnement virtuel.

Caractéristiques du Pathfinder

Ces caractéristiques peuvent être statiques. Si le *Pathfinder* est par exemple un véhicule qui se déplace dans la forêt, la largeur du véhicule pourrait être une caractéristique statique importante puisque certains sentiers ou chemins forestiers pourraient ne pas être assez larges pour laisser passer le véhicule. D'autres caractéristiques sont dynamiques. Par exemple, si l'objectif principal d'un véhicule est d'« atteindre une destination A » mais s'il n'a pas assez de carburant pour y parvenir, il peut changer son objectif initial par un autre : « atteindre la destination A via une station service ». Le niveau de carburant dans le réservoir du véhicule pourrait ainsi être considéré comme une caractéristique dynamique du véhicule puisque ce niveau change au cours du processus de Pathfinding.

Dans cet exemple nous pouvons noter également l'importance de considérer les objectifs du *Pathfinder* qui peuvent changer durant la génération du chemin. Ceci justifie le second facteur lié au *Pathfinder* : les objectifs.

Objectifs du Pathfinder

L'objectif le plus commun pour un *Pathfinder* est d'atteindre une destination précise. Toutefois, l'objectif peut être plus complexe. Par exemple, une unité militaire dans une zone hasardeuse pourrait avoir comme objectif d'atteindre « n'importe quel endroit sûr ». Dans tous les cas, considérer les objectifs du *Pathfinder* est fondamental pour trouver le chemin le plus approprié. L'objectif du *Pathfinder* peut être aussi bien statique que dynamique. En effet, l'objectif peut changer durant le processus du Pathfinding comme nous l'avons illustré dans l'exemple du véhicule qui change son chemin selon le niveau de carburant dans son réservoir.

5.1.3. Facteurs liés à la procédure de recherche

Pour introduire ce dernier volet, nous empruntons un exemple du domaine des jeux par ordinateur et plus particulièrement des jeux de stratégie militaire. Pour pouvoir décider comment bouger d'un point à un autre, les planificateurs militaires distinguent trois sources de contraintes [Forbus et al., 01] : (i) traficabilité (la facilité avec laquelle une unité peut se déplacer le long d'un chemin donné). (ii) visibilité (elle décrit jusqu'à quel

point une unité peut être vue par l'ennemi et percevoir elle-même cet ennemi). (iii) champ de tir (la zone dans laquelle l'unité peut être la cible des tirs ennemis). La plupart des jeux d'ordinateur ne considèrent que la traficabilité, ce qui rend le jeu moins réaliste. Forbus et ses collègues [Forbus et al., 01] nomment ces contraintes *soft constraints*.

Un autre type de contraintes émerge du problème classique du « *massed fires* » [Forbus et al., 01] dans lequel trois unités sont déployées pour attaquer une cible précise. On suppose que le chemin le plus rapide (fourni par l'algorithme A*) pour atteindre l'ennemi mène à un passage étroit obligeant les trois unités à passer l'une après l'autre. Si une telle solution est adoptée dans un contexte réel, l'ennemi pourrait très bien détruire les trois unités à tour de rôle. Dans [Forbus et al., 01], les auteurs ont noté que pour un meilleur réalisme et immersion, le joueur (dans un contexte de jeu) devrait spécifier son intention d'entourer l'ennemi autrement que d'emprunter cet étroit passage en donnant une esquisse des chemins (les chemins que doivent emprunter les trois unités pour encercler l'ennemi).

Nous pouvons conclure que les algorithmes doivent considérer les contraintes spatiales rencontrées le long du chemin. Nous distinguons des contraintes spatiales qualitatives et quantitatives.

Contraintes spatiales quantitatives

Les contraintes quantitatives sont formelles et strictes. Elles peuvent donc être négociées par des algorithmes. Seules les contraintes sur le type du chemin [Reese et Stout 99] sont généralement considérées par les applications courantes et spécialement par les jeux. Parmi ces contraintes liées au type du chemin, la plus commune est probablement « le chemin le plus court ». Certains problèmes doivent être résolus par des moyens différents comme pour trouver le chemin à couverture maximale d'une zone donnée [Emmanuel et al., 94] ou pour trouver le chemin avec le minimum d'exposition [Hoff et al., 95], etc. D'autres contraintes qui ne sont pas liées au type du chemin telles que les *soft constraints* [Forbus et al., 01] peuvent aussi être considérées comme quantitatives, mais elles sont rarement utilisées.

Contraintes spatiales qualitatives

Si dans l'exemple du "Massed fires" nous nous proposons de trouver le chemin qui « suit au mieux » la « forme générale » de l'esquisse proposée par le joueur, un algorithme classique tel que A* ne peut pas être appliqué. Nous pensons que de telles contraintes qualitatives ne peuvent pas être implémentées par des algorithmes formels puisque les entrées du problème ne sont pas précises.

Contraintes temporelles

Les contraintes spatiales ne sont pas les seules à influencer le choix des techniques de Pathfinding à utiliser, mais aussi les contraintes temporelles. Nous pouvons distinguer en gros entre deux types de contraintes temporelles : *temps réel* et *temps non réel*.

Lorsque la contrainte du temps réel est absente, la technique reconnue comme la plus efficace peut être utilisée sans problème pour trouver le meilleur chemin. En effet, il y a suffisamment de temps pour chercher la meilleure solution. Toutefois, quand le temps disponible n'est pas suffisant, un compromis doit être fait. Les approches utilisées pour de telles situations et leurs limites sont discutées plus tard dans la Section 5.5. Les différents facteurs présentés et discutés ci dessus, sont récapitulés dans le graphe suivant (Figure 5.1):

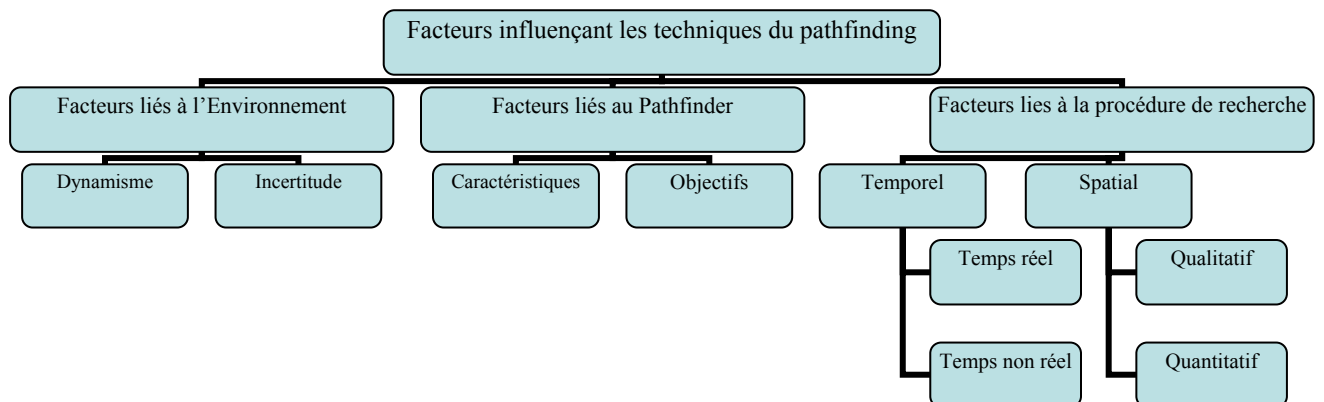


Figure 5.1. Hiérarchie des différents facteurs influençant les techniques du Pathfinding

Nous nous intéressons à faire du Pathfinding sous les contraintes suivantes :

- L'environnement est réel (la précision des données est fondamentale).
- Le *Pathfinder* a des objectifs vagues (p.ex. pas de destination précise) et plusieurs caractéristiques (statiques et dynamiques) sont à considérer.
- Les contraintes spatiales qualitatives doivent être prises en compte.

D'après nos recherches bibliographiques, la plupart des travaux sur le Pathfinding ne tiennent pas compte de la combinaison de toutes ces contraintes. De plus, ces travaux se concentrent surtout sur le Pathfinding dans les jeux d'ordinateurs exclusivement. Nous visons par contre à aider les *Pathfinders* réels dans des espaces réels et dans des situations critiques.

5.2. Pathfinding par Simulation basée sur les agents : approche proposée

Les agents logiciels sont surtout utilisés en Intelligence Artificielle, mais aussi récemment dans d'autres domaines comme la géographie [Frank 00]. Certains chercheurs ont démontré un intérêt croissant pour l'application des agents dans le *wayfinding*. Un des travaux les plus connus dans ce domaine est celui de Raubal [Raubal 01]. Il a utilisé les agents pour simuler comment les gens font du *wayfinding* dans les aéroports. Cependant, à notre connaissance, très peu de chercheurs ont appliqué les agents logiciels sur les problèmes de Pathfinding. Nous étudions dans cette section comment les agents peuvent être utilisés afin d'aider les agents réels (humains ou machines) à trouver leur chemin dans le monde réel.

5.2.1. Principes

Comme mentionné dans la section précédente, une technique efficace de Pathfinding doit considérer trois aspects: l'environnement, le *Pathfinder*, et les contraintes de la procédure de recherche.

Dans cette section, nous essayons de montrer comment ces trois volets sont traités par une approche qui est totalement basée sur les agents logiciels. Étant donné que l'objectif principal est d'aider un sujet réel à trouver son chemin dans un espace géographique réel, nous proposons de construire un monde virtuel qui est similaire au monde réel. Ce monde virtuel encapsule les informations pertinentes sur le monde réel. Les agents du monde virtuel peuvent ainsi avoir accès à une connaissance qui est augmentée (les données sont précises et connues pour chaque cellule de l'environnement virtuel), comparée à celle des agents réels qui est limitée (dans le monde réel). Les agents logiciels utilisent cette connaissance augmentée du monde de simulation afin de trouver un chemin approprié qu'ils communiquent ensuite au *Pathfinder* réel.

Comment modéliser l'environnement?

L'environnement doit encapsuler toute l'information (statique et dynamique) nécessaire au processus du Pathfinding. Puisque nous nous intéressons aux applications réelles, les données doivent être les plus fiables possibles. Nous utilisons ainsi avec la première couche (*plate-forme de simulation*) de l'architecture ENCASMA (Chapitre 4) des données fournies par un SIG. Une approche classique basée sur la modélisation de l'environnement virtuel par une grille de cellules peut être adoptée. Cependant, les coûts des cellules ne sont pas directement calculés et assignés à chaque cellule. En effet, pour plus de flexibilité, chaque cellule doit renfermer plusieurs types de données (pour l'exemple des feux des forêts, il faudrait avoir le type de l'essence de bois, l'élévation, la pente, etc.) et dépendamment des objectifs du *Pathfinder*, un ou plusieurs types de données seront considérés.

Le dynamisme peut également être modélisé suivant la deuxième couche (*modèle simulé*) de l'architecture ENCASMA. Les changements au niveau de l'environnement peuvent être modélisés dans le monde virtuel par un agent ou un ensemble d'agents. Par exemple, un nouvel obstacle impliquerait l'ajout d'un nouvel *Agent de Simulation* qui renferme les mêmes caractéristiques de l'obstacle. Un obstacle mobile par contre, serait modélisé par un agent mobile (mobile dans le sens qu'il est capable de se déplacer dans l'environnement de simulation). Les phénomènes naturels tels que les feux de forêt, pourraient être modélisés par un ensemble d'agents qui renferment les caractéristiques du

feu à différents points et qui se déplacent dans le terrain simulé selon un modèle physique prédéfini.

Comment modéliser le Pathfinder?

Dans une approche basée sur les agents, il est naturel de penser à modéliser le *Pathfinder* par un agent logiciel que nous nommons ici l'*Agent Pathfinder*. Dans la Section 5.1, nous avons déjà attiré l'attention sur la nécessité de considérer les caractéristiques et objectifs du *Pathfinder* réel pour certains problèmes complexes.

Ces besoins sont bien supportés par le paradigme agent. En effet, l'agent logiciel est capable de supporter aussi bien les caractéristiques du *Pathfinder* que ses objectifs. Le processus de Pathfinding peut ainsi tenir compte des besoins spécifiques du *Pathfinder*. La Figure 5.2 illustre l'instanciation de l'architecture ENCASMA pour les problèmes de Pathfinding dans un environnement géographique réel.

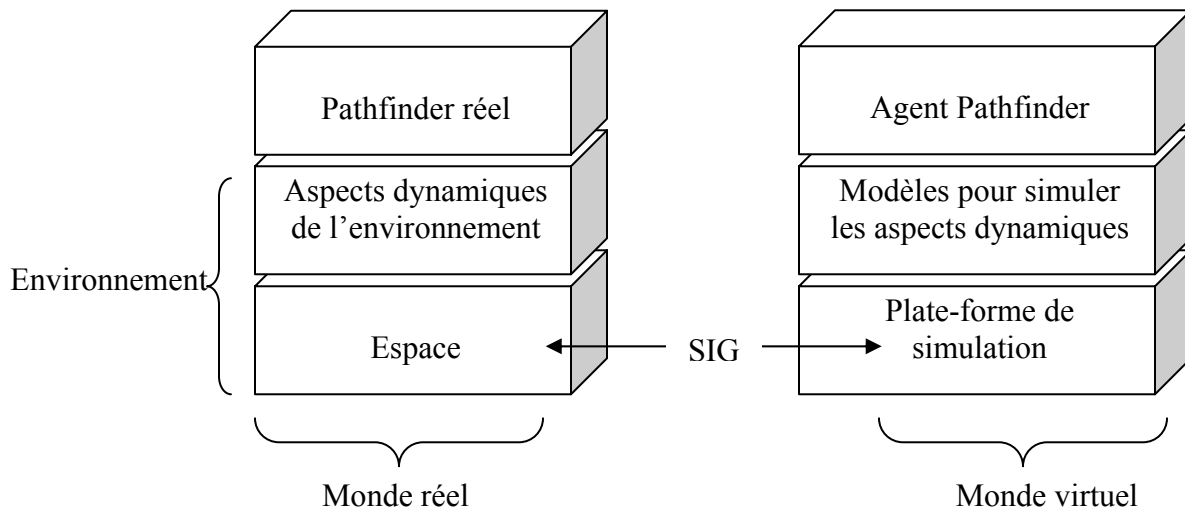


Figure 5.2. ENCASMA pour les problèmes du Pathfinding

Comment supporter les contraintes sur la procédure de recherche?

Comme nous l'avons mentionné précédemment dans la Section 5.1, les contraintes spatiales quantitatives sont efficacement supportées par les algorithmes tels que A*. Cependant, gérer des contraintes qualitatives n'est pas une tâche facile puisqu'un tel algorithme est trop formel pour pouvoir traiter des aspects qualitatifs. De plus, les

problèmes impliquant des contraintes qualitatives varient d'une application à une autre. Concevoir un algorithme générique pour un ensemble de ces problèmes, semble donc difficile à réaliser. Cependant, l'*Agent Pathfinder* peut avoir une certaine connaissance qui lui permettrait de supporter de telles contraintes. Nous nous proposons dans ce qui suit d'identifier les principaux mécanismes qui pourraient aider l'agent à acquérir cette connaissance à partir du monde virtuel.

5.2.2. Architecture interne de l'agent de Pathfinding

Dans un article célèbre [Stout 96], Stout fait la distinction entre le « Pathfinding pendant le déplacement » (“*Pathfinding on the Move*”) et « Regarder avant de chercher le chemin » (“*Looking before you leap pathfinding*”). Le dernier consiste à planifier le chemin au complet avant que la première étape ne soit entreprise. Ceci constitue la base des algorithmes classiques tels que *Breadth-first search*, l'algorithme de Dijkstra, et A*. Étant donné que cette approche n'est pas appropriée pour le genre de situations qui nous étudions, du Pathfinding par “*on the move*” est une bonne alternative. En fait, c'est ce qu'un sujet humain fait lorsqu'il n'est pas capable d'appréhender tout l'espace dans lequel il se trouve. Ceci est un problème de *wayfinding* pour lequel plusieurs modèles basés sur la perception, la décision et l'action, ont été proposés. À titre d'exemple, le modèle de *wayfinding* de Raubal [Raubal 01] intègre le schéma cognitif de l'agent et les structures de perception dans une approche *Sense-Plan-Act* (SPA) [Gat 98]. Un schéma cognitif interne [Neisser 76] guide les processus de perception de l'agent, de décision et d'action de l'agent durant la tâche du *wayfinding*. Dans un processus classique de *wayfinding*, un sujet humain ne peut pas percevoir tout l'espace à partir d'un seul point de vision. Il a besoin de naviguer dans l'espace pour y parvenir. Cependant, un *Agent Pathfinder* dans un monde virtuel, a accès à toutes les données. Ceci est la principale différence entre le processus du *wayfinding* et le type de Pathfinding que nous étudions. Un modèle similaire à *Sense-Plan-Act* est toutefois adopté pour l'*Agent Pathfinder*. Dans ce cadre, et afin d'assurer les processus de perception, de décision et d'action, nous avons besoin de doter l'*Agent Pathfinder* d'un mécanisme évolué de perception, d'un mécanisme qui gère les règles et les objectifs (pour la décision), et de capacités de navigation adéquates (pour agir, puisque agir dans un contexte de Pathfinding revient à se

déplacer). L'architecture interne d'un *Agent Pathfinder* est décrite par les trois composantes principales suivantes :

Composante de Perception

La perception de l'agent est généralement simulée dans les environnements virtuels pour imiter le comportement réel de l'entité simulée (p.ex. comportement humain). Ce que l'agent doit percevoir dans ce cas, doit être aussi similaire que possible à ce que l'*Agent Réel* aurait pu percevoir dans un environnement réel. L'*Agent Pathfinder* par contre, perçoit dans le but de planifier un chemin. Ses capacités de perception peuvent être plus évoluées que celle du *Pathfinder* (le réel) puisque l'objectif principal n'est pas de simuler une réalité, mais plutôt de profiter de l'environnement virtuel en tant que source de connaissances augmentées. Cela ressemble plus à un accès aux données qu'à un processus de perception. Toutefois, nous gardons le terme « perception » car l'*Agent Pathfinder* n'a quand même accès qu'aux données dans un certain rayon autour de lui (pour des raisons de performance).

Composante de Planification Spatiale

Après avoir perçu et avant de bouger, l'*Agent Pathfinder* a besoin de savoir à tout moment quel est le meilleur déplacement à faire. Pour y parvenir, l'*Agent Pathfinder* doit avoir des objectifs à atteindre. Ces objectifs sont associés à des actions que l'agent doit exécuter. Ils peuvent aussi changer durant le processus de planification. Dans un contexte de Pathfinding, un exemple typique d'objectif est d'« atteindre une destination *A* ». L'action correspondante à cet objectif est d'« aller à *A* ». Quand l'objectif est plus complexe, il est associé à des règles qui indiquent les contraintes à respecter pour atteindre l'objectif. Les règles peuvent refléter comment les contraintes de la procédure de recherche sont supportées.

Un être humain faisant face à une situation où il doit déterminer le meilleur déplacement à faire, pourrait avoir deux comportements différents : un comportement réactif ou un comportement délibératif. Quand il a suffisamment de temps pour planifier son chemin ou au moins une portion de ce chemin, il essaye de trouver la meilleure solution possible. Ceci est une planification délibérative. Par contre, quand un problème survient

soudainement, le sujet humain n'a pas souvent suffisamment de temps pour planifier son déplacement, alors il se trouve contraint à réagir le plus vite possible. C'est la planification réactive.

Par analogie, l'*Agent Pathfinder* peut adopter une approche délibérative pour trouver son chemin, ou encore une approche réactive quand il doit répondre immédiatement à des changements qui surviennent dans l'environnement. L'avantage de l'approche réactive est qu'elle permet de réagir vite aux changements lorsque le temps est critique. L'inconvénient, est qu'en général ni des objectifs à long terme ni à court terme ne peuvent être supportés. De plus, aucune garantie n'est donnée quant à trouver une solution finale. C'est la raison pour laquelle plusieurs chercheurs comme Jensen et Veloso [Jensen et Veloso 98] ont suggéré comme approche idéale pour des domaines dynamiques et à temps réel, la combinaison de la planification délibérative classique et la planification réactive.

La composante de *planification spatiale* de l'*Agent Pathfinder* s'inspire de cette idée et assure une sorte d'entrelacement entre la planification délibérative et la planification réactive.

Dépendamment du temps disponible et des contraintes courantes, l'une des deux stratégies décrites ci-dessous peut être appliquée. Le mécanisme est assuré par les composantes suivantes :

- Composante *Contrôle Rapide (Quick Control)*: Suite à un événement imprévu, cette composante est responsable de l'affectation d'un objectif à court terme à la composante de navigation (décrite plus loin). *Contrôle Rapide* est utilisée seulement dans un contexte de temps réel. Par exemple, quand l'*Agent Pathfinder* doit replanifier (suite à un obstacle imprévu ou parce qu'un bulldozer s'est déplacé autrement que ce qui a été prévu), il essaye de trouver une solution rapide et la communiquer au *Pathfinder* réel (le bulldozer) et ce dans les plus brefs délais. Cette solution (chemin trouvé rapidement) n'est pas forcément la meilleure. Elle pourrait même être incomplète. Cependant, elle respecte vaguement les contraintes de la situation et permet au *Pathfinder* de ne pas

attendre longtemps pour une meilleure solution (optimale et/ou complète). Pratiquement, quand une replanification est jugée nécessaire, l'*Agent Pathfinder* envoie quasi immédiatement un chemin possible au *Pathfinder* réel. Plus de détails sont présentés dans la Section 5.5.

- Composante *Contrôle raffiné (Refined Control)* : Après avoir envoyé un chemin rapide (par la composante *Contrôle Rapide*), l'*Agent Pathfinder* peut raffiner cette solution si nécessaire. Il pourrait par exemple essayer plusieurs autres possibilités en créant des duplicata de soi-même (des clones) qui se chargent donc de tester d'autres stratégies ou d'autres possibilités. Les agents clones doivent interagir entre eux afin d'éviter les plans redondants ou très similaires. Par exemple, si un agent détecte qu'un autre clone a déjà trouvé un meilleur chemin qui mène à la position dans laquelle il se trouve, il s'autodétruit. Le mécanisme est décrit en détail plus loin dans ce chapitre (sous-Section 5.3.4.1).

Si maintenant le processus de raffinement mène à une meilleure solution (que celle proposée par *Contrôle Rapide*), cette dernière est communiquée au *Pathfinder* réel. Cette nouvelle solution doit bien sûr tenir compte de la nouvelle position courante du *Pathfinder*.

Le processus de *Contrôle raffiné* est aussi appliqué de la même façon lorsque la contrainte du temps réel est absente (p.ex. lorsque l'utilisateur humain veut tester des stratégies sur l'environnement de simulation avant l'exécution effective du plan).

Navigation

Cette composante complète la précédente. Elle est en charge de l'exécution des déplacements de l'*Agent Pathfinder* dans l'environnement de simulation, telles que suggérées par le mécanisme de *Planification Spatiale*.

La Figure 5.3 illustre donc ces différentes composantes et qui constituent l'architecture interne de l'*Agent Pathfinder*. Le modèle est appelé *Perceive, Plan and Move PPM*.

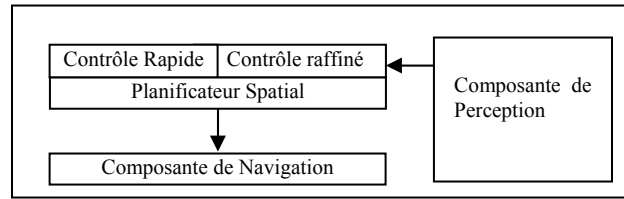


Figure 5.3. Architecture interne de l'Agent Pathfinder : modèle PPM.

La Figure 5.4 décrit le fonctionnement de l'Agent Pathfinder: l'agent commence par percevoir son environnement pour ensuite agir en conséquence. Le *Planificateur Spatial* se charge d'amorcer le *Contrôle Rapide* ou bien le *Contrôle Raffiné* selon les circonstances (discutées précédemment). Ceci permet d'indiquer à l'agent comment se déplacer tout en respectant ses objectifs. Étant donné le dynamisme de l'environnement, l'Agent Pathfinder doit recommencer le cycle par la perception afin de détecter d'éventuels changements qui ont pu survenir entre temps.

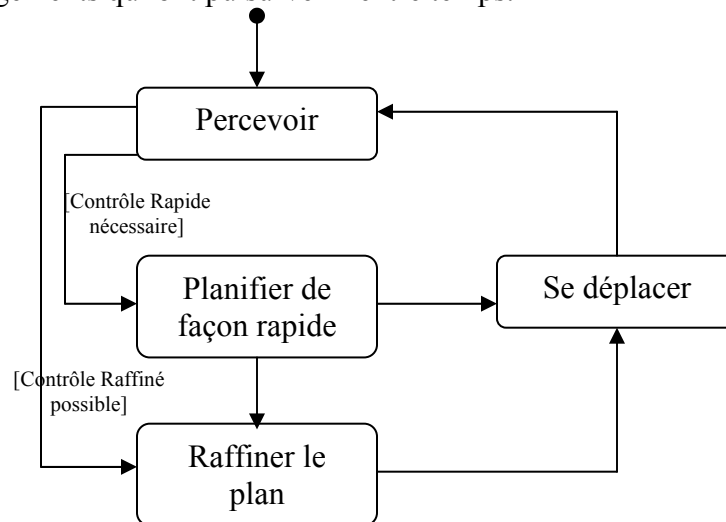


Figure 5.4. Diagramme d'état de l'Agent Pathfinder selon le modèle PPM

5.3. Illustration de l'approche sur la construction de la ligne d'arrêt

5.3.1. Scénario

Enlever l'essence forestière est la méthode la plus commune pour attaquer les feux de forêt. Cette méthode n'éteint pas le feu. Le feu continue à brûler jusqu'à ce qu'il n'y plus

d'essence à consommer. Écarter l'essence du passage du feu empêche en effet le feu de progresser.

Selon la taille du feu, l'attaque peut être directe ou indirecte. Pour les petits feux, les pompiers attaquent généralement le feu par son flanc (le côté) et progressent vers le front du feu jusqu'à encercler tout le feu : c'est l'*attaque directe*. Quand par contre le feu est plus intense, une *attaque indirecte* s'impose. Les pompiers construisent des lignes assez profondes et larges (d'environ 3m de large) qui entourent le périmètre du feu [Nicolas et Beebe 99]. Ce genre d'attaque nécessite aussi bien l'habileté et l'expérience que les ressources adéquates. Les intervenants peuvent utiliser des outils à main pour construire les lignes mais de préférence les bulldozers (si cette ressource est disponible). La ligne d'arrêt doit être construite à une distance sécuritaire des feux et doit profiter le plus possible des barrières naturelles ou artificielles disponibles (les incorporer dans la ligne) [Firewise 05]. La construction de la ligne d'arrêt commence généralement par un point d'ancrage et passe par des portions de terrain où l'essence est moins dense. Les virages étroits doivent être évités sur la ligne d'arrêt car ceux-ci peuvent causer des spots de feu (quand le feu saute par-dessus la ligne).

Pratiquement, des experts de lutte contre les feux décident vaguement de l'endroit où il faut construire la ligne en se basant sur les prédictions de progression des feux ainsi que sur leur expérience personnelle. Sur le terrain, les bulldozers arrivent sur l'endroit choisi par les experts et essayent de se trouver un bon passage pour avancer le plus vite possible. Pour ce faire, des éclaireurs (des hommes à pied) devancent les bulldozers afin d'explorer le terrain et guider ainsi les machines à travers les endroits les plus faciles à passer. Cette façon de faire présente plusieurs inconvénients : l'endroit choisi par les experts n'est pas forcément le meilleur; les passages trouvés par les éclaireurs ne sont pas forcément les meilleurs pour la prochaine étape et peuvent même mener plus tard les machines vers une impasse ou un obstacle difficile à franchir; Les pompiers ne sont jamais sûrs de pouvoir finir la ligne à tant même en absence d'imprévu; En cas d'imprévu (avancée plus rapide que prévue des feux), les hommes sur le terrain ne disposent pas de beaucoup de données pour pouvoir s'adapter au changement; etc.

Nous nous proposons donc d'assister les gestionnaires de feu en utilisant un environnement virtuel (virtuel mais aussi réaliste que possible) dans lequel des agents logiciels essaient de trouver un chemin convenable pour les bulldozers dans le cas de la construction d'une ligne d'arrêt par la machinerie lourde. Un chemin approprié est un chemin qui suit le plus possible l'esquisse proposée par le gestionnaire humain, qui tient compte de la progression du feu dans le temps et qui passe par les endroits les moins denses en essence tout en respectant les contraintes particulières du domaine (p.ex. pas de virages serrés).

5.3.2. Architecture

Pour atteindre les objectifs que nous nous sommes fixés, nous utilisons la plate-forme MAGS [Moulin et al., 03]. Les agents MAGS sont dotés de capacités cognitives (navigation, perception, planification spatiale, mémoire, etc.) qui leur permettent d'évoluer de façon autonome durant la simulation.

L'architecture de la solution proposée est décrite par la Figure 5.5. Seules les composantes importantes de MAGS sont représentées.

Dans ce qui suit nous décrivons ces composantes.

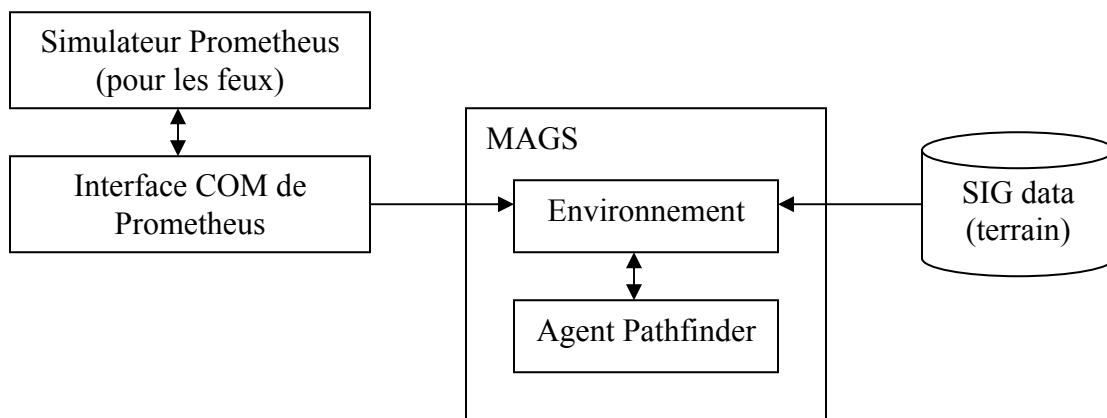


Figure 5.5. L'architecture de la solution proposée.

5.3.2.1. Simulateur Prometheus

Prometheus est le modèle de propagation des feux de forêt élaboré à partir du système *Canadian Forest Fire Behaviour Prediction* (FBP) ainsi que la plus récente vague d'algorithmes de propagation des feux développés à l'université de Brandon. Le système FBP est un système complexe et semi empirique qui exprime et intègre de façon mathématique, plusieurs facteurs qui influencent le comportement des feux de forêt tels que l'essence, le climat, la topographie, etc. Pour pouvoir utiliser les résultats de Prometheus dans un programme (comme MAGS), nous faisons appel à son interface COM qui donne un accès direct au moteur de simulation.

Prometheus fournit à chaque période de temps le périmètre de feu prévu. Chaque périmètre est donné sous forme d'un ensemble de sommets. Chacun de ces sommets encapsule les caractéristiques du feu à ce point précis (la position géoréférentielle, l'intensité du feu, la direction de propagation, la vitesse de propagation, etc.).

5.3.2.2. Environnement

L'environnement est représenté par une grille de cellules. Chaque cellule correspond à un carré réel du terrain (25 m² dans notre implémentation). Elle encapsule toute sorte d'informations pertinentes sur cette portion du terrain : élévation, degré de la pente, direction de la pente, type d'essence, etc. Les routes, les lacs et les rivières sont aussi représentés dans l'environnement de simulation, étant donné leur pertinence pour le domaine étudié. Les lacs et les rivières sont en effet des sources potentielles d'eau et de bonnes barrières naturelles qui peuvent être intégrées à une ligne d'arrêt. Les routes et les chemins forestiers permettent d'acheminer les ressources et peuvent servir parfois de ligne d'arrêt pour le feu (dépendamment de leur largeur). Chaque cellule de l'environnement de simulation contient donc l'information si la portion du terrain correspondante fait partie d'un lac, d'une rivière ou d'une route. Au besoin, des lacs et des rivières pourraient être représentés aussi par des agents logiciels : *Agents Lac* et *Agents Rivière*. L'*Agent Lac* par exemple représenterait tout le lac et encapsulerait des informations générales sur le lac telles que la superficie, la profondeur moyenne, etc.

Dans notre exemple, nous avons aussi des données dynamiques à considérer : les périmètres du feu. Comme mentionné dans la sous-section précédente, Prometheus fournit à l'environnement de simulation ces périmètres sous forme d'ensembles de sommets. Nous modélisons par la suite ces sommets par des agents logiciels. Chacun de ces agents (nommés *Agents Fire*) représente un sommet de feu ainsi que ses caractéristiques (position, intensité, vitesse de propagation, etc.).

Tous ces agents (*Agent Lac*, *Agent Pathfinder*, *Agent Fire*, etc.) interagissent entre eux afin de collaborer ou d'échanger des informations. À titre d'exemple, l'*Agent Pathfinder* qui perçoit les *Agents Fire* dans son voisinage peut communiquer avec ces agents avant de décider du prochain déplacement à faire (un déplacement qui doit être sécuritaire : assez loin des feux). Autre exemple : lors de la construction de la ligne d'arrêt, le gestionnaire des feux peut déployer deux bulldozers. Dans ce cas, si nous supposons que chacun des deux bulldozers est représenté par un *Agent Pathfinder*, chaque *Agent Pathfinder* doit commencer à partir d'un point d'ancrage (les deux points d'ancrage seront ainsi les deux extrémités de la ligne à construire) et essayer de joindre l'autre agent pour encercler ainsi le feu. Les deux *Agents Pathfinder* collaborent donc pour atteindre le même point de rendez-vous. Chacun de ces agents fait appel à ses capacités cognitives (surtout ici la perception) afin de détecter l'autre agent.

5.3.3. Agent Pathfinder

Dans notre exemple, l'*Agent Pathfinder* est un agent logiciel MAGS qui perçoit l'environnement décrit précédemment, planifie ses déplacements et se déplace dans le but d'atteindre sa destination. Son objectif ultime est de trouver un chemin approprié qui suit le plus possible l'esquisse proposée par l'expert humain et qui respecte certaines contraintes qualitatives.

La Figure 5.6 illustre le problème à résoudre. Un *Agent Pathfinder* (représentant le bulldozer) partant d'un point du voisinage de l'une des extrémités de l'esquisse dessinée, planifie son chemin comme suit (Figure 5.6):

- (1) détecter la direction de la portion de l'esquisse la plus proche.

- (2) Percevoir les *Agents Fire* (si lieu) et déterminer la *direction limite* (Figure 5.7) qui éloignerait l'*Agent Pathfinder* des *Agents Fire*.
- (3) Percevoir l'environnement en lançant un rayon de balayage sur une certaine distance et selon la direction déterminée en (1) ainsi que les directions du voisinage (l'angle entre la direction qui est la plus sur la gauche et celle qui est la plus sur la droite, est appelé l'angle de recouvrement). Toute direction choisie doit également être comparée à la *direction limite* déterminée en (2). Elle doit se trouver à droite ou bien à gauche de cette *direction limite* (selon que le feu soit à droite ou à gauche de l'agent) (voir Figure 5.7).
- (4) Calculer le coût de chaque rayon lancé (le coût relatif au déplacement du bulldozer selon la direction du rayon).
- (5) Si aucun rayon de balayage ne retourne un coût raisonnable (ceci peut être dû à un obstacle par exemple ou encore à cause de la présence du feu en face de l'*Agent Pathfinder*), alors élargir l'angle de recouvrement (cela revient à considérer plus de directions) et aller à (3) sinon (un coût raisonnable est trouvé) aller à (6).
- (6) Choisir la direction qui offre le moindre coût et se déplacer selon cette destination et jusqu'à faire la même distance que le rayon de balayage a parcourue.
- (7) Aller à (1).

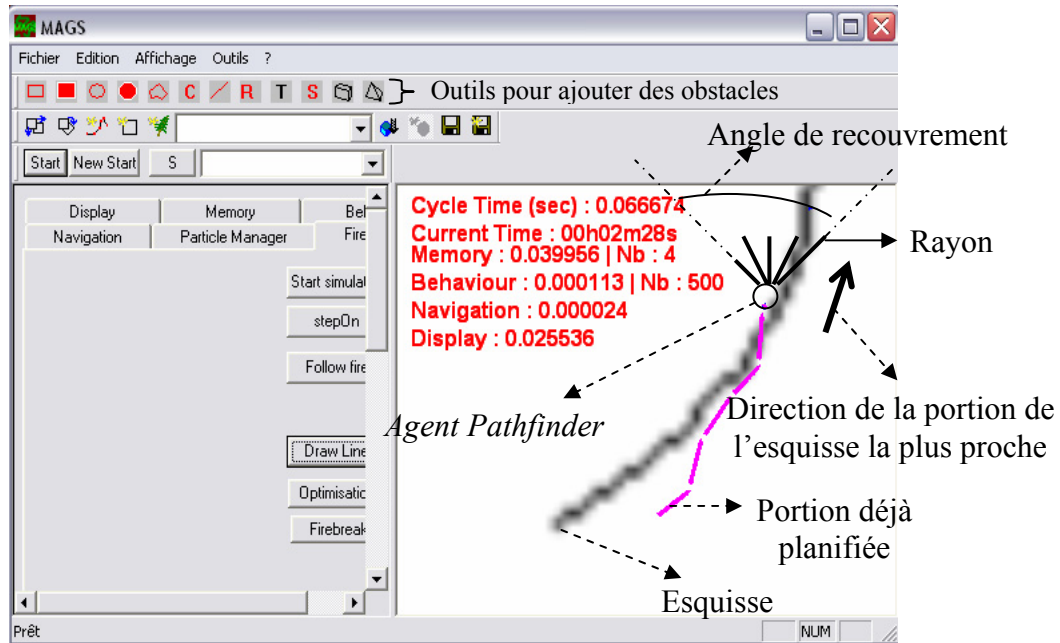


Figure 5.6. Procédure suivie par l'*Agent Pathfinder* pour suivre une esquisse (snapshot)

Dans l'étape (6), choisir la direction qui offre le moindre coût implique la prise en compte des caractéristiques du terrain, des contraintes spatiales et de la position du feu.

Les principales caractéristiques du terrain qui influencent le coût des déplacements sont : l'élévation, le degré de la pente, la direction de la pente et le type d'essence.

Étant donné que l'objectif de l'*Agent Pathfinder* est de suivre une esquisse sans faire de virages étroits, le degré de déviation par rapport à la direction donnée par l'esquisse ainsi que le voisinage de l'*Agent Pathfinder* (caractéristiques du terrain dans cette zone), sont les deux principaux facteurs qui influencent le coût d'un rayon donné.

Enfin, la proximité du feu constitue un danger pour les bulldozers. L'*Agent Pathfinder* doit alors prendre en considération ce facteur lors du choix de la direction à suivre. Tout d'abord, l'agent doit absolument éviter de se diriger vers le feu, ceci est assuré par la l'utilisation de la *direction limite* (Figure 5.7) comme seuil à ne pas franchir. Ensuite, même si l'agent ne se dirige pas directement vers le feu, il peut y être plus ou moins proche, ce qui rend son mouvement plus ou moins dangereux et risqué. Une direction qui mène ainsi le bulldozer proche du feu doit être plus coûteuse qu'une autre qui maintient le bulldozer à une distance sécuritaire des flammes. C'est la raison pour laquelle, et mise

à part l'utilisation de la *direction limite*, nous introduisons dans le coût de la direction un facteur lié au feu.

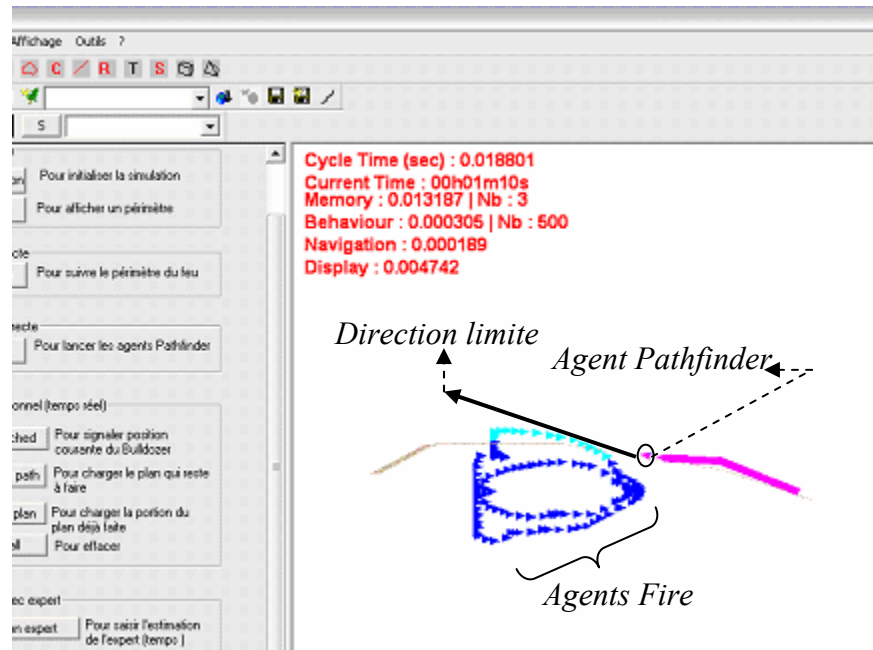


Figure 5.7. Définition de la *direction limite* (dans ce cas les directions choisies dans l'étape (3) doivent être à droite de la *direction limite*)

Si nous appelons *BeamCost* le coût d'un rayon donné, *TerrainCost*¹⁴ le coût lié aux caractéristiques du terrain à cet endroit là, *SpatialConstraintsCost* le coût lié aux contraintes spatiales, et *FireClosenessCost* le coût lié à la proximité du feu, l'intensité du feu et la direction de propagation, *BeamCost* est calculé selon l'expression suivante:

$$\begin{aligned} \text{BeamCost} = & \text{TerrainCost}(\text{elevation}, \text{slope}, \text{slope_direction}, \text{fuel_type}) + \\ & \text{SpatialConstraintsCost}(\text{deviation}, \text{sketch_nearby}) + \text{FireClosenessCost}(\text{fire_closeness}, \\ & \text{Fire_Spreading_direction}, \text{Fire_intensity}) \end{aligned}$$

L'*Agent Pathfinder* doit aussi supporter une autre contrainte spatiale: le fait qu'aucun point de départ ou d'arrivée précis n'est donné. En effet, le point de départ de la ligne à construire peut être n'importe lequel dans une certaine zone géographique au tour du point de départ donné par l'esquisse de l'expert. Il faut alors considérer tous les points de

cette zone comme d'éventuels points de départ (ou d'arrivée). Ceci pourrait augmenter nos chances de trouver un meilleur chemin pour le problème. En effet, commencer par un point au lieu d'un autre pourrait faire éviter à l'*Agent Pathfinder* un obstacle majeur qui aurait augmenté de façon significative le coût du chemin trouvé.

Si S est l'ensemble¹⁵ des points de départ possibles, l'*Agent Pathfinder* doit chercher un chemin à partir de chaque élément de S . Puisque nous sommes dans un environnement virtuel, nous pouvons très bien créer un *Agent Pathfinder* pour chaque élément de S . Ceci permet d'explorer plusieurs chemins possibles en moindre temps. Un *Agent Pathfinder* considère qu'il atteint sa destination s'il est dans une certaine zone prédéfinie (zone de l'ensemble des points d'arrivée possibles).

5.3.4. Éléments d'optimisation

5.3.4.1. Parallélisme par clonage

Voir Section 7.3.3.6. (*Agent Pathfinder*) du Chapitre 7.

5.3.4.2. Pré-Pathfinding

Le gestionnaire des feux en dessinant l'esquisse, ne propose pas nécessairement la meilleure position possible pour cette esquisse car il n'a pas les capacités nécessaires pour satisfaire toutes les contraintes mentalement et de façon aussi précise que la machine. Il ne donne en fait qu'une vague allure de la ligne d'arrêt ainsi qu'un emplacement approximatif de cette ligne. Il serait intéressant d'avoir une meilleure esquisse qui faciliterait par la suite le travail des *Agents Pathfinder*. En effet, une meilleure esquisse peut être obtenue par exemple juste en effectuant une petite translation ou rotation (ou une combinaison des deux) de l'esquisse initiale (proposée par l'expert humain). Une telle amélioration pourrait faire éviter à un *Agent Pathfinder* de contourner des obstacles majeurs (s'il se conformait à l'esquisse initiale).

¹⁴ Pour avoir un coût minimal, l'*Agent Pathfinder* doit éviter les portions à végétation dense.

¹⁵ MAGS peut supporter des milliers d'agents logiciels actifs en temps réel de simulation (et en mode accéléré). Avec la version courante, nous pouvons lancer des milliers d'*Agents Pathfinder* en parallèle, ce qui est largement plus que ce qui est nécessaire.

Afin d'évaluer chaque modification possible de l'esquisse générale, une approximation pourrait avoir lieu. Par exemple, si l'environnement est modélisé par une grille de cellules, nous pouvons évaluer le coût des cellules formant l'esquisse ainsi que son voisinage (Figure 5.8). L'esquisse qui correspond au moindre coût serait retenue pour le processus de recherche du meilleur chemin.

- Cellule de l'esquisse
- Cellule du voisinage de l'esquisse

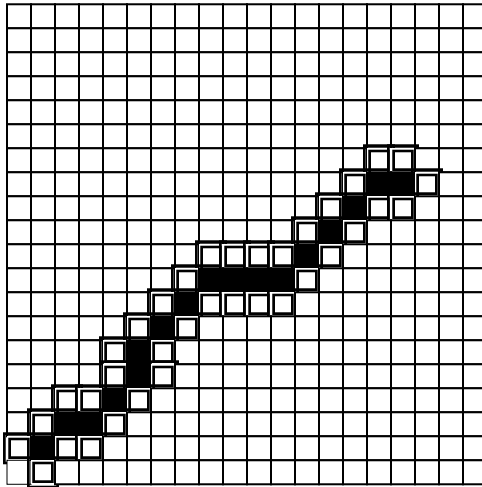


Figure 5.8.a. Esquisse initiale

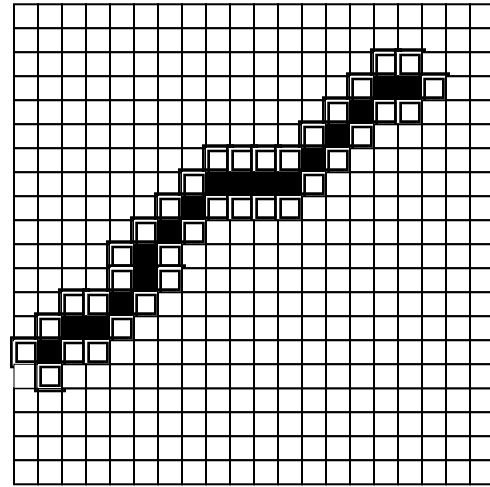


Figure 5.8.b. Esquisse après translation

Dans la Figure 5.8.a., l'esquisse initiale est évaluée en faisant la somme des coûts des cellules formant l'esquisse et son voisinage. Le coût de l'esquisse obtenue par translation (Figure 5.8.b.) est évalué de la même façon. Si cette dernière évaluation donne un coût plus bas que celui de l'esquisse initiale, l'esquisse translatée sera donc préférable à l'esquisse initiale.

5.4. Évitement des obstacles

5.4.1. État de l'art

Beaucoup de chercheurs ont proposé différentes approches pour la planification autonome de chemin. Nous présentons ici un bref aperçu des travaux les plus connus.

Une des premières approches proposées est l'approche du Champ de Potentiel (*Potential Field*). Elle a été introduite par Khatib [Khatib 86] en 1986. Elle est basée sur le principe intuitif de placer un champ attractif à la destination et un champ répulsif sur les objets (généralement les obstacles) et de superposer leurs effets pour que l'agent responsable de la recherche du chemin se déplace vers la destination en évitant les obstacles. La principale difficulté de cette approche est d'atteindre le potentiel global minimum (l'objectif) sans être bloqué dans les minimums locaux. Plusieurs améliorations ont été proposées par la suite pour éviter le problème du minimum local [Valavanis et al., 00; Koditschek 87; Koditschek and Rimon 88]. Toutefois, ces nouveaux algorithmes ne garantissent pas toujours l'existence d'un minimum global unique. De plus, ils ne fonctionnent pas très bien en présence d'obstacles mobiles. D'autres raffinements ont été proposés à la méthode du *champ de potentiel* tels que la méthode du *champ de force virtuelle* (*virtual force field*) [Borenstein and Koren 89] ou encore la méthode de l'*histogramme à champ vectoriel* (*vector field histogram*) [Borenstein and Koren 91].

À part ces approches qui sont inspirées de l'électromagnétisme, il existe d'autres méthodes inspirées cette fois de la biologie de l'être humain ou de l'animal. Dans [Tang et al., 01], les auteurs incluent des connaissances heuristiques de l'évitement des obstacles dans un ensemble flou (*fuzzy set*). Une approche d'apprentissage basée sur les réseaux de neurones est aussi utilisée pour éviter les obstacles. Dans [Rathbun et al., 02], un algorithme génétique est appliqué sur l'évitement des collisions.

Une approche basée sur une fonction de courant (*stream function*) [Akishita et al., 93] semble prometteuse puisqu'elle n'a pas le problème du minimum local. Cependant, elle nécessite la disponibilité en avance de l'information complète sur les positions des obstacles.

Nous trouvons également dans la littérature une large panoplie d'autres approches : approche de [Hu et al., 00], approches à systèmes hybrides telles que dans [Tomlin et al., 00] où les auteurs utilisent la théorie des jeux et les concepts des automates, la méthode du *Champ Total* (*total field*) [Sigurd et How 03], les *cartes routières probabilistes* (*probabilistic roadmaps*) [Kavraki et al., 96] et [Latombe 91], ou encore l'approche à *Fenêtre Dynamique* (*dynamic window*) [Fox et al., 97].

La plupart des techniques que nous venons de citer sont couplées avec la planification des chemins (Pathfinding). À titre d'exemple, la méthode *probabilistic roadmaps* est dédiée à la recherche de chemin tout en évitant les obstacles. Toutefois, pour le genre de problème qui nous intéresse dans ce travail de thèse, aucune de ces approches n'est appropriée pour planifier un chemin (ou difficile à appliquer). Par conséquent, nous ne pouvons pas utiliser ces méthodes comme telles pour résoudre notre problème d'évitement d'obstacles. Nous pouvons par contre nous en inspirer pour appliquer certaines idées et les adapter à notre problème.

Étant donné que nous nous intéressons à l'évitement d'obstacles durant la navigation, il semble être plus approprié de considérer les approches utilisées pour l'évitement d'obstacles en temps réel par les robots mobiles. En effet, ces robots essayent d'éviter les obstacles tout en se déplaçant. Ceci implique que les obstacles sont détectés au fur et à mesure (*on the fly*) et non en avance.

Les approches utilisées par les robots mobiles sous ces contraintes varient des algorithmes primitifs qui détectent un obstacle et arrêtent le robot sur le champ pour éviter la collision, jusqu'aux algorithmes sophistiqués qui permettent au robot de contourner les obstacles. Il va sans dire que ces derniers algorithmes sont beaucoup plus complexes puisqu'ils comportent certaines métriques qualitatives concernant les dimensions des obstacles.

Une des approches pour la navigation autonomes est la méthode du *wall-following* [Bauzil et al., 81]. La navigation du robot consiste à se déplacer le long d'un « mur » à une distance prédéfinie. Si jamais un obstacle est détecté, le robot considère cet obstacle comme un nouveau « mur » et contourne donc l'obstacle. Ce type de navigation est technologiquement moins exigeant puisqu'un problème majeur des robots mobiles (la détermination de leur propre position) est largement facilité. Bien entendu, la navigation robotique par la méthode du *wall-following* est moins polyvalente et n'est appropriée qu'à des applications bien spécifiques [Borenstein et Koren 89].

Une méthode plus générale et plus employée pour l'évitement des obstacles, est basée sur la détection des bords (*edge detection*). Dans cette méthode, l'algorithme essaye de

déterminer la position des arrêtes verticales de l'obstacle et par la suite de conduire le robot pour passer autour de l'une de ces arrêtes. La ligne qui connecte deux arrêtes verticales est utilisée pour représenter une des limites de l'obstacle.

Cette méthode a été utilisée dans plusieurs projets de recherches tels que [Borenstein 87] et [Crowley 84]. Selon [Borenstein et Koren 89], un des inconvénients de l'évitement des obstacles basé sur la détection des bords, est le fait que les robots doivent toujours s'arrêter devant l'obstacle pour prendre des mesures plus précises de l'obstacle.

5.4.2. Solutions possibles pour l'*Agent Pathfinder*

À la lumière de cette brève revue de littérature, nous allons déterminer les stratégies possibles que nous pourrions adopter pour notre problème.

Une première solution serait de garder la même stratégie que lors de la navigation (Section 5.3.3), c'est-à-dire, suivre la direction qui garantit le coût minimal de déplacement (le coût étant calculé sur les cellules du voisinage). Cette façon de faire revient en fait à faire du *wall-following*. En effet, l'agent va avancer jusqu'à se heurter au « mur » (l'obstacle). À partir de là, il va suivre l'obstacle jusqu'à le contourner.

Contrairement au cas du robot qui en temps réel doit décider de suivre l'obstacle par la droite ou par la gauche, l'agent logiciel est capable de tester les deux possibilités en même temps. L'*Agent Pathfinder* doit alors se cloner. L'*Agent Pathfinder* peut donc emprunter le chemin par la gauche par exemple pendant que son clone essaye de passer par la droite. Ceci est dû au fait que ces agents évoluent dans un environnement de simulation. L'anticipation étant permise, les agents logiciels pourraient se multiplier afin de tester le plus de chemins possibles avant que l'*Agent Réel* (qui est sur le terrain) n'ait besoin du plan. Ce parallélisme ne retarde donc en rien les actions sur le terrain.

Une deuxième approche possible serait de détecter les extrémités de l'obstacle en avance afin de le contourner d'une manière un peu plus « intelligente ». L'inconvénient de cette solution pour les robots (le fait qu'ils doivent s'arrêter pour prendre les mesures nécessaires) n'est plus valide pour les agents logiciels car même si ces derniers sont contraints à ralentir, ceci n'affecte en rien l'avancée de l'*Agent Réel* (ceci est dû toujours

au fait que les agents logiciels planifient de façon anticipée les actions qui seront exécutées plus tard sur le terrain, voir Chapitre 6). L'*Agent Pathfinder* peut ainsi faire appel à ses capacités de perception afin de détecter les extrémités de l'obstacle. Il pourrait même élargir son rayon de perception au besoin afin d'essayer de détecter les extrémités. Si les extrémités ne peuvent toujours pas être déterminées, l'agent logiciel pourrait revenir en arrière afin de mieux percevoir l'obstacle. Cette manière de faire est très courante chez l'être humain. En effet, lorsqu'une personne veut mieux observer un objet qui lui est très proche, elle recule afin de voir l'objet en globalité. Cette théorie a bien sûr ses fondements en géométrie : pour un même rayon de perception, la surface observée est d'autant plus grande que l'observateur est loin. L'agent logiciel pourrait ainsi avec le même rayon de perception mieux observer l'obstacle en reculant. Cependant, il doit allonger la portée de sa perception afin d'atteindre l'obstacle. L'avantage de faire reculer l'agent c'est de trouver un chemin meilleur pour contourner l'obstacle. En effet, en détectant de loin ses extrémités, l'agent peut chercher un meilleur passage pour aller jusqu'à l'une de ces extrémités.

De la même façon que pour la première approche, l'*Agent Pathfinder* pourrait se cloner afin d'augmenter ses chances de trouver la meilleure façon de contourner l'obstacle (se diriger vers une extrémité et laisser le clone se diriger vers l'autre extrémité).

5.4.3. Illustration sur la construction de la ligne d'arrêt

5.4.3.1. Les contraintes du problème

Afin de choisir la bonne approche, il est nécessaire de définir les contraintes du problème. En fait, le facteur qui pourrait le plus influencer le choix, est la position de l'obstacle par rapport à la position réelle du bulldozer au moment de la détection de cet obstacle. Ce dernier ne pourrait être perçu que lorsque le bulldozer est déjà très proche. Ceci est possible par exemple quand c'est le chauffeur du bulldozer qui rapporte l'existence d'un obstacle dans son chemin. La présence d'un obstacle pourrait par contre être signalée alors que le bulldozer est encore loin de cet obstacle. Il est clair qu'a priori, plus l'obstacle détecté est loin de la position de bulldozer, plus l'*Agent Pathfinder* a de possibilités d'optimiser le chemin qui permettrait de contourner l'obstacle.

5.4.3.2. Solution proposée

Compte tenu de la contrainte sur le moment de détection effective de la présence de l'obstacle (quand l'*Agent Réel* est déjà proche de l'obstacle ou non), le choix de l'approche adoptée par l'*Agent Pathfinder* va différer.

Nous allons traiter les deux cas et proposer une solution pour chacune des situations possibles.

1^{er} cas : l'obstacle n'est ajouté sur l'environnement de simulation que lorsque l'*Agent Réel* est déjà bloqué par l'obstacle sur le terrain.

Ceci survient lorsque l'*Agent Réel* rencontre sur le terrain un obstacle imprévu (qui n'est pas représenté sur l'environnement de simulation). Il notifie alors l'environnement de simulation qui est aussitôt mis à jour. Dans ce cas de figure, l'*Agent Pathfinder* doit trouver un chemin de passage à partir de la position courante de l'*Agent Réel* et le communiquer à ce dernier pour qu'il puisse contourner l'obstacle de la meilleure façon. Cette position courante de l'*Agent Réel* (sur le terrain) est une position de blocage puisqu'il ne peut avancer sans contourner l'obstacle. Pour ce faire, l'*Agent Pathfinder* (au niveau du monde de simulation) peut adopter l'approche du *wall-following*. Étant donné que la meilleure direction est a priori inconnue, l'*Agent Pathfinder* crée autant de clones que de possibilités. Généralement, un agent essaye de contourner l'obstacle par la droite et un autre par la gauche. Le meilleur chemin est ensuite communiqué à l'*Agent Réel* qui peut alors continuer sa progression sur le terrain.

2^{ème} cas : l'obstacle est ajouté sur l'environnement de simulation alors que l'*Agent Réel* est encore loin de l'obstacle.

Une fois que le nouvel obstacle est ajouté dans l'environnement de simulation, l'*Agent Pathfinder* doit vérifier si ce changement affecte ou non son ancien plan. Pour ce faire, il n'a qu'à essayer d'exécuter tout le plan en mode accéléré et voir s'il se heurte ou non au nouvel obstacle. Si l'obstacle n'a aucun effet sur le plan déjà établi, aucune action n'est entreprise par l'*Agent Pathfinder* (ceci ne veut pas dire que l'*Agent Pathfinder* n'aura jamais à contourner cet obstacle, car il se peut que plus tard le plan soit changé et que

l'obstacle devienne alors important à considérer). Dans le cas contraire (l'obstacle affecte le plan initial), L'*Agent Pathfinder* doit déterminer sa position de blocage, c'est-à-dire, la position à partir de laquelle il ne peut plus avancer sans avoir à contourner l'obstacle. Soit P_b cette position. L'approche *wall-following* n'est pas recommandée dans ce cas car l'*Agent Pathfinder* a la possibilité de trouver un chemin pour contourner l'obstacle de façon plus « intelligente » et optimale. En effet, il est généralement moins coûteux de « penser » à se diriger vers une extrémité de l'obstacle avant d'en être très proche. Nous proposons un algorithme pour que l'*Agent Pathfinder*¹⁶ puisse trouver un chemin moins coûteux pour contourner l'obstacle.

L'algorithme ainsi que les détails d'implémentations sont présentés dans le Chapitre 7, sous-section 7.3.3.5.

5.5. Vers un Pathfinding en temps réel

5.5.1. Motivations

Breadth-First, *Bidirectional Breadth-First*, *Depth-First*, *Dijkstra* et A^* sont tous des algorithmes de Pathfinding. Cependant, plusieurs chercheurs tels que LaMothe [LaMothe 99], affirment qu'aucun de ces algorithmes n'est applicable au Pathfinding en temps réel.

Même si le plus rapide des *Pathfinders* ne mettra pas beaucoup de temps pour trouver son chemin dans une application temps réel, il demeure inacceptable d'attendre qu'une solution soit élaborée [Stejo 01]. Le problème est encore plus accentué lorsque plusieurs demandes de recherche de chemin arrivent en même temps, ce qui est souvent le cas.

Pour remédier à de telles situations, le processus de Pathfinding peut tourner en arrière plan pendant que le *Pathfinder* humain attend le plan. Durant cette attente, nous pourrions laisser les choses comme telles (laisser le *Pathfinder* attendre) ou encore le laisser se déplacer dans la direction la plus probable, qui est souvent la direction directe vers le but [Stejo 01].

¹⁶ Nous verrons dans le Chapitre VII que l'*Agent Pathfinder* délègue un autre agent pour lui trouver les extrémités de l'obstacle à contourner.

Toutefois, une telle solution n'est pas acceptable pour des applications critiques telles que les applications militaires ou de secours. Par exemple, si une unité militaire est en train de chercher son chemin pour fuir d'autres unités ennemies, il serait inacceptable qu'elle avance tout droit vers un point donné jusqu'à ce que l'algorithme puisse donner un chemin optimal.

Des efforts ont déjà été entamés pour trouver une solution au problème de Pathfinding en temps réel [Niederberger et Gross 04] et [Serban 02]. Ces travaux de recherche s'appliquent bien sur certains problèmes temps réel tels que les jeux et les environnements virtuels. Cependant, ces approches ne traitent pas les problèmes résultant du fait que l'environnement est réel. Pour cette raison, elles ne peuvent pas être appropriées à des applications réelles critiques.

Pourquoi utiliser les SMA pour des applications spatiales en temps réel?

Pour une application du monde réel, nous avons besoin d'interroger un SIG afin d'obtenir certaines données ou pour demander des instructions de navigation. La complexité computationnelle d'une telle tâche est souvent un obstacle. Ceci est dû au grand nombre d'objets dans le SIG ainsi qu'aux délais nécessaires pour déduire de l'information qualitative à partir des données SIG qui sont pour la plupart métriques [Kray 01]. Pour assurer un temps de réponse tolérable, une approche multiagent pourrait être intéressante.

Puisque les agents logiciels sont supposés être autonomes, ils peuvent être affectés à des sous problèmes. Ils peuvent même se dupliquer pour une meilleure performance. Quand il s'agit de traiter une grande quantité d'informations, plusieurs instances d'un agent spécifique peuvent travailler de façon concurrente sur des petits ensembles de données.

Si maintenant la plate-forme courante est incapable de supporter toutes ces tâches dans les temps, un agent pourrait très bien décider de migrer vers un site plus puissant [Kray 01]. La communication inter-agents (p.ex. par messages) peut également améliorer la collaboration entre les agents.

Kray [Kray 01] ressort aussi d'autres avantages de l'utilisation des approches basées sur les agents dans un monde réel et où un grand nombre de facteurs sont importants : la géométrie, les facteurs contextuels, les langages, et les restrictions sur les ressources.

Informatique omniprésente

Dans le contexte d'un monde réel, le processus de Pathfinding peut être effectué par un système de simulation, parfois très loin (physiquement) du terrain réel où les *Pathfinders* réels essayent de trouver un chemin en temps réel. Les *Pathfinders* réels et les virtuels (des agents logiciels) doivent alors communiquer en continu. En raison des contraintes temporelles (temps réel) et spatiales (données spatiales réelles et précises), la communication entre le monde réel et le monde de simulation doit tenir compte de deux facteurs principaux : la précision spatiale et le temps de réponse.

L'avancée des technologies liées à l'informatique omniprésente (ou diffuse) nous offre l'opportunité de permettre à un *Pathfinder* réel de « voir à travers les yeux des agents logiciels » (en parlant des agents de l'environnement de simulation). Retracer un *Pathfinder* réel par GPS peut par exemple assurer la précision spatiale. Les agents de l'environnement de simulation sont notifiés automatiquement (ou semi automatiquement) par des données du monde réel grâce aux nouvelles technologies sans fil. Le projet Siren [Jiang et al., 04] est un bon exemple d'utilisation de l'informatique omniprésente pour faire profiter les agents réels (dans Siren, se sont des pompiers) de la connaissance spatiale augmentée que les agents logiciels peuvent avoir dans le monde simulé.

5.5.2. Scénario

Le scénario décrit dans la Section 5.3.1 montre comment trouver un chemin convenable pour les bulldozers avant que ces derniers ne commencent l'exécution. Étant donné que nous nous intéressons à une application réelle dans un terrain réel, le dynamisme de l'environnement influence beaucoup la solution recherchée (le chemin). Par exemple, un vent imprévu peut accélérer la propagation des feux et rendre ainsi le plan initial (chemin) inapproprié. De même, un bulldozer peut faire face à un obstacle imprévu. La

surveillance (*monitoring*) du *Pathfinder* et de la validité de son chemin est alors une tâche cruciale dans de telles situations réelles.

Nous supposons que les bulldozers sont équipés de GPS et qu'ils sont capables de suivre un chemin précis (en coordonnées géoréférencées) dicté par les agents. Cette technologie nécessaire existe et est actuellement utilisée dans le domaine de l'agriculture de précision [O'Connor 96; Marchant et al., 97; Zhu et al., 98]

Nous nous proposons de trouver un moyen de surveiller le chemin établi initialement (par les *Agents Pathfinder*) et de le mettre à jour en cas de besoin. Pour plus de réalisme et de précision, la cohérence doit être maintenue entre ce qui se passe dans le monde réel et ce que le gestionnaire des feux pourrait visualiser sur l'environnement de simulation.

5.5.3. Extension de l'approche au temps réel

5.5.3.1. Principes

Afin d'assurer une bonne cohérence entre le monde réel et le monde de simulation, la correspondance illustrée précédemment par la Figure 5.2 dans l'approche initiale, doit être révisée comme suit (Figure 5.9).

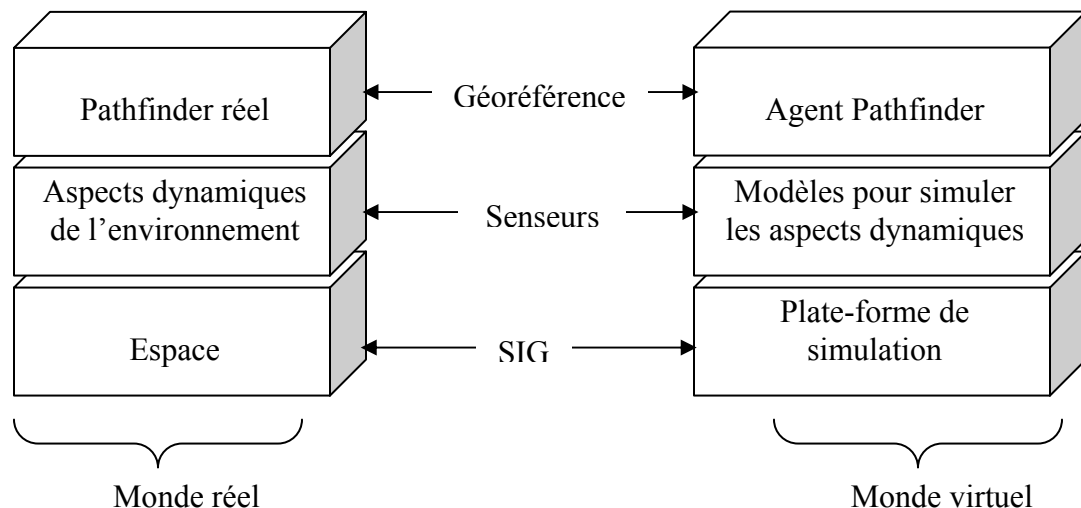


Figure 5.9. Correspondance entre le monde réel et le monde de simulation pour des applications temps réel

Les changements au niveau de l'environnement réel peuvent survenir à tout moment. C'est la raison pour laquelle des senseurs doivent être utilisés afin de notifier l'environnement de simulation de ces changements. Les senseurs peuvent être des radars (pour localiser des ennemis), des caméras, ou tout autre appareil capable de rapporter des informations pour mettre à jour les données de l'environnement simulé.

Puisque le *Pathfinder* réel est en mouvement, l'*Agent Pathfinder* correspondant doit lui aussi se déplacer dans son environnement virtuel de façon similaire. Ceci est désormais possible grâce au progrès réalisé par l'informatique omniprésente. Dans l'environnement virtuel, nous devons pouvoir visualiser la position courante (envoyée par un équipement GPS à partir du monde réel) du *Pathfinder* réel.

Étant responsable de trouver un chemin convenable, l'*Agent Pathfinder*, doit continuellement dicter à l'*Agent Réel* ses prochains mouvements. Toutefois, puisque la réalité est souvent différente de ce qui a été prévu, le *Pathfinder* réel pourrait se déplacer de façon différente de ce que lui a été suggéré par l'*Agent Pathfinder*. Par exemple, le *Pathfinder* réel peut rencontrer soudainement un obstacle imprévu qui ne figure pas dans l'environnement simulé (ceci peut arriver quand par exemple les données du SIG ne sont pas mises à jour) et décider de réagir en conséquence (en évitant l'obstacle par exemple). Dans ce cas, et afin de maintenir la cohérence entre le *Pathfinder* réel et son correspondant de simulation, l'environnement virtuel doit être ajusté (en ajoutant cet obstacle) et l'*Agent Pathfinder* doit ajuster également sa position dans l'environnement de simulation avant de relancer une nouvelle recherche de chemin basée sur les derniers changements. La Figure 5.10 décrit le fonctionnement de l'*Agent Pathfinder* dans un contexte de temps réel.

Le modèle agent décrit à la Section 5.3 (Figure 5.3) doit à son tour être modifié : un module de communication doit être ajouté. Cette composante assurera aussi bien la communication que la cohérence entre le *Pathfinder* réel et son *Agent Pathfinder* correspondant (la position de l'*Agent Pathfinder* dans le monde de simulation doit toujours correspondre à la position réelle du *Pathfinder* sur le terrain). Ceci est illustré par la Figure 5.11. Ce modèle est nommé *Perceive, Plan, Move, and Communicate* (PPMC).

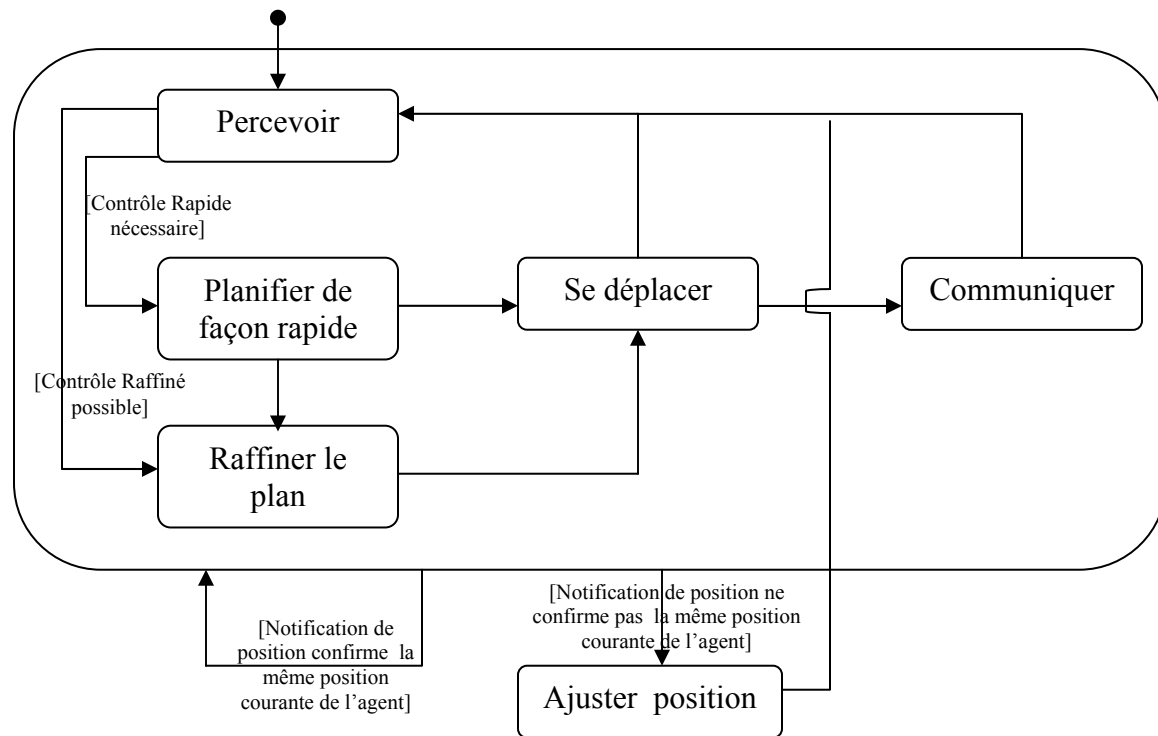


Figure 5.10. Diagramme d'état du fonctionnement normal de l'Agent Pathfinder dans un contexte de temps réel.

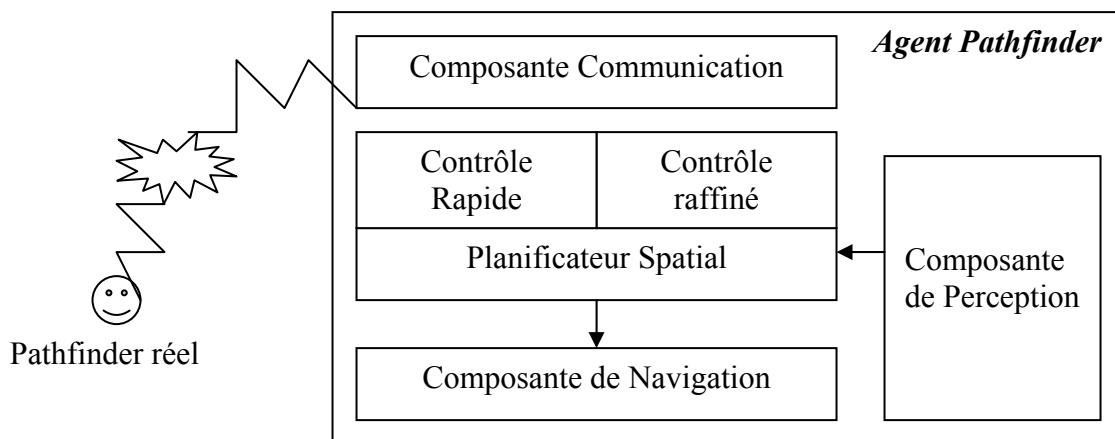


Figure 5.11. Modèle agent PPMC dans un contexte temps réel

5.5.3.2. Solution proposée pour la construction de la ligne d'arrêt pour les feux

Dans cette section nous présentons la conception des *Agents Pathfinder* pour le scénario spécifié en 5.5.2 et conformément au modèle PPMC et aux principes de la *planification anticipée en continu* (Section 6.2.2 du Chapitre 6).

L'*Agent Pathfinder* doit être lié au *Pathfinder* réel (bulldozer) grâce à une communication sans fil. Le bulldozer est aussi équipé d'un appareil géoréférencé qui indique périodiquement à l'*Agent Pathfinder* la position courante du bulldozer. Ceci est illustré dans la Figure 5.12.

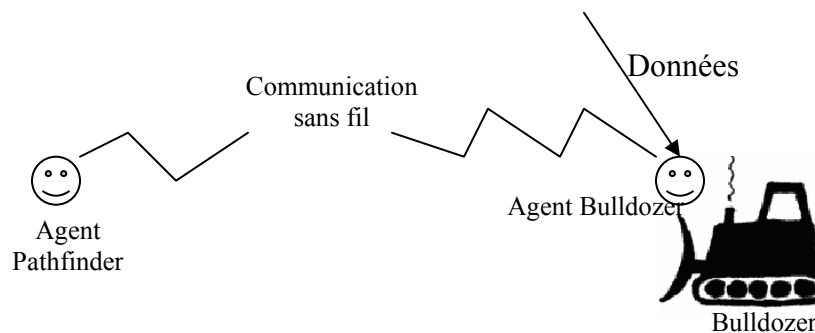


Figure 5.12. Relation entre l'*Agent Pathfinder* et le bulldozer

Selon le principe de la *planification anticipée en continu* (Chapitre 6), un plan global doit être proposé avant même que le bulldozer ne commence son travail. Ceci peut être fait comme on l'a décrit dans la Section 5.4. Désormais, nous supposons qu'un chemin a été trouvé pour acheminer le bulldozer d'un certain point de départ jusqu'à sa destination finale. Le bulldozer commence à travailler selon ce plan. L'*Agent Bulldozer* (l'agent logiciel qui accompagne le bulldozer réel) notifie périodiquement l'*Agent Pathfinder* de la position courante du bulldozer. Ainsi, si ce dernier décide de se déplacer différemment (suite à un imprévu), l'*Agent Pathfinder*, aussitôt prévenu, ajuste en conséquence sa position dans l'environnement de simulation. Un tel déplacement imprévu pourrait (donc pas toujours) rendre le plan initial irréalisable. L'*Agent Pathfinder* relance alors une nouvelle planification globale en partant de la nouvelle position courante du bulldozer.

Même si aucun ajustement n'est nécessaire, le processus de replanification doit être quand même lancé périodiquement. Ceci reflète l'aspect anticipatoire de la *planification anticipée en continu*. En effet, plusieurs facteurs qui ont été considérés pour trouver le plan initial pourraient changer durant l'exécution du plan. Par exemple, la progression des feux peut se révéler différente de ce qu'a été prédit pendant l'élaboration du plan. Le processus de replanification doit être mené à partir de la position courante du bulldozer. Si le nouveau plan est différent du plan initial, le bulldozer doit se conformer aux changements (Figure 5.13).

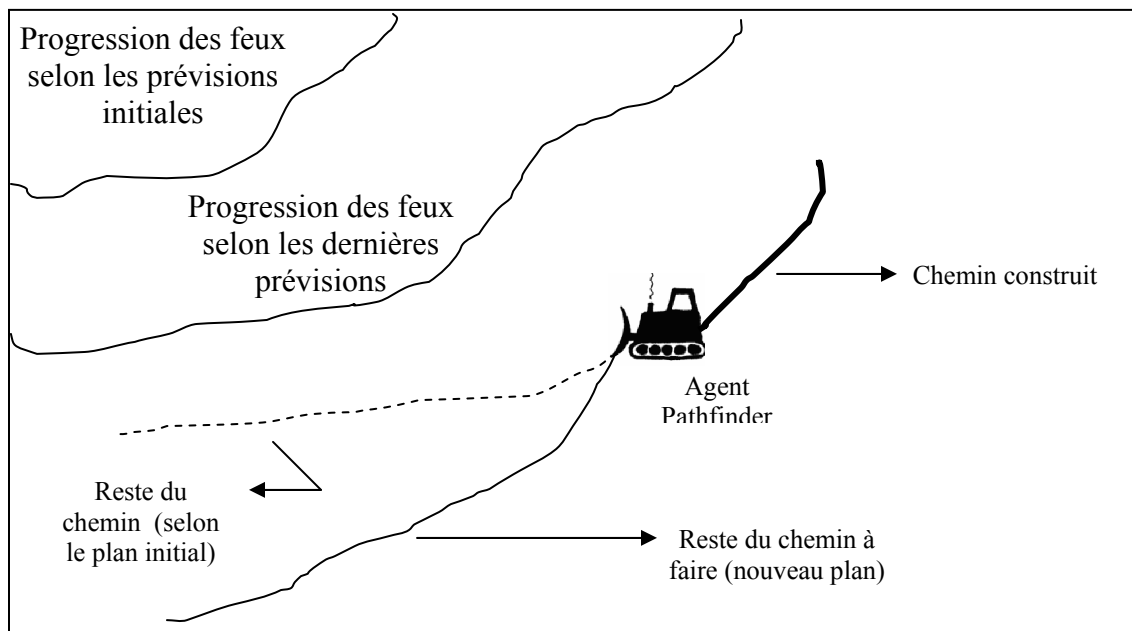


Figure 5.13. Application de la planification anticipée en continue pour la recherche de chemin

5.6. Discussion

Avant d'entamer une discussion sur les apports techniques et cognitifs des *Agents Pathfinder* proposés dans ce chapitre, nous faisons une petite récapitulation sur les deux modes de fonctionnement de l'*Agent Pathfinder* discutés en détail respectivement dans les Sections 5 et 6, tout en indiquant cette fois l'implication du gestionnaire humain dans le processus de planification.

L'*Agent Pathfinder* opère donc selon deux modes : un mode de pré-exécution et un mode en temps réel.

a) Mode *pré-exécution* : Un gestionnaire humain peut tester différentes stratégies par simulation de différents plans avant que l'exécution effective ne commence. Par exemple, il peut tester une stratégie d'attaque directe en demandant à l'*Agent Pathfinder* de suivre la progression des feux le long de la ligne de feu (par le flanc). Le chemin résultant est évalué par l'expert humain, ce qui va entraîner son refus ou son acceptation. Lorsqu'il s'agit d'une attaque indirecte, le gestionnaire humain demande aux *Agents Pathfinder* de chercher un chemin qui suit au mieux l'esquisse qu'il a proposée (dans notre application, nous offrons un outil visuel pour que l'utilisateur puisse dessiner cette esquisse dans l'environnement de simulation). Ces agents ont les capacités nécessaires (surtout de perception) qui leur permettent de considérer toutes les caractéristiques du terrain en détails lors de la recherche du meilleur chemin. Les *Agents Pathfinder* peuvent également se dupliquer (par clonage) et tester ainsi plusieurs scénarios possibles simultanément. Chaque *Agent Pathfinder* qui se trouve face à plusieurs choix, crée des clones qui vont tester les autres possibilités offertes. Afin de contrôler une éventuelle explosion combinatoire des agents clonés, ces agents doivent collaborer afin d'éviter les solutions redondantes ou similaires. Nous notons ici que ces agents logiciels responsables de trouver un meilleur chemin qui respecte des contraintes qualitatives (suivre au mieux une esquisse, éviter les virages étroits, etc.) doivent être sensibles à l'espace dans lequel ils évoluent (sensibilité spatiale). Les algorithmes classiques de Pathfinding tels que A* ne peuvent pas supporter de telles contraintes contrairement à nos *Agents Pathfinder* qui bénéficient de mécanismes avancés pour la résolution des problèmes spatiaux.

b) Mode *Temps réel*: Quant le plan est en train de s'exécuter dans le monde réel, l'*Agent Pathfinder* est utilisé pour d'autres fins. Premièrement, il est responsable de retracer dans l'environnement de simulation les déplacements effectifs des bulldozers. Deuxièmement, il doit replanifier de façon périodique et continue le reste du plan en commençant à chaque fois par la position courante du bulldozer à cet instant là. Ceci est fait afin d'anticiper tout problème qui peut survenir parce que l'environnement réel est tellement dynamique que les données (telles que la progression effective des feux) peuvent être très

différentes de ce qu'a été prévu quelques minutes auparavant. L'approche est décrite avec plus de détails dans le Chapitre 6. Finalement, l'*Agent Pathfinder* doit replanifier dès qu'un imprévu est rapporté. Par exemple, un bulldozer peut notifier l'*Agent Pathfinder* de la présence d'un obstacle imprévu. L'agent essaye alors de trouver un nouveau plan qui tient compte de ce nouvel obstacle, ainsi que la position courante du bulldozer. Le nouveau plan est par la suite acheminé au bulldozer. Étant donné que le processus de replanification s'effectue dans l'environnement de simulation, l'*Agent Pathfinder* peut créer autant de clones que nécessaire pour tester le plus de stratégies possibles. Ceci dépend aussi du temps disponible qui déterminera si l'*Agent Pathfinder* doit solliciter sa composante à *Contrôle Rapide* ou celle à *Contrôle Raffiné*.

Rappelons que l'un des objectifs de cette thèse est de proposer une complémentarité cognitive entre le planificateur humain et les agents logiciels planificateurs. Nous examinons ici ce point.

Dans la stratégie d'attaque indirecte, la procédure courante consiste à déployer des pompiers éclaireurs sur le terrain. Ces éclaireurs (des hommes à pied) devancent les bulldozers afin d'appréhender le terrain et guider ainsi les machines à travers les endroits les plus faciles à passer (les éclaireurs accrochent des rubans rouges sur les arbres pour indiquer le chemin aux bulldozers). La qualité du chemin proposé dépend ainsi de la perception de ces éclaireurs. Cependant, puisque ces derniers n'ont pas une vision globale de la situation (ils ne peuvent percevoir que leur voisinage), ils ne peuvent qu'au mieux proposer une portion du plan au lieu d'un plan global qui mène à destination. Ceci pourrait entraîner bien entendu l'échec du plan, ce qui n'est pas toléré pour de telles situations critiques. Les décisions prises par les éclaireurs qui sont déjà limitées par les capacités humaines de calcul, peuvent être biaisées. L'*Agent Pathfinder*, au contraire, est capable (en utilisant surtout ses capacités de perception) de considérer les principales caractéristiques du terrain en détail dans l'environnement de simulation afin de trouver un chemin convenable. Il peut ainsi proposer un plan global. Cependant, cet agent a besoin de suivre une esquisse proposée par le gestionnaire humain. Pour proposer une telle esquisse, l'expert humain fait appel à son expérience, à sa compréhension de l'espace ainsi qu'à son raisonnement spatial intuitif (ce qui ne peut être assuré par des agents

logiciels, du moins pour l'instant). Nous avons ici un exemple de complémentarité entre les cognitions spatiales des agents logiciels sensibles à l'espace d'une part et des planificateurs humains d'autre part, et ce dans le cadre d'une géosimulation multiagent.

Puisque l'être humain a des limitations au niveau de ses capacités de simulation (Section 2.1 du Chapitre 2), l'*Agent Pathfinder* lui offre un bon support car il est capable de simuler des plans complexes avec précision et en très peu de temps. Cependant, étant donné la complexité du domaine et sa nature (situation réelle et critique), les agents logiciels ont à leur tour besoin des planificateurs humains pour valider leurs plans. En effet, les agents logiciels de façon générale et les *Agents Pathfinder* particulièrement, n'ont pas le même sens raffiné d'anticipation et de jugement que les experts humains. Un plan élaboré par des agents logiciels, même si dans son apparence il semble bien construit, peut ainsi ne pas être faisable dans le monde réel ou peut contredire certaines doctrines. Les experts humains peuvent proposer de changer ce plan selon leur propre expérience et leur sens de l'anticipation. Le phénomène est plus ressenti chez les militaires comme il est décrit dans [Lewis et al., 01]. En effet, les commandants ont une vaste expérience qui n'est pas facile à quantifier. Ils doivent considérer des facteurs très caractéristiques de la situation en cours tels que les informations non quantifiables, les objectifs complexes ou vaguement spécifiés, les situations dynamiques (information incomplète ou changeante, obstacles, actions ennemies, etc.), etc. La géosimulation multiagent peut donc compléter les compétences des planificateurs humains pour résoudre des problèmes complexes.

5.7. Travaux connexes

Comme nous l'avons mentionné au début du chapitre, le Pathfinding est surtout utilisé pour les jeux d'ordinateur. A* est toujours l'un des algorithmes les plus utilisés. Quand il s'agit d'un environnement dynamique, A* est remplacé par D* [Stentz 94]. Les réseaux de neurones artificiels (*Artificial Neural Networks*) et les algorithmes génétiques [Graham et al., 03] sont les deux approches d'apprentissage les plus utilisées pour ce genre de situation. Cependant, très peu de travaux considèrent les contraintes énoncées dans la Section 5.1.3 : L'environnement est réel; Le Pathfinder a des objectifs vagues et

plusieurs caractéristiques (statiques et dynamiques) à considérer; Les contraintes spatiales qualitatives doivent être prises en compte.

Nous nous sommes également intéressés un peu à l'aspect temps réel. Nous avons alors remarqué que le Pathfinding en temps réel est principalement utilisé pour la navigation des robots et aussi pour les jeux d'ordinateur. Plusieurs approches ont été proposées dans ce cadre, la plupart de ces travaux sont basés sur D* comme dans [Yahya et al., 98]. Toutefois, l'espace considéré est souvent fictif (comme dans les jeux) ou simple (comme dans la zone d'action du robot). Très peu de recherches ont ciblé la recherche de chemin dans un espace géographique réel, complexe et à grande échelle. Dans un tel espace, le processus de Pathfinding ne peut être assuré dans l'environnement réel (pour des raisons que nous avons déjà citées) mais plutôt dans un monde virtuel parallèle au monde réel.

L'idée de faire du Pathfinding par simulation n'est pas nouvelle. Elle a été déjà utilisée par [Lee et Fishwick 95] pour décider du chemin d'un *rover* en temps réel. Le système proposé intègre plusieurs approches et algorithmes incluant A*. Le même système a été par la suite appliqué au problème de planification des missions militaires aériennes [Lee and Fishwick 97] où les auteurs ont adopté une approche « multiagent ». Cependant, l'environnement considéré n'est pas très réaliste (ne se base pas sur des données SIG), les agents logiciels ne sont pas autonomes et sensibles à l'espace, et enfin la complémentarité entre le monde réel et le monde simulé est pratiquement négligée.

Justement, pour ce qui est de l'utilisation des agents logiciels pour faire du Pathfinding, une revue de littérature montre que très peu de chercheurs ont essayé d'appliquer les SMA au processus de Pathfinding. Parmi ces chercheurs, Arikan et ses collègues [Arikan et al., 01] ont proposé une méthodologie de Pathfinding conçue pour un grand nombre d'agents qui se déplacent simultanément les uns autour des autres ainsi qu'autour des obstacles fixes. Quand un agent reçoit l'ordre de se déplacer vers une destination donnée, un premier plan est construit : il ne tient compte que des obstacles fixes. L'agent planifie ensuite comment contourner les autres agents présents dans l'environnement et qui se sont arrêtés sur son chemin. Il peut par la suite commencer son déplacement tout en évitant les collisions avec les autres agents qui sont cette fois en mouvement. Les obstacles fixes sont représentés dans l'environnement de simulation par de simples

polygones. Le chemin le plus court entre deux sommets donnés est pré-calculé par l'algorithme de Dijkstra jusqu'à ce que tous les plus courts chemins soient déterminés pour tout autre sommet. Cette approche ne peut pas évidemment être appliquée sur le type de problème que nous traitons car les contraintes du problème sont différentes surtout au niveau de :

- L'environnement : Arikan et ses collègues [Arikan et al., 01] supposent que tous les objets de l'environnement sont pareils et simples, tandis que nous considérons un environnement géographique réel et à grande échelle.
- Type de contrainte : Les contraintes d'Arikan sont simples alors que dans notre problème elle sont spatiales et qualitatives.
- Obstacles : Dans [Arikan et al., 01] les obstacles sont bien déterminés et dénombrés alors que dans notre problème toute cellule peut être plus ou moins un obstacle.
- Temps réel : Le pré-calcul des obstacles dans [Arikan et al., 01] est très coûteux surtout si on adopte un environnement réel, ce qui empêche la contrainte du temps réel d'être supportée.
- Etc.

Le seul travail qui ressemble un peu à ce que nous avons proposé, est celui de Lewis, Lenox, Payne et Sycara [Lewis et al., 01]. Les auteurs proposent un simulateur de champ de bataille (MokSAF) où chaque commandant participant doit planifier un chemin qui commence à un point de départ différent à chaque fois mais qui arrive au même point de rendez-vous que les autres chemins proposés par les autres commandants. La planification des chemins est supportée par des agents logiciels nommés *route-planning agents RPA*. Trois types de RPA interagissant avec les commandants humains ont été développés [Lewis et al., 01]:

- *Le RPA autonome* : il guide les commandants dans la tâche de Pathfinding mais effectue la majeure partie du travail tout seul. L'agent crée le chemin en faisant

- appel à sa connaissance du terrain en utilisant un algorithme de planification IA (non détaillé par les auteurs) qui a pour but de trouver le plus court chemin.
- Le *RPA coopératif*: il analyse les chemins dessinés par les commandants, sélectionne les points optimaux de ces chemins et aide ces planificateurs humains à raffiner leurs plans. Dans ce cas, le participant humain et l'agent travaillent conjointement. Les tâches sont distribuées de telle façon que chacun se charge du travail dans lequel il excelle. Le commandant, qui a une compréhension privilégiée des contraintes *intangibles* (les zones du terrain vaguement localisées par les commandants et qui sont à éviter) et les particularités de la mission, se charge ainsi de diriger le chemin selon ces contraintes et comme il le désire. Cependant, le commandant ne possède pas une connaissance détaillée du terrain. C'est la raison pour laquelle, l'agent se charge de lui indiquer les portions du chemin qui violent les contraintes physiques. L'agent est donc responsable de la révision du chemin dessiné par le commandant et de son amélioration si nécessaire.
 - *RPA naïf*: il offre une assistance minimale aux commandants lors du dessin et du raffinement des chemins.

Des expériences menées sur des commandants montrent que la recherche de chemin par des agents coopératifs donne les meilleurs résultats. Les commandants ayant expérimenté l'environnement avec le support des *RPA autonomes* ont manifesté leur frustration envers cette autonomie des agents et ont souhaité pouvoir « juste dessiner le chemin à la main » [Lewis et al., 01]. Ceci démontre en effet, que déléguer toute la planification aux agents n'est pas une bonne idée mais que la complémentarité entre les agents logiciels et les humains est plus appropriée pour ce genre de planification. Cependant, la complémentarité proposée par les auteurs est moins profonde que ce que nous présentons dans notre approche. En effet, nous mettons plus l'emphase sur les capacités cognitives spatiales des agents et des humains. Les commandants qui ont essayé de planifier les chemins par leurs propres moyens (cas du *RPA naïf*) ont été confrontés à beaucoup de difficultés. La procédure a été fastidieuse et les chemins trouvés contenaient beaucoup d'erreurs. Les auteurs arrivent ainsi à des conclusions semblables à ce que nous avons

établi dans notre travail : les agents logiciels peuvent offrir une grande assistance aux planificateurs humains pour trouver de meilleurs chemins.

Bien que ce travail [Lewis et al., 01] soit assez différent du notre (pas de contraintes qualitatives supportées, pas de données réelles, pas de parallélisme entre un monde virtuel et un monde réel pour supporter des applications réelles en temps réel, les agents n'ont pas des capacités de perception, de mémoire, etc.), il reste toutefois très intéressant car il aboutit à des conclusions très similaires à celles que nous avons présentées dans ce chapitre : les agents logiciels sont très bénéfiques pour le Pathfinding, la simulation est un bon moyen pour faire du Pathfinding pour des applications réelles et complexes dans un grand espace géographique et enfin la complémentarité entre agents logiciels et planificateurs humains est à la base de la réussite du processus de Pathfinding assisté par agents.

5.8. Contributions du chapitre

Dans ce chapitre nous avons présenté une approche de Pathfinding dans un environnement géographique réel, à forte composante spatiale, et soumis à des contraintes complexes (quantitatives et qualitatives). L'approche est basée sur la simulation par des agents logiciels. Nous avons démontré également que l'utilisation des agents logiciels est très bénéfique pour résoudre de tels problèmes complexes de Pathfinding. En effet, les agents logiciels, munis de capacités cognitives développées, sont capables de supporter des contraintes plus complexes que les algorithmes classiques de Pathfinding ne peuvent pas gérer. Les caractéristiques du *Pathfinder* sont aussi prises en considération par les agents logiciels et des chemins mieux adaptés aux besoins des *Pathfinders* peuvent alors être proposés. L'utilisation de la simulation multiagent dans le processus du Pathfinding permet également de meilleures performances lors de la recherche du meilleur chemin (possibilité d'évaluer plusieurs stratégies et chemins en même temps grâce au clonage des agents), d'éviter les obstacles dynamiques et même de supporter des contraintes temporelles très exigeantes telles que le temps réel. En effet, nous avons proposé dans ce chapitre une extension de l'approche qui supporterait la recherche de chemin en temps réel.

CHAPITRE 6

ENCASMA : support pour la DCP

Dans ce chapitre, nous discutons l'apport d'une architecture basée sur la simulation multiagent, telle que ENCASMA pour une meilleure implémentation de l'approche DCP. Dans la Section 6.1, nous faisons un bref aperçu sur l'approche DCP afin de relever ses exigences et ses limites. Nous mettons l'emphase sur un aspect particulier de l'approche : le processus d'entrelacement de la planification avec l'exécution. Dans la Section 6.2, nous proposons une nouvelle façon d'implémenter la DCP (par un SMA dans un environnement géospatial de simulation) qui donne des solutions aux limites relevées dans la Section 6.1. La nouvelle approche est illustrée sur le cas des feux de forêt et plus précisément, sur la construction d'une ligne d'arrêt lors d'une attaque indirecte.

6.1. DCP: état de l'art

Comme nous l'avons souligné dans le Chapitre 2, la planification distribuée en continu (ou encore la DCP) est une approche bien adaptée aux environnements dynamiques et incertains. Dans la sous-Section 6.1.1, nous faisons une petite synthèse de l'approche DCP tout en spécifiant les exigences à satisfaire pour son implémentation. Étant donné que l'un des principes de base de cette approche est d'entrelacer la génération du plan avec le processus d'exécution, nous présentons dans la sous-Section 6.1.2 une brève revue de littérature de cette technique. La dernière sous-section par contre, révèle certaines des limites de la DCP actuelle.

6.1.1. Planification Distribuée en Continu : synthèse

6.1.1.1. Planification distribuée

Dans la communauté de planification, les chercheurs [Desjardins et al., 99] font souvent la distinction entre deux approches de planification distribuées : la planification distribuée coopérative (*Cooperative Distributed Planning* (CDP)) et la planification distribuée négociée (*Negotiated Distributed Planning* (NDP)). Afin de concevoir un système de planification coopérative distribuée, DesJardins et ses collègues [DesJardins et al., 99] affirment que nous devons prendre en compte six points essentiels :

- Comment générer et représenter un plan (les méthodes) ?
- Comment allouer les tâches ?
- Comment assurer la communication entre les agents ?
- Comment permettre aux différents agents de se coordonner ?
- Comment collaborer ?
- Comment négocier ?

Étant donné que l'aspect de distribution n'est pas traité dans cette thèse, nous nous donnons pas plus de détails ici.

6.1.1.2. Planification en Continu : synthèse

Un agent agissant dans un monde dynamique doit continuellement être sur ses gardes vis-à-vis des changements qui peuvent affecter son plan. Puisque le plan doit être réévalué et révisé de façon continue, l'agent doit entrelacer les deux processus de planification et d'exécution.

La planification réactive [Agree et Chapman 87] pourrait être vue comme un cas particulier de planification en continu, puisque le processus de planification ne se soucie que de la prochaine action à accomplir.

Quelques systèmes à exécution flexible comme PRS [Georgeff et Lansky 86] et RAPS [Firby 87], entrelacent le processus de raffinement de plan avec le processus d'exécution. Ce raffinement est différé le plus possible afin de permettre à l'agent de prendre des décisions avec le maximum de certitude. Le problème d'une telle approche est que le raffinement de bas niveau peut générer des conflits insolubles durant l'exécution, ce qui entraîne un échec total de l'ensemble du processus.

Le framework *Continuous Planning and Execution Framework* (CPEF) [Myers 99] est un système typique de planification en continu des années 90.

Que faut-il pour assurer la continuité? Selon [Pollack et Horty 99], un agent planificateur doit être capable d'accomplir au moins les tâches suivantes quand il agit dans un monde dynamique.

Génération de plan

L'agent doit certainement faire de la génération classique de plan.

Gestion des engagements

L'un des gros problèmes est de savoir comment gérer les nouveaux engagements¹⁷ (apprécier la pertinence d'un nouvel engagement, vérifier l'existence de conflits avec les engagements précédents, etc.).

Une solution possible est donnée par le *Intelligent Resource-Bounded Architecture* (IRMA) [Pollack et Horty 99]. Dans cette architecture, l'agent est doté d'un mécanisme de filtrage à deux composantes. Le principe de ce mécanisme s'organise comme suit :

Quand l'environnement impose une nouvelle option à l'agent, la première composante, appelée *Compatibility check*, vérifie la compatibilité de la nouvelle option avec le plan en cours. La deuxième composante, appelée *Override mechanism*, vérifie si cette nouvelle option est suffisamment importante pour nécessiter une délibération, même sous une incompatibilité. Ceci évite évidemment de perdre du temps à délibérer inutilement pour chaque nouvelle option.

¹⁷ Un agent s'engageant pour un plan P, tend à exclure toute action en conflit avec P [Pollack et Horty 99]

D'autres solutions existent, comme l'architecture dMars proposée par Rao et Georgeff [Rao et Georgeff 95].

Surveillance de l'environnement

L'un des premiers travaux dans ce domaine est *Triangle Tables* [Fikes et al., 72]. Knoblock [Knoblock 95] a développé beaucoup plus tard le système SAGE pour la collecte d'informations, Les décisions de SAGE sont basées sur ce qui est à surveiller dans la structure du plan.

Plusieurs efforts de recherche ont mené au processus *Rational-based Monitoring* [Pollack et Horty 99], qui est un processus en trois étapes :

- Génération des moniteurs [Pollack et Horty 99], ce qui représente les caractéristiques environnementales qui affectent le plan.
- Délibération (pour décider s'il est approprié de répondre au moniteur ou non) aussitôt que le moniteur envoie sa notification.
- Transformation du plan (si l'agent décide de répondre)

Le *Rational-based Monitoring* a été utilisé dans plusieurs travaux. Veloso et ses collègues [Veloso et al., 95] ont ainsi proposé le modèle PRODIGY de surveillance et Pollack et McCarthy [Pollack et McCarthy 99] ont conçu le planificateur *Partial-Order Causal-Link Planner*.

Néanmoins, d'autres travaux affirment qu'un agent ne devrait pas seulement surveiller l'environnement mais aussi garder trace des plans rejetés. En effet, sauvegarder les décisions antérieures (des plans rejetés) pourrait aider l'agent lors de ses futures prises de décisions.

Malgré ces travaux de recherche, les approches de surveillance (*monitoring*) demeurent incomplètes. En effet, il serait préférable que l'agent puisse surveiller seulement ce qui est requis et quand c'est nécessaire.

Évaluations alternatives

Quand une nouvelle option survient, un agent doit évaluer les coûts et les bénéfices possibles de cette nouvelle option. Parmi les théories disponibles qui traitent ce problème, nous pouvons nommer *Rational Choice Theory* (dans l'économie et la philosophie) basée sur la maximisation de l'utilité [Keeny et Raiffa 93].

Élaboration de plan

Dans un monde dynamique, la planification et l'exécution doivent être entrelacées [Pollack et Horty 99]. Les agents doivent alors décider du niveau de détail qui doit être inclus dans leurs plans et quand ils doivent chercher ces détails. Cette question a été longtemps ignorée. Onder et Pollack [Onder et Pollack 99] ont exploré en partie ce problème.

Si un grand nombre de contingences peuvent apparaître durant l'exécution du plan dans un monde dynamique et incertain, il est important de déterminer quelles contingences doivent être traitées durant l'élaboration du plan initial et quelles autres peuvent être considérées plus tard (si nécessaire). Afin de résoudre ce problème, chaque contingence est associée à une action qui peut être potentiellement en échec. En général, quand un agent détecte que son plan inclut une action qui pourrait échouer, il peut agir en conséquence selon trois approches :

- Réparation corrective : nécessite la planification d'une solution en cas d'échec.
- Réparation préventive : nécessite l'amélioration du plan en incluant des étapes qui réduisent le risque d'échec.
- Remplacement : nécessite la modification du plan en remplaçant les actions problématiques par des alternatives.

Afin de déterminer sur quelles contingences il faut se concentrer, deux principaux critères peuvent être pris en compte :

- La probabilité d'occurrence de la contingence

- Les coûts de réparation et l'impact sur le plan.

Coordination

Quand différents agents doivent accomplir des activités dans un environnement commun, ils doivent coordonner leurs plans. Beaucoup de travaux de recherche traitent ce problème de coordination comme dans [Grosz et al., 99].

6.1.1.5. Récapitulation et orientations du chapitre

D'après cette petite synthèse, nous remarquons que les voies de recherches dans la DCP sont diverses : génération et représentation du plan, allocation des tâches, communication entre les agents, coordination des agents, collaboration, négociation, gestion des engagements, surveillance de l'environnement, évaluations alternatives, élaboration de plan, etc. Comme il a été déjà avancé dans les objectifs de la thèse (Chapitre 1), nous ne nous intéressons qu'à quelques aspects spécifiques de la planification distribuée en continu. Nous avons déjà abordé quelques uns de ces aspects dans les chapitres précédents : la communication entre agents et la surveillance de l'environnement (discutés de façon générale dans le Chapitre 4), et la génération du plan et l'évaluation des alternatives (présentés dans le Chapitre 5 ; le plan dans ce cas est en fait un chemin). Dans le présent chapitre, nous mettons plutôt l'emphase sur un autre aspect très caractéristique de l'approche DCP : le processus de continuité, et plus précisément sur l'entrelacement de la planification et de l'exécution. En effet, cet entrelacement entre les deux processus caractérise de façon nette l'approche DCP par rapport aux autres approches. C'est surtout à ce niveau précis de la théorie sur la DCP que notre travail de recherche se situe. Dans la sous-section suivante, nous présentons un petit survol bibliographique sur l'évolution de cette idée d'entrelacement de la planification et de l'exécution.

6.1.2. Entrelacement de la planification et de l'exécution dans un environnement dynamique et incertain

Dans un environnement dynamique, les longs plans élaborés de façon classique ne sont pas suffisants pour un agent autonome à cause de tous les changements qui peuvent survenir sur l'environnement. Dans de tels cas, le plan doit être modifié ou même reconstruit. La reconstruction ou la modification d'un plan est une tâche fastidieuse, puisque ce plan peut ne pas être applicable aussitôt que l'état des ressources requises change [Au et Parameswaran 98]. Dans un système multiagent le problème est encore plus important. En effet, à cause de l'aspect distribué de la connaissance et les différences de capacités entre les agents, un agent souvent ne peut pas décider de la prochaine action sans échange d'information avec d'autres agents. Ceci n'est qu'un exemple d'un problème plus générique : un agent peut réaliser qu'il a besoin d'autres agents pour atteindre ses objectifs. Il peut déléguer des tâches à d'autres agents ou même former un groupe et élaborer ensuite un plan commun.

Dans un système multiagent, il n'est pas pratique de garder la distinction entre les processus d'exécution (l'agent construit le plan en entier avant d'agir). En effet, dans un environnement dynamique, ce qu'un agent considère comme vrai, peut être faux après que d'autres agents aient exécuté leurs actions.

En particulier, la planification classique présente un sérieux problème pour les applications quand le temps est critique [Jonsson et Backstorm 95]. D'une part, nous devons attendre le planificateur jusqu'à ce qu'il génère le plan en entier avant que nous puissions commencer l'exécution. D'autre part, quand l'exécution d'une action échoue ou que d'autres changements surviennent sur l'environnement, nous devons invoquer le planificateur afin de trouver un nouveau plan qui convient. Cette approche (replanification) est tellement coûteuse qu'elle peut être considérée comme infaisable pour une application dans laquelle le temps d'exécution est critique [Jonsson et Backstrom 95].

La planification réactive peut être considérée dans certains cas comme une meilleure alternative. Toutefois, du point de vue coût de traitement, un planificateur réactif n'est

pas plus efficace qu'un planificateur classique (basé sur la planification et la replanification), surtout si le problème est complexe [Jonsson et Backstrom 95].

En séparant la « planification et l'exécution », l'agent risque de voir son plan devenir inapplicable. Ambros-Ingerson et Steel [Ambros-Ingerson et Steel 88] proposent alors une nouvelle approche de planification : entrelacer le processus de planification et le processus d'exécution. L'idée consiste à concevoir un planificateur qui est capable de générer une séquence d'actions (appelées préfixes), en vue d'une éventuelle solution jusqu'à ce qu'il ne puisse plus planifier. En conséquence, les actions peuvent être immédiatement exécutées tandis que le planificateur essaye de produire les prochaines actions exécutables. Ambros-Ingerson et Steel [Ambros-Ingerson et Steel 88] notent deux grands avantages pour cette approche :

- Même si la génération d'une solution complète prend plus de temps, nous pouvons toujours commencer l'exécution des actions préfixes relativement tôt. Cette flexibilité est parfois très importante pour certaines applications où le temps est critique : Nous sommes parfois contraints à débiter l'exécution du plan aussitôt que possible même s'il n'est pas complet. A titre d'exemple, dans un contexte de feu de forêt, il est souvent fondamental d'agir vite (p.ex. évacuer d'urgence une zone menacée par le feu dans l'immédiat) même si nous ne savons pas ce qu'il faut faire après (p.ex. où se réfugier après l'évacuation).
- Dans une approche à base d'entrelacement, si nous sommes contraints à replanifier, nous devons juste attendre les préfixes exécutables du nouveau plan avant de continuer l'exécution. Nous ne perdons ainsi –pour chaque échec d'exécution- pas plus que le temps que le planificateur mettrait pour chercher les préfixes du nouveau plan. Nous évitons par conséquent les problèmes de la replanification classique.

Dans [Ambros-Ingerson et Steel 88], les auteurs ont appelé leur approche par entrelacement, planification incrémentale. Depuis ce temps, plusieurs chercheurs ont exploré le problème d'entrelacement [Dean et al., 90, Georgeff et Ingrand 89, Nourbakhsh 97]. Par exemple, Stone et Veloso [Stone et Veloso 95] ont exploré de

nouvelles méthodes afin d'améliorer l'algorithme de planification robotique PRODIGY [Carbonell et al., 92]. Dans ce sens, ils ont réduit le temps total de planification et d'exécution lorsque ces deux processus sont entrelacés, et ils ont permis aux informations collectées pendant le processus d'exécution d'être ajoutées à la connaissance du planificateur.

Récemment, Paolucci et ses collègues [Paolucci et al., 00] ont proposé un *Hierarchical Task Network Planner* (HITaP) qui entrelace la planification et l'exécution dans un système multiagent. Le principe est le suivant : L'agent construit son plan jusqu'à ce qu'il se rende compte qu'il a besoin d'information ou de synchronisation avec d'autres agents. L'agent suspend alors sa planification sur le moment et exécute les actions qui vont mener à trouver les informations manquantes ou à compléter la tâche qu'il ne peut pas accomplir par lui-même.

Bien que la planification par entrelacement résolve les problèmes de la planification classique, elle présente certains inconvénients. Jonsson et Backstorm [Jonsson et Backstorm 95] ont souligné certains de ces points. En effet, avec la planification incrémentale, il n'y a aucune garantie que le planificateur puisse générer des préfixes à exécuter avant d'arriver à la solution finale. Pire encore, avant d'arriver à la fin du plan, l'existence d'une solution finale n'est jamais assurée. Exécuter des préfixes n'est en fait qu'un essai vers une éventuelle solution. Jonsson et Backstorm ont alors proposé une solution restrictive qui supporte certains types de problèmes appelés *3S class*. Pour de tels problèmes, il est possible de connaître à l'avance si le problème admet ou non une solution finale.

Un problème complexe relié à l'entrelacement de la planification et de l'exécution est de décider quand et comment entrelacer. Parmi les planificateurs qui supportent l'entrelacement, IPeM [Ambros-Ingerson et Steel 88], XII [Golden et al., 94] et SAGE [Knoblock 95] utilisent la même technique d'entrelacement (de base) : les actions ne sont exécutées que lorsque la planification devient impossible (à cause du manque d'information par exemple). Une telle technique ne permet pas de décider à l'avance quand exactement la planification doit être reprise. D'autres stratégies d'entrelacement ont été étudiées par Olawsky et Gini [Olawsky et Gini 90] lors de la conception de

BUMP. Par exemple, avec la stratégie « *Continue Elsewhere* », le maximum de la planification est fait avant de commencer l'exécution. Avec la stratégie « *Stop and Execute* », l'exécution d'un but –défini en termes de résultat du senseur- est lancée aussitôt que possible. Aucune de ces stratégies n'a un net avantage sur l'autre. Les deux produisent des plans non optimaux ou parfois même des plans qui vont échouer.

6.1.3. Limites de la DCP actuelle

La planification en continu semble inévitable dans un environnement réel, complexe et dynamique. Toutefois, dans son état actuel, elle présente quelques inconvénients que nous avons déjà discutés dans le Chapitre 1. Nous rappelons brièvement ces limites :

- Les plans sont construits étape par étape sans aucune garantie quant à l'aboutissement du plan à une solution finale acceptable. Cette approche est inacceptable surtout pour les situations critiques telles que les champs de bataille ou les feux de forêts car elle peut emmener le plan vers une impasse. Le plan est alors condamné à l'échec alors qu'on en a peut-être déjà exécuté une bonne partie.
- L'entrelacement de la planification et de l'exécution à améliorer : Dans une approche DCP classique, nous sommes contraints à attendre l'exécution d'une étape du plan avant de pouvoir entamer la planification de l'étape suivante. Bien que dans certains cas, nous n'avons pas d'autres alternatives, il nous semble que cet aspect pourrait très bien être amélioré pour certains types d'applications.
- Même si nous nous intéressons pas vraiment à la distribution de la planification, il est intéressant de noter que lorsque nous avons une telle distribution le flux de communication distante est souvent important. Or, l'environnement ne peut pas toujours assurer une communication fiable et à grande bande passante. L'approche DCP basée en général sur des communications distantes entre les différents planificateurs doit donc absolument trouver des moyens efficaces pour réduire le flux de ces communications.

Étant donné que notre approche se base sur la simulation multiagent pour implémenter la DCP, nous allons démontrer que l'utilisation de cette technique (la simulation multiagent) peut résoudre la plupart des problèmes cités ci-dessus et surtout améliorer le processus d'entrelacement.

6.2. Implémentation de la DCP par un SMA dans un environnement géospatial de simulation

Dans cette section, nous présentons une façon d'implémenter l'approche DCP par simulation multiagent (ou orientée agent). Cette technique devrait résoudre certains des problèmes actuels de la DCP. Notre approche est illustrée dans un premier temps sur le problème des feux de forêt (sous-Section 6.2.2) avant d'être présentée de façon plus générique dans la sous-Section 6.2.3.

6.2.1. Problème à résoudre

6.2.1.1. Scénario

Lors de la lutte contre un incendie de forêt, le responsable des opérations (le CP) opte souvent pour une stratégie indirecte (surtout quand le feu est de grande ampleur, Chapitre 3 Section 3.3.3). Cette stratégie consiste à construire une ligne d'arrêt pour encercler le feu. Nous avons mentionné dans un chapitre antérieur (Chapitre 3, Section 3.3.3.5) que la construction d'une ligne d'arrêt débute souvent à partir d'un point d'ancrage (une barrière naturelle, une autoroute, etc.) pour minimiser les pertes d'espace forestier et aussi pour accélérer la construction de la ligne.

Le responsable des opérations a suffisamment d'expérience pour choisir les bons points d'ancrage. Il est nécessaire cependant de savoir s'il est possible de construire une ligne à partir du point d'ancrage en question dans les délais impartis (avant l'arrivée du feu). Plusieurs outils commerciaux de gestion des feux peuvent assister le responsable en lui indiquant les prévisions à long terme de la propagation des feux et en calculant approximativement le temps nécessaire aux bulldozers pour construire la ligne.

Dans le chapitre 3, nous avons également constaté que le choix de la trajectoire de la ligne est très important et que plusieurs contraintes doivent être prises en compte :

- Minimiser l'espace sacrifié.
- Éviter les courbes aiguës.
- Éviter des passages difficiles pour les bulldozers.
- Éviter de passer dans un espace à forte densité d'arbres.
- Tenir compte de l'avancée des feux, etc.

Même si le responsable des opérations est capable de donner l'allure générale de la ligne d'arrêt à construire, il n'est toutefois pas en mesure de donner aux bulldozers les détails du chemin à suivre. Pour pouvoir donner un chemin précis, le responsable des opérations (planificateur humain) a besoin d'avoir une connaissance détaillée de la topographie de la région. Il est aussi difficile pour un être humain de vérifier toutes les possibilités en un temps raisonnable tout en respectant les contraintes citées ci-dessus.

Il s'agit donc d'un problème de recherche de chemin (Pathfinding) qui prend en compte la topographie de l'espace et qui est influencée par la situation (avancée des feux). Dans ce chapitre, nous ne mettons pas l'accent sur les détails du processus de Pathfinding (le chapitre 5 est consacré à cet aspect là), mais plutôt sur l'application des principes de la DCP (surtout l'entrelacement de la planification avec le processus d'exécution).

Dans la stratégie indirecte, une fois le chemin des différents bulldozers planifié, il est très probable que l'évolution de la situation empêche ces bulldozers de suivre le plan tracé. En effet, plusieurs imprévus peuvent entraver le bon déroulement de l'exécution du plan. Nous allons citer quelques problèmes qui peuvent survenir :

- Le premier tracé de la ligne d'arrêt est basé sur des prédictions de la propagation des feux ; or l'évolution de l'incendie peut changer de façon imprévue et rapide : changement météorologique (changement dans la direction ou vitesse du vent, etc.), mauvaise estimation des essences présentes, etc.

- Un bulldozer peut tomber en panne ou rencontrer des problèmes sur le terrain, ce qui l'empêche de suivre son plan.
- Le responsable des opérations peut estimer l'évolution de la situation très dangereuse et décider de changer de stratégie en conséquence.

Quant à la stratégie directe, plusieurs imprévus peuvent survenir également et contraindre les différents intervenants (les pompiers) à changer leurs plans. Parmi ces événements :

- Une avancée du feu plus rapide (ou plus lente) que prévue vers la ressource en péril.
- Un camion de pompier en panne.
- Ressources (en eau) insuffisantes une fois sur les lieux, ce qui nécessite donc la demande de renfort.

Nous constatons ainsi que la planification, dans le cas d'une stratégie directe ou indirecte, n'est pas une tâche facile en raison des changements qui peuvent survenir durant l'exécution et même parfois durant le processus de planification même.

Nous nous proposons de trouver un chemin que les bulldozers doivent suivre pour creuser la ligne d'arrêt lors d'une attaque indirecte, de faire le suivi du plan pendant son exécution effective et de gérer les événements imprévus.

6.2.1.2. Hypothèses

Le responsable des opérations possède suffisamment d'expertise et d'informations pour choisir un point d'ancrage à partir duquel la ligne d'arrêt va se construire.

Ce responsable a les moyens (par sa connaissance du domaine et son expérience personnelle ou bien par l'aide d'outils logiciels) de définir un plan stratégique (la stratégie à utiliser ici est la stratégie indirecte.) et global (le nombre de bulldozers nécessaires dans le cas d'une stratégie indirecte, etc.) de l'opération. Cette expertise se traduit, entre autres, par la proposition d'une esquisse de ce que pourrait être la ligne d'arrêt (voir le Chapitre 5 pour plus de détails).

Les bulldozers ainsi que les camions des pompiers sont équipés d'un dispositif GPS qui leur permet de suivre exactement sur le terrain un chemin spécifié en coordonnées géospatiales. Cette technologie existe déjà et est appliquée dans le domaine de l'agriculture de précision [O'Connor 96; Marchant et al., 97; Zhu et al., 98].

Ces bulldozers et ces camions sont également équipés d'un dispositif de communication GPS qui assure la liaison avec le centre de commandement.

6.2.2. Solution sur scénario : Planification de la construction de la ligne d'arrêt

Afin de répondre aux besoins du problème de construction de la ligne d'arrêt des feux, nous adoptons une approche de planification à base d'agents logiciels.

Nous illustrons la solution sur le problème énoncé à la sous-section précédente 6.2.1.1. Nous supposons que le responsable des opérations a dessiné une esquisse (qui part d'un point d'ancrage) qui représente l'allure générale de la ligne d'arrêt à construire et qu'il a ordonné le déploiement d'un bulldozer pour cette tâche (nous considérons un scénario à un seul bulldozer afin de simplifier l'exemple et ne pas avoir à trop détailler les contraintes très particulières du domaine). La Figure 6.1 illustre le scénario de départ. Les prochaines sous-sections décrivent les différentes étapes du processus de planification de la ligne d'arrêt selon notre approche basée sur la simulation multiagent.

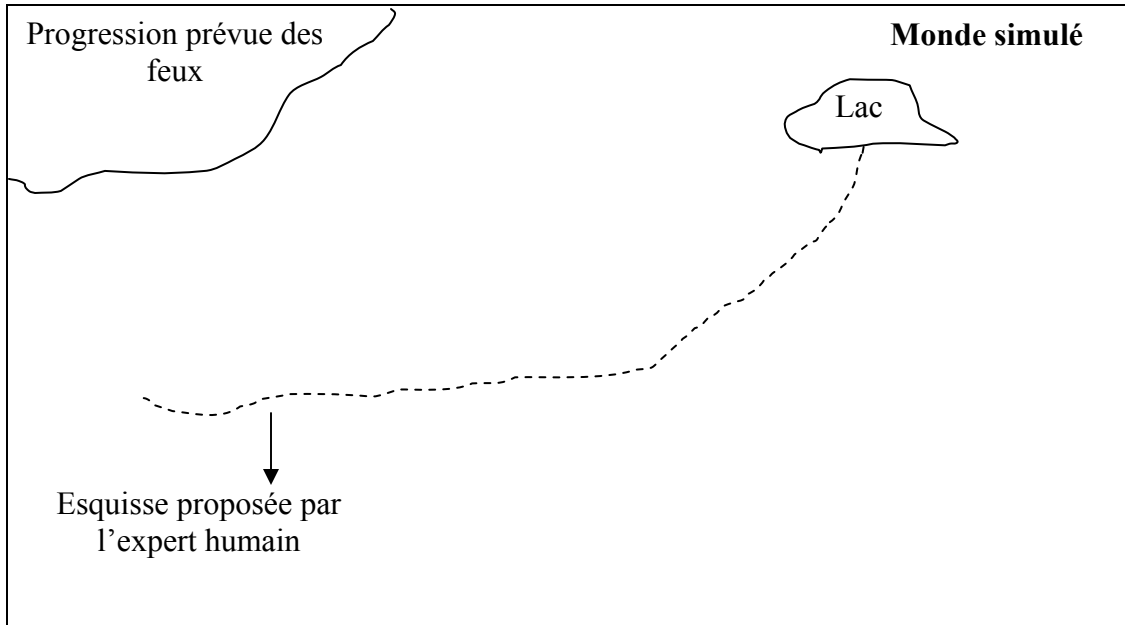


Figure 6.1. L'expert humain propose une esquisse de la ligne d'arrêt dans le monde simulé

6.2.2.1.1. Recherche d'un plan global

Le bulldozer sur le terrain a besoin d'un plan détaillé pour construire la ligne d'arrêt. Une planification doit se faire alors au niveau de l'environnement de simulation où la connaissance spatiale est disponible. Un agent *AAS* (*Agent Acteur de Simulation*, Chapitre 4) à qui nous donnons ici un nom plus spécifique : *AAS-Bulldozer* est chargé donc de représenter le bulldozer réel au niveau de l'environnement de simulation. La planification (effectuée par des *Agents Pathfinder*¹⁸, Chapitre 5) se fait pendant que le bulldozer réel est en train de rejoindre le point de départ spécifié par le responsable des opérations.

Les *Agents Pathfinder* trouvent alors un plan (un chemin) qui suit au mieux l'esquisse proposée par l'expert humain, mais qui de plus, tient compte des caractéristiques du

¹⁸ L'*Agent de Simulation* qui se trouve dans l'environnement de simulation à la position correspondante à la position courante du bulldozer est appelé *AAS-Bulldozer*. Ce dernier déploie des *Agents Pathfinder* pour lui chercher un chemin approprié. Ces différentes appellations sont faites dans ce chapitre pour des raisons de simplification afin de distinguer les deux fonctionnalités de l'agent planificateur.

terrain et de l'avancée des feux (Figure 6.2). Le meilleur chemin global trouvé est retenu. La génération de ce plan a fait l'objet du chapitre précédent (Chapitre 5).

Nous nous sommes permis ici de faire un plan global de l'opération car nous disposons suffisamment de temps pour le faire (le bulldozer est en train de rejoindre la zone de départ). Même si ce plan va possiblement échouer, il aura donné une idée globale sur le déroulement de l'opération. Comme nous allons le voir dans la prochaine section, ce plan global peut aider le planificateur humain à choisir sa stratégie globale.

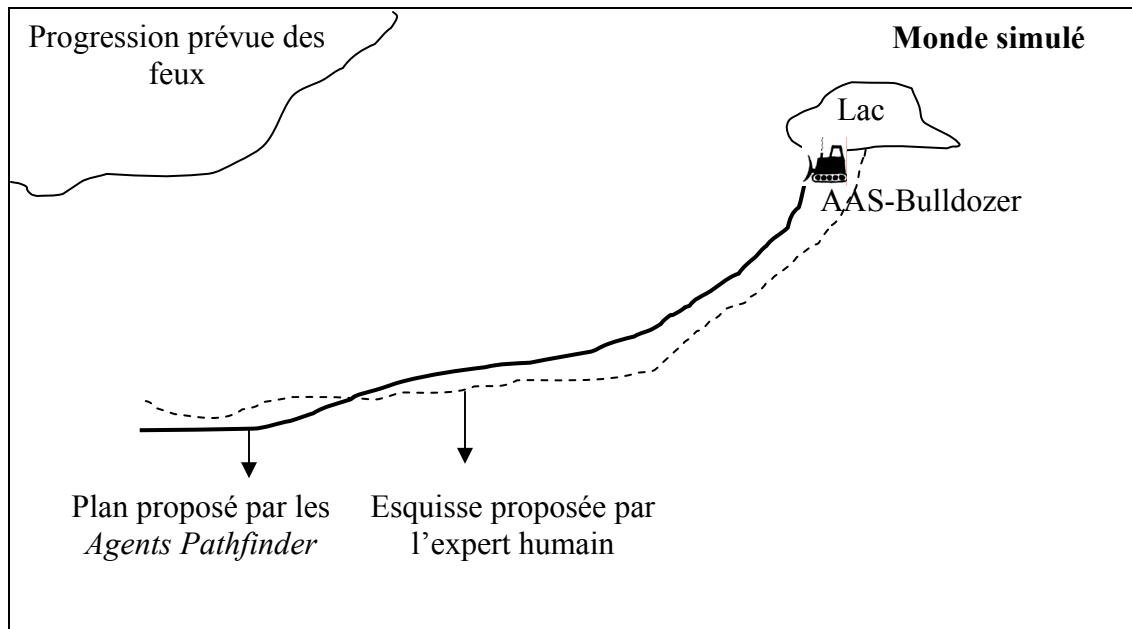


Figure 6.2. L'agent *AAS-Bulldozer* (via ses *Agents Pathfinder*) trouve un meilleur plan global.

6.2.2.1.2. Test et validation des plans

Lorsqu'il s'agit de planifier dans un environnement aussi chaotique, complexe et risqué qu'un incendie de forêt, il est important que les plans élaborés soient validés et vérifiés par des experts humains (les responsables des opérations). Les agents logiciels ne peuvent pas garantir l'efficacité du plan, même s'ils opèrent dans un milieu à connaissance augmentée (environnement de simulation). L'expertise, le jugement et l'anticipation des humains sont donc indispensables.

Le responsable des opérations devrait donc pouvoir participer à la planification en se basant sur son expérience dans le domaine et sur ses capacités cognitives (anticipation, jugement, etc.) qui lui permettent d'apporter un plus à la planification logicielle.

Nous pensons donc qu'un agent humain doit toujours vérifier et valider les plans élaborés par des agents logiciels, surtout lorsque les effets de ces plans peuvent être très critiques (p.ex. présenter des risques pour des vies humaines).

Dans notre problème de lutte contre les feux de forêt, afin de valider et vérifier un plan par un agent humain (responsable des opérations), une simulation de l'exécution du plan doit être faite. Étant donné que le problème est très complexe et à connotation spatiale, nous pensons qu'une simulation visuelle et spatiale dans l'environnement de simulation est mieux adaptée qu'une simple simulation basée seulement sur des calculs. Ceci permet au responsable des opérations de visualiser l'exécution du plan dans un espace semblable à celui du terrain. Il pourrait ainsi relever des anomalies, juger de la faisabilité du plan et anticiper les problèmes que les agents n'ont pas pu relever.

Dans l'exemple de la construction de la ligne d'arrêt, rien ne garantit que le premier plan optimal trouvé par les *Agents Pathfinder* du *AAS-Bulldozer* serait efficace et faisable. En effet, le responsable des opérations, en visualisant en mode simulation accélérée le plan établi par les agents (il visualise en fait l'avancée des feux en même temps que la simulation de la construction de la ligne d'arrêt), pourrait juger le plan très risqué (en se basant sur son expérience personnelle) ou l'espace sacrifié trop important. Le planificateur humain devrait alors tester une autre stratégie, par exemple, il peut appeler un deuxième bulldozer en renfort.

Nous supposons pour la suite de notre scénario, que le responsable des opérations a fini par approuver la première stratégie (à un Bulldozer) car il n'est pas arrivé à trouver un deuxième bulldozer disponible pour l'opération.

6.2.2.1.3. Exécution (réelle) du plan et replanification périodique anticipée

Le premier plan global trouvé par les *Agents Pathfinder* étant approuvé par le planificateur humain, l'*AAS-Bulldozer* communique le plan au *AAR-Bulldozer*

correspondant (c'est l'agent logiciel implanté dans le bulldozer réel). Le bulldozer commence alors son avancée sur le terrain selon le plan établi. Au fur et à mesure que le bulldozer avance sur le terrain réel, son représentant *AAS-Bulldozer* se déplace sur l'environnement de simulation (Chapitre 4). Il est clair que l'agent *AAR-Bulldozer* doit communiquer périodiquement sa position par GPS au *AAS-Bulldozer*.

Il est toutefois important de rappeler que le plan global en question a été construit sur la base de données incertaines. En effet, les agents planificateurs (*Agents Pathfinder*) ont cherché un chemin selon des prévisions de progression des feux. Ces prévisions ne correspondent presque jamais exactement à ce qui va se passer réellement sur le terrain. Ceci est dû à plusieurs facteurs : imprécision des prévisions météorologiques à long terme, les erreurs du modèle physique de propagation des feux, incertitude sur la connaissance des données relatives au terrain, etc. Le plan a été également bâti selon des approximations sur la progression des bulldozers sur le terrain. Bien entendu, ces approximations ne peuvent pas toujours correspondre exactement à ce qui se passe sur le terrain.

Il est ainsi nécessaire de maintenir la validité du plan durant son exécution. Pour ce faire, nous proposons une replanification périodique anticipée, périodique parce qu'elle devrait se faire régulièrement après un certain temps d'exécution du plan et anticipée parce qu'elle n'attend pas que le plan en cours échoue pour qu'elle soit déclenchée.

La *replanification périodique anticipée* est une planification en continu et dont le principe est le suivant : Lors de l'exécution du plan (déplacement effectif du bulldozer) le *AAS-Bulldozer* suit dans l'environnement de simulation le mouvement de *AAR-Bulldozer* sur le terrain. Après chaque période de temps (déterminée par les experts du domaine d'après l'expérience), le *AAS-Bulldozer* (via ses *Agents Pathfinder*) lance une replanification du chemin qui reste à faire (toujours selon le même principe présenté dans le Chapitre 5) à partir du point (point A1¹⁹ de la Figure 6.3) que le bulldozer va atteindre bientôt (conformément au plan de départ). Pendant que *AAR-Bulldozer* se dirige vers le

¹⁹ Le point A1 pourrait être très proche physiquement de la position courante du bulldozer dépendamment de la productivité du bulldozer (sa vitesse d'avancement) et le temps nécessaire au planificateur humain d'approuver un nouveau plan.

point A1, *AAS-Bulldozer* doit avoir suffisamment de temps pour trouver un chemin optimal à partir de A1. Si le nouveau chemin trouvé est le même que l'ancien, aucune nouvelle révision n'est faite. Si par contre ce nouveau plan est différent (cela veut dire que durant la dernière période de temps, la situation n'a pas évolué comme prévu), le *AAR-Bulldozer* est avisé du changement du plan et en tient compte une fois arrivé au point A1. Le même processus est répété pour chaque période de temps jusqu'à la fin du plan.

La période qui sépare deux replanifications est nécessaire pour que les agents planificateurs mettent à jour les données dont ils se servent pour construire les plans telles que :

- La vitesse et la progression effectives des bulldozers sur le terrain.
- L'état courant des feux (cette mise à jour est obtenue à partir des nouvelles images satellite, des senseurs et des constats rapportés par les témoins sur le terrain).
- Les nouvelles données météorologiques courantes et les dernières prévisions. En fait, les prévisions à court terme sont souvent beaucoup plus précises et moins incertaines que celles à long terme – qui ont servi pour construire le premier plan global.

Cette replanification périodique a donc pour but d'augmenter les chances de réussir l'opération globale en améliorant régulièrement le plan selon les données les plus récentes. Elle anticipe ainsi les échecs de plan avant que cela n'arrive. Les nouveaux plans (si jamais ils sont différents des anciens) sont également validés par le planificateur humain, ce qui permet encore une fois une meilleure anticipation des éventuels problèmes. Il est important de noter que ces replanifications périodiques ne tardent en aucun cas l'exécution du plan puisqu'elles sont faites en parallèle avec l'exécution et dans un monde de simulation.

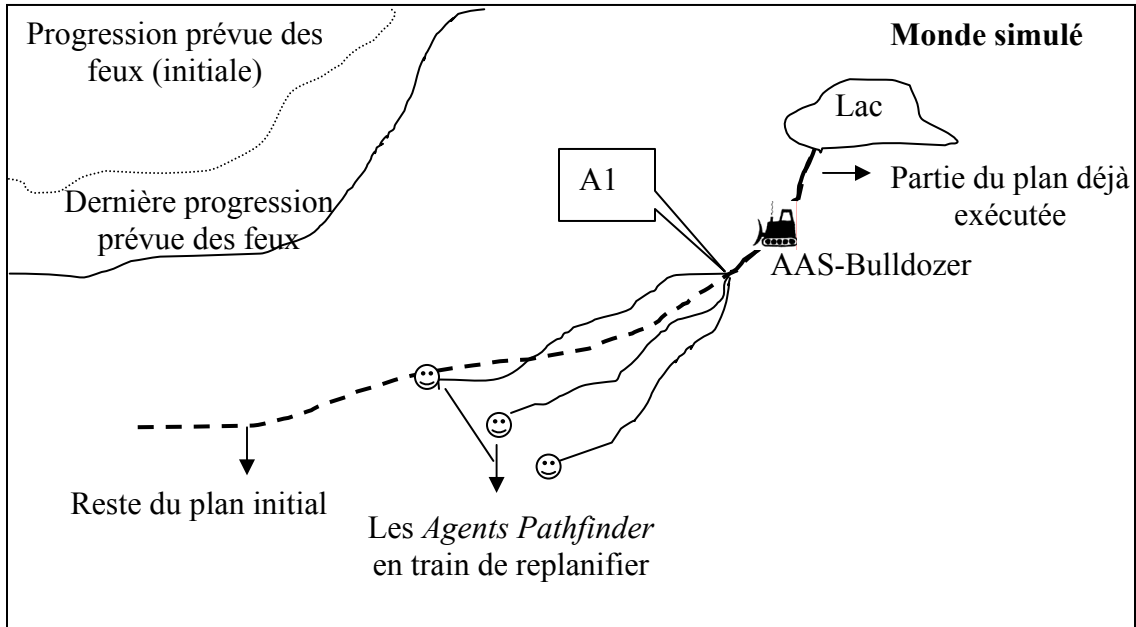


Figure 6.3. Illustration du principe de la replanification périodique anticipée

6.2.2.1.4. Replanification en cas d'événement imprévu

Même si le plan global est régulièrement mis à jour et même si plusieurs problèmes peuvent être anticipés, un imprévu peut toujours survenir à n'importe quel moment (p.ex. pendant la période entre deux replanifications). En fait, comme nous l'avons mentionné dans 3.1.1, rien ne garantit que ce plan va rester valable jusqu'à la fin ou même jusqu'à la prochaine replanification périodique : plusieurs imprévus peuvent survenir. L'environnement de simulation doit alors assurer la réparation du plan, et ce le plus vite possible.

Supposant que dans notre scénario, le bulldozer a déjà fait environ le 1/3 de la ligne lorsqu'un nouveau feu (causé par les spots de feu par exemple) vient d'être repéré à quelques kilomètres devant le front du premier feu. Ce changement affecte bien sûr les prédictions sur la progression des feux. Si le bulldozer continue sur le même plan, les feux risquent probablement de dépasser la ligne d'arrêt avant qu'elle ne soit terminée, et plus le bulldozer tarde à changer de plan, plus la situation peut devenir critique et dangereuse.

Le plan (le reste de la ligne à construire) doit donc être modifié. Pour ce faire, une replanification se fait dans l'environnement de simulation comme dans le cas de la *replanification périodique anticipée*, mais cette fois à partir de la position courante du bulldozer (réel). Il arrive même que le bulldozer change de trajectoire sur le terrain (devant un danger par exemple) et ne se conforme pas au plan déjà établi. Une replanification doit alors être reprise à partir de la nouvelle position du bulldozer.

6.2.3. Généralisation de la solution : Une DCP spatiale basée sur des agents et à exécution anticipée en mode simulation

Dans cette section, nous allons généraliser l'approche et synthétiser l'apport scientifique derrière les solutions proposées initialement pour le domaine de lutte contre les feux.

6.2.3.1. DCP spatiale à base d'agents logiciels dans un environnement simulé

Dans un environnement spatial et dynamique, l'approche DCP est inévitable. Pour implémenter la DCP nous proposons une approche par agents logiciels. Les principes de cette planification sont illustrés par les points suivants :

- *Agent de Simulation versus Agent Réel*

Les acteurs actifs (les planificateurs) dans le monde réel ainsi que certains acteurs passifs (p.ex. une infrastructure) sont équipés d'agents logiciels appelés Agents Réels qui ont pour principal rôle d'assurer le lien avec le monde de simulation. Ces agents logiciels ne planifient donc pas ; ils ont plutôt un rôle d'interface entre le monde réel et le monde de simulation.

Ces agents réels ont des correspondants dans le monde de simulation, qui eux sont des planificateurs. Ils disposent en effet de plus de connaissance (sur les autres agents et sur l'espace qui les contient) et sont donc capables de mieux planifier.

- Planification par exploration spatiale

Afin de pouvoir raisonner sur l'espace, l'agent planificateur de simulation (p.ex. *Agent Pathfinder*) a besoin de connaître l'espace qui l'entoure. Si de telles connaissances ne sont pas déjà à la disposition de l'agent (via l'interaction avec d'autres agents par exemple), ce dernier doit les chercher en explorant l'espace qui l'entoure. Pour ce faire, il fait appel à ses capacités de perception, de mémoire et de navigation. Nous avons ainsi des agents plus sensibles à l'espace.

- Planification en continu

Lorsqu'il s'agit de planifier dans un environnement réel dynamique et complexe, la planification en continu est inévitable.

Cependant, la planification en continu proposée jusqu'à présent par les chercheurs, présente quelques sérieux problèmes (discutés précédemment) auxquels nous espérons avoir apporté quelques améliorations. Nous proposons donc une nouvelle façon d'élaborer le plan et plus particulièrement, une nouvelle manière d'entrelacer les processus de planification et d'exécution. L'idée de l'approche par *Replanification Périodique Anticipée* se résume en les étapes suivantes :

- **Génération d'un plan initial global** : Bien que ce plan soit basé sur des hypothèses et sur des informations incertaines, il permet de donner une vue globale du plan final et éviter ainsi des débuts de plan catastrophiques. En effet, si nous appliquons une planification bout par bout au problème de construction de la ligne d'arrêt, un scénario catastrophe serait très probable : Le début du chemin proposé par les agents planificateurs pourrait par exemple mener les bulldozers vers une impasse, ce qui mettrait en échec tout le plan et pourrait même présenter un danger pour la vie des pompiers. Le fait d'avoir une idée sur le plan global minimise le risque de tomber dans de telles situations. En d'autres termes, il vaut mieux se dire : « si le début du plan réussit et si la situation évolue comme prévu, nous savons comment nous rendre à la fin du plan. » que: « Si le premier bout de plan réussit, nous devons penser à la suite. »

- **Test et validation du plan** : Dans des situations critiques (p.ex. feu de forêt, champ de bataille, etc.), les planificateurs (ou parfois des superviseurs) humains doivent vérifier, corriger et valider les plans construits par les agents logiciels. L'environnement de simulation est un bon moyen pour y arriver. Pendant que le plan s'exécute, les agents logiciels planifient (ou plutôt replanifient) le reste en se basant sur les données les plus récentes du problème. Cette partie du plan est alors exécutée en mode de simulation accélérée avant d'être exécutée dans le monde réel. L'agent humain peut ainsi visualiser comment les choses devraient se réaliser dans le monde réel. Il peut alors apporter des changements sur le plan, approuver ou désapprouver son exécution. Lorsqu'il s'agit de faire une replanification préventive (*périodique anticipée*, voir le point 3) ou une replanification corrective (suite à un imprévu, voir point 4), une simulation en mode accéléré doit se faire dans le monde de simulation, ce qui permettrait au planificateur humain de contribuer au processus de planification comme nous allons le voir dans la prochaine section.
- **Replanification périodique anticipée** : Le fait de proposer un plan initial global ne résout pas le problème relié à l'aspect dynamique de l'environnement. En effet, notre plan global de départ a de faibles chances de rester valide jusqu'à la fin. C'est la raison pour laquelle nous proposons une mise à jour continue de ce plan global. Dans un contexte militaire, nous avons vu une problématique semblable et nous avons alors dit que Wass de Czege [Wass de Czege 99] incite à planifier le plus possible des plans (autant que les différentes actions possibles de l'ennemi dans le cas d'un conflit militaire, par exemple) et créer ensuite un plan pour les situations les plus probables et les plus dangereuses. Les contraintes de notre problème étant différentes, nous avons pensé à une autre manière de faire mais qui s'inspire de l'approche anticipatrice de Wass de Czege : Lorsque l'exécution du plan commence, une replanification est déclenchée périodiquement pour mettre à jour le plan global. Durant chaque période d'exécution, les planificateurs peuvent collecter de nouvelles données sur l'évolution effective de la situation, sur l'état d'exécution du plan et sur les hypothèses sur lesquelles ils ont basé la planification. Dans le cas de la construction de la ligne d'arrêt, ceci

correspond respectivement à la progression effective des feux, à la progression effective des bulldozers sur le terrain et aux nouvelles prévisions sur la propagation des feux. L'idée derrière cette replanification périodique, est qu'au fur et à mesure que l'exécution du plan avance, des données qui étaient manquantes ou incertaines deviennent disponibles ou moins incertaines. Une replanification peut ainsi améliorer le plan global et augmenter ses chances de réussite. Contrairement à la planification en continu standard qui arrête le processus de planification pour lancer l'exécution de ce qui vient d'être planifié, la replanification périodique se fait en parallèle avec l'exécution. Cette approche permet donc de maintenir à jour le plan, mais aussi d'anticiper les problèmes. En effet, elle essaye de corriger le plan avant qu'il n'échoue, d'où son nom : *replanification périodique anticipée*.

- **Replanification en cas d'imprévu** : L'approche par *replanification périodique anticipée* est donc une planification en continu qui a un rôle préventif. Le même principe est aussi utilisé pour corriger le plan lorsqu'un imprévu survient. Si un événement inattendu survient, les agents planificateurs n'attendent pas la prochaine replanification périodique anticipée pour corriger le plan, mais déclenchent aussitôt une replanification immédiate à partir de la situation en cours.

Pour résumer, la planification en continu que nous proposons se ramène à une replanification préventive (*périodique anticipée*) et une autre corrective (suite à un imprévu). Dans les deux cas, un plan global est élaboré en parallèle (pendant que le plan s'exécute dans le monde réel) par des agents logiciels dans un environnement de simulation à connaissance (spatiale entre autres) augmentée.

6.2.3.2. Complémentarité entre les capacités humaines et logiciels pour une meilleure planification

Dans la section 2.1 du chapitre 2 nous avons énuméré les points forts et les points faibles d'un être humain qui essaye de planifier dans un environnement spatial et complexe. Nous avons dit que le planificateur humain a une cognition spatiale limitée et biaisée

ainsi que des capacités de simulation limitées. Nous avons vu par contre qu'il dispose d'autres capacités cognitives très développées et qui ne peuvent (ou difficilement) être remplacées par la machine telles que l'anticipation, le jugement, etc. Les agents logiciels dans ENCASMA par contre, ont une sensibilité spatiale qui peut remédier aux limites humaines. Ils peuvent ainsi faire des raisonnements détaillés sur l'espace dans lequel ils doivent agir. Grâce à l'environnement de simulation, ils peuvent également simuler des plans avant leur exécution. Ceci étant dit, le planificateur humain garde un rôle principal. En effet, c'est lui qui, grâce à ses capacités cognitives et à son expérience, peut corriger, approuver ou désapprouver des plans. Sa capacité d'anticiper les problèmes ne peut qu'être d'un grand support pour les planificateurs logiciels.

Nous pensons que, même en disposant de toute la connaissance qu'il faut, les agents logiciels ne peuvent planifier sans l'appui des agents humains. De la même façon, les planificateurs humains ne peuvent pas planifier sans l'aide des agents logiciels dans des environnements spatiaux complexes et dynamiques. Cette complémentarité est assurée par une DCP basée sur la validation (en mode simulation) par un opérateur humain des plans construits par les agents logiciels.

6.2.3.3. Infrastructure de rencontre spatiale : un support pour la planification

Si maintenant nous nous situons dans un contexte de planification plus distribuée (impliquant plusieurs planificateurs géographiquement distribués) nous pourrions noter que les agents logiciels sont souvent invités à échanger des messages afin de collaborer, négocier ou se coordonner. Vu que ces agents peuvent être géographiquement distants, la plupart de ces messages seront distants et doivent donc transiter par un réseau informatique. Ce réseau peut ne pas être fiable (faible bande passante, non sécuritaire...), ce qui cause un énorme problème à la communication inter-agents, surtout si l'échange de messages est très fréquent comme dans le cas d'une négociation (Chapitre 1). Il serait alors préférable si cet échange de messages peut se faire localement (au niveau d'une seule machine par exemple). Étant donné que les agents sont dispersés, nous devons trouver un moyen de les réunir.

Une première alternative serait de doter les agents d'une capacité de mobilité (ils deviennent donc des agents mobiles), l'échange de messages en local est donc possible si ces agents mobiles arrivent à se rencontrer dans un seul endroit (une seule machine). Cet endroit de rencontre est appelé « Infrastructure de Rencontre » (IR).

[Maamar et al., 02] définissent l'IR comme « une plate-forme logicielle dans laquelle des agents résident et interagissent dans un environnement local et sécuritaire. » Le principe de l'IR est illustré par les Figures 4 et 5.

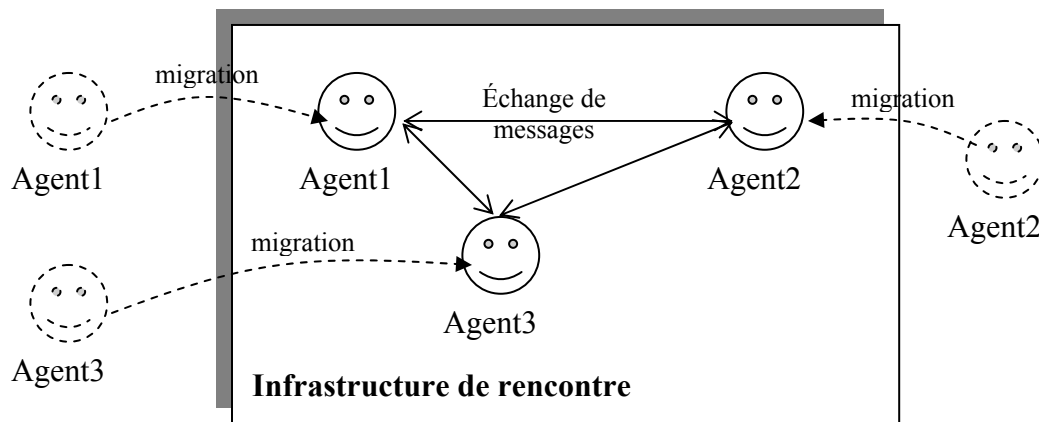


Figure 6.4. Principe de l'infrastructure de rencontre par migration des agents.

Lorsqu'il s'agit de se rencontrer dans une IR, un agent logiciel peut s'y rendre en personne (Figure 6.4), comme il peut très bien déléguer un autre agent pour faire le travail à sa place. Cet agent délégué doit hériter de son agent délégant les capacités (de négociation, de planification, de raisonnement...) nécessaires pour pouvoir accomplir sa tâche. On a souvent recours à cette approche lorsque l'agent délégant est contraint à rester dans son environnement d'origine (et donc ne peut pas migrer vers l'IR). Le principe de cette deuxième approche basée sur la délégation est illustré par la Figure 6.5.

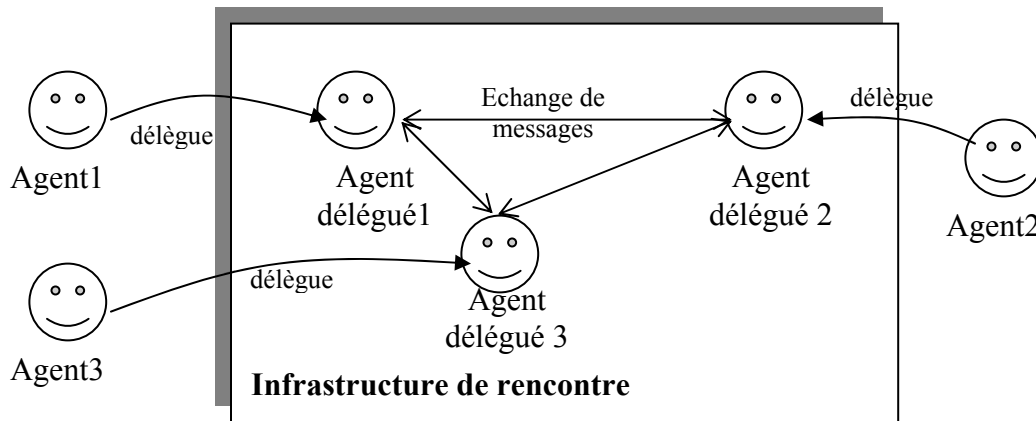


Figure 6.5. Principe de l'infrastructure de rencontre par délégation agents.

Dans un contexte de M-services, [Maamar et al., 02] utilisent l'IR comme un lieu de rencontre où des agents logiciels fournisseurs annoncent leurs services respectifs et où les agents clients peuvent venir négocier avec les fournisseurs l'attribution de ces services. Les agents fournisseurs et clients envoient leurs agents délégués sur l'IR au lieu de se déplacer par eux-mêmes ; Ils pourront ainsi rester dans leurs environnements d'origine afin d'y accomplir d'autres tâches. Un agent superviseur au niveau de l'IR assure son bon fonctionnement et la coordination entre les différents agents qui y entrent.

Nous avons également utilisé la notion d'IR dans un contexte de planification [Sahli 01] et [Moulin et al., 01]. L'IR est le lieu de rencontre d'agents logiciels planificateurs (qui planifient une évacuation aéromédicale dans [Sahli 01]). Dans un environnement distribué et à réseau de communication non fiable comme c'est le cas dans un contexte militaire ou dans un feu de forêt, des agents planificateurs peuvent se rencontrer dans une IR afin de pouvoir échanger des messages de façon plus sécurisée et fiable. Une telle approche permet donc à des planificateurs distribués (des agents mobiles) de mener des négociations gourmandes en échanges de messages, dans un environnement local (où la communication est plus fiable et sécurisée). Lors de la conception de notre solution au problème de planification d'itinéraire d'évacuation aéromédicale, nous avons utilisé une approche d'IR hybride : par agents délégués et par migration des agents mêmes selon les circonstances.

Pour revenir au contexte de ce chapitre et au problème de planification dans un environnement réel à forte composante spatiale, nous pouvons dans ce sens considérer l'environnement de simulation comme une infrastructure de rencontre pour les planificateurs agents. Pour des scénarios plus complexes que celui considéré (planification de la construction de la ligne d'arrêt par une seule équipe de bulldozers) et impliquant plusieurs équipes distribuées, les agents de simulation peuvent communiquer en local sans problème. Les communications distantes se réduisent alors à quelques messages (requête demandant de trouver un chemin, envoi du plan, notifications lors des imprévus). L'implémentation de l'approche DCP par simulation multiagent, est ainsi un bon moyen pour contourner le problème des communications distantes.

Mieux encore, il s'agit d'une infrastructure de rencontre spatiale qui fournit aux différents agents planificateurs une connaissance spatiale fiable. Notre environnement de simulation n'est donc pas juste un lieu de rencontre des agents planificateurs, mais aussi un support de connaissances spatiales. Ceci permet donc aux agents de planifier en étant conscients de l'espace dans lequel ils doivent agir. Nous dotons ainsi les agents logiciels d'une sensibilité spatiale qui leur permet de raisonner sur l'espace et de construire des plans qui tiennent compte des contraintes du terrain. Nous offrons également aux agents planificateurs un support non négligeable pour une planification distribuée. En effet, grâce à l'environnement de simulation nous donnons une alternative à un environnement réel dans lequel la dispersion géographique des planificateurs ne facilite pas le processus de planification.

6.3. Contributions du chapitre

Au début du chapitre nous avons relevé les limites de l'approche DCP au niveau du processus d'entrelacement de la planification et de l'exécution.

Nous avons ensuite proposé une nouvelle approche d'entrelacement du processus de planification avec l'exécution : approche par *Replanification Périodique Anticipée*. Cette façon de faire améliore le processus d'entrelacement (l'agent n'est pas obligé d'arrêter l'exécution pour planifier, mais la planification se fait toujours en parallèle), améliore la qualité du plan (le plan est périodiquement mis à jour, les éventuels problèmes sont

anticipés et pris en compte des fois avant même de constater les répercussions de ces problèmes sur le plan), mais surtout donne plus de garanties quant à l'aboutissement à une solution finale (un plan global est toujours disponible, contrairement à l'approche classique étape par étape qui ne garantit jamais l'atteinte d'une solution finale).

L'idée de planifier dans une infrastructure de rencontre (environnement de simulation) semble pertinente aussi pour remédier au problème des communications distantes coûteuses lorsque la planification est plus distribuée.

Il est à noter que toutes ces limites de la DCP ont été franchies grâce à la présence d'un environnement de simulation parallèle au monde réel. Si jamais, pour une raison ou une autre (non disponibilité des données, nature de l'application, etc.), un tel monde virtuel ne peut pas être construit, notre approche ne peut pas être appliquée et par conséquent, les limites de la DCP citées ci dessus persistent.

CHAPITRE 7

Réalisation

Dans le Chapitre 1 nous avons justifié notre intérêt pour le domaine des feux de forêts (environnement dynamique à grande échelle géographique et nécessitant une planification continue). Nous avons également présenté des spécificités du domaine (Chapitre 3) afin de montrer les contraintes particulières (spatiales, stratégiques, et humaines) qui doivent être prises en compte dans les opérations de lutte contre les feux. Dans les Chapitre 5 et 6, nous avons présenté une application particulière (« la construction d'une ligne d'arrêt ») afin d'illustrer les différents aspects apportés par notre approche théorique et plus particulièrement l'architecture ENCASMA (Chapitre 4).

Dans ce chapitre, nous décrivons les aspects techniques de la réalisation de l'application. La Section 1 présente les scénarios que nous avons développés. Étant donné que la réalisation du prototype se base essentiellement sur les fonctionnalités offertes par la plateforme de géosimulation MAGS, nous consacrons la deuxième section à la description détaillée des principaux modules et concepts de MAGS utilisés lors de la réalisation de notre prototype. Enfin, la Section 3 décrit la réalisation de l'application.

7.1. Scénarios

Selon la taille du feu, l'attaque peut être directe ou indirecte (nous mettons de côté la stratégie hybride (Chapitre 3) pour des raisons de simplification). Pour les petits feux, les pompiers attaquent généralement le feu par son flanc (le côté) et progressent vers le front du feu jusqu'à encercler tout le feu : c'est l'*attaque directe*. Quand par contre le feu est plus intense, une *attaque indirecte* s'impose. Les pompiers construisent des lignes assez profondes et larges (d'environ 3m de large) qui entourent le périmètre du feu [Nicolas et

Beebe 99]. Nous avons essayé d'implémenter les deux stratégies avec cependant un intérêt particulier pour la stratégie indirecte.

7.1.1. Stratégie directe

Scénario : Dans ce scénario, il s'agit de

- Simuler la progression des feux dans l'espace et sur une échelle de temps en mode accéléré.
- Concevoir un agent logiciel qui simule les déplacements²⁰ d'un bulldozer aux abords des feux et qui suit la progression du feu (lorsqu'un nouveau périmètre est affiché sur l'environnement de simulation, il doit automatiquement le suivre).
- Doter cet agent de capacités de perception afin d'appréhender l'espace qu'il traverse. Il peut ainsi simuler les variations de la vitesse de progression du bulldozer selon les caractéristiques du terrain.

La simulation de la construction de la ligne d'arrêt (par l'agent représentant le bulldozer) permet aux décideurs humains (chefs des opérations) de vérifier la faisabilité de l'opération par rapport à l'éventuelle progression des feux. Si par exemple, la simulation montre que le bulldozer pourrait construire une ligne d'arrêt jusqu'à un certain point d'ancrage (p.ex. lac) sans que les feux ne traversent de l'autre côté du point d'ancrage, cela signifie que la stratégie est a priori efficace dans ce cas.

7.1.2. Stratégie indirecte

Ce genre d'attaque nécessite aussi bien l'habilité et l'expérience que les ressources adéquates. Les intervenants peuvent utiliser des outils à main pour construire les lignes mais de préférence les bulldozers (si cette ressource est disponible). La ligne d'arrêt doit être construite à une distance sécuritaire des feux et doit profiter le plus possible des barrières naturelles ou artificielles disponibles (les incorporer dans la ligne) [Firewise 05].

²⁰ Il ne s'agit pas ici d'un simple déplacement, mais aussi du nettoyage d'une ligne d'arrêt au fur et à mesure que le bulldozer avance.

La construction de la ligne d'arrêt commence généralement par un point d'ancrage et passe par des portions de terrain où l'essence est moins dense. Les virages étroits doivent être évités sur la ligne d'arrêt car ceci peut causer des spots de feu (quand le feu saute par-dessus la ligne).

Pratiquement, des experts de lutte contre les feux décident vaguement de l'endroit où il faut construire la ligne en se basant sur les prédictions de progression des feux ainsi que sur leur expérience personnelle. Sur le terrain, les bulldozers arrivent sur l'endroit choisi par les experts et essayent de se trouver un bon passage pour avancer le plus vite possible. Pour ce faire, des éclaireurs (des hommes à pied) devancent les bulldozers afin d'explorer le terrain et guider ainsi les machines à travers les endroits les plus faciles à passer. Cette façon de faire présente plusieurs inconvénients : l'endroit choisi par les experts n'est pas forcément le meilleur; les passages trouvés par les éclaireurs ne sont pas forcément les meilleurs pour la prochaine étape et peuvent même mener plus tard les machines vers une impasse ou un obstacle difficile à franchir; Les pompiers ne sont jamais sûrs de pouvoir finir la ligne à temps même en absence d'imprévu; En cas d'imprévu (avancée plus rapide que prévue des feux), les hommes sur le terrain ne disposent pas de beaucoup de données pour pouvoir s'adapter au changement; etc.

Nous nous proposons d'implémenter un prototype pour aider les décideurs humains (les chefs opération) de deux façons différentes : avant la construction de la ligne et durant la construction de la ligne d'arrêt.

7.1.2.1. Scénario « avant la construction de ligne d'arrêt »

Il s'agit de trouver un chemin approprié pour le passage des bulldozers en tenant comptes des contraintes décrites précédemment et de l'avancée prévue des feux. L'utilisateur humain doit être capable de tracer dans l'environnement de simulation une esquisse de ce que devrait être la ligne d'arrêt.

Le but principal est de simuler le déplacement du bulldozer sur le chemin trouvé afin de permettre au décideur humain de vérifier la faisabilité du tracé par rapport à la progression des feux et de s'assurer de l'utilité de la ligne sur le processus du contrôle des feux.

La solution théorique et pratique pour implémenter ce scénario a déjà été discutée dans le Chapitre 4 et 5. Nous présentons ici la façon dont elle a été implémentée.

7.1.2.2. Scénario « pendant la construction de la ligne d'arrêt »

Une fois le plan trouvé et validé par les experts des feux, l'exécution effective du plan peut commencer (les bulldozers commencent à dégager la ligne d'arrêt). L'environnement de simulation doit alors assurer les tâches suivantes :

- Mettre à jour de façon continue l'environnement de simulation selon l'évolution de la situation sur le terrain réel. Ceci revient à déplacer les *agents acteurs* et *agents ressources* sur l'environnement virtuel selon leurs vraies positions dans le monde réel.
- Détecter toute déviation des bulldozers par rapport au plan initial. Si ce changement rend le plan invalide, il faut relancer une nouvelle planification du chemin selon les nouvelles données.
- Les pompiers sur le terrain pourraient rapporter un obstacle qui ne figure pas dans l'environnement de simulation (à cause de données SIG erronées ou non mises à jour). L'utilisateur de l'environnement de simulation doit alors pouvoir rajouter cet obstacle à l'endroit indiqué. Le système doit ensuite vérifier si cet obstacle nuit au plan initial. Si tel est le cas, le chemin doit être changé afin de contourner l'obstacle.
- Appliquer les principes de la replanification périodique anticipée décrite dans le Chapitre 6 comme une amélioration de l'approche DCP. Il faut donc relancer la planification du chemin de façon périodique, en tenant compte à chaque fois des dernières données concernant la situation en cours (particulièrement les positions effectives des bulldozers et la progression effective des feux).

7.2. MAGS

Afin de comprendre un peu comment MAGS fonctionne, nous avons synthétisé à partir du guide du développeur de MAGS [MAGS 2003] les principales composantes de la plateforme.

7.2.1. Architecture

MAGS est une plateforme générique de simulation qui permet la simulation des comportements de milliers d'agents dans un environnement géographique. Il est donc nécessaire d'avoir un module pour l'environnement, un module pour les agents, une interface et un contrôleur. En plus de ces modules, MAGS comprend un moteur d'affichage (Display module) ainsi qu'un moteur de génération de particules. L'architecture globale de MAGS (la version originale de MAGS, avant l'implémentation de l'application des feux de forêts) est illustrée par la Figure 7.1. Dans ce qui suit, nous donnons une brève présentation de chacun des principaux modules de MAGS.

Initialisation

La première tâche de MAGS est d'initialiser les différents modules à l'aide des fichiers de configuration. C'est la tâche de ce module d'ouvrir les fichiers de configuration et de distribuer les paramètres d'initialisation aux différents modules pour qu'ils s'initialisent. Le fichier d'initialisation de la version 1.2 est une base de données qui contient toute l'information nécessaire (environnement, scénario, comportements).

Environnement

Le module d'environnement contient toutes les classes servant à décrire et à gérer l'environnement géographique dans MAGS. L'environnement est découpé en couches (layer) où chaque *layer* contient les agents qui s'y trouvent.

Les agents

Le modèle d'agent implanté dans MAGS est assez complexe. Il y a le module d'agents (*AgentsModule*) qui contient tous les fichiers relatifs aux classes d'agents. Ensuite, il y a le module des états des agents. Chaque agent contient un ensemble d'états servant à lui fournir des fonctionnalités (navigation, perception, etc.). Les agents doivent aussi posséder des comportements et il y a un module qui s'y consacre (*BehaviourModule*).

Contrôleur

La simulation de milliers d'agents doit être gérée efficacement et le module *ProcessesModule* s'occupe de tout. Le temps de la simulation, la synchronisation des processus, ne sont que quelques fonctions dédiées à ce module de contrôle.

Moteur de Particules

MAGS offre aussi la possibilité de simuler des phénomènes de fumées à l'aide du moteur de particules (*ParticleModule*).

Moteur d'affichage

Pour être en mesure d'afficher la simulation en 2D ou en 3D, un moteur d'affichage est nécessaire (*DisplayModule*). Il permet l'affichage, via l'interface, de la simulation en cours.

Interface

Le module d'interface est dédié à l'interaction entre l'utilisateur et MAGS. Il permet de visualiser et d'interagir avec la simulation.

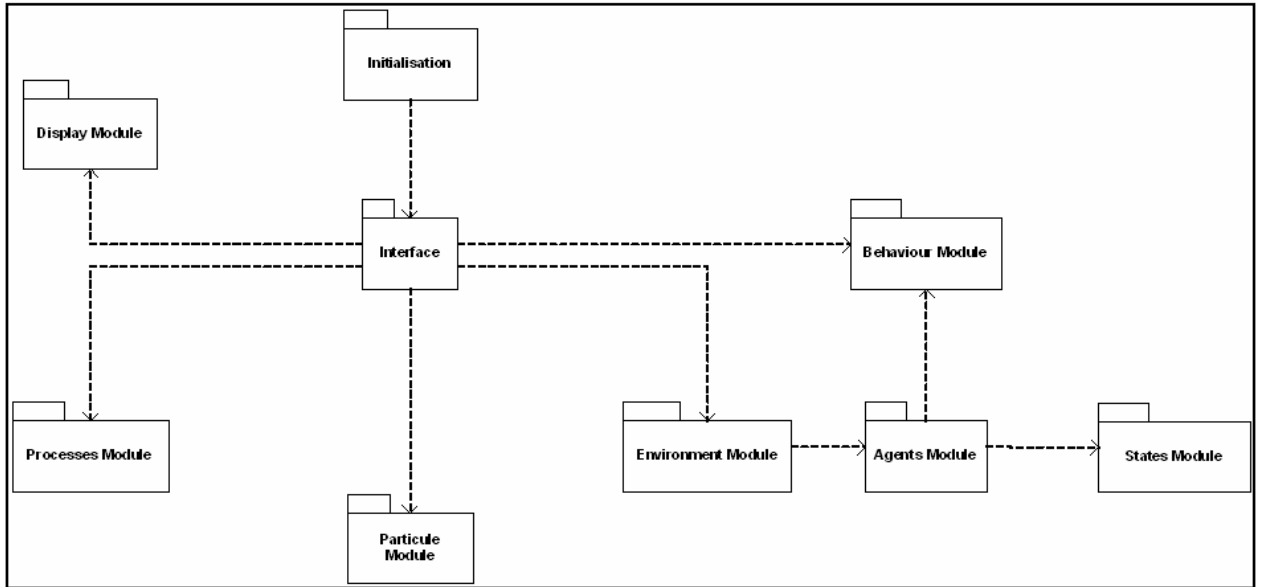


Figure 7.1. Architecture générale de MAGS (version originale)

Dans la suite, nous présentons plus en détail les modules les plus importants qui vont servir pour implémenter les scénarios décrits à la section précédente.

7.2.2. Module Environnement

Ce module est très important dans MAGS et son instantiation est incontournable. Il permet de définir l'environnement de simulation dans lequel les agents vont pouvoir opérer. Les deux classes les plus importantes du module (Figure 7.2) sont la classe *TEnvironment* et la classe *TRasterLayer*.

L'objet principal de l'application de simulation (*TMagsApp*) contient un et un seul objet instancié de la classe *TEnvironment* qui représente l'environnement de simulation dans lequel les agents opèrent. L'attribut principal de *TEnvironment* est *m_RasterLayers* qui est un tableau (de taille maximale 10 pour l'instant) d'objets *TRasterLayer*.

En effet, la structure de l'environnement a été conçue pour pouvoir supporter la simulation de plusieurs couches dans la même application. Dans un centre d'achat, par exemple, chaque étage peut être simulé sur une couche à part, ainsi chaque étage aura son propre objet *TRasterLayer*.

La classe *TRasterLayer* contient les données de l'environnement ainsi que les agents d'une couche de simulation. Parmi les attributs d'un objet *TRasterLayer* :

m_Height : Hauteur du raster en pixels

m_Width : Largeur du raster en pixels

m_Scale : L'échelle

m_HeightMap : la matrice d'élévations du modèle 3D. Cette matrice donne pour chaque pixel (x,y) du Raster, l'élévation du terrain (le sommet) en ce point.

m_NetworkMap : Le *NetworkMap* est une matrice d'éléments *NetworkElt*. Chaque *NetworkElt* contient les champs suivants :

IsInSideWalk : trottoir

IsInCrossWalk : passage pour piétons

IsInRoad : route

IsInRiver : rivière

IsInTransitPoint : point de transit (il permet de gérer la transition des agents d'une couche vers une autre)

IsInAccessPoint : point d'accès (à un bâtiment généralement)

IsInJunctionPoint : nœud du réseau piétonnier

IsObstacle : obstacle pour la navigation des agents

m_AgentsMap : C'est la carte de localisation des agents. Le *AgentsMap* est aussi une matrice. Chaque élément *AgentsMap(x,y)* de cette matrice est une liste chaînée de pointeurs vers les agents qui sont localisés à la position (x,y).

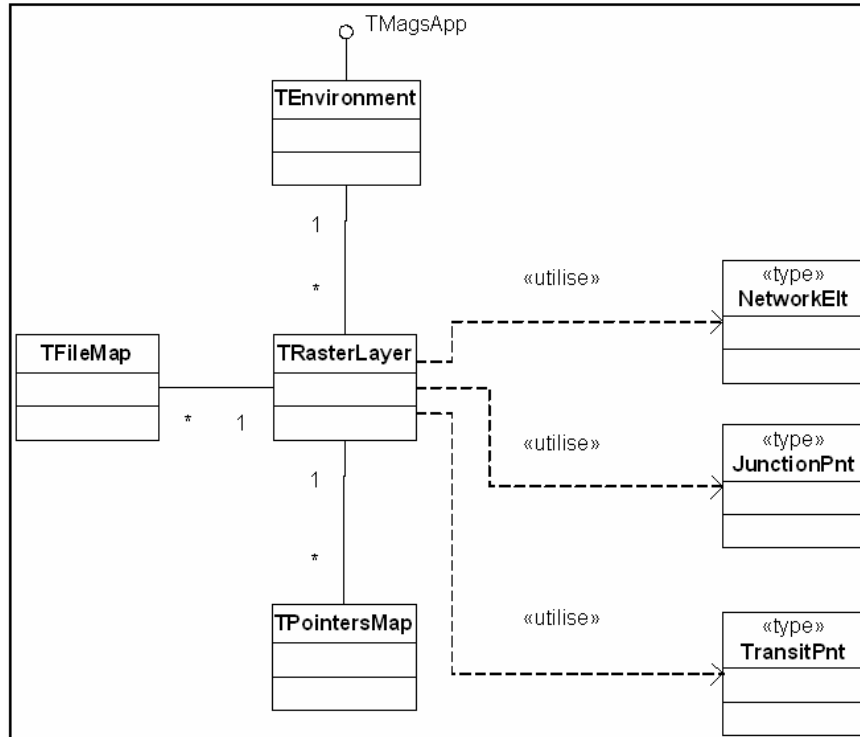


Figure 7.2. Diagramme de classe simplifié du module *Environnement*

7.2.3. Module Agents

MAGS définit quatre types d'agents (Figure 7.3) soit les *agents*, les *agents clusters*, les *agents groupes* et les *agents scénarios*. Tous ces agents dérivent de la classe de base *TBasicAgent*.

Les Agents

Un agent de type *Agent* peut représenter presque n'importe quoi comme par exemple un piéton, une voiture, un bâtiment, un objet, etc. L'agent est caractérisé plus particulièrement par ses états. Un agent peut posséder jusqu'à 11 états. Par exemple, une voiture devra posséder l'état *DriveState* tandis qu'un piéton devra posséder l'état *NavigationState*. Les états seront expliqués en détails dans la prochaine section.

Nous ne donnons pas de détails sur les trois autres types d'agents car ils ne sont pas pertinents pour notre application (ils le seront plus tard lorsque nous aurons intégré, dans nos travaux futurs (Chapitre Conclusion), la notion de hiérarchie parmi les agents).

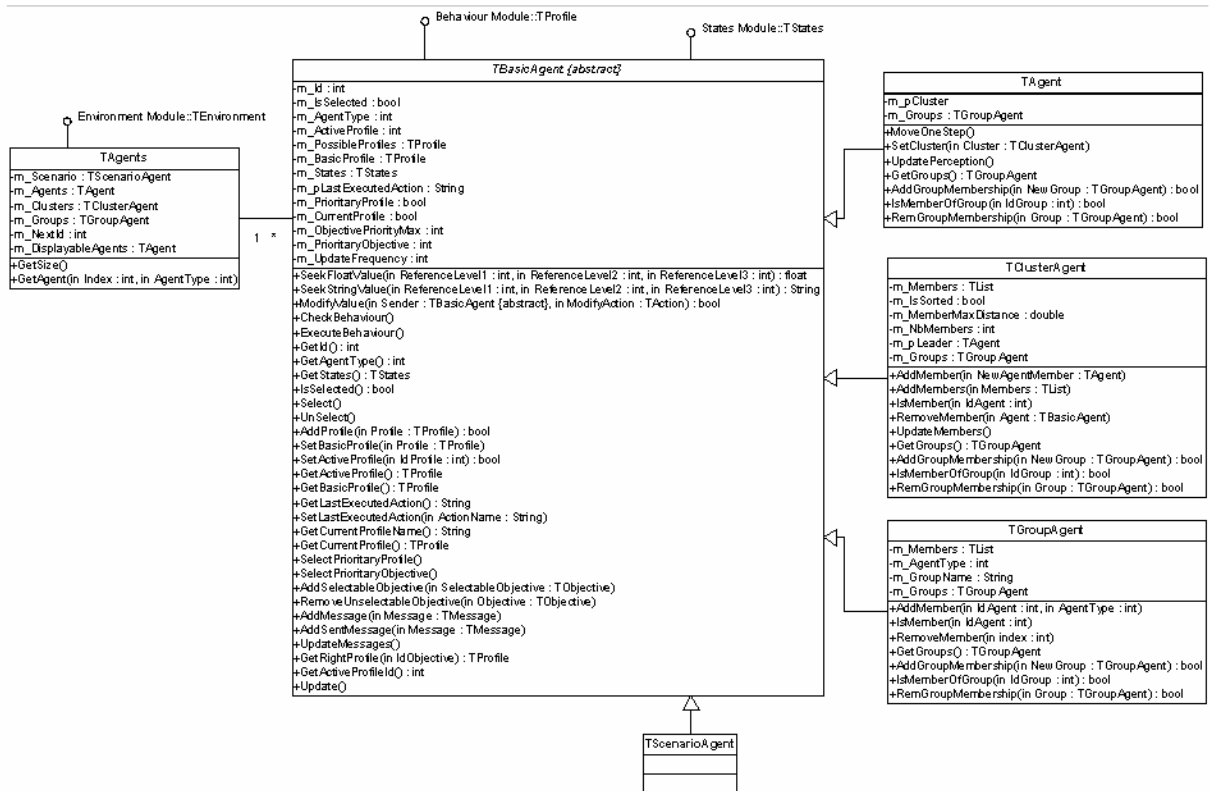


Figure 7.3. Diagramme de classes du module *Agents*

Tous les agents de la simulation se trouvent dans la classe *TAgents* qui gère les listes, les créations, les destructions et l'accès aux agents.

La classe *TAgents* est le conteneur de tous les agents d'une simulation. Lorsqu'un agent est créé dans MAGS, il se retrouve dans l'une des listes de la classe *TAgents* qui gère l'accès et la distribution des identifiants. La classe *TAgents* comprend 7 listes pour contenir les agents :

- *m_Agents* : Cette liste contient des pointeurs vers tous les agents de type *Agent* de la simulation.

- *m_Groups* : Cette liste contient des pointeurs vers tous les agents de type *Group_Agent* de la simulation.
- *m_Clusters* : Cette liste contient des pointeurs vers tous les agents de type *Group_Agent* de la simulation.
- *m_AgentsNav* : Cette liste contient des pointeurs vers tous les agents de type *Agent* qui possèdent un état de navigation (*TNavigationState*).
- *m_AgentsVis* : Cette liste contient des pointeurs vers tous les agents de type *Agent* qui possèdent un état de visualisation (*TVisualisationState*).
- *m_AgentsDrv* : Cette liste contient des pointeurs vers tous les agents de type *Agent* qui possèdent un état *TDriveState*.
- *m_AgentsMem* : Cette liste contient des pointeurs vers tous les agents de type *Agent* qui possèdent un état de mémorisation (*TMemoryState*).

Les trois premières listes contiennent tous les agents, sans exception, de la simulation. Les autres listes sont seulement présentes pour regrouper certains agents ayant des particularités communes. Il y a évidemment de la redondance mais le gain en performance est considérable. Par exemple, lorsque la boucle principale de la simulation doit mettre à jour la navigation des agents, elle doit cibler seulement les agents ayant un état de navigation (*TNavigationState*). Il serait donc lourd de parcourir toutes les listes des agents en testant s'ils ont un état de navigation. C'est dans cette optique que plusieurs listes ont été créées.

7.2.4. Module États

Chaque agent peut posséder les états décrits dans la Figure 7.4. Tous ces états sont gérés par la classe « conteneur » *TStates*. C'est la classe qui contient tous les états que l'agent peut posséder. Nous y retrouvons l'état de possession (*TPossessionState*), d'occupation (*TOccupantsState*), de mémoire et de perception (*TMemoryState*), de communication (*TCommunicationState*), de position (*TLocationState*), de navigation (*TNavigationState*),

ainsi qu'une liste d'états dynamiques (*TDynamicStates*) et une liste d'états statiques (*TStaticStates*).

- La classe *TPossessionState* comprend les possessions de l'agent.
- Le module de mémoire (*TMemoryState*) représente la perception et la mémoire d'un agent.
- La classe *TCommunicationState* contient les différents messages qu'un agent a reçus et envoyés.
- La classe *TLocationState* concerne le positionnement de l'agent dans son environnement. Grâce à cet état, il est possible d'accéder à la carte de l'environnement qui comprend des pointeurs sur les agents positionnés dans l'environnement géographique virtuel. Cet état nous permet entre autres de vérifier si un agent se trouve près d'un autre agent ou de vérifier si un ou plusieurs agents possédant une certaine caractéristique se trouvent près de l'agent.

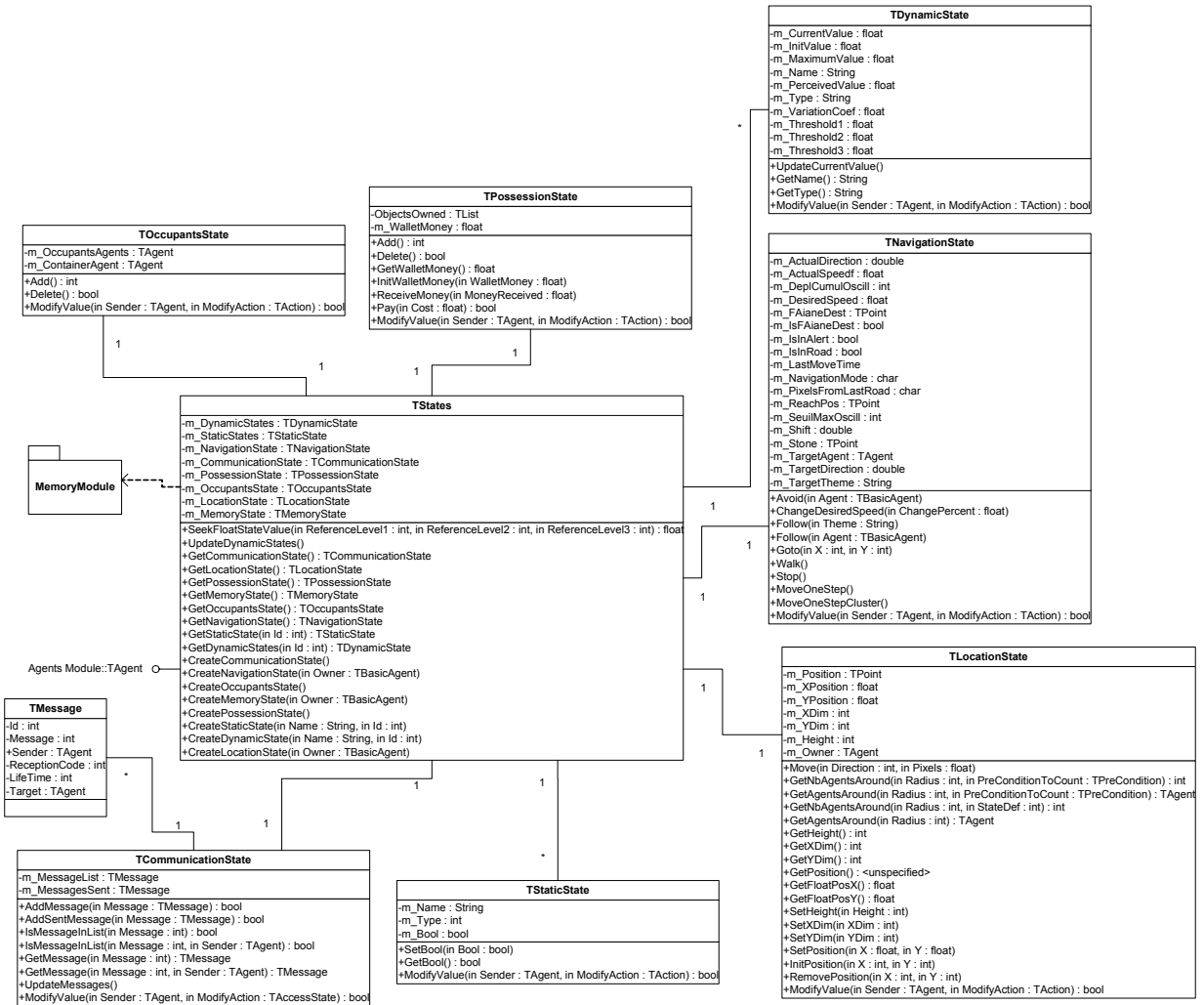


Figure 7.4. Diagramme de classes du module des états

- La classe *TNavigationState* s'occupe de la gestion des déplacements de l'agent au bas niveau. Il ne s'agit pas de prendre des décisions sur la destination d'un agent, car c'est le module comportemental qui s'en occupe. Cet état s'occupe plutôt d'éviter les collisions et de contourner les obstacles afin que l'agent soit en mesure de se rendre d'un point à l'autre. Plusieurs modes de navigation sont possibles. L'agent peut, par exemple, marcher sans but précis ou vers une destination donnée, suivre un autre agent ou fuir un agent.
- La classe *TDynamicState* représente des variables qui évoluent dynamiquement.

- La classe *TStaticState* est prévue pour ajouter différents types d'états à l'agent. Par exemple, un état statique pourrait contenir le sexe ou l'âge d'un agent.

7.3. Implémentation

Afin d'implémenter les scénarios décrits dans la Section 1 selon l'approche décrite dans le Chapitre 4, il faut spécifier les différentes couches de l'architecture ENCASMA selon le contexte de l'application. Nous définissons alors dans la sous-section 7.3.2 l'environnement (la 1^{ère} couche d'ENCASMA) qui va servir pour planifier la construction de la ligne d'arrêt. Nous intégrons le modèle de simulation des feux Prometheus (la 2^{ème} couche d'ENCASMA) dans la sous-section 7.3.4. Les agents qui vont devoir agir dans l'environnement afin de pouvoir planifier la construction de la ligne d'arrêt (selon la stratégie directe ou indirecte) auront un nouvel état MAGS (Section 7.2.3.) appelé *TRoutePlanningState* et présenté dans la sous-section 7.3.3. Cependant, et avant d'entamer ces détails, nous présentons l'architecture globale de l'application ainsi que les différents modules qui ont été ajoutés ou modifiés dans MAGS.

7.3.1. Architecture globale

L'architecture globale de l'application en tant que telle a été déjà présentée à la Figure 5.5 du Chapitre 5. Dans la présente sous-section, nous décrivons (Figure 7.5) l'architecture de la plate-forme MAGS après intégration des modules spécifiques à notre application.

Nous remarquons alors que par rapport à l'architecture initiale de MAGS (Figure 7.1), un nouveau module a été ajouté (*TFire Module*) qui sera discuté à la sous-section 3.4. Ce module utilise entre autre l'interface COM du simulateur des feux Prometheus. Toutefois, il ne s'agit pas de la seule accommodation faite sur MAGS pour supporter notre application. Plusieurs modules préexistants ont été modifiés ou améliorés afin de supporter toutes les fonctionnalités que nous désirons offrir aux agents planificateurs. Les prochaines sous-sections décrivent ces changements.

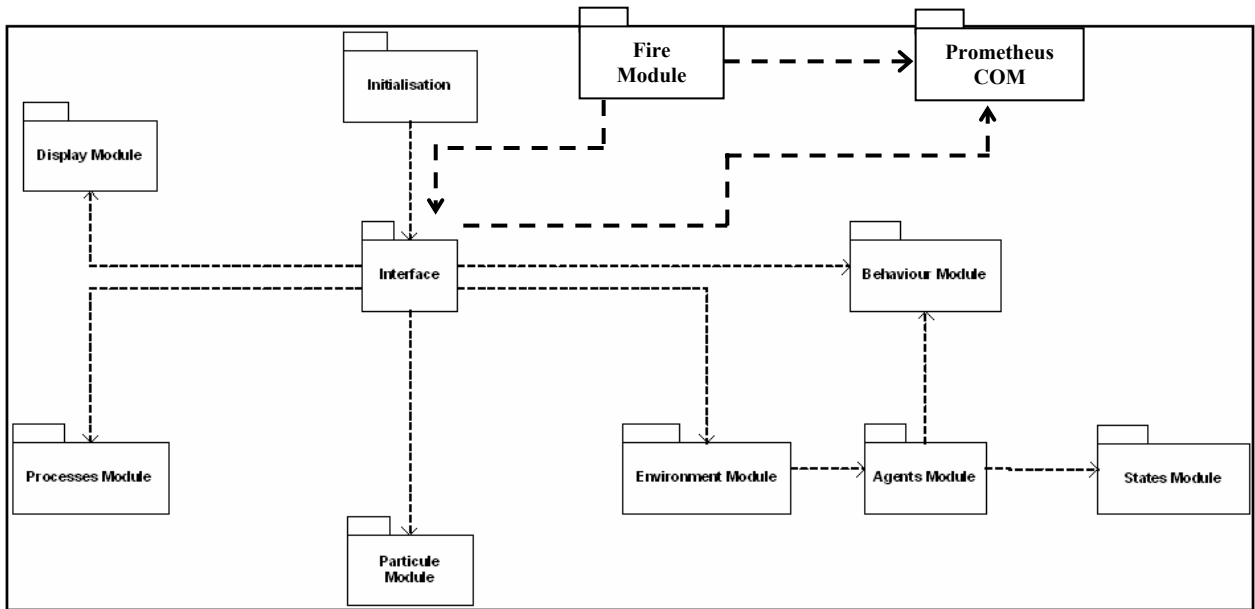


Figure 7.5. Architecture de MAGS après intégration des modules spécifiques à l'application de construction des lignes d'arrêt.

7.3.2. Environnement

Comme cela a été déjà décrit dans la Section 2 de ce chapitre, MAGS fait appel à une classe *TRasterLayer* qui contient les données de l'environnement ainsi que les agents d'une couche de simulation. Cette classe s'alimente en données essentiellement grâce à deux matrices de données : *m_HeightMap* (la matrice d'élévations du modèle 3D et qui donne pour chaque pixel (x,y) du Raster, l'élévation du terrain en ce point) et *m_NetworkMap* (une matrice d'éléments *NetworkElt* qui indique si ce point fait partie d'un trottoir, d'une rivière, d'une route, ..., ou enfin d'un obstacle). Cependant, les besoins de notre application sont plus complexes. En effet, l'élévation (dans *m_HeightMap*) n'est pas la seule donnée topographique pertinente. Le degré de la pente, la direction de la pente, le type d'essence, etc. ; sont tous des facteurs qui influencent le comportement du feu et des agents dans notre application. Nous avons le même problème avec la matrice *NetworkElt* qui ne tient pas compte de certains types de paramètres spécifiques au problème des feux de forêt. De plus, le fait de dire de façon dichotomique qu'un point représente un obstacle ou non, n'est pas très approprié dans notre application. En effet, quand un bulldozer est en

train de creuser sa ligne d'arrêt, son but n'est pas de contourner les obstacles, mais plutôt de passer par le terrain le plus facile d'accès. Tout peut donc être considéré comme obstacle, mais de difficultés différentes. Ces limites dans MAGS sont dues au fait que la plate-forme est plutôt destinée à des applications de simulation à caractère urbain (p.ex. simulation de la foule). Toutefois, la conception de l'environnement dans MAGS est suffisamment flexible pour permettre de supporter la simulation de plusieurs couches dans la même application et par la suite la simulation de données très spécifiques de l'espace. Il suffit donc de rajouter des couches ou encore de modifier la structure des couches existantes pour ces besoins de l'application.

Pour notre application, nous avons remplacé la matrice *m_HeightMap* par une autre (*m_FuelMap*) qui encapsule (à part l'altitude) les données sur le degré de la pente, la direction de la pente, le type d'essence, etc. Pour chaque point de la matrice (correspondant à une cellule de l'environnement virtuel), toutes ces données sont rassemblées dans une structure de données de taille 64 bits. La manipulation de simples opérateurs de logique binaire permet de coder et de décoder ses données au besoin.

Pour ce qui est des données relatives au réseau routier et maritime, nous avons construit une matrice semblable à la matrice *m_NetworkMap* et que nous avons adaptée aux besoins spécifiques des problèmes des feux de forêts. Les données contenues dans cette nouvelle matrice appelée *m_ArianeMap* sont :

- Les rivières
- Les lacs
- Les ruisseaux
- Les routes
- Les sentiers
- Les chemins de gravier
- Les obstacles (infranchissables ou difficilement franchissable)

La connaissance de ces informations par les agents présents dans l'environnement de simulation est très importante car les chemins sont des passages privilégiés pour les bulldozers. Ils sont une sorte de fils d'Ariane [Moulin et al., 2003] que les *agents*

Bulldozers doivent suivre lors de leurs déplacements, ce qui explique l'appellation *m_ArianeMap*. Quant aux obstacles (infranchissables ou difficilement franchissable), ils constituent les cellules que les agents doivent absolument éviter (p.ex. un rocher).

Lors de l'initialisation de l'environnement (par le module initialisation Figure 7.1), les deux matrices *m_FuelMap* et *m_ArianeMap* sont chargées en mémoire. Les agents présents dans l'environnement ont donc accès (par le mécanisme de perception) aux données de ces matrices et plus particulièrement aux cellules contenues à l'intérieur du cône de perception de ces agents. Plusieurs fonctionnalités ont été ajoutées au module *Environment* et plus particulièrement aux classes *TEnvironment* et *TRasterLayer* pour permettre aux agents d'accéder à ces données.

Étant donné que l'environnement est dynamique, la mise à jour des données sur l'environnement a été prévue. Cette mise à jour peut se dérouler en ligne (pendant la simulation) ou hors ligne (en dehors de la simulation). Dans le dernier cas, il suffit de changer les données des matrices *m_FuelMap* ou *m_ArianeMap*. Quant à la mise à jour durant la simulation, le cas le plus probable est l'ajout d'un obstacle (infranchissable ou difficilement franchissable) durant la simulation. Ce scénario se produit lorsque l'utilisateur humain veut ajouter à l'environnement un obstacle qui a été signalé sur le terrain par les pompiers (sous-section 5.4.3.1 du Chapitre 5). Pour ce faire, nous avons prévu des outils graphiques pour ajouter de tels obstacles sur l'environnement même durant la simulation. En fait, l'utilisateur peut dessiner la forme de l'obstacle par simples cliques de souris. Afin de préserver les données initiales de l'environnement (au cas où l'utilisateur voudrait revenir à l'état initial de l'environnement) la matrice *m_ArianeMap* chargée en mémoire est changée (le champ *Obstacle* des cellules qui constituent l'obstacle dessiné par l'utilisateur, est modifié) alors que les données initiales de la matrice restent inchangées sur les fichiers physiques jusqu'à confirmation finale de la part de l'utilisateur (l'utilisateur doit sauvegarder le changement pour qu'il devienne persistant).

7.3.3. État *TRoutePlanningState*

MAGS offre la possibilité de créer des agents logiciels à n'importe quel moment de la simulation. Lors de la création, le développeur peut choisir les états (sous-section 7.2.4) dont le nouvel agent pourrait disposer. Dans l'exemple de la Figure 7.6, le code indique que l'agent ayant l'identifiant '2500' a été créé avec les états de navigation (A_NAVIGATION_STATE), de mémoire (A_MEMORY_STATE), de localisation (A_LOCATION_STATE), de possession (A_POSSESSION_STATE) et de visualisation (A_VISUALISATION_STATE).

```

TAction *action = new TAction;
TBasicAgent* agent;

action->Init(CREATE_AGENT,"Create Mobile Agent");

action->AddParameter(2500); // Identifiant
action->AddParameter(100); // Posx
action->AddParameter(146); // PosY
action->AddParameter(20); // rayon de perception
action->AddParameter(
    A_NAVIGATION_STATE      |
    A_MEMORY_STATE          |
    A_LOCATION_STATE        |
    A_POSSESSION_STATE      |
    A_VISUALISATION_STATE );
action->AddParameter(1.0f);
action->AddParameter(1.0f);
action->AddParameter(1.0f);

```

Figure 7.6. Exemple de code de création d'un agent dans MAGS

Pour les besoins spécifiques de notre application, nous avons défini un nouvel état nommé *TRoutePlanningState* (Figure 7.7) qui caractérisera les agents responsables de la planification des lignes d'arrêt.

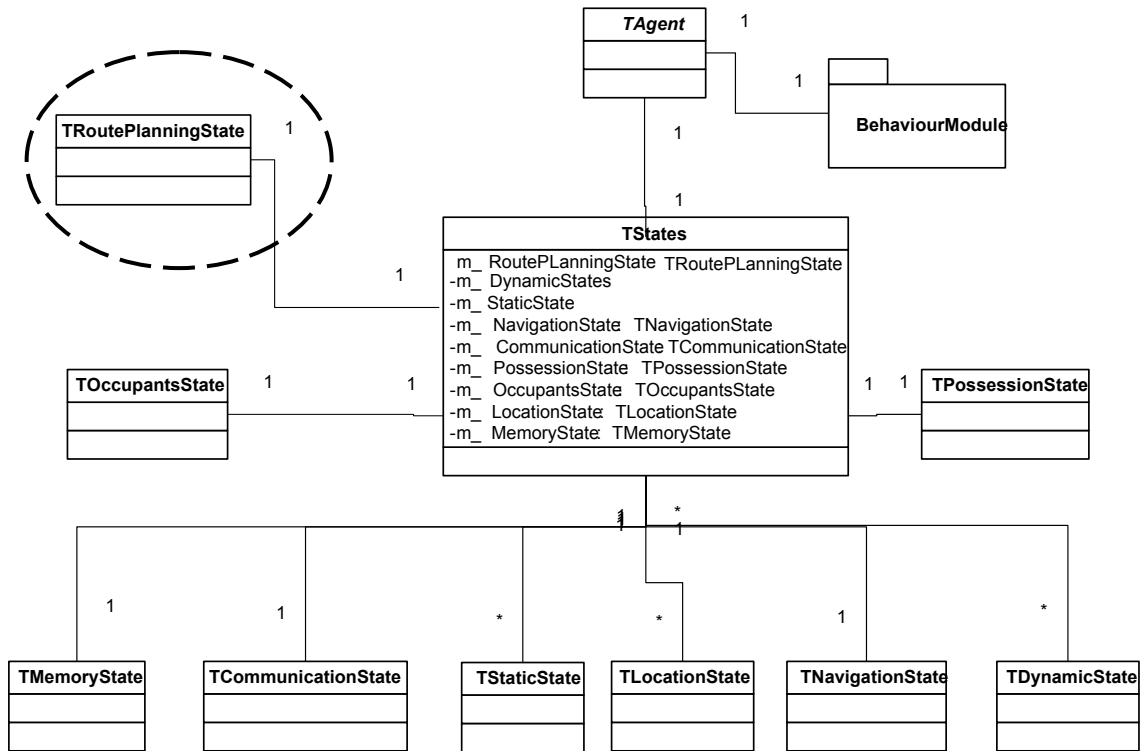


Figure 7.7. Intégration de l'état *TRoutePlanningState* dans le module États des agents MAGS

Pour créer un agent ayant ce nouvel état, il suffit de le rajouter dans la liste des états lors de la création de l'agent (Figure 7.8). Le système de simulation se rendra compte alors à chaque unité de temps de simulation que cet agent a un état `A_ROUTEPLANNING_STATE` et qu'il doit par conséquent agir selon ce que la classe *TRoutePlanningState* lui indique de faire.

Nous remarquons que sur l'exemple de la Figure 7.8, l'agent ne dispose pas de l'état `A_NAVIGATION_STATE` nécessaire pour la navigation de l'agent. En fait, la navigation agent telle que définie dans MAGS suppose que l'agent se déplace sans plan précis. Il essaye juste de maintenir le cap sur une destination précise tout en évitant les obstacles qui peuvent lui faire face. Ce mode de navigation ne correspond pas aux besoins de nos agents qui devront disposer de capacités de planification pour faire les meilleurs choix lors du processus de recherche de chemin.

```

TAction *action = new TAction;
TBasicAgent* agent;

action->Init(CREATE_AGENT,"Create Mobile Agent");

action->AddParameter(2501); // Identifiant
action->AddParameter(100); // Posx
action->AddParameter(146); // PosY
action->AddParameter(20); // rayon de perception
action->AddParameter(
    A_ROUTEPLANNING_STATE |
    A_MEMORY_STATE |
    A_LOCATION_STATE |
    A_POSSESSION_STATE |
    A_VISUALISATION_STATE );
action->AddParameter(1.0f);
action->AddParameter(1.0f);
action->AddParameter(1.0f);

```

Figure 7.8. Exemple de code de création d'un agent dans MAGS ayant l'état
A_ROUTEPLANNING_STATE

La classe *TRoutePlanningState* est ainsi responsable de définir pour les agents planificateurs de notre application la façon dont ils vont pouvoir planifier et par la suite se déplacer. Cependant, plusieurs sortes d'agents interviennent dans le processus de la planification d'une ligne d'arrêt. Nous avons alors défini pour chaque type d'agent ce qu'il doit faire, et ce dans la même classe²¹ *TRoutePlanningState*.

Parmi les agents intervenants les plus importants :

- L'agent qui suit le périmètre des feux pour trouver le chemin le plus adéquat dans une stratégie d'attaque directe. Nous appelons cet agent *Agent_Attaque_Directe*.
- L'agent qui cherche un chemin dans la stratégie indirecte selon les contraintes décrites dans le Chapitre 5. Nous avons déjà appelé auparavant cet agent : *Agent_Pathfinder*, et nous le notons ici *Agent_Pathfinder* (notation utilisée dans le code).

²¹ Nous aurions pu définir un état à part pour chaque type d'agent mais nous avons préféré intégrer le tout dans un seul état pour ne pas augmenter de façon exagérée le nombre d'états possibles.

- L'agent qui simule le meilleur chemin trouvé. Nous appelons cet agent *Agent_Meilleur_Chemin*.
- L'agent qui simule la partie du plan déjà exécutée sur le terrain (afin de montrer à l'utilisateur humain ce qui a déjà été fait). Nous appelons cet agent *Agent_Déjà_Fait*.
- L'agent qui vérifie si le plan initial est toujours valide à chaque fois qu'un bulldozer envoie une notification indiquant sa localisation courante. Nous appelons cet agent *Agent_Vérificateur_Position*.
- L'agent qui vérifie si un obstacle imprévu et qui vient d'être ajouté à l'environnement affecte ou non le plan initial. Si jamais il l'affecte, il doit déterminer les extrémités de cet obstacle. Nous appelons cet agent *Agent_Vérificateur_PlanVsObstacle*.

Nous allons dans ce qui suit donner plus de détails sur les agents évoqués ci-dessus.

7.3.3.1. Agent_Attaque_Directe

Cet agent a pour tâche de suivre la progression des feux tout en restant très proche du périmètre du feu. Le chemin qu'il aurait parcouru est celui que devrait emprunter le bulldozer lors d'une attaque directe. Chaque périmètre de feu correspond à un instant t de la simulation et est composé d'*Agents_Fire* (sous-section 3.4.). Parmi les attributs de l'*Agent_Fire*, son *identifiant* (un index pour reconnaître l'emplacement de l'agent par rapport à ses voisins) et l'*étape de la simulation* (qui correspond à un instant précis de la simulation). Lorsque l'*Agent_Attaque_Directe* se retrouve près d'un *Agent_Fire* d'identifiant n , il essaye de bouger vers l'*Agent_Fire* ayant pour identifiant $n+1$. La vitesse de progression de l'*Agent_Attaque_Directe* entre les deux *Agents_Fire*, dépend de la nature du terrain. Nous obtiendrons ainsi une simulation plausible de l'avancée du bulldozer par rapport à la progression des feux lors d'une attaque directe.

Lorsqu'un nouveau périmètre de feu est disponible et affiché sur l'environnement de simulation (ceci revient à dire que nous simulons le feu à un instant $t+1$),

l'*Agent_Attaque_Directe* doit suivre le nouveau périmètre. Pour ce faire, et dès que la présence d'un nouveau périmètre lui a été signalée par l'environnement de simulation, l'*Agent_Attaque_Directe* détecte l'*Agent_Fire* qui a comme attribut *step* (algorithme de la Figure 7.20) le numéro du dernier périmètre et qui est le plus proche de sa position. Cette détection se fait grâce au mécanisme de perception dont dispose l'*Agent_Attaque_Directe*. Ce dernier finit alors par se diriger vers le nouvel *Agent_Fire* détecté et suivre le nouveau périmètre (le dernier périmètre à avoir été généré par le simulateur Prometheus).

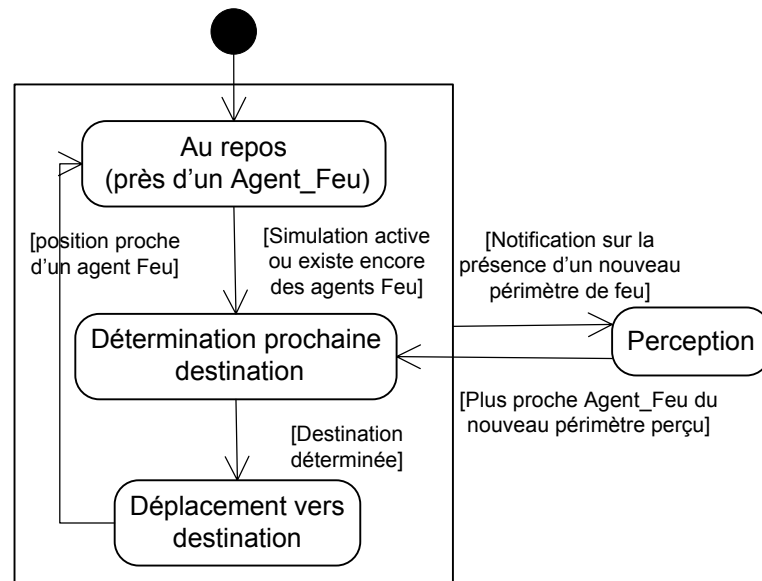


Figure 7.9. Diagramme d'état du fonctionnement de l'*Agent_Attaque_Directe*

Le processus illustré par le diagramme d'état de la Figure 7.9, est répété jusqu'au dernier périmètre généré par le simulateur Prometheus ou lorsque l'utilisateur humain décide d'arrêter la simulation.

Le diagramme de séquence de la Figure 7.10 donne un peu plus de détails sur l'implémentation de l'*Agent_Attaque_Directe*. L'utilisateur humain commence par lancer la simulation du feu à partir de l'interface (plus précisément la classe *TFirePPage*). La classe *TFire* (Section 3.4.), modérateur de la plupart des agents planificateurs dans l'application, crée alors un *Agent_Attaque_Directe* (instance de la classe *TBasicAgent*) ainsi que les états qui lui sont associés (surtout le *TRoutePlanningState* et le *TNavigationState*). L'itération des fonctions *MoveAroundAgent* et *MoveAgentAround*

permet à l'agent (sous la supervision de la classe TFire) de décider des mouvements à faire pour pouvoir suivre le périmètre du feu. La fonction *Goto* permet de déplacer l'agent étape par étape tout autour du feu. Lorsqu'un nouveau périmètre est disponible sur l'environnement de simulation, une fonction *FindAgent* permet de retrouver le pointeur sur l'*Agent_Fire* (Section 3.4.) le plus proche et que l'*Agent_Attaque_Directe* avait perçu (ceci n'est pas décrit dans la Figure 7.10 car complexe). C'est ainsi que l'*Agent_Attaque_Directe* peut se diriger vers le nouveau périmètre.

Le résultat de la simulation de l'avancée du bulldozer par l'*Agent_Attaque_Directe* donne au décideur humain une idée sur l'exécution du plan (l'attaque directe du feu) en mode simulé et accéléré, ce qui lui permet de prendre de meilleures décisions quant à la stratégie utilisée, les ressources déployées, le meilleur endroit pour creuser la ligne d'arrêt, etc.

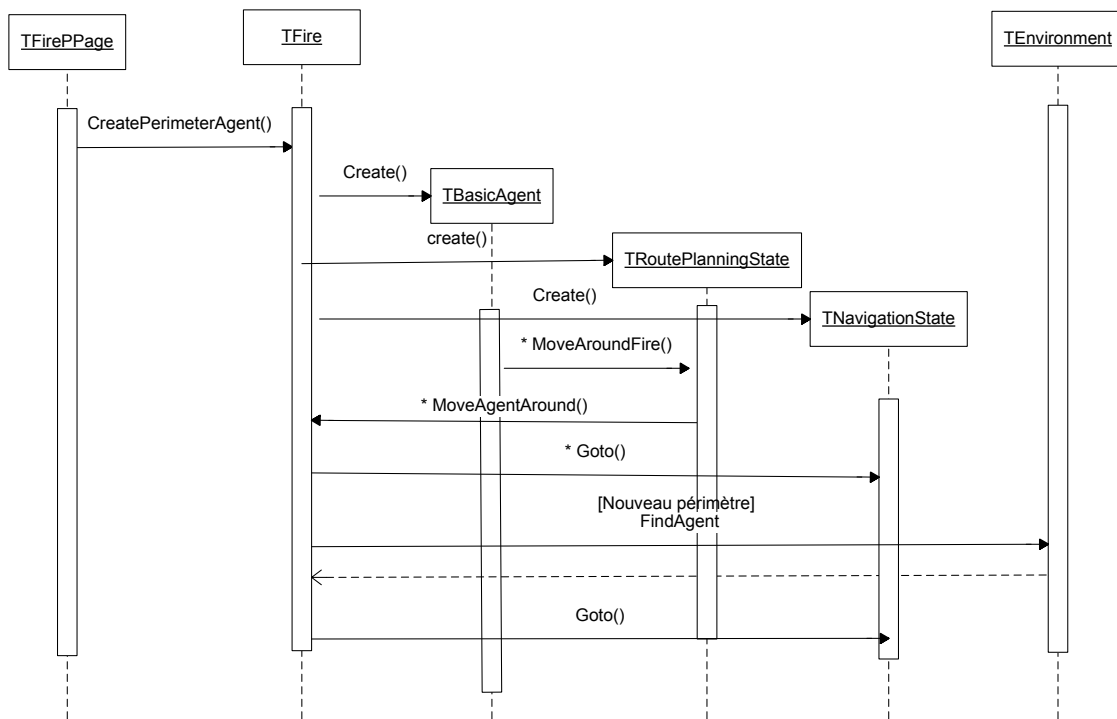


Figure 7.10. Diagramme de séquence décrivant le fonctionnement de l'*Agent_Attaque_Directe* lorsqu'un nouveau périmètre s'ajoute à l'environnement.

7.3.3.2. *Agent_Meilleur_Chemin*

Cet agent est assez simple. Lorsque le meilleur chemin est trouvé pour construire la ligne d'arrêt (surtout dans la stratégie indirecte), son tracé est sauvegardé dans un fichier nommé *BestPath.txt*. Il est à signaler que ce tracé peut changer pendant l'exécution du plan (si la replanification du chemin donne un chemin différent du plan initial, par exemple lors d'un imprévu ou après une *replanification anticipée périodique*, voir Chapitre 6). L'*Agent_Meilleur_Chemin* peut être sollicité à n'importe quel moment de la simulation (bien entendu, après qu'un premier plan sera établi) par l'utilisateur humain afin de lui permettre de visualiser le plan qui est en cours d'exécution.

Le fichier *BestPath.txt* est alimenté par les *Agents_Pathfinder* (voir Section 3.3.6. pour plus de détails) durant la phase « avant la construction de ligne d'arrêt » ou celle « pendant la construction de la ligne d'arrêt » (voir Scénarios, Section 1).

7.3.3.3. *Agent_Déjà_Fait*

Cet agent a la même structure que le précédent. Il accède à un fichier (*PathAlreadyDone.txt*) pour chercher les coordonnées du tracé d'un chemin que l'utilisateur humain désire visualiser. La seule différence est que le chemin est celui qui a été déjà exécuté, c'est à dire, déjà construit par les bulldozers sur le terrain.

L'autre différence est que le fichier *PathAlreadyDone.txt* est alimenté par les notifications périodiques qui proviennent des bulldozers sur le terrain et qui indiquent les positions courantes des bulldozers. Dans notre application nous simulons l'arrivée de ces notifications par une simple fenêtre de saisie de coordonnées.

L'*Agent_Déjà_Fait* bien évidemment, n'intervient que pendant la phase « pendant la construction de la ligne d'arrêt » car c'est là que les bulldozers auraient déjà exécuté une partie du plan.

7.3.3.4. *Agent_Vérificateur_Position*

Lorsqu'une notification sur la localisation courante du bulldozer arrive à l'environnement de simulation, elle est certes enregistrée dans le fichier *PathAlreadyDone.txt*, mais elle est surtout l'objet d'une vérification par un agent spécial nommé *Agent_Vérificateur_Position*. Ce dernier vérifie si le bulldozer sur le terrain est en train de suivre le plan qui lui a été acheminé ou non. Dans le premier cas, aucune action n'est déclenchée car l'exécution du plan est en train de se dérouler comme prévue. Si par contre, les données reçues signalent que le bulldozer est en train de dévier de son plan initial pour une raison ou pour une autre, une replanification du plan doit être déclenchée en tenant compte de la nouvelle position réelle du bulldozer sur le terrain. Le processus est illustré par le diagramme d'activité de la Figure 7.11. L'*Agent_Vérificateur_Position* intervient ainsi durant la phase « pendant la construction de la ligne d'arrêt ».

La première replanification déclenchée est une replanification rapide (assurée par la composante *Contrôle Rapide* de l'*Agent_Pathfinder*, Section 5.2.2 du Chapitre 5). Un raffinement du plan est ensuite effectué. Le processus de replanification est, dans les deux cas, assuré par des *Agents_Pathfinder*, et détaillé dans la sous-section 3.3.6.

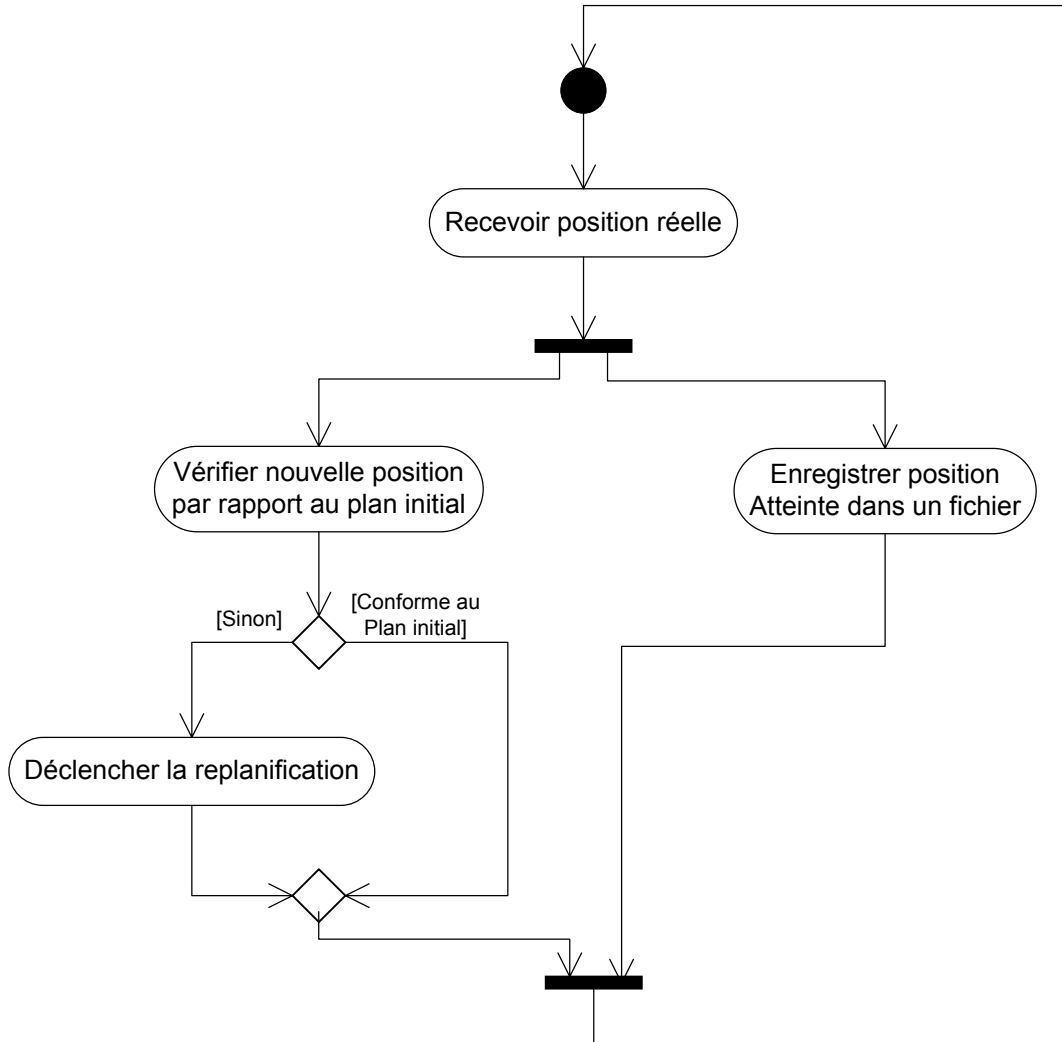


Figure 7.11. Diagramme d'activité décrivant le fonctionnement de l'Agent_Vérificateur_Position

7.3.3.5. Agent_Vérificateur_PlanVsObstacle

Comme nous l'avons déjà mentionné dans le Chapitre 5, l'utilisateur de l'environnement virtuel peut ajouter pendant la phase « pendant la construction de la ligne d'arrêt » un obstacle (infranchissables ou difficilement franchissable). Ceci se produit généralement suite au signalement d'un obstacle imprévu sur le terrain par les pompiers qui sont sur place. L'opérateur humain reproduit alors l'obstacle signalé sur l'environnement virtuel (grâce à un outil de dessin interactif que nous avons mis à disposition de l'utilisateur). Plus

précisément, une classe du module *Interface* nommée *TLineList* capte l'événement graphique (dessin de l'obstacle) et permet la création d'un agent logiciel particulier nommé *Agent_Vérificateur_PlanVsObstacle* qui sera chargé de trouver une solution au nouveau problème (diagramme de séquence de la Figure 7.13). Cet agent interpelle l'environnement (représenté par la classe *TEnvironment* dans la Figure 7.13) via la fonction *GetObstacle* afin de déterminer les extrémités de l'obstacle à contourner. Si l'obstacle ajouté n'affecte pas le plan initial (mémoire dans le fichier *BestPath.txt*), aucune action n'est entreprise. Dans le cas contraire, l'*Agent_Vérificateur_PlanVsObstacle* essaye de trouver un moyen (ou plusieurs moyens possibles) pour contourner l'obstacle. Pour ce faire, il se charge de percevoir l'obstacle afin d'en déterminer les extrémités. Le principe d'évitement de l'obstacle a déjà été annoncé à la Section 5.4.3 du Chapitre 5. Nous présentons ici l'algorithme à suivre.

Nous supposons que le plan initial (mémoire dans le fichier *BestPath.txt*) est l'ensemble des points destinations déterminés par l'*Agent Pathfinder* lors du processus de recherche du chemin. Ces points sont des minimums locaux puisque l'*Agent Pathfinder* s'est basé sur un calcul des coûts pour trouver à chaque étape le point vers lequel il doit se diriger (sous-section 3.3.6.). La position de blocage étant le $n^{\text{ième}}$ minimum local par lequel l'*Agent Pathfinder* serait passé s'il n'y avait pas l'obstacle (Section 5.4.3 du Chapitre 5). L'*Agent_Vérificateur_PlanVsObstacle* est donc un agent qui est délégué par l'*Agent_Pathfinder* afin de lui trouver les extrémités de l'obstacle. Lorsque l'obstacle affecte le plan initial, cet agent (*Agent_Vérificateur_PlanVsObstacle*) fonctionne selon l'algorithme de la Figure 7.12.

```

L'Agent_Vérificateur_PlanVsObstacle est initialement à Pb (Position de blocage de
l'Agent Pathfinder);
Tant que extrémité_déTECTÉE=faux et 1er minimum local non atteint Faire
    L'Agent_Vérificateur_PlanVsObstacle observe devant lui;
    Si une extrémité de l'obstacle est détectée Alors extrémité_déTECTÉE=Vrai
    Sinon
        L'Agent_Vérificateur_PlanVsObstacle recule vers le minimum local
        précédent;
    Fin Si
Fin Tant que
Si extrémité_déTECTÉE=Vrai alors
Faire un clone Agent Pathfinder Clone et le diriger vers l'extrémité trouvée;
Fin Si
Tant que 2ème_extrémité_déTECTÉE=faux 1er minimum local non atteint Faire
    L'Agent_Vérificateur_PlanVsObstacle observe devant lui;
    Si La 2ème extrémité de l'obstacle est détectée Alors
        2ème_extrémité_déTECTÉE=Vrai
    Sinon
        L'Agent_Vérificateur_PlanVsObstacle recule vers le minimum local
        précédent;
    Fin Si
Fin Tant que
Si 2ème_extrémité_déTECTÉE=Vrai alors
Agent Pathfinder se dirige vers la 2ème extrémité;
Fin Si
L'Agent_Vérificateur_PlanVsObstacle s'auto-détruit.
Fin.

```

Figure 7.12. Algorithme de recherche des extrémités d'un obstacle par
l'Agent_Vérificateur_PlanVsObstacle

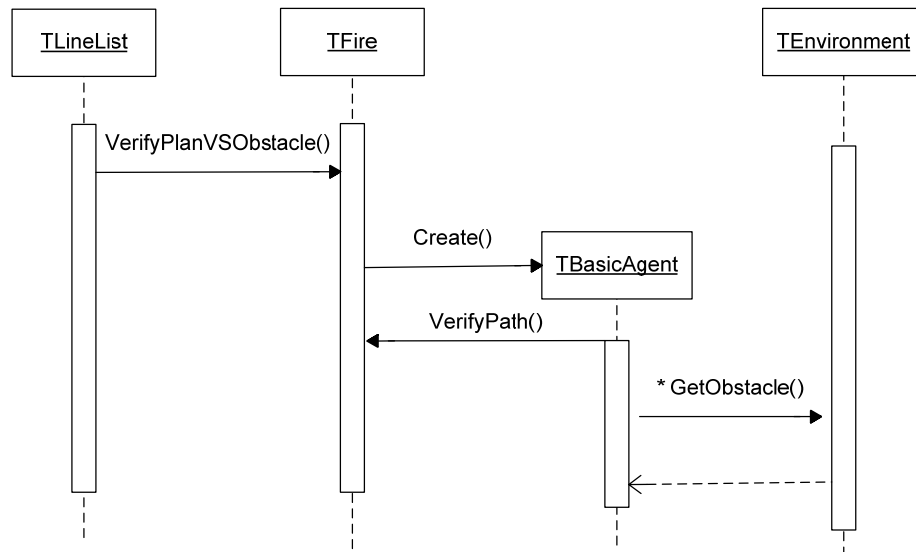


Figure 7.13. Diagramme de séquence décrivant le déclenchement de l'*Agent_Vérificateur_PlanVsObstacle*

7.3.3.6. *Agent_Pathfinder*

Le fonctionnement de l'*Agent_Pathfinder* a déjà été discuté dans le Chapitre 5. Nous nous contentons ici de présenter les détails de l'implémentation. Cet agent intervient essentiellement dans la stratégie d'attaque indirecte et pendant les deux phases : « avant la construction de ligne d'arrêt » et « pendant la construction de la ligne d'arrêt ».

L'utilisateur humain en interagissant avec l'interface de l'application (plus particulièrement la classe *TFirePPage*), peut lancer le processus de recherche du meilleur chemin pour une stratégie indirecte. La classe *TFire* (classe principale du module *Fire*) crée alors plusieurs *Agents_Pathfinder* (des instances de la classe *TBasicAgent*, qui définit l'agent MAGS) ayant pour points de départ des points différents de la zone qui entoure le début de l'esquisse dessinée auparavant par le planificateur humain. En spécifiant dans les paramètres de création l'état *A_ROUTEPLANNING_STATE*, chaque *Agent_Pathfinder* possèdera l'état indiqué (la classe *TRoutePlanningState*). Il peut alors évoquer la classe d'un autre état (*TNavigationState*) afin de pouvoir se déplacer dans l'environnement,

comme il peut se cloner afin d'investiguer d'autres possibilités (dans le cas d'une planification raffinée). Ce processus est décrit par le diagramme de séquence de la Figure 7.14. La vitesse de progression de l'*Agent_Pathfinder* dépend de la nature du terrain. Nous obtiendrons ainsi une simulation plausible de l'avancée du bulldozer par rapport à la progression des feux lors d'une attaque indirecte.

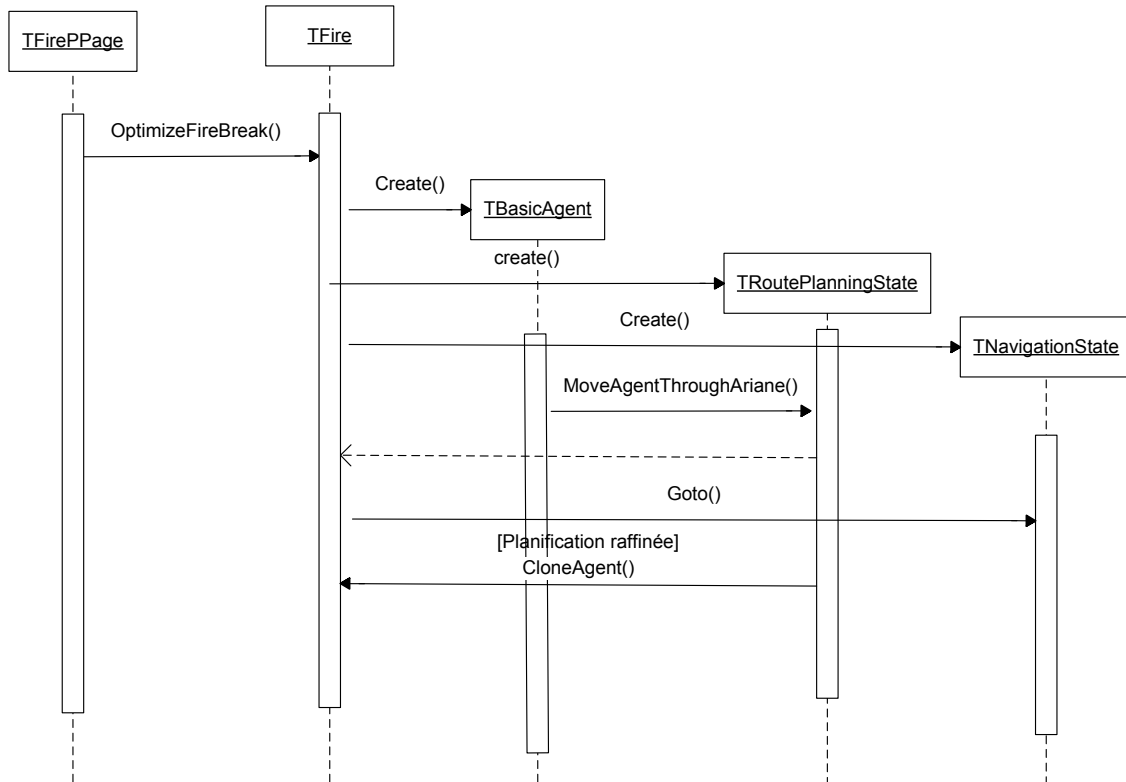


Figure 7.14. Diagramme de séquence décrivant le fonctionnement de l'*Agent_Pathfinder* en cas de planification *raffinée*

Toutefois, l'utilisateur n'est pas le seul à pouvoir lancer (directement) une planification du chemin. Comme nous l'avons déjà indiqué précédemment, l'*Agent_Vérificateur_Position* peut déclencher la replanification du chemin (si la position courante du bulldozer ne coïncide pas avec les données du plan initial). Dans ce cas, l'*Agent_Vérificateur_Position* initialise la planification rapide (via la classe *TaskXY*) en appelant la fonction *QuickReplan* de la classe *TFire* (Section 4.3.). Cette dernière crée alors un *Agent_Pathfinder* avec l'état

TRoutePlanningState qui lui permet de replanifier le reste du chemin (via la fonction *MoveAgentThroughAriane*). Une fois le plan établi, et avant de s'auto-détruire, cet *Agent_Pathfinder* initialise la replanification raffinée (via un appel de la fonction *RePlan*) qui, bien entendu, va nécessiter la création de plusieurs *Agents_Pathfinder* (clones) comme le montre le diagramme de séquence de la Figure 7.15.

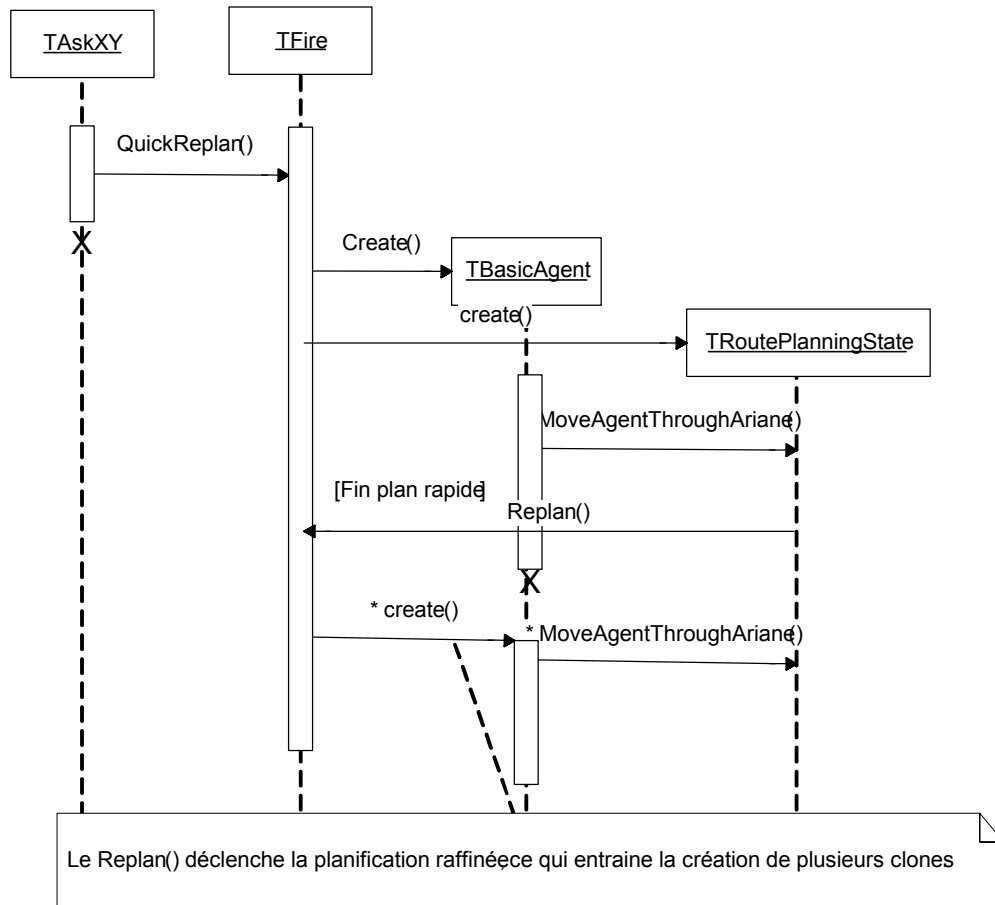


Figure 7.15. Diagramme de séquence décrivant le cas où l'*Agent_Vérificateur_Position* déclenche une replanification

Lorsqu'un *Agent_Pathfinder* possède l'état *A_ROUTEPLANNING_STATE*, l'objet de la classe *TRoutePlanningState* qui lui est associé, est mis à jour à chaque unité de temps de simulation. Cette mise à jour est en fait l'exécution d'une fonction qui détermine pour l'*Agent_Pathfinder* ce qu'il doit faire après chaque unité de temps écoulée de la simulation. Cette fonction nommée *MoveAgentThroughAriane* est évoquée si l'agent en question est un

Agent_Pathfinder (nous avons vu précédemment qu'une autre fonction de *TRoutePlanningState* est évoquée lorsqu'il s'agit d'un *Agent_Attaque_Directe*). L'algorithme de *MoveAgentThroughAriane* est présenté dans la Figure 7.17.

Comme le montre cet algorithme, l'*Agent_Pathfinder* doit chercher n directions que l'agent pourrait suivre et qui respectent certaines contraintes géométriques. Ces contraintes évoquées au Chapitre 5 peuvent être résumées comme suit :

- Une direction ne doit pas éloigner beaucoup l'agent de l'esquisse dessinée par le planificateur humain.
- Une direction doit aller dans le même sens que l'objectif final (et non pas dans le sens contraire).
- Une direction doit être pertinente, donc assez différente des autres $n-1$ directions.
- Une direction ne doit pas varier beaucoup par rapport à la dernière direction empruntée par l'agent (pour éviter les courbes étroites).

Toutes ces contraintes ont été traduites par une fonction *GetBestDirection* (correspond à 'Chercher les n directions possibles' dans l'algorithme de la Figure 7.16) de l'*Agent_Pathfinder*. Le n est déterminé selon le contexte dans lequel évolue l'agent courant (l'*Agent_Pathfinder* qui est en train d'exécuter l'algorithme). S'il s'agit d'une planification rapide (Section 5.2.2. du Chapitre 5), le n a tendance à être petit (dans notre prototype il est égal à 2). Si par contre la planification est raffinée, le n est un peu plus grand (dans notre prototype il est égal à 5). En effet, dans ce dernier cas, le système dispose de plus de temps et peut donc se permettre d'examiner plusieurs chemins possibles.

Par la suite, les coûts relatifs aux n directions possibles sont calculés. Cependant, seules les m ($m \leq n$) directions ayant les meilleurs coûts, sont retenues. En fait, dépendamment de la nature de la planification menée (rapide ou raffinée), l'*Agent_Pathfinder* peut décider ou non de cloner d'autres agents comme lui, afin d'investiguer le maximum de chemins (ou directions) possibles. Si la planification est raffinée, il peut donc se permettre de générer des clones comme le montre l'algorithme de la Figure 7.17. Si par contre, la planification

est de type rapide, il se contente de choisir le meilleur chemin et de l'emprunter lui-même (dans ce cas, le m est égal à 1). Dans le premier cas (planification raffinée), m est supérieur à 1 mais sa valeur varie pendant la simulation. En effet, étant donné que le clonage est récursif (un clone peut par la suite à son tour créer un autre clone), nous risquons de voir le nombre de clones augmenter de façon exponentielle. Il est nécessaire donc de contrôler cette éventuelle explosion combinatoire. Dans ce cadre, le m agit comme un régulateur de clonage. Par exemple, si le nombre de clones présents dans l'environnement de simulation est bas, le m peut avoir une valeur égale à 3 (c'est à dire que l'*Agent_Pathfinder* a le droit de générer jusqu'à 3 clones à chaque fois). Si par contre, le nombre est élevé (ce qui risque de ralentir considérablement la simulation et de dépasser les limites permises), le m verra sa valeur diminuer.

Toutefois, il ne s'agit pas du seul mécanisme de contrôle de clonage. Il est parfois plus judicieux de tuer des clones (inutiles) que de limiter le nombre des nouveaux clones. En effet, il arrive souvent que des clones se rencontrent dans le même point mais avec des coûts différents. Nous avons alors pensé à un autre mécanisme complémentaire au premier, et qui va débarrasser la simulation des clones inutiles. Ce mécanisme est explicité dans l'algorithme de la Figure 7.17. Il s'agit tout simplement de tuer le clone qui arrive au même point qu'un autre agent, mais avec un coût cumulatif (coût du chemin déjà parcouru avant d'arriver à ce point) plus élevé. Il est à noter qu'une autre condition doit être vérifiée : il faut que les deux agents aient des directions assez similaires. En effet, dans le cas contraire (les deux agents arrivent au même point avec des coûts différents mais avec des directions futures différentes), les deux agents sont maintenus en vie car ils vont peut-être emprunter des chemins non similaires et donc celui qui avait un coût plus élevé pourrait très bien finir avec un coût total moins élevé s'il emprunte une autre direction très différente de l'autre agent.

Enfin, nous notons (algorithme de la Figure 7.17) que le processus de clonage implique que l'agent père (l'agent qui a créé le clone) passe à son fils (l'agent qui vient d'être créé) certains paramètres lui permettant de continuer son chemin comme s'il était là dès le début de la simulation. Les paramètres les plus importants sont :

- Le coût cumulé jusqu'au point où le clone a été créé.
- Les points déjà parcourus par l'agent père.
- La première direction que doit emprunter le clone et qui vient juste d'être calculée par le père.

Ces paramètres sont passés entre le père et le fils grâce aux états statiques de l'agent (sous-section 7.2.4). Il s'agit de la classe *TStaticState* prévue pour ajouter différents types d'états à l'agent. L'agent père conserve déjà les informations sur les points parcourus et sur le coût du chemin parcouru (coût cumulé) grâce à cette structure.

Par exemple, la Figure 7.16 montre comment attribuer à un nouvel agent clone le coût cumulé de son père (*StartingCost*).

Cette structure est aussi utilisée pour reconnaître le type de l'*Agent_Pathfinder* présent. En effet, un attribut nommé *AgentMode*, conserve le type de l'agent. Si la valeur de *AgentMode* est à 1, cela veut dire que cet agent est en train de contourner un obstacle. Si la valeur est 2, cela veut dire qu'il est en train de mener une planification raffinée, etc.

```

/*Pour enregistrer le coût dans un état statique*/
agent→GetStates()→CreateStaticState("cost", 1, M_INTEGER);
agent→GetStates()→GetStaticState(1) →SetValue(StartingCost);

/*Pour récupérer le coût*/
STATEVALUE CumulativeCost= agent→GetStates()→GetStaticState(1)→
GetValue(M_INTEGER, 1);
int CumulativeCost Int= CumulativeCost.iValue;

```

Figure 7.16. Utilisation du *TStaticState* pour conserver et consulter les paramètres des agents

Ajouter la position courante de l'agent à la liste des points parcourus

*/*Mécanisme de contrôle de l'explosion combinatoire des clones */*

Si il s'agit d'une planification raffinée **alors**

Chercher les agents coexistants (les agents qui sont sur la même cellule que l'agent courant)

Pour chaque agent coexistant **Faire**

Si il a presque la même direction destination que l'agent courant **alors**

Si coûtCumulatif(agent coexistant) < coûtCumulatif(agent courant) **alors**

L'agent courant s'auto-détruit

Sortir de la boucle

Sinon Tuer l'agent coexistant

Fin Si

Fin Pour chaque

Fin Si

Si il s'agit d'un contournement d'obstacle (cf. explication sur

Agent_Vérificateur_PlanVsObstacle) **alors**

Se diriger vers l'extrémité proposée par *Agent_Vérificateur_PlanVsObstacle*

Vérifier si l'agent a atteint ou non cette extrémité

Fin Si

/ Mécanisme de recherche de la meilleure direction à emprunter par l'agent*/*

Tant que l'agent n'est pas encore rendu à destination **Faire**

Si il s'agit d'un contournement d'obstacle **alors**

meilleure direction est celle déjà donnée par

Agent_Vérificateur_PlanVsObstacle

SINON

Déterminer la *direction limite* (voir Section 5.3.3 du Cahpitre 5) qu'il faut respecter pour ne pas aller en direction du feu.

Chercher les *n* directions possibles (ceci fait appel à un calcul géométrique et à la considération de l'esquisse existante)

Fin Si

Calculer les coûts relatifs à chaque direction possible

Conserver les *m* meilleures directions (cf. explications pour le *m*)

Avancer l'agent selon la première meilleure direction

Si il s'agit d'une planification raffinée **alors**

Cloner (*m-1*) Agents_Pathfinder

Passer à ces clones en paramètres : le coûtCumulatif(agent courant), les points déjà parcourus par l'agent courant et la 1ère direction à suivre.

Fin Si

Fin Tant que

Si l'agent est arrivé à sa destination finale **alors**
 Il s'arrête et attend que tous les autres *Agents_Pathfinders* terminent pour s'auto-détruire (il attend parce qu'il conserve son plan au cas où il serait choisi comme meilleur chemin)

Fin Si

Figure 7.17. Algorithme de *MoveAgentThroughAriane* suivi par les *Agents_Pathfinder*

7.3.4. Module Fire

Dans cette section, nous présentons les deux principales tâches dont le module *Fire* est responsable, à savoir : orchestrer les agents intervenant dans la planification de la ligne d'arrêt et assurer le lien avec le simulateur des feux Prometheus.

7.3.4.1. 'Chef d'orchestre des agents'

Le module *Fire* (constitué principalement de la classe *TFire*) est en quelque sorte le 'chef d'orchestre' des agents participant dans la planification de la ligne d'arrêt. C'est lui qui est responsable de la création de ces agents et de leur attribuer les tâches à réaliser. Par exemple :

- La fonction *QuickReplan* crée l'*Agent_Pathfinder* avec le mode 1 (pour une planification raffinée)
- La fonction *QuickReplan* crée l'*Agent_Pathfinder* avec *AgentMode* à 3 (pour une planification rapide).
- La fonction *Replan* crée l'*Agent_Pathfinder* avec *AgentMode* à 2 (pour une planification raffinée).
- La fonction *VerifyPlanObstacle* crée l'*Agent_Verificateur_PlanVsObstacle* afin de déterminer si l'obstacle qui vient d'être ajouté affecte ou non le plan initial. Dans le cas où l'obstacle affecterait le plan initial, elle crée un *Agent_Pathfinder* avec *AgentMode* à 1 (pour une planification avec contournement d'obstacle).

- La fonction *LoadDonePath* crée l'*Agent_déjà_fait* qui se charge de reproduire la portion de plan déjà exécutée.
- La fonction *LoadBestPath* crée l'*Agent_Meilleur_Chemin* qui se charge de reproduire le plan qui reste à exécuter.
- La fonction *MoveAgentAround* crée l'*Agent_Attaque_Directe* qui se charge de suivre les périmètres de feu lors de l'application de la stratégie directe.

7.3.4.2. Intermédiaire entre Prometheus et MAGS

L'une des fonctions principales du module est d'assurer le lien entre le simulateur des feux Prometheus et la plate-forme de simulation. La fonction *CreateScenarioTFirePrometheus* fait ainsi appel à l'interface COM de Prometheus afin de créer un scénario de feu (initialiser les données sur l'environnement : le terrain, la météo et le feu déclencheur). Pour ce faire, il faut tout d'abord importer la librairie de Prometheus (Figure 7.18). Ensuite, *CreateScenarioTFirePrometheus* définit tous les paramètres nécessaires pour que Prometheus puisse simuler un feu (Figure 7.19)

```
# import "..\data\wildfire\prometheusCOM.dll"
using namespace PROMETHEUSCOMLib;
```

Figure 7.18. Importation de la librairie de Prometheus en code C++

La fonction *CreatePerimeterAgent* importe par la suite les résultats de la simulation effectuée par le moteur de Prometheus (Figure 7.20). Ces résultats se présentent en fait sous forme d'un ensemble de périmètres. Chaque périmètre est composé d'une suite de sommets. Chaque sommet est une structure de données renfermant les paramètres les plus pertinents du feu à un point précis sur le périmètre. Parmi ces données : l'intensité du feu, la vitesse de propagation, la direction d'avancement du feu, etc. *CreatePerimeterAgent* crée alors pour chaque sommet un agent appelé *Agent_Fire* (Figure 7.20). Les *Agents_Fire* créés, en encapsulant les informations pertinentes sur le feu à chaque point du périmètre, permettent aux agents planificateurs et notamment les *Agents_Pathfinder* de bénéficier de

ces données. En effet, lors de la recherche de chemin, un *Agent_Pathfinder* percevant un *Agent_Fire*, en extrait les informations qui lui sont nécessaires pour calculer le coût des directions (Section 5.3.3 du Chapitre 5). Les *Agents_Fire* peuvent être plus proactifs en envoyant des messages d'alertes aux autres agents planificateurs présents dans l'entourage.

```

/* Initialisation de l'appel à l'interface COM de Prometheus*/
CoInitialize(NULL);
pApp.CreateInstance(CLSID_PromeApp);
/* Initialisation de la position géographique de la région en feu*/
HRESULT hr=pApp→Initialize(51.7,-115.4,3,23);
/* Initialisation de la date de la simulation*/
CString csScenarioStart("20/06/2000:10:00:00"),csScenarioEnd("20/06/2000:20:00:00");
...
/* Initialisation des données sur le terrain */
pApp→ImportProjection("C:\\Program Files\\prometheus\\examples\\Dogrib-
1TestDataSet\\projection.prj");
/* Initialisation des données sur l'essence */
pApp→ImportFuelGrid("Test", "C:\\Program Files\\prometheus\\examples\\Dogrib-
1TestDataSet\\FBPfueltype25.asc");
/* Initialisation des données sur la météo */
pApp→CreateWeatherStation("First", 52, -115, 500);
pApp→CreateWeatherStream("First", "First", 85, 15, 25, 200,"20/06/2000:00:00:00");
pApp→EditWeatherDaily( "First", "20/06/2000:00:00:00", 15, 25, 15, 25, 20, 0, 180);
/* Initialisation du scénario */
pApp→SetInitialFFMC( "First", "First", 85, 15);
pApp→SetStreamFFMCMethod( "First", "First", 0 );
pApp→CreateScenario("First");
pApp→SetScenarioIntervals("First", 600, 600);
pApp→SetScenarioGrid("First","Test");
pApp→SetScenarioTime("First", bScenarioStart,bScenarioEnd);
/* Initialisation du point de feu déclencheur d'incendie */
pApp→CreateIgnitionPoint("FirstFire", "20/06/2000:9:00:00", 2485.36, 2900);
pApp→SetScenarioFire( "First", "FirstFire");
pApp→SetScenarioStream("First", "First", "First");
/* Lancement de simulation : simulation d'une seule étape ici */
pApp→SimulateRestart( "First" );
long stepResult, LastIndex= 0;
pApp→SimulateStep( "First", &stepResult);

```

Figure 7.19. Extrait du code C++ de la fonction *CreateScenarioTFirePrometheus*, lors de l'appel des fonctions de l'interface COM de Prometheus


```

while( stepResult == 0) /* Vérifie si le moteur de Prometheus donne un résultat*/
{
    long NumPoints;
    /* Fonction COM qui donne le nbre de sommets d'un périmètre*/
    NumPoints = pApp→GetNumOfNewVertices("First");
    for(int i = 0;i<NumPoints; i++) /* pour tout sommet du périmètre*/
    {
        double X,Y, ros, fi, raz, sfc, cfb, cfc, tfc;
        /* Fonction COM qui donne la position X Y d'un sommet d'index donné*/
        pApp→GetIndexVertice("First", i + LastIndex, &X, &Y);
        /* Fonction COM qui donne les caractéristiques du feu au sommet X Y*/
        pApp→GetXYParameters("First", X, Y, &ros, &fi, &raz, &sfc,
        &cfb, &cfc, &tfc);
        Vertex=new ChampsFeu;
        Vertex→X=X; Vertex→Y=Y;
        Vertex→step=step;
        Vertex→RateOfSpread=ros;
        Vertex→Intensity=fi;
        m_ChampsFeu.Add(Vertex);
        int Xi, Yi;    Xi=(int)X;        Yi=(int)Y;
        /* Création de l'agent Agent_fire*/
        TAction *action = new TAction;
        TBasicAgent* agent;
        action→Init(CREATE_AGENT,"Create Mobile Agent");
        action→AddParameter(i+ LastIndex); // ID
        action→AddParameter(Xi); // Posx
        action→AddParameter(Yi); // PosY
        action→AddParameter(20); // Perception
        action→AddParameter(A_NAVIGATION_STATE |
        A_MEMORY_STATE | A_LOCATION_STATE|
        A_PROFILE_STATE | A_VISUALISATION_STATE ); ...
        /* Attribution d'un des paramètres de feu (numéro étape) à l'Agent_Feu*/
        agent→GetStates()→CreateStaticState("step", 0, M_INTEGER);
        TStaticState* staticState = agent→GetStates()→etStaticState(0);
        staticState→SetValue(step); ....
    }
    LastIndex += NumPoints; step+=1;
    /* Simulation de la prochaine étape (prochain périmètre)*/
    pApp→SimulateStep ("First", &stepResult);
}

```

Figure 7.20. Extrait du code C++ de la fonction *CreatePerimeterAgent* lors de l'extraction des résultats de Prometheus et la création des *Agents_Fire*

7.4. Discussion et Conclusion

Dans ce chapitre, nous avons présenté les modules les plus importants de la réalisation de notre application. Pour ce faire, nous avons utilisé une plate-forme de géosimulation à base d'agents logiciels (MAGS). Notre travail a consisté donc à intégrer l'application dans la plate-forme. Le produit de notre réalisation reste cependant un simple prototype qui a besoin de plusieurs améliorations pour pouvoir servir dans des situations réelles. En effet, il va sans dire qu'une application de planification des opérations de lutte contre les feux de forêts, ne peut être réalisée par aussi peu de ressources (humaines et financières). Une réalisation plus professionnelle nécessiterait la collaboration de plusieurs experts (dont des experts en interaction personne-machine) pour assurer au produit livrable la fiabilité, la convivialité, l'efficacité, et la précision des résultats. De toute façon, ceci n'était pas notre but de départ. Notre objectif était cependant d'illustrer notre approche de planification par simulation à base d'agents, et non pas d'implémenter une application complète pour les opérations de lutte contre les feux.

Bien qu'il ne s'agisse que d'un prototype, sa réalisation était loin d'être facile. L'intégration de nouveaux modules et fonctionnalités dans MAGS était assez difficile à faire et ce pour plusieurs raisons dont les deux suivantes:

- MAGS est une plate-forme de simulation, donc les agents MAGS sont conçus pour simuler un comportement ou un phénomène. Les agents ont donc besoin de capacités « primitives » pour percevoir, mémoriser, naviguer, etc. Dans notre application par contre, nous faisons de la simulation non pas pour le seul but de simuler, mais surtout pour planifier. La plupart des comportements de base des agents MAGS ne peuvent ainsi être utiles sans une réadaptation. En ce qui concerne la navigation des agents par exemple, MAGS permet aux agents de naviguer selon plusieurs modes, mais des modes qui restent primitifs (sans planification). A l'opposé, la navigation des agents pour notre application doit être réfléchie et planifiée.

- La spécification de l'environnement dans MAGS est assurée par une interface. Cette dernière permet de définir les différentes couches qui constituent l'environnement physique. Cependant, les besoins de notre application en terme de données sur l'environnement sont différents et surtout très spécifiques. Ceci nous a donc privé d'utiliser cette interface et nous a poussé à implémenter la partie qui nous permet de définir les données sur l'environnement. Dans ce cadre, il serait intéressant d'améliorer la plate-forme MAGS en offrant la possibilité de définir les couches sur les données de l'environnement de façon plus flexible et générique. Par exemple, dans la version actuelle de MAGS, une des couches constituant l'environnement, est dédiée à la définition des obstacles. Une cellule de l'environnement de simulation est donc soit un obstacle, soit elle ne l'est pas. Cependant, pour certaines applications particulières, les choses sont beaucoup moins simples. En effet, ce qui peut être un obstacle pour un agent, peut ne pas l'être pour un autre type d'agent. Si nous considérons notre application comme exemple, une cellule qui fait partie d'une rivière est considérée comme un obstacle pour un *Agent_Pathfinder* (représentant un bulldozer), mais ne l'est pas pour un *Agent_Barque*. De plus, les obstacles ne sont pas tous de même niveau de difficulté. A titre d'exemple, une cellule qui a pour végétation (ou type d'essence) du 'sapin' est plus difficile à franchir pour un *Agent_Pathfinder* qu'une autre cellule ayant pour végétation 'herbe'. Il serait donc intéressant de définir les données sur l'environnement de façon objective, c'est à dire, juste décrire le terrain, et distinguer plus tard ce qui est obstacle de qui ne l'est pas dépendamment de l'agent en question. Ceci revient à inclure la définition des obstacles dans les caractéristiques des agents.

Ceci étant dit, la plate-forme MAGS, malgré ces difficultés, nous a été d'une grande utilité pour la réalisation de notre application, surtout pour assurer les propriétés du paradigme agent au niveau computationnel. Nous avons déjà mentionné au Chapitre 2 les limites de la Simulation Orientée Agent. Nous avons entre autres, cité les études de Drogoul et ses collègues [Drogoul et al., 02] qui démontrent que les propriétés des agents (autonomie, proactivité et interaction) définies à un niveau métaphorique, ne sont pas traduites en propriétés computationnelles. MAGS, par contre, utilise vraiment des agents

computationnels tels que décrit par [Jennings 00a]. Toutefois, les propriétés de l'agent n'ont pas toutes eu la même considération dans MAGS. Par exemple, l'interaction entre les agents est encore (dans la version actuelle de MAGS) assez basique et doit donc faire l'objet de plusieurs améliorations afin d'offrir une meilleure interaction entre les agents.

Les résultats obtenus par notre prototype ainsi que la procédure de validation, sont présentés dans le chapitre suivant.

CHAPITRE 8

Résultats et Validation

Le but de ce chapitre est de valider une partie de notre approche. Pour ce faire, nous implémentons un outil logiciel (basé sur la simulation) qui permet de comparer les solutions fournies par les agents de notre application à celle élaborées par des planificateurs humains.

Lorsqu'on utilise la simulation pour atteindre un certain objectif, une condition fondamentale pour l'adoption du modèle est la garantie que les résultats du modèle de simulation soient acceptés en toute confiance. L'établissement de cette confiance vis-à-vis des résultats est associé à deux activités distinctes à savoir la *vérification* et la *validation* [Pegden et al., 95]. Certains chercheurs rajoutent une troisième activité : la mesure de la *fidélité*. Dans la première section de ce chapitre, nous présentons un bref aperçu bibliographique sur ces trois aspects (*fidélité*, *vérification* et *validation*) tout en indiquant comment nous avons procédé dans notre cas pour satisfaire ces critères de validité. La deuxième section est consacrée à la présentation des scénarios que nous avons montés pour l'étape de la *validation*. La troisième est dernière section, présente les résultats de cette validation.

8.1. Critères de validité

8.1.1. Fidélité de la simulation

La *fidélité* est le niveau de réalisme qu'offre la simulation [Feinstein et Cannon 02]. Ce concept constitue une composante intégrale dans la simulation.

La simulation est généralement utilisée pour imiter un comportement social ou un phénomène. La *fidélité* dans ce cas consiste à ressembler le plus possible à ce qui se passe en réalité. Cependant, le but de la simulation dans notre approche, est de planifier et non

pas de simuler au sens propre du terme. La simulation dans notre cas est un moyen et non pas un objectif. La mesure de la *fidélité* de la simulation revient par contre à mesurer la fidélité de l'environnement de simulation (aussi bien son aspect statique que dynamique) par rapport à l'environnement réel.

Dans notre application, la *fidélité* de la simulation dépend des facteurs suivants:

- La fiabilité et le degré d'exactitude des données SIG utilisées pour former l'environnement physique (la première couche de l'architecture ENCASMA).
- L'efficacité du modèle Prometheus dans la prédiction de l'avancée des feux.
- La fiabilité des moyens de communication reliant le monde réel au monde de simulation.

Ces trois facteurs sont en fait extrinsèques à notre système car le degré de performance ou de fiabilité de ses éléments dépend fortement du progrès technologique (surtout en matière de communication sans fil, observation et mise à jour des données géographiques, élaboration des formules mathématiques de propagation des feux).

8.1.2. Vérification de la simulation

La vérification est le processus d'évaluation qui indique si le modèle de simulation fonctionne comme prévu [Feinstein et Cannon 02]. Le processus de vérification implique le débogage du modèle en isolant et en éliminant le plus d'erreurs possible. Typiquement, les développeurs utilisent tout d'abord les tests alpha et par la suite font exécuter plusieurs simulations sous différentes conditions à des utilisateurs indépendants (tests Béta). Dans des termes plus simples, la vérification être faite par plusieurs moyens :

- Utiliser un débogueur interne du logiciel de simulation.
- Observer les rapports en sortie.
- Évaluer étape par étape les traces de l'exécution de la simulation.
- Engager des individus pour évaluer la simulation. etc.

Pour notre prototype décrit dans le chapitre précédent, nous avons évalué les traces de l'exécution et nous avons surtout observé les résultats en sortie.

Pour ce qui est du scénario de la stratégie directe (voir Section 1.1 du Chapitre 7), nous avons vérifié si l'*Agent_Attaque_Directe* suit bien le périmètre du feu (Figure 8.1). Nous avons également noté que lorsqu'un nouveau périmètre s'affiche, l'*Agent_Attaque_Directe* perçoit avec succès l'*Agent_Fire* le plus proche de lui et se dirige donc vers cet agent.

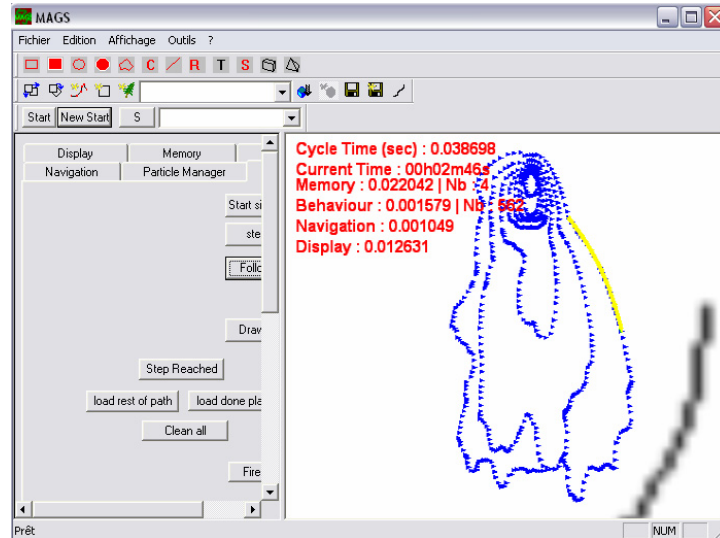


Figure 8.1. L'*Agent_Attaque_Directe* qui suit un périmètre de feu avec succès (snapshot)

Pour ce qui est de la stratégie indirecte et plus précisément du scénario « avant la construction de ligne d'arrêt », nous avons examiné la trace de la simulation (la trace des chemins trouvés par les *Agents_Pathfinder*). Nous avons essayé de lancer des *Agents_Pathfinder* sans la capacité de clonage (Figure 8.2) et ensuite avec l'option clonage (Figure 8.3). Lorsque les agents ne se clonent pas, nous obtenons tout de même un chemin acceptable (le jugement ici est qualitatif car il tient compte non seulement du coût du chemin mais aussi de sa conformité à l'esquisse dessinée par l'utilisateur). Le temps de simulation nécessaire pour qu'une dizaine d'*Agents_Pathfinder* terminent leur recherche de chemin sans se cloner, est de quelques minutes. Ce temps dépend fortement de la longueur du chemin à tracer, mais très peu du nombre d'*Agents_Pathfinder* utilisés (tant que ce nombre reste petit bien entendu).

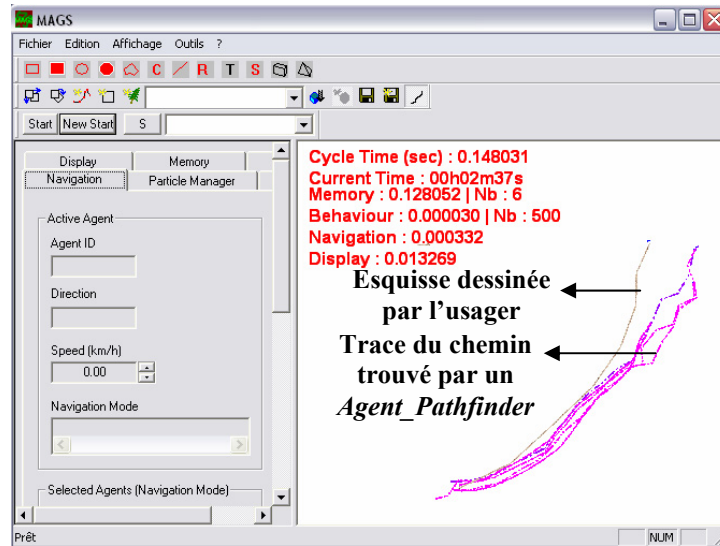


Figure 8.2. Exemple de recherche de chemin sans clonage.

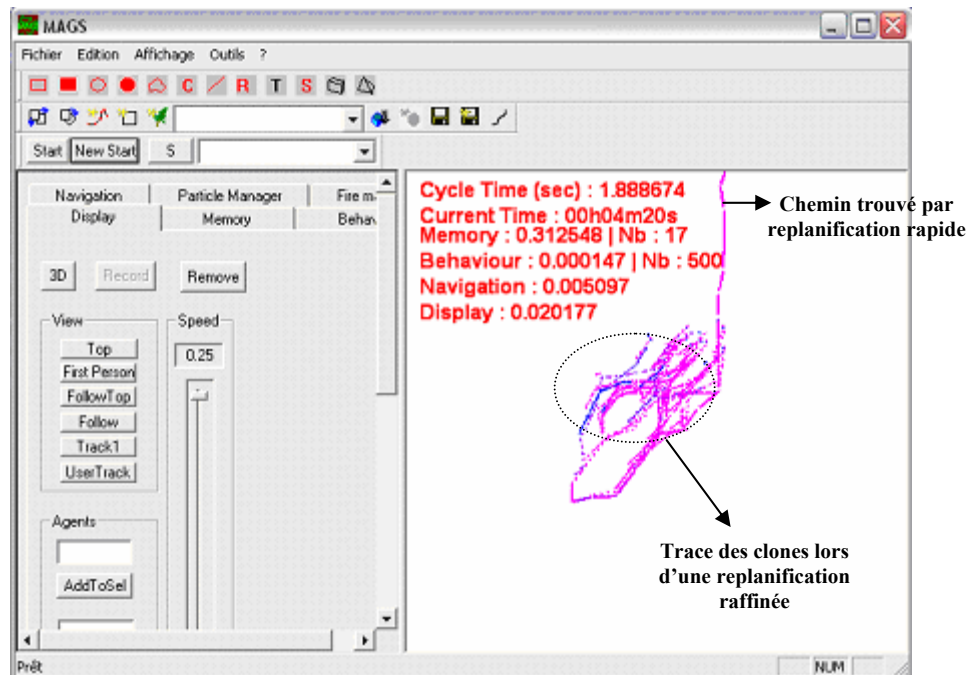


Figure 8.3. Exemple de recherche de chemin avec clonage (ici cas d'une replanification raffinée)

Quant à la recherche de chemin avec possibilité de clonage, le temps de simulation est plus important. A titre d'exemple, si chaque *Agent_Pathfinder* peut cloner 2 agents à chaque étape (selon l'algorithme de *MoveAgentThroughAriane* de la Figure 7.17 du Chapitre 7,

ceci correspond à $m=3$), le temps de simulation est d'environ 10 fois plus qu'avec des *Agents_Pathfinder* sans clonage, et ce pour des chemins de longueur raisonnable. En effet, dans le cas où le clonage est permis, la fonction temps de simulation n'est pas une fonction linéaire par rapport à la longueur de l'esquisse. Plus le chemin recherché est long, plus le temps de simulation croît considérablement si nous ne contrôlons pas l'explosion combinatoire des clones. Avec les mécanismes de contrôle de clonage décrits dans le Chapitre 7 (Section 7.3.3.6), nous arrivons à rendre la fonction de simulation quasi linéaire (par rapport à la longueur du chemin).

Pour ce qui est du scénario « pendant la construction de la ligne d'arrêt », nous avons vérifié que l'*Agent_Pathfinder* (et ses clones) réagit bien lorsque l'utilisateur humain rajoute un obstacle qui affecte le plan initial (Figure 8.4).

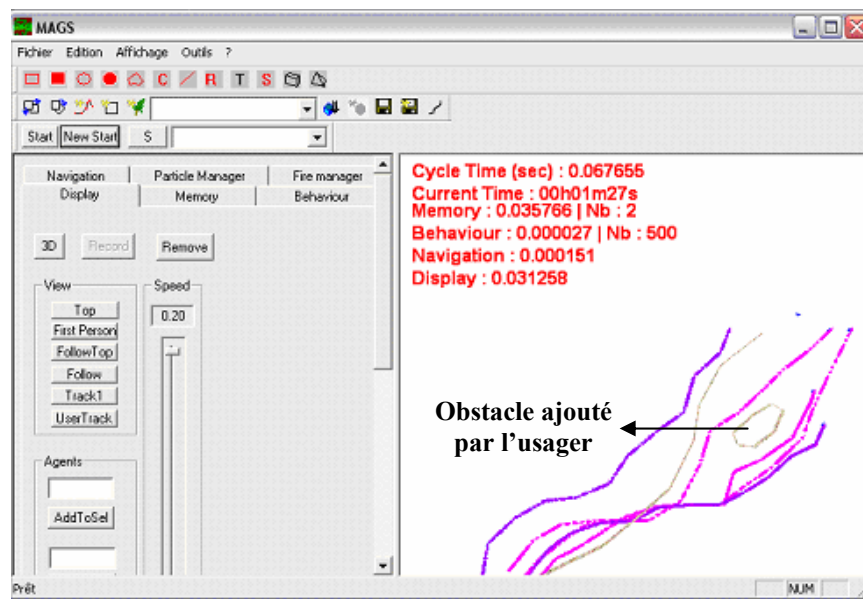


Figure 8.4. Évitement d'obstacle des *Agents_Pathfinder*

La vérification des traces de simulation de notre prototype montre dans tous les cas de figures, que les résultats obtenus sont corrects (les résultats sont conformes aux attentes). Toutefois, la vérification est une étape nécessaire mais pas suffisante pour assurer la validité. Une deuxième phase doit alors avoir lieu : la *validation*.

8.1.3. Validation de la simulation

La validation est le processus qui consiste à évaluer les conclusions de la simulation pour voir si oui ou non elles sont similaires aux conclusions observées sur le monde réel [Feinstein et Cannon 02]. Pour différencier la vérification de la validation, Pegden et ses collègues [Pegden et al., 95] notent que « *la validation est le processus qui détermine si on a construit le bon modèle, alors que la vérification est conçue pour voir si on a construit le modèle correctement* ».

Plusieurs concepts sont reliés au processus de validation et font l'objet de plusieurs recherches, nous citons ici quelques exemples de sous-domaines de recherche qui s'intéressent à un ou plusieurs aspects particuliers de la validation :

- Exactitude [Dukes et Walter 76] : Est-ce que la simulation reflète la réalité avec exactitude.
- Validité algorithmique [Wolfe et Jackson 89]: Est-ce que la simulation donne les valeurs escomptées.
- Croyance [Pegden et al., 95] : Est-ce que l'utilisateur final a suffisamment confiance dans les résultats de la simulation.
- Validité éducationnelle [Feinstein et Cannon 02] : Est-ce que la simulation fournit une expérience d'apprentissage.
- Etc.

Ces aspects sont généralement discutés dans le cadre des simulations pour les jeux, mais peuvent très bien s'appliquer à n'importe quel type de simulation.

Nous pouvons par contre classer les approches de validation selon deux grands types : validation objective et validation subjective. L'approche objective implique souvent l'utilisation de tests statistiques et de procédures [Balci 88; Barlas 90; Youngblood 93]. L'approche subjective, par contre, est basée sur d'autres moyens tels que l'affichage graphique, l'intuition, les opinions ou les expériences antérieures [Schruben 90; Turing 50].

L'approche objective est utilisée seulement lorsque le système est complètement observable tandis que l'approche subjective est privilégiée lorsque le système est partiellement ou complètement non observable [Birta et Ozmizrak 96]. Il est clair que notre application (les feux de forêts) forme un système partiellement observable, car lors d'un feu, le chaos, le danger, la complexité du phénomène font qu'il est très difficile de tout observer sur le terrain. Nous devons donc faire, s'il y a lieu, une validation subjective.

Quant à la procédure de validation, un modèle de prédiction peut être scientifiquement validé en comparant ses prédictions avec les observations futures ou passées ou encore en vérifiant la similarité en terme de structure entre le modèle et les connaissances théoriques ou empiriques que nous avons du système (ou monde) réel [Gross et Strand 00]. Deux procédures peuvent donc s'appliquer pour ce genre de problème, dépendamment des données utilisées: *validation par prédiction* et *validation par retrodiction*. En général, dans le cas de la validation par prédiction, on considère que lorsque les prédictions fournies par le modèle de simulation donnent des résultats satisfaisants lors de tests répétitifs, il est rationnel de s'attendre que le modèle reste fiable sous des conditions similaires dans le futur. Cependant, plusieurs problèmes et spécialement dans les situations *post-normales* (des problèmes pratiques écologiques ou sociaux, à grande échelle, complexes et très incertains [Funtowicz et Ravetez 93]), ne permettent pas d'effectuer des tests répétitifs. La validation par *retrodiction*²² [Gross et Strand 00] est alors mieux adaptée. Ayant à disposition un historique de qualité suffisante et un modèle qui reproduit cet historique correctement, le modèle pourrait alors être jugé fiable pour le futur. La reproduction de l'historique joue ainsi un rôle analogue à celui de la prédiction en temps réel.

En ce qui concerne notre application, il va sans dire que le domaine des feux de forêts est un problème *post-normal* selon la définition de Funtowicz et Ravetez. Nous allons par conséquent utiliser une validation par *retrodiction*. Pour ce faire, nous disposons d'un cas de feu de forêt majeur DOGRIB²³ pour lequel nous avons des informations précises sur ce qui s'est vraiment passé lors de cet incendie.

²² Faire des tests sur des données prises de l'historique.

²³ C'est un feu qui a commencé le 25 Septembre 2001 au centre de l'Alberta, et a brûler pendant 22 jours [Prometheus 05]

La validation dans le sens philosophique classique de l'empirisme logique, nécessite la vérification des correspondances (une à une) entre les éléments du modèle de simulation et les entités observables du monde réel [Nagel 61; Hempel 66]. Toutefois, de nos jours, plusieurs scientifiques et philosophes affirment qu'un modèle peut aussi être très utile même si cette procédure de vérification des correspondances n'est pas respectée [Gross et Strand 00] et ce pour plusieurs raisons. En effet, Gross et Strand [Gross et Strand 00] pensent que sous certaines circonstances, il est parfois préférable de prendre sa chance et de croire en un modèle prédictif même si on ne peut pas le valider complètement. De plus, selon la philosophie de Popper [Popper 59], une vérification complète est impossible et la vérité ne peut qu'être approchée négativement en éliminant les erreurs. Casti [Casti 97] va encore plus loin en disant que la fidélité parfaite d'un modèle de simulation n'est ni suffisante ni nécessaire pour garantir le succès lors de son utilisation. Pour notre application nous nous contentons de valider certains aspects de l'approche.

En général, la validité « absolue » ne peut être établie [Birta et Ozmizrak 96]. Par conséquent, le but de la validation est surtout d'atteindre un niveau de *confiance* raisonnable pour que les résultats de la simulation puissent être acceptés avec confiance [Shannon 75; Herskovitz 91]. Nous nous contenterons dans ce qui suit de faire une validation objective, par *retrodiction*, et incomplète. Notre stratégie est la suivante : nous disposons de données réelles (à partir de l'historique d'un vrai feu). Nous devons alors valider de façon objective par des experts du domaine, une série d'aspects quantitatifs et qualitatifs. Ces aspects sont exprimés à travers des petits scénarios que nous avons développés pour des fins de validation. La section suivante propose donc ces différents aspects que nous avons implantés.

8.2. Scénarios

Deux scénarios seront mis à l'œuvre pour valider notre travail:

1^{er} scénario : planification préalable de la ligne d'arrêt.

Cette planification a pour but de déterminer le chemin que va emprunter le bulldozer pour construire la ligne avant même que le travail effectif sur le terrain ne soit entamé.

Ce scénario se déroule en trois phases :

1^{ère} phase : prédictions de l'expert

Le but de cette phase est de récupérer des données sur les capacités de l'expert à planifier sans assistance de la part des agents logiciels. Nous présentons alors à ce planificateur humain une reproduction réelle de la forêt (sur carte papier ou sur l'environnement de simulation que nous avons développé). La représentation de l'environnement doit être assez significative pour que l'expert puisse distinguer les facteurs ayant le plus d'influence, à savoir la végétation, l'élévation et l'inclinaison du terrain, etc. Des prédictions sur la progression des feux sont reproduites dans l'environnement (prédictions fournies par le simulateur Prometheus).

Face à cette situation, et connaissant les ressources qui sont à sa disposition (dans notre cas, une seule équipe mécanisée), l'expert doit fournir son plan d'action. Ce plan doit contenir les éléments suivants :

- La ligne que les bulldozers doivent construire et qui doit emprunter le plus possible des parties du terrain à faible essence et à accès facile.
- Le point de départ prévu pour l'équipe mécanisée.
- Une prévision dans le temps de l'avancée de la construction. En d'autres termes, l'expert doit fournir une estimation du temps nécessaire pour construire la ligne qu'il a proposée.

2^e phase : Prévisions des agents sur les propositions de l'expert

Les agents logiciels planificateurs doivent donner une estimation (en s'appuyant sur leurs capacités de planification et de simulation) de l'avancée de l'équipe mécanisée en se basant sur les données relatives au terrain et à la productivité des ressources (p.ex. la vitesse des bulldozers).

3^e phase : Amélioration du plan proposé par l'expert

Dans un deuxième temps, les agents logiciels planificateurs (les *Agents_Pathfinder*) vont essayer d'améliorer le plan proposé par l'expert. Pour ce faire, plusieurs *Agents_Pathfinder* (des clones) essaient de trouver un chemin pour le bulldozer qui respecte la forme générale de l'esquisse proposée par l'expert mais qui nécessite moins de temps pour le construire.

2^e Scénario : La planification en ligne.

Dans ce scénario, nous supposons que l'exécution effective du plan est en train de se faire sur le terrain. En d'autres termes, ce scénario se déroule pendant que le bulldozer construit la ligne d'arrêt. Ce scénario suppose que nous disposons déjà d'un plan à suivre accepté par l'expert. Nous supposons que la position effective du bulldozer est périodiquement mise à jour sur l'environnement de simulation. L'expert peut ainsi visualiser l'avancée de l'équipe sur le terrain en temps quasi réel. Le but de ce scénario est de simuler les changements imprévus sur le plan de départ. Nous prenons comme exemple d'imprévu l'avancée plus lente que prévue de l'équipe sur le terrain. En d'autres termes, nous simulons sur l'environnement une construction de la ligne qui prend plus de temps que prévu en raison de la progression lente de l'équipe. Deux phases successives sont prévues et qui évaluent respectivement le comportement de l'expert et celui de l'*Agent_Pathfinder*.

1^e phase : L'expert en action

L'expert est invité à observer le déroulement des événements et à prendre les bonnes décisions pour assurer le succès de l'opération. Il aura en fait à sa disposition un outil lui permettant de changer le chemin que l'équipe devra suivre tout en tenant compte bien sûr qu'elle a déjà réalisé une portion du plan.

2^e phase : Les Agents_Pathfinder en action

L'exécution du plan est reprise à zéro, mais cette fois ce sont les *Agents_Pathfinder* qui sont chargés de contrôler le déroulement du plan. Ils doivent replanifier en cas de besoin pour que la ligne soit construite tout en respectant les contraintes de la situation (progression des feux, choix stratégique de l'expert, etc.)

8.3. Implémentation et Résultats

Dans cette section, nous présentons l'implémentation de l'outil logiciel que nous avons développé pour valider quelques aspects de notre approche. Nous mettons l'emphase sur le scénario 1 (« planification préalable de la ligne d'arrêt ») tel que spécifié dans la section précédente. Nous décrivons dans ce qui suit la réalisation des trois phases du scénario. Les résultats seront présentés au fur et à mesure. Par faute de temps et de moyens, le deuxième scénario n'a pas pu être implémenté.

Phase préliminaire : préparation de l'environnement.

Avant d'entamer les trois phases principales du processus de validation, il était nécessaire de fournir à l'utilisateur humain un environnement plus riche en information afin de lui permettre de prendre les meilleures décisions. Pour ce faire, nous affichons sur l'environnement de simulation les données sur la végétation (principale information pour choisir la ligne d'arrêt) et les sources d'eau de la région incendiée, et ce sous une forme ergonomique. Des couleurs (reconnaissables par les experts des feux) sont attribuées donc à chaque cellule de l'environnement virtuel selon le type de l'essence végétale présent dans cette cellule (Figure 8.5). Des courbes de niveaux (dessinées à la main) indiquent en plus l'élévation du terrain.

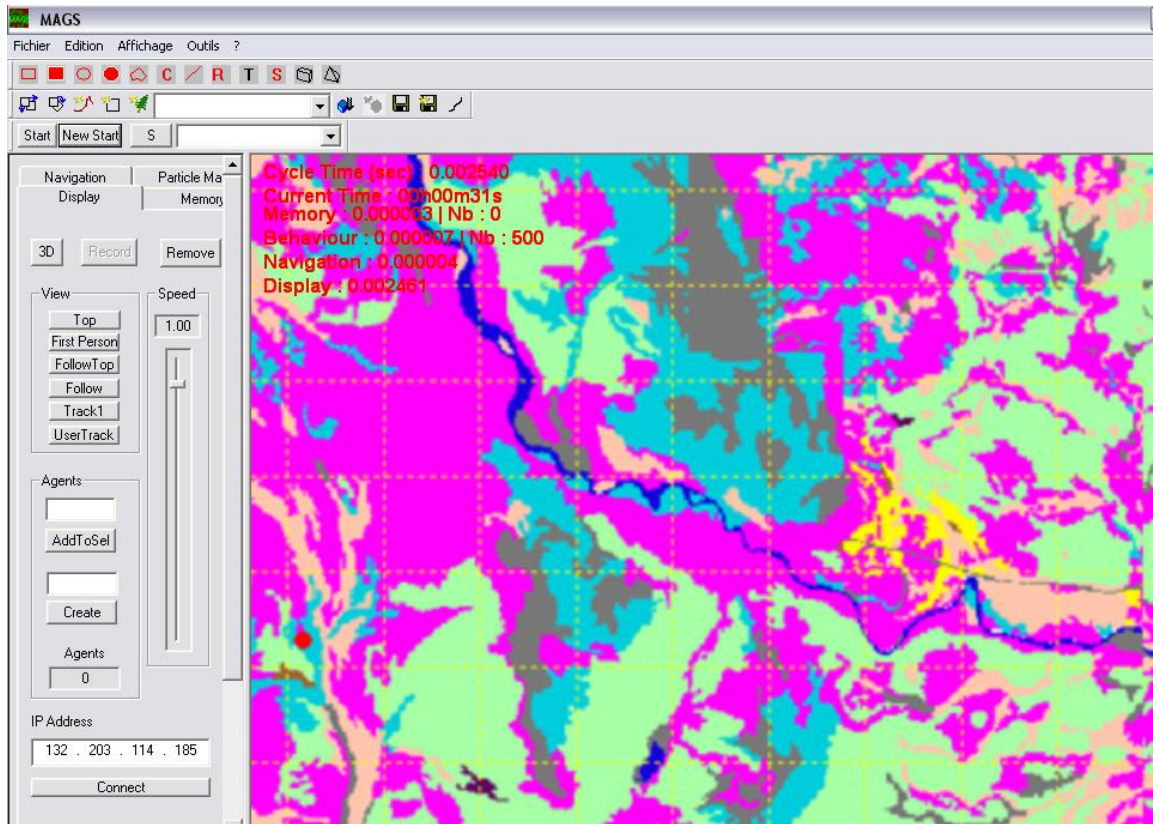


Figure 8.5. Représentation des données du terrain sur l'environnement virtuel

1^{ère} phase : prédictions de l'expert

Selon la spécification du scénario, l'expert doit fournir :

- Le tracé de la ligne d'arrêt
- Le point de départ et le point d'arrivée prévus pour l'équipe mécanisée.
- Une prédiction sur le temps nécessaire à la réalisation de la ligne d'arrêt.

Pour ce faire, l'expert commence tout d'abord par observer la situation. Il dispose ainsi d'un contrôle pour lancer la simulation de la progression des feux prévue par Prometheus. La visualisation de cette simulation sur l'environnement virtuel lui permet de comprendre la situation et par la suite de proposer une esquisse de la ligne d'arrêt. Un outil de dessin est alors prévu pour cette fin. Le point de départ et le point d'arrivée de la ligne correspondent respectivement au premier et au dernier point dessiné par l'expert. La Figure 8.6 donne

deux exemples différents d'esquisses proposées par l'utilisateur : esquisse (1) et esquisse (2). Les différents périmètres de feu affichés correspondent aux prédictions données par Prometheus.

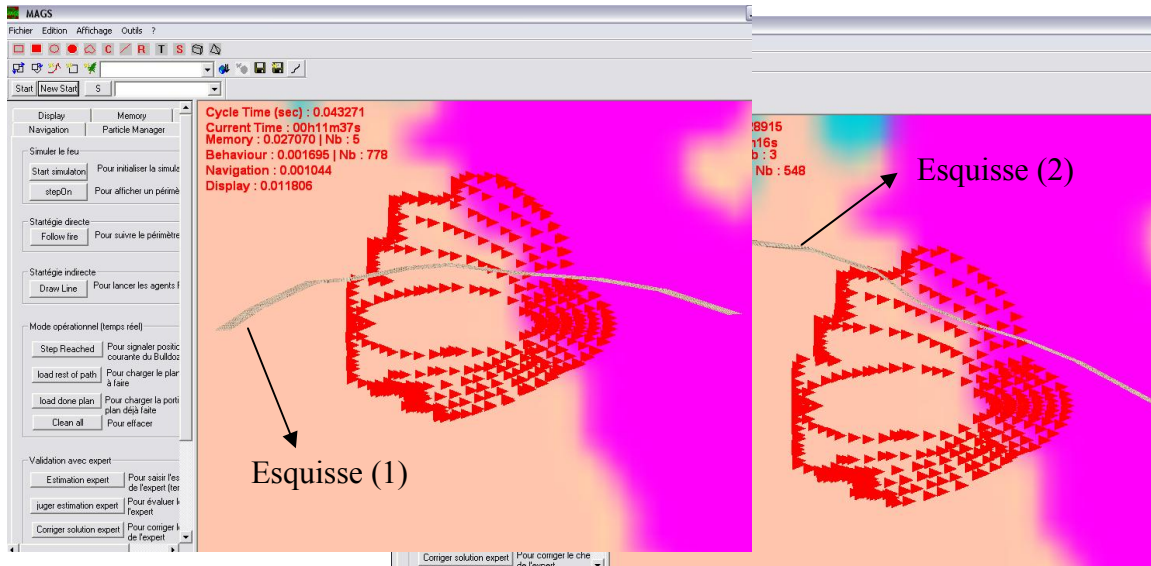


Figure 8.6. Esquisse (1) (à gauche) et Esquisse (2) proposée par l'utilisateur.

L'expert doit par la suite saisir son approximation du temps qu'il juge nécessaire pour réaliser la ligne effectivement sur le terrain.

2^e phase : Prévisions des agents sur les propositions de l'expert

Dans cette deuxième phase, il s'agit en fait d'évaluer la proposition de l'expert. En d'autres termes, notre système doit juger la faisabilité de la ligne proposée et dans le cas où cette solution est faisable, juger l'approximation donnée par l'expert quand à la durée prévue pour la réalisation de la ligne. Pour ce faire, nous avons conçu un agent spécial : *Agent_EstimerLigneExpert*. Cet agent a trois tâches principales :

- Suivre le tracé de l'expert tout en ajustant sa vitesse de progression selon les caractéristiques des portions du terrain par lesquelles il passe.
- Calculer le coût (le temps) de son déplacement.

- Vérifier de façon continue que la ligne de feu n'atteint pas une portion de la ligne d'arrêt qui n'a pas été encore construite (donc une portion que l'*Agent_EstimerLigneExpert* n'a pas encore atteint).

Si l'*Agent_EstimerLigneExpert* arrive à destination (la fin du tracé) sans que le feu ne dépasse la ligne d'arrêt, le tracé est jugé faisable. Dans ce cas, le coût calculé par l'*Agent_EstimerLigneExpert* est comparé à l'estimation donnée auparavant par l'expert.

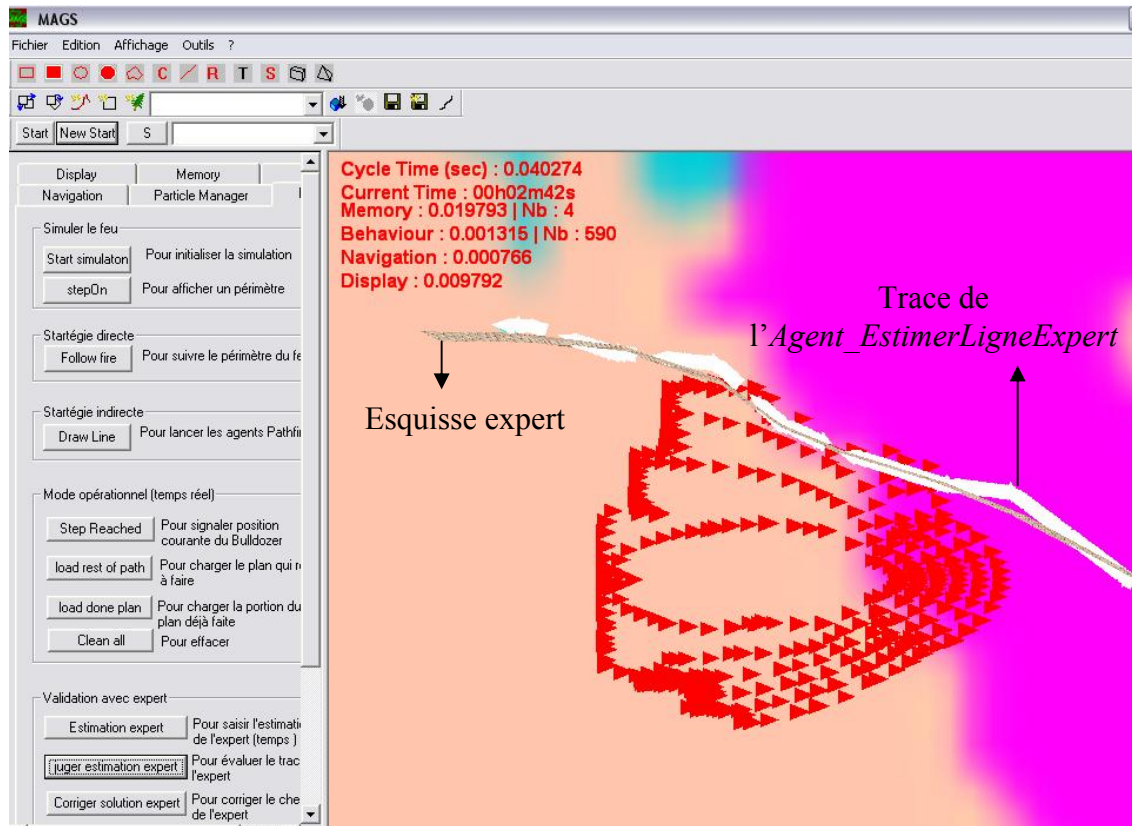


Figure 8.7. Cas où le tracé proposé par l'utilisateur est jugé faisable par l'*Agent_EstimerLigneExpert*

Il est à noter que pour plus de réalisme, nous mettons à jour les données d'entrée de Prometheus surtout pour ce qui est de la portion de la ligne d'arrêt déjà construite (que l'*Agent_EstimerLigneExpert* a déjà passée), et ce pendant la simulation et de façon continue. Au fur et à mesure que la ligne d'arrêt se construit (virtuellement bien sûr), le modèle Prometheus génère le nouveau périmètre de feu en tenant compte de cette nouvelle donnée (présence d'une portion de la ligne d'arrêt). La Figure 8.7 est un exemple de ligne

d'arrêt jugée faisable par l'*Agent_EstimerLigneExpert*. En comparant le comportement du feu dans ce cas par rapport à celui de la Figure 8.8, nous remarquons qu'effectivement Prometheus a tenu compte en temps réel de la construction de la ligne d'arrêt. En effet, la propagation des feux a été stoppée²⁴, l'allure des périmètres est changée par rapport à ce qu'elle devrait être s'il n'y avait pas la ligne d'arrêt (comme est le cas dans la Figure 8.8).

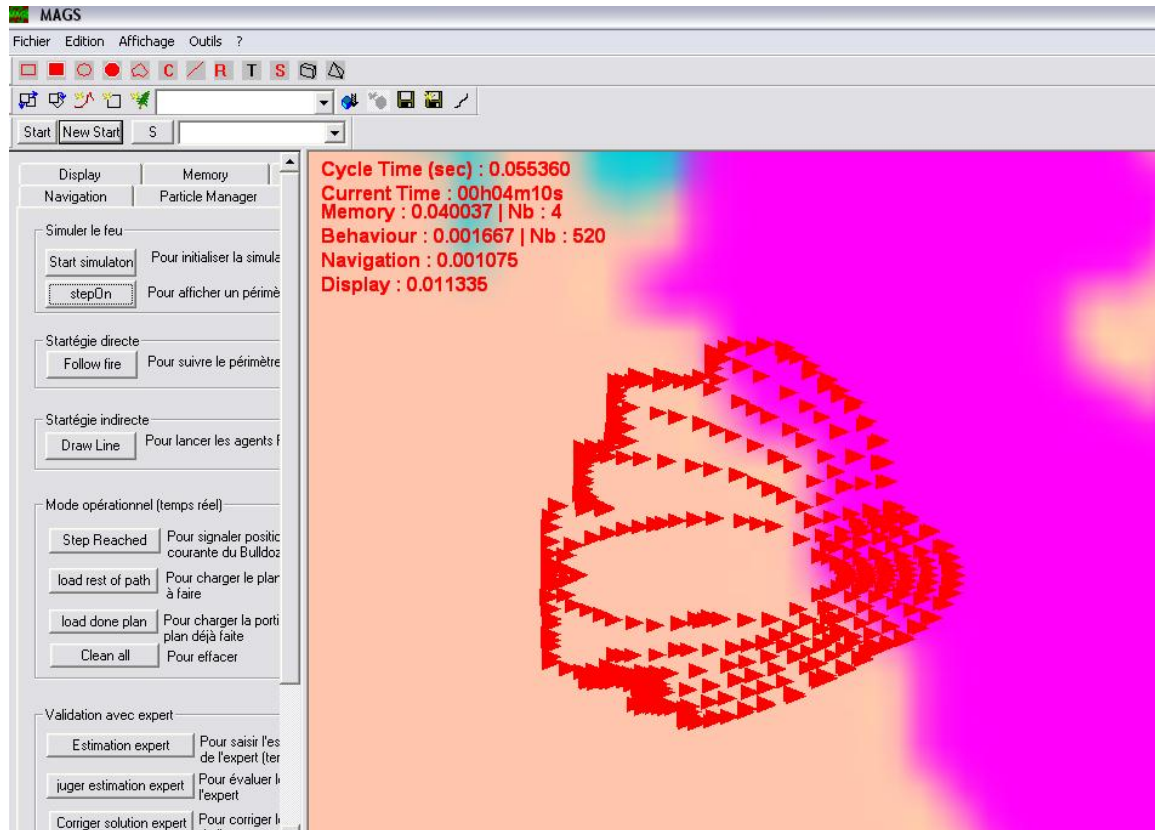


Figure 8.8. Les périmètres de feu générés par Prometheus en l'absence de ligne d'arrêt.

Si par contre, l'*Agent_EstimerLigneExpert* s'est arrêté avant d'arriver à sa destination finale (cela veut dire qu'il a détecté que le feu a dépassé une portion de la ligne qu'il allait construire), le système affiche un message avertissant l'utilisateur du fait que l'esquisse proposée n'est pas faisable sous les conditions courantes. La Figure 8.9 illustre le cas de figure où l'*Agent_EstimerLigneExpert* s'est arrêté parce que l'esquisse que l'utilisateur voulait construire n'est pas faisable, il s'agit dans ce cas précis de l'esquisse (1) de la

²⁴ Dans la figure 8.8, quelques points représentant les *Agents Fire* semblent dépasser de quelques millimètres la ligne d'arrêt. Ceci est en fait dû à la précision d'affichage : les coordonnées des points représentant le feu sont exprimées en nombres réels, alors que celles de la ligne en nombres entiers.

Figure 8.6. La portion du périmètre du feu qui aurait dépassé le tracé est alors affichée avec une couleur différente afin de bien montrer à l'utilisateur le niveau de gravité du problème.

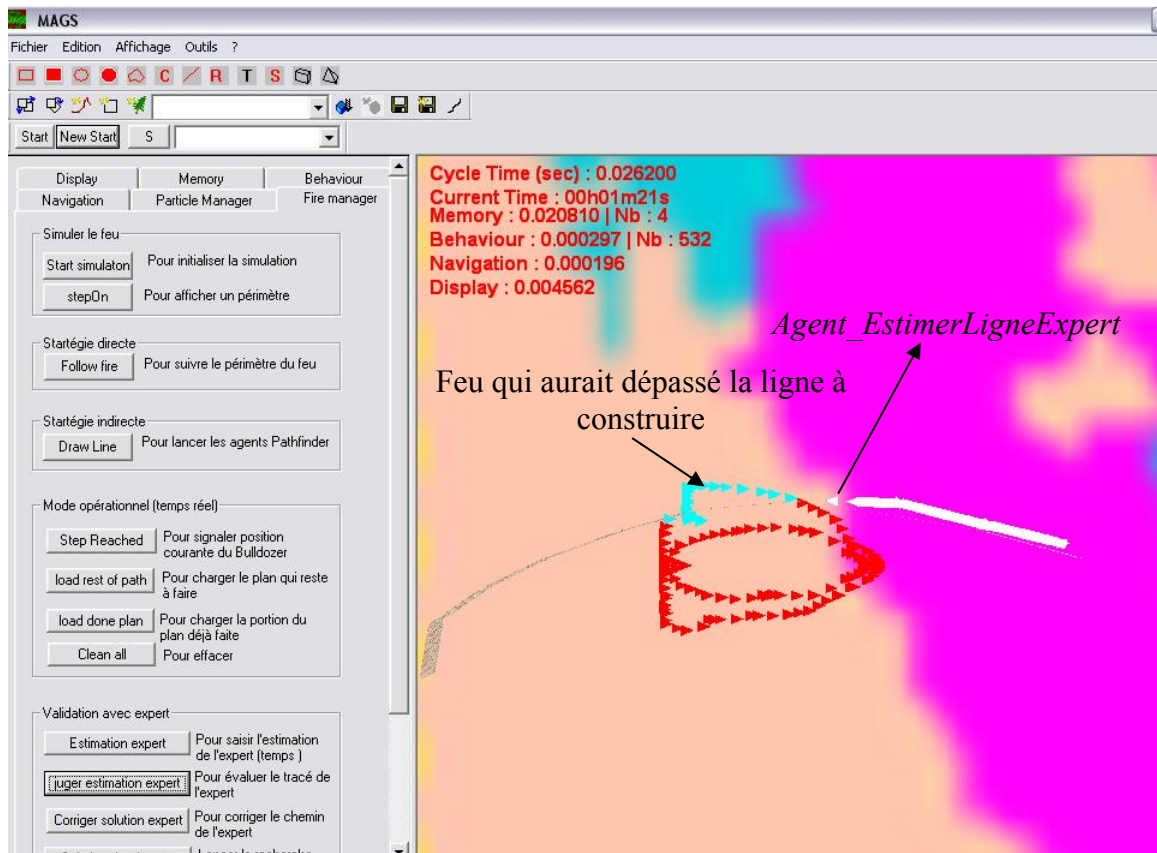


Figure 8.9. Cas où le tracé proposé par l'utilisateur est jugé infaisable par l'*Agent_EstimerLigneExpert*

Selon les tests effectués, l'utilisateur qui essaye de minimiser au maximum la longueur de la ligne et l'espace qui sera sacrifié (espace qui sera brûlé entre la ligne d'arrêt et la position courante des feux) voit souvent son plan refusé par l'*Agent_EstimerLigneExpert* à cause de son infaisabilité. En effet, même par des tests répétitifs sur le même feu, il était très difficile de trouver un chemin qui soit à la fois faisable et qui minimise les pertes aussi bien au niveau de l'espace qu'au niveau des coûts de réalisation. Quant à l'utilisateur qui ne préfère pas prendre de risque, il a souvent tendance à proposer une esquisse bien loin des feux et par conséquent plus coûteuse. En effet, si l'*Agent_EstimerLigneExpert* arrive à terme de ce plan (donc le plan est faisable), le coût serait beaucoup plus élevé que si la l'emplacement

de la ligne avait été mieux optimisé. De plus, ses estimations quant à la durée de réalisation de la ligne sont souvent très différentes des calculs de l'agent. En effet, il s'est avéré qu'il est très difficile d'estimer cette durée. Ceci était en fait prévisible compte tenu de la forte corrélation entre le coût du chemin et les diverses caractéristiques du terrain.

3^e phase : Amélioration du plan proposé par l'expert

Lorsque l'utilisateur arrivait à donner un plan faisable (après évaluation de l'*Agent_EstimerLigneExpert*), il était intéressant de tester les capacités des *Agents Pathfinder* à trouver de meilleures solutions. L'utilisateur pouvait alors lancer le processus de recherche d'un meilleur plan par simple clique de bouton. Plusieurs *Agents Pathfinder* sont alors créés afin de chercher un chemin qui ressemble à l'esquisse proposée mais avec un coût moindre. Ces agents utilisent donc la procédure décrite dans la Section 5.3.3 du Chapitre 5 ainsi que le mécanisme de clonage présenté dans l'algorithme de la Figure 7.17 du Chapitre 7. Le meilleur chemin est alors retenu. Ce chemin est d'après les tests toujours meilleur (moindre coût) que celui qui a été proposé au début par l'utilisateur. Ce résultat était encore une fois prévisible. En effet, étant donné le grand nombre de chemin évalués, les *Agents Pathfinder* trouvent, au pire des cas, un plan qui ressemble quasiment à celui proposé par l'expert (si par hasard l'expert avait trouvé un « très bon » plan dès le départ), sinon mieux. Dans les chapitres 5 et 7, nous avons donné justement des exemples de plans que les *Agents Pathfinder* proposent comme alternatives au plan initial de l'expert, soit par planification rapide ou raffinée. Le coût de la solution agent est évidemment inférieur (de façon significative, car sinon, d'après le mécanisme de recherche établi, l'*Agent Pathfinder* aurait préféré de suivre la solution de l'expert).

Dans le cas où la solution proposée par l'expert n'aboutit pas (jugée infaisable par l'*Agent_EstimerLigneExpert*), l'utilisateur peut solliciter l'aide des agents pour trouver une solution faisable. Il est clair que dans ce cas de figure, la solution agent est toujours meilleure que celle proposée par l'expert car nous comparons une solution faisable (celle des agents) à une autre infaisable (celle de l'expert). Cependant, nous avons noté d'après les tests effectués, que la solution agent est souvent considérablement différente (le chemin est assez loin du tracé initial) de l'esquisse de départ, ce qui est bien sûr logique. En effet,

la solution de l'expert avait certainement échoué parce qu'elle s'approchait trop des feux. Il est alors légitime que la solution agent soit assez différente afin de garantir la faisabilité.

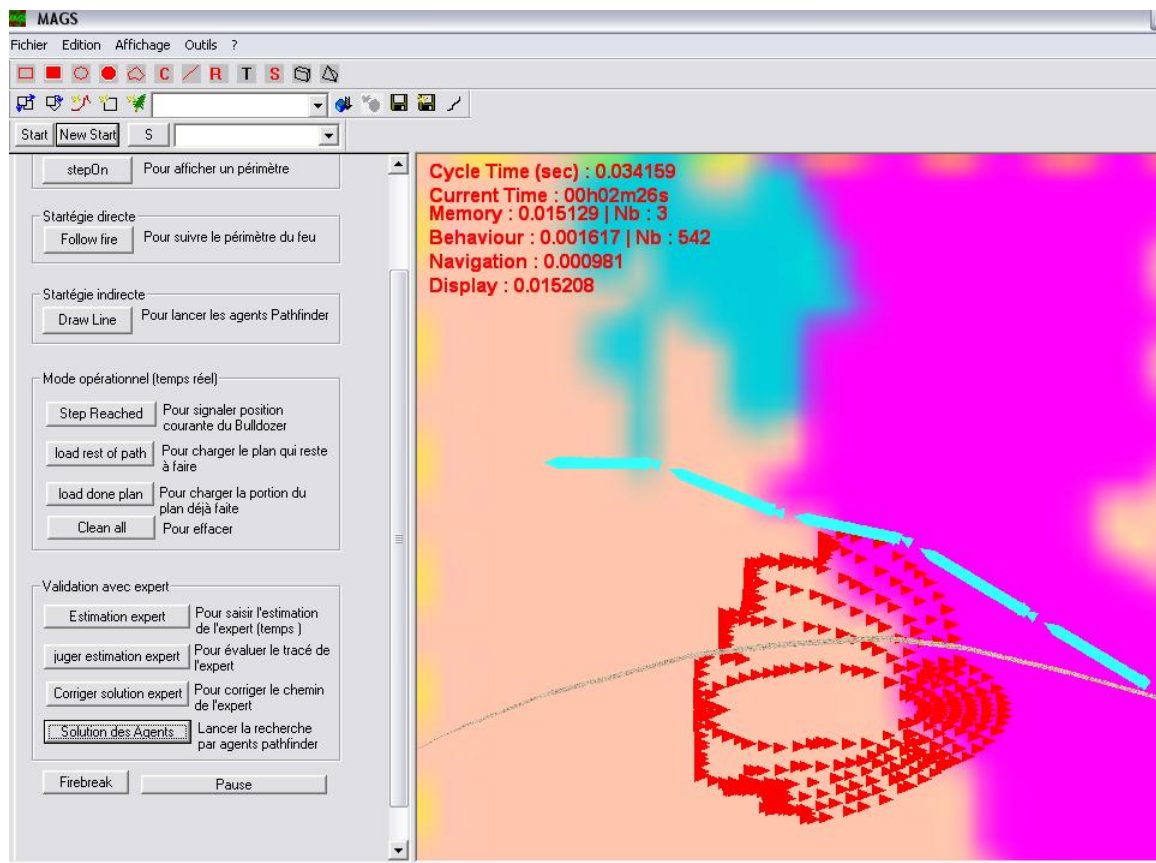


Figure 8.10. Cas où les agents reprennent la recherche du plan à zéro.

Pour demander l'aide des agents, l'utilisateur humain a le choix entre une solution complète et une correction du plan qui vient d'être jugé infaisable. Dans le cas d'une demande de solution complète, les *Agents Pathfinder* construisent leur plan au complet depuis la zone qui entoure l'extrémité de l'esquisse utilisateur. La Figure 8.10 montre effectivement ce que l'utilisateur obtiendrait si à partir du blocage de la Figure 8.9, il choisit de solliciter les agents pour une solution complète (au lieu d'essayer de corriger manuellement le plan initial).

Dans le cas d'une correction de plan, les agents essaient de corriger le plan initial de l'expert. Pour ce faire, ils commencent par déterminer le point du plan à partir duquel il y a

eu un problème. Ce point est appelé *point de blocage*. Par la suite, la recherche du reste du plan est entamée à partir de ce point. La Figure 8.11 montre ce qu’obtiendrait l’utilisateur en demandant une correction de son plan initial à partir de la position de blocage obtenue dans la Figure 8.9.

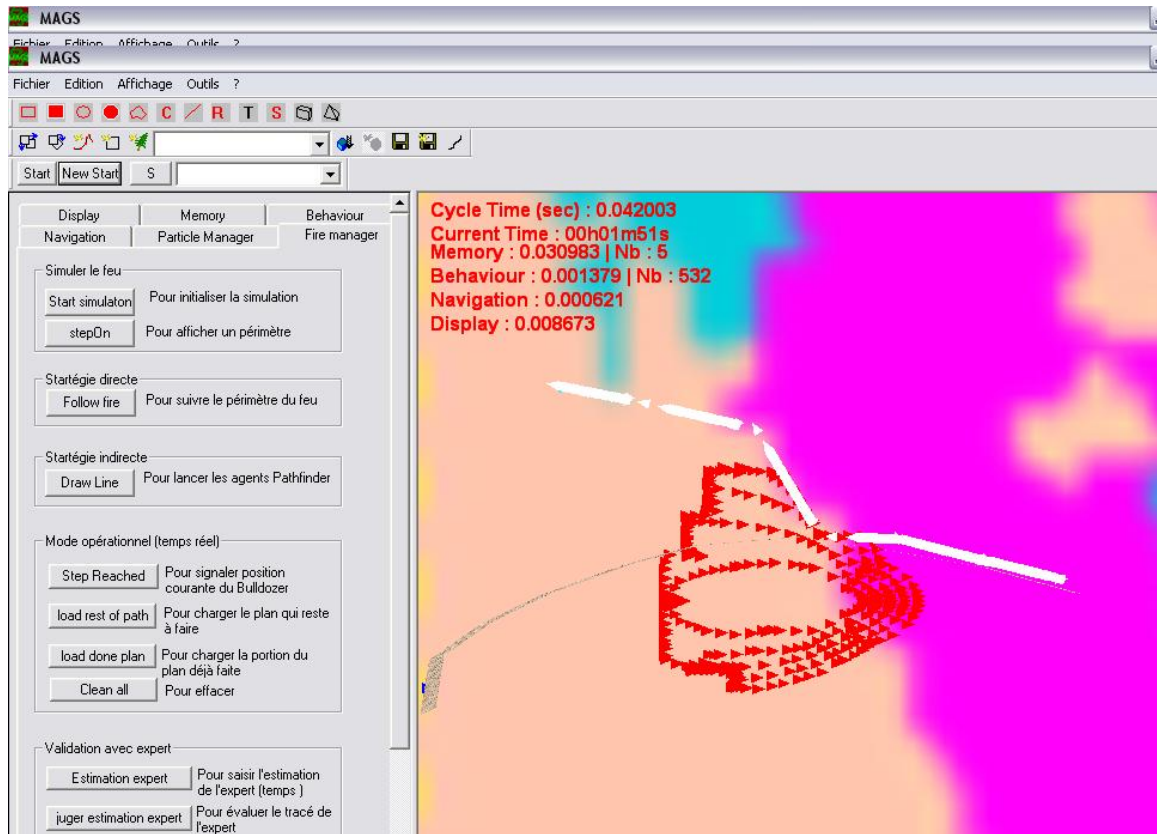


Figure 8.11. Cas où les agents corrigent le plan de l’expert humain.

8.4. Analyse des résultats et conclusion

Tout d’abord, par manque de moyens et de temps, il nous a été impossible d’implémenter le deuxième scénario. Ce dernier nécessitait surtout des données réelles des interventions effectives sur le terrain lors des opérations de lutte. Malheureusement, nous n’avons pas pu mettre la main sur de telles données. En effet, les planificateurs des opérations ne gardent pas trace de leurs plans. Il aurait fallu participer à une vraie opération de lutte contre un feu majeur pour pouvoir collecter ces informations, ce qui n’a pas pu être réalisé. Toutefois, les expériences au laboratoire sont quand même prometteuses.

Quant au scénario implémenté, il était question de valider deux aspects relatifs à la planification d'une ligne d'arrêt : un aspect quantitatif et un autre qualitatif. Pour ce qui est de l'aspect quantitatif, il était très facile de constater l'efficacité des agents dans l'estimation du coût du plan (le chemin) ainsi que celui du temps nécessaire pour le réaliser. Ceci est parfaitement compréhensible car ces agents disposent de toute l'information complète et détaillée concernant le terrain, et peuvent par conséquent donner des estimations fiables (si bien sûr les données utilisées dans le calcul sont fiables et les règles de calcul sont réalistes) du temps que mettrait un bulldozer pour exécuter le plan.

Pour ce qui est du volet qualitatif, et en comparant les résultats obtenus par les agents à ceux des utilisateurs humains, il faut noter tout d'abord que dans un premier cas, la différence est nette. Il s'agit du cas où l'expert humain propose un plan infaisable. La solution proposée par les agents est évidemment meilleure puisqu'elle est au moins faisable. Dans le cas où l'expert réussit à proposer un plan faisable, les agents réussissent dans la plupart du temps de trouver une solution quantitativement meilleure (de moindre coût). La question qui se pose ici : est-elle toujours qualitativement meilleure? En fait, trouver un chemin moins coûteux pour les agents, revient à s'éloigner un peu de l'esquisse de l'utilisateur à l'affût de portions de terrains plus faciles à traverser. Ceci nous entraîne des fois à sacrifier plus ou moins de terrain au profit des feux ou encore à se détourner un peu de la stratégie proposée par l'expert. Cependant, chaque critère qualitatif d'évaluation du plan peut avoir des résultats différents selon les circonstances : Le fait de sacrifier plus de terrain au profit des feux peut sembler désavantageux si la priorité de l'expert est de minimiser au maximum l'espace perdu ou encore de sauver certaines ressources des flammes, mais peut aussi être avantageux si par exemple la situation est très critique et que la sécurité des équipes d'intervention est prioritaire.

La leçon que nous pourrions tirer de ces expériences, est que, lorsque l'expert humain est capable de proposer une solution faisable, les agents ne donnent pas forcément une meilleure solution (au niveau qualitatif), mais plutôt une alternative moins coûteuse (quantitativement). Dépendamment des circonstances, l'expert humain pourrait choisir l'une ou l'autre des deux solutions. Dans tous les cas, nous l'aurions aidé à trouver la solution la plus adéquate. Cette conclusion confirme en fait l'un des principes de base de

notre approche : la planification dans des situations dynamiques et complexes doit être menée conjointement par les humains et par les agents logiciels en ralliant les capacités cognitives des planificateurs humains aux capacités avancées des agents.

Il est à noter que l'un des facteurs influençant les résultats obtenus est la fonction de calcul des coûts des déplacements des agents et plus précisément la fonction *BeamCost* décrite à la Section 5.3.3 du Chapitre 5. Nous avons certes pu déterminer les principaux facteurs qui influencent le coût. Toutefois, les pondérations respectives de ces facteurs dans la fonction ne sont pas évidentes à estimer. Par exemple, pour les caractéristiques du terrain, quelle pondération donner à l'élévation, au degré de la pente, etc.? Pour ce qui est des contraintes spatiales, faudrait-il privilégier plus une direction moins coûteuse mais qui dévie un peu de l'esquisse ou bien une autre direction qui respecte mieux la forme de l'esquisse mais qui est un peu plus coûteuse? Etc. Ce qui est sûr par contre, c'est que les pondérations sont importantes et que des heuristiques doivent être faites pour trouver une fonction avec des pondérations plus réalistes, et par conséquent de meilleurs plans. Il est donc fondamental de mieux observer les opérations de lutte sur le terrain pendant les incendies afin de collecter les données nécessaires pour fournir au modèle de planification des agents des règles plus réalistes.

Enfin, nous tenons à souligner que l'outil qui a été développé ici pour valider les aspects discutés dans ce chapitre s'est aussi avéré utile pour assister le planificateur humain dans la prise de ses décisions. En effet, l'option « corriger le plan » par exemple, pourrait être très utile dans un vrai système de gestion des opérations de lutte contre les feux.

CHAPITRE 9

Conclusion

9.1. Discussion générale

Dans cette thèse, nous avons proposé une approche originale pour assister les êtres humains dans la résolution de problèmes complexes de planification dans un espace réel, dynamique et à grande échelle géographique. L'approche se base sur plusieurs principes dont nous citons ici les principaux :

Utilisation de la simulation pour planifier : Nous avons repris le principe de la *Planification Basée sur la Simulation* [Lee et Fishwick 94] et nous l'avons adapté aux contraintes de notre problème. Nous avons ainsi utilisé tout d'abord une géosimulation (au lieu d'une simple simulation) afin de considérer les données réelles de l'espace et par la suite garantir un meilleur réalisme à l'application. Nous avons ensuite déployé des agents logiciels pour assurer la planification dans un environnement spatial et dynamique.

Connaissance spatiale augmentée : L'environnement de simulation construit en parallèle avec le monde réel, est considéré comme une source de connaissances spatiales augmentées.

Sensibilité spatiale des agents logiciels planificateurs : Nous avons doté les agents logiciels agissant dans l'environnement de simulation, de capacités avancées qui leur permettent d'être plus sensibles à l'espace et de raisonner selon les contraintes physiques du terrain.

Nous discutons dans ce qui suit certains des apports (autres que l'objectif principal de l'architecture) de notre approche.

Rallier les capacités cognitives des planificateurs humains à celles des planificateurs logiciels pour construire de meilleurs plans (plus réalistes, plus efficaces et plus précis) :

Comme nous l'avons discuté dans le Chapitre 5 (Section 5.6), la coopération entre planificateur humain et agents planificateurs, constitue un exemple de la complémentarité entre les cognitions spatiales des agents logiciels sensibles à l'espace d'une part et des planificateurs humains d'autre part, et ce dans le cadre d'une géosimulation multiagent. Elle est également un exemple de complémentarité qu'apporte la géosimulation par agent aux capacités humaines de planification lors de la résolution de problèmes complexes.

Pouvoir assister le planificateur humain avec du raisonnement spatial qualitatif (et non

seulement quantitatif) : Nous avons démontré dans le Chapitre 5 qu'il est possible d'utiliser les agents logiciels pour assister le planificateur humain avec du raisonnement spatial qualitatif. Bien que l'exemple des agents qui suivent une esquisse proposée par l'expert humain, soit spécifique au problème des feux de forêt, il illustre bien les possibilités qu'offrent les agents logiciels pour la résolution des problèmes qualitatifs.

Planifier de façon plus efficace dans un environnement dynamique et complexe :

Nous avons pu utiliser notre architecture ENCASMA afin d'améliorer l'approche DCP classique. En effet, le fait que l'environnement de simulation constitue à la fois une infrastructure de rencontre pour les agents et aussi une source de connaissances spatiales augmentées, a permis d'améliorer le processus général de planification DCP afin d'assurer une meilleure communication, un meilleur parallélisme et une meilleure anticipation des imprévus.

En ce qui concerne le domaine d'application (les feux de forêts), il serait intéressant de situer notre recherche par rapport aux autres travaux qui s'intéressent aussi aux opérations de lutte contre les incendies de forêts. Il existe en fait, comme nous l'avons déjà mentionné auparavant, plusieurs applications de planification dans le domaine des feux de forêts telles que PHOENIX [Cohen et al., 89], CHARADE [Avesani et al., 00] et SIADEx [de la Asunción 04]. Phoenix, par exemple, est aussi basé sur une architecture à quatre couches similaire à ENCASMA. Le processus de planification dépend d'une librairie de plans. Les agents doivent en choisir le plan le plus approprié. Ces agents sont organisés de façon hiérarchique afin de supporter la hiérarchie des acteurs de l'environnement. Malgré tous ces

atouts, Phoenix est resté un outil d'essai au laboratoire. Il ne propose aucune extension pour supporter les situations du monde réel. Les agents dans Phoenix n'ont pas la sensibilité spatiale requise pour raisonner sur les caractéristiques spatiales du terrain et respecter les contraintes spatiales qualitatives du problème. De plus, le processus de planification et de replanification est assez simple et ne profite pas des récentes approches de planification telles que la DCP. Même le principe d'*anticipation des actions futures* (qui est une bonne initiative) présent dans Phoenix, est limité puisqu'il ne profite pas des avantages que peut lui offrir l'environnement de simulation. Il aurait été plus judicieux dans Phoenix de simuler les actions futures (comme nous le faisons dans ENASMA), au lieu de simplement les calculer.

CHARADE, un système interactif de planification à base de cas (pour les feux de forêts), offre une panoplie d'outils pour assister les gestionnaires des feux à gérer les ressources et faire des plans. Cependant, ce système n'est pas basé sur les agents logiciels et ne permet pas de construire des plans mais plutôt de trouver un plan dans une liste prédéfinie de plans sauvegardés.

La plupart des systèmes à l'image de PHOENIX et CHARADE, n'offrent pas aux agents logiciels (au cas où il y en a) la sensibilité géospatiale et ne mettent pas l'accent sur la complémentarité cognitive entre les planificateurs humains et les agents logiciels.

Récemment, SIADEX, un système intelligent d'aide à la décision pour la planification des opérations de lutte contre les feux de forêts, a proposé d'utiliser des techniques de planification délibérative sans faire appel à une librairie prédéfinie de plans déjà mémorisés. SIADEX offre également suffisamment d'interactivité pour impliquer les planificateurs humains dans le processus de la planification. Toutefois, cette implication de l'être humain n'est pas aussi poussée que dans notre approche où l'intervenant humain peut par exemple faire des suggestions qualitatives lors de l'élaboration du plan (dessiner une esquisse). SIADEX ne se base pas sur les agents logiciels, mais plutôt sur des techniques de l'IA, et par conséquent, il lui est difficile d'atteindre une sensibilité spatiale suffisante. Il ne supporte pas les opérations temps réel non plus (pas de liens entre le monde de simulation et le monde réel). Cependant, SIADEX propose plusieurs types d'incertitudes (temporelles

et spatiales) pour gérer le dynamisme de l'environnement, ce qui n'est pas encore fait dans notre approche.

Enfin, et pour clore la discussion sur les feux de forêts, nous devons préciser que notre approche ENCASMA peut très bien s'appliquer sur d'autres domaines que les feux de forêts. ENCASMA pourrait ainsi être appliquée sur la gestion des marées noires. Les données sur le terrain seraient remplacées par les données maritimes, le modèle de Prometheus par un modèle de propagation du pétrole sur la surface de l'eau, etc. Les agents planificateurs agissant dans l'environnement de simulation auront cette fois pour tâche de planifier comment encercler la nappe de pétrole et de simuler son exécution en tenant compte des contraintes maritimes (les courants d'eau, les profondeurs, etc.) et de la météo.

Pour ce qui est des résultats de notre prototype, les expériences faites au laboratoire sur la version courante du prototype (développé en C++ et intégré à MAGS²⁵) montrent de bons résultats. Par exemple, les *Agents Pathfinder* mettent de quelques secondes à quelques minutes (dépendamment du nombre de clones utilisés et de la longueur de la ligne d'arrêt) pour trouver un chemin qui suit la forme générale de l'esquisse proposée par le planificateur humain. Un tel chemin ne peut être trouvé par l'application des algorithmes classiques de Pathfinding tels que A* ou D* (ces algorithmes peuvent donner des chemins sinusoïdaux, ce qui n'est pas acceptable pour une ligne d'arrêt). Nous avons également simulé les événements imprévus lors de l'exécution du plan. Un exemple d'évènement imprévu est d'ajouter un obstacle ou de changer la position courante des bulldozers sur le terrain. La replanification rapide se fait alors en quelques secondes avec succès. Plus tard, une replanification raffinée est menée en quelques minutes (encore une fois dépendamment du nombre permis de clones et de la longueur de la ligne à construire).

Enfin, nous nous sommes posé des questions sur les limites réelles de notre approche, du moins de son application aux feux de forêts. Nous avons alors rencontré des experts des feux de forêts (en Alberta) qui ont montré leur enthousiasme envers notre projet. Ils étaient convaincus que la géosimulation basée sur les agents peut très bien faire partie des futurs outils qui gèrent les opérations de lutte contre les incendies de forêts. Cependant,

²⁵ Sur plate-forme Microsoft Visual C++.

l'utilisation de cette approche dans des situations réelles dépendra du développement des liens de communication entre les agents (surtout les *agents réels* et les *agents de simulation*), de la disponibilité des données SIG, de l'efficacité des senseurs, de la fiabilité des communications, de la qualité des données géoréférencées utilisées, etc. Certains de ces défis sont déjà en voie d'être surmontés : Les données SIG sont de nos jours de plus en plus utilisées; Quelques bulldozers sont déjà équipés de dispositifs GPS leur permettant de suivre automatiquement et avec précision un chemin géoréférencé; etc. Ces technologies commencent également à être appliquées dans des domaines similaires comme dans la navigation par suivi de chemin des véhicules terrestres autonomes (*path tracking navigation of autonomous ground vehicle*) [Hellstrom 02]. Enfin, le projet Siren [Jiang et al., 04] est un bon exemple de l'utilisation de l'informatique omniprésente pour permettre aux agents humains (les pompiers) de profiter de la connaissance spatiale augmentée que peuvent offrir les agents logiciels.

9.2. Contributions

Les principales contributions de cette thèse sont :

1. Une architecture (ENCASMA) pour la conception et l'implémentation d'applications (typiquement des applications de lutte contre les désastres naturels) dans un espace géographique réel à grande échelle et dynamique. L'architecture offre un monde virtuel similaire au monde réel, dans lequel des agents logiciels représentant les acteurs et les ressources du monde réel peuvent interagir entre eux et avec l'espace afin de planifier des actions, les tester et les évaluer avant de les exécuter sur le terrain. ENCASMA n'est cependant pas dédiée exclusivement aux problèmes de planification, mais elle peut être utilisée pour d'autres applications qui ont besoin d'une connaissance spatiale augmentée. L'idée de l'approche a été tout d'abord introduite dans [Moulin et Sahli 02] et publiée ensuite dans [Sahli et Moulin 03]. Dans [Sahli et Moulin 05b] nous avons raffiné l'approche en mettant plus d'emphasis sur la complémentarité cognitive entre les agents et les humains.

2. Une approche basée sur les agents logiciels pour des problèmes de Pathfinding (recherche de chemin) particuliers (dans un environnement réel et à forte composante spatiale, soumis à des contraintes qualitatives). Cette approche a été publiée dans [Sahli et Moulin 05a].
3. Une amélioration de l'approche de planification DCP afin de remédier à certaines limites de la DCP classique. Les principes de la solution ont été évoqués dans [Sahli et Moulin 03] et [Sahli et Moulin 05a].
4. Une solution pratique pour un problème réel et complexe : la lutte contre les feux de forêts. Cette nouvelle solution permet aux experts du domaine de mieux planifier d'avance les actions de lutte et aussi de surveiller l'exécution du plan en temps réel.

Les contributions 1 et 3 répondent ainsi à la question de recherche « *Comment peut-on implémenter une approche DCP pour que les agents logiciels puissent planifier de façon plus efficace (cognitivement) en temps réel (ou quasi-réel) tout en tenant compte des contraintes géospatiales de l'environnement?* ». La contribution 2 donne un exemple de comment les agents logiciels peuvent être sensibles à l'espace et résout par l'occasion un problème particulier de Pathfinding. Enfin, la contribution 4 est un premier pas vers l'amélioration de la gestion des opérations de lutte contre les feux de forêts, qui reste encore jusqu'à date un problème majeur auquel le Canada (comme certains autres pays à travers la planète) doit faire face chaque été.

9.3. Travaux futurs

La partie sur laquelle nous avons mis le moins d'emphase dans ce travail, par manque de temps, est certainement la communication (entre agents de simulation et agents réels, ou entre agents logiciels et agents humains). Cependant, il va sans dire que des applications telles que celles décrites dans les Chapitres 5 et 6 (« construction d'une ligne d'arrêt ») ou dans l'Annexe C (« Assistance sur le terrain »), ne peuvent être opérationnelles dans des situations réelles sans une bonne communication. Mis à part la partie matérielle informatique qui est en plein progrès (GPS, communication sans fils, senseurs web, etc.), beaucoup de travail reste à faire au niveau logiciel et surtout au niveau de la spécification

d'un langage approprié utilisable par les agents. Ce langage doit être léger (car les réseaux de communication dans des situations d'urgence sont souvent non fiables et à faible bande passante) et doit supporter l'aspect spatial des données qui seront acheminées dans les messages.

Nickerson et Lu ont récemment proposé un langage pour les senseurs web sans fil, appelé SWL (Sensor Web Language) [Nickerson et Lu 04]. SWL est utilisé pour transmettre des informations à travers un réseau sans fil à faible puissance. Le langage a été conçu simple et léger afin de faciliter la communication entre les senseurs et la centrale ou entre les senseurs eux-mêmes. Les auteurs notent que des agents logiciels peuvent très bien communiquer avec les senseurs via le SWL, ce qui est très prometteur pour les applications réelles telles que celles que nous avons proposées dans cette thèse. Ce qui serait par contre intéressant de faire, c'est de concevoir un langage spatial que les agents peuvent utiliser pour interagir entre eux ou avec les planificateurs humains et qui serait simple et léger à l'image du SWL. Ce langage doit supporter les contraintes spatiales aussi bien quantitatives que qualitatives. Nous avons déjà pensé à une grammaire qui pourrait être la base d'un tel langage spatial (non présentée dans cette thèse). Il serait alors intéressant de retravailler cette grammaire et de l'enrichir, puis de l'implémenter afin de permettre une meilleure communication entre les agents de simulation et surtout entre les agents de simulation et les agents réels (sur le terrain) ou les senseurs. Le langage doit enfin être compatible avec le SWL car la communication agent/senseur est essentielle pour assurer à l'environnement de simulation sa conformité et sa cohérence avec ce qui se passe sur le terrain.

Étant donné la nature des applications auxquelles est destinée l'architecture ENCASMA, la hiérarchie au niveau des intervenants humains est très importante. En effet, pour le cas de la lutte contre les feux de forêts, nous distinguons plusieurs niveaux de commandement : chef d'opération, chef, d'unité, etc. Il est clair que la nature de la planification (stratégique ou technique) et la portée de cette planification (à toute la région ou à une zone limitée) dépendent fortement des grades (pour les pompiers, les policiers ou les militaires). Il serait ainsi intéressant d'intégrer cet aspect à notre approche et d'attribuer par exemple à chaque agent intervenant un niveau hiérarchique qui lui donne plus ou moins de droits d'action et de décision. Certains travaux ont déjà intégré cette hiérarchie parmi les agents logiciels

intervenant dans la planification. Par exemple, Horn et Baxter [Horn et Baxter 00] utilisent une hiérarchie de commandement des agents pour planifier et exécuter en continu des petites unités de tâches (pour planifier un assaut d'un bataillon de chars). Un autre travail qui s'est intéressé à cet aspect et qui porte sur le même domaine d'application que le nôtre (les feux de forêts), est Phœnix [Cohen et al., 89]. Les agents dans Phœnix ont une organisation hiérarchique et centralisée (même si le modèle centralisé n'est pas robuste). Un *agent chef-pompier* coordonne les activités de tous les *agents bulldozer* en leur envoyant des directives et en recevant des rapports sur les observations du feu, les mises à jour des positions et les actions accomplies. Le chef-pompier maintient ainsi une vue globale de la situation et se base donc sur ces rapports pour choisir des plans globaux à partir de sa librairie de plans. Les actions de ces plans sélectionnés sont transmises aux agents bulldozers, qui à leur tour sélectionnent des plans de leurs propres librairies afin de mettre en œuvre les directives du plan global. Le chef-pompier maintient la cohérence globale, coordonne les ressources disponibles afin de contrôler le feu. Il est également responsable de coordonner les agents en calculant leurs points de rendez-vous, en décidant comment déployer les ressources, en notant quand le feu est sous contrôle, etc. Quant aux autres agents (bulldozers), chacun d'entre eux a une vision locale de l'environnement basée sur ses propres capacités sensorielles. Il a aussi un accès aux cartes décrivant les caractéristiques statiques du parc. Il n'a cependant comme connaissance sur le processus dynamique (progression des feux) que ce qu'il perçoit par ses propres moyens ou ce qu'il reçoit dans les directives de son chef.

Il est clair qu'adopter une telle architecture hiérarchique des agents nécessite la résolution de plusieurs problèmes reliés. Le plus important peut-être est la représentation multidimensionnelle de l'environnement. En effet, les agents d'un bas niveau hiérarchique se contenteraient d'une vision locale (mais peut-être détaillée d'une partie de l'espace) alors que des agents commandants seraient plus intéressés par une vision plus globale de l'espace. Il est alors nécessaire de pouvoir représenter l'espace avec des niveaux de granularité différents selon la nature de l'agent qui s'y trouve. Certains chercheurs se sont déjà penchés sur le problème. Nous avons relevé récemment la proposition de Rodriguez et

ses collègues [Rodriguez et al., 05] pour représenter l'environnement de façon holonique²⁶. Les auteurs représentent un réseau routier avec des holons. La structure holonique d'une route est la suivante : une route est divisée en liens ; Un lien représente une ligne à sens unique de la route ; Un segment est composé de deux points échangeurs de voies et au moins un lien. L'environnement à son tour est composé d'un ensemble de zones qui contiennent des bâtiments et des segments. Une zone peut être également décomposée en plusieurs sous-zones. Durant la simulation, les agents véhicules peuvent bouger d'un holon à un autre et la granularité est choisie selon certaines contraintes. Chaque niveau de granularité de l'environnement est responsable de stocker les informations pertinentes sur la topologie, sur les caractéristiques de l'environnement et ses lois telles que les signaux routiers et les liens adjacents.

Il serait alors intéressant d'adopter pour notre environnement une structure holonique (ou autre) afin de pouvoir simuler plusieurs niveaux de granularités différents. Par exemple, nous pouvons offrir au chef des opérations une simulation globale de la progression des feux dans la forêt (le niveau de détail sera alors bas) et présenter par contre au chef d'une équipe d'intervention donnée, la simulation de l'avancée de son équipe dans la région où elle se trouve, donc une vision plus locale et détaillée.

²⁶ Un holon [Koestler 1967] est une entité qui ne peut être considérée ni comme un tout ni comme une partie. C'est une structure similaire à elle-même qui consiste en plusieurs holons en tant que sous-structures.

Bibliographie

[Agarwal et Abrahart 03] P. Agarwal, R. Abrahart, (2003) “Agent-based simulation of cognitive neighbourhoods in urban environments.” Proceedings of *GeoComputation*, September 8-10, 2003, Southampton, UK.

[Agree et Chapman 87] P. Agree, D. Chapman (1987). « Pengi: An implementation of a theory of activity. » In Proceedings of the Sixth National Conference on *Artificial Intelligence*, pp. 268-72.

[Akishita et al., 93] S. Akishita, T. Hisanobu, and S. Kawamura, (1993). “Fast path planning available for moving obstacle avoidance by use of laplace potential,” in Proc. IEEE/RSJ International Conference on *Intelligent Robots and Systems*, vol. 1, pp. 673–678.

[Al Timimi 98] K. Al-Timimi, (1998). « Co-operative Intelligent Agents for Command and Control ». Édité par Al-Timimi à partir de deux articles: un par Peter Martin, et l’autre par Christopher Dee, Paul Millington, Ben Walls, et Tim Ward, Logica UK Limited, Site Web <http://www.datamation.co.uk/samples/edn/vol6/e604sp2.html>

[ALCIF 02] Aviation et Lutte Contre les Incendies de Forêt (2002). Site Web : <http://www.mnr.gov.on.ca/MRN/affmb/Fire/FFpkg%5Cfrcontent.htm>

[Ambros-Ingerson et Steel 88] J.A. Ambros-Ingerson, S. Steel, (1988), “Integrating planning, execution and monitoring.” In proceedings of the 7th (US) National Conference on *Artificial Intelligence (AAAI 88)* pp. 83-88, St Paul, MN, USA. American Association for Artificial Intelligence, Morgan Kaufmann.

[Andrew et al., 01] U.F. Andrew, S. Bittner, M. Raubal, (2001). “Spatial and Cognitive Simulation with Multi-agent Systems.” *COSIT 2001* pp. 124-139

[Appleby et Steward 94] S. Appleby, S. Steward, (1994). “Mobile Software Agents for Control in Telecommunications Networks”, *BT Technological Journal* vol. 12(2), pp. 104-113.

[**Arikan et al., 01**] O. Arikan, S. Chenney, D.A. Forsyth, (2001). “Efficient Multi-Agent Path Planning.” Proceedings of the *Eurographics Workshop on Animation and Simulation*.

[**Atkin et al., 98**] M. Atkin, D.L. Westbrook, P.R. Cohen, G.D. Jorstad, (1998). « AFS and HAC: Domain-general agent simulation and control. » *AAAI-98 Workshop on Software Tools for Developing Agents*, pp. 89-95.

[**Au et Parameswaran 98**] S. Au, N. Parameswaran, (1998). « Progressive Plan Execution in a Dynamic World ». In Virtual Proceedings of the Workshop *Integrating Planning, Scheduling and Execution in Dynamic and Uncertain Environments* AAAI Technical Report WS-98-02, AAAI Press.

[**Avesani et al., 00**] P. Avesani, A. Perini, F. Ricci, (2000). « Interactive case-based planning for forest fire management.” *Applied Intelligence*, vol. 13(1). pp. 41–57.

[**Axtell 00**] R. Axtell, (2000). “Why agents ? on the varied motivations for agent computing in the social sciences.” Technical Report 17, Center on Social and Economics Dynamics - The Brookings Institution.

[**Baird 70**] J.C. Baird, (1970). *Psychophysical analysis of visual space*. Oxford: Pergamon Press.

[**Balci 88**] O. Balci, (1988). “Credibility assessment of simulation results: The state of the art.” In *Methodology and Validation*, O. Balci, Ed. SCS vol. 19(1), pp. 19-25.

[**Bandini et al., 02**] S. Bandini, S. Manzoni, C. Simone (2002). « Dealing with Space in MultiAgent Systems: a model for Situated MAS.” In *AAMAS 2002*, Bologna, Italie.

[**Barlas 90**] Y. Barlas, (1990). “An autocorrelation function test for output validation.” *Simulation* vol. 55(1), pp. 7-16.

[**Barruffi et al., 98**] R. Barruffi, E. Lamma, M. Milano, P. Mello (1998). “Planning with incomplete and dynamic knowledge via Interactive Constraint Satisfaction”. In Virtual Proceedings of the Workshop *Integrating Planning, Scheduling and Execution in Dynamic and Uncertain Environments* AAAI Technical Report WS-98-02, AAAI Press.

[Bartlett 32] F.C. Bartlett, (1932). *Remembering: A study in experimental and social psychology*. Cambridge, England; Cambridge University Press.

[Basch et al., 97] J. Basch, L.J. Guibas, J. Hershberger, (1997). "Data Structures for Mobile Data." In the 8th Symposium on *Discrete Algorithms*, pp. 747-756.

[Bauzil et al., 81] G. Bauzil, M. Briot, P. Ribes, (1981). "A Navigation Sub-System Using Ultrasonic Sensors for the Mobile Robot HILARE." 1st Int. Conf. on *Robot Vision and Sensory Controls*, Stratford-upon-Avon, UK., pp. 47-58 et pp. 681-698.

[Beaver 01] A. Beaver, (2001). "Evaluating Risk and Reward Relationships in Wildland Firefighter Safety", Proceedings of the 2001 *International Wildland Fire Safety Summit*.

[Beckman 97] R.J. Beckman (edt.) (1997) "The Dallas – Fort Worth Study", Los Alamos unclassified report LAUR-97-4502LANL, Los Alamos National Laboratory, valable sur Site Web : <http://transims.tsasa.lanl.gov/> - Dernière mise à jour 2005.

[Bellifemine et al., 00] F. Bellifemine, A. Poggi, G. Rimassa, P. Turci. (2000). "An object-oriented framework to realize agent systems." Technical report, Université de Parma/Arizona, Italy.

[Ben Said et al., 02] L. Ben Said, T. Bouron, A. Drogoul, (2002) "Agent-based interaction analysis of consumer behavior." Proceedings of the First International Joint Conference on *Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, Bologna, Italy.

[Benenson et Torrens 04] I. Benenson, P.M. Torrens (2004). *Geosimulation: Automata-Based Modeling of Urban Phenomena*. Edited by John Wiley & Sons.

[Bilgili et Baskent 97] E. Bilgili, E.Z. Baskent. (1997). « Fire management planning and geographic information systems », Actes du XIe congrès *Forestier Mondial*, Antalya, Turquie.

[Birta et Ozmizrak 96] L.G. Birta, F.N. Özmizrak, (1996). "A knowledge-based approach for the validation of simulation models: the foundation." *ACM Transactions on Modeling and Computer Simulation (TOMACS)*. vol. 6(1), pp. 76-98

[BLM 03] Bureau of Land Management office of fire and aviation. Interagency Standards for Fire and Aviation operations (2003). Site web: <http://www.fire.blm.gov/Standards/redbook.htm>

[Bond et Gasser 88] A.H. Bond, H. Gasser, (1988). "An analysis of problems and research in DAI." In A.H. Bond, H. Gasser, eds., *Readings in distributed artificial intelligence*. San Manteo, CA: Morgan Kaufmann.

[Borenstein 87] J. Borenstein, (1987). "The Nursing Robot System." Ph. D. Thesis, Technion, Haifa, Israel.

[Borenstein et Koren 89] J. Borenstein, Y. Koren, (1989). "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19(5), pp. 1179–1187.

[Borenstein et Koren 91] J. Borenstein, Y. Koren, (1991). "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7(3), pp. 278–288.

[Bousquet et Lepage 01] F. Bousquet, C. Le Page (2001). « Systèmes multi-agents et écosystèmes. » Article dans : J. P. Briot et Y. Demazeau, *Agent et systèmes multiagents* chez Hermès.

[Boutilier et al., 99] C. Boutilier, T. Dean, S. Hanks, (1999). "Decision-theoretic planning: structural assumptions and computational leverage." *Journal of AI Research*, vol. 11, pp. 1-94.

[Braudaway 93] W. Braudaway (1993). « A Blackboard Approach to Computer Generated Forces ». In Proceedings of the Third Conference on *Computer Generated Forces and Behavioral Representation*, Orlando, FL, pp. 11-20.

[Brooks 86] R.A. Brooks (1986). « A robust layered control system for a mobile robot. » *IEEE Journal of Robotics and Automation*, RA-2(1) pp. 14-23.

[Brooks 91a] R.A. Brooks (1991). "Intelligence without representation". *Artificial Intelligence*, vol. 47(1-3) pp. 139-159.

[Brooks 91b] R.A. Brooks (1991). "Intelligence without reason." In Ray Myopoulos, John Reiter, editor, Proceedings of the 12th International Joint Conference on *Artificial Intelligence*, pp. 569-595, . Morgan Kaufmann, Sydney, Australia.

[Canakcioglu et Ozkazanc 97] H. Canakcioglu, O. Ozkazanc, (1997). « What can we do to reduce forest fires in Mediterranean region? » Actes du XIe congrès *Forestier Mondial*, Antalya, Turquie.

[Carbonell et al., 92] J. G. Carbonell, J. Blythe, O. Etzioni, Y. Gil, R. Joseph, D. Kahn, C. Knoblock, S. Minton, A. Pérez, S. Reilly, M. Veloso, X. Wang (1992). "PRODIGY4.0 : The manual and the tutorial". Technical report CMU-CS-92-150, Department of Computer Science, Carnegie Mellon University.

[Carr 02] J. Carr, (2002). "On the Fireline-Handcrews." Paru sur le site Web: http://www.firehouse.com/wildfires/fireline/2002/0904_handcrews.html

[Casti 97] J. Casti, (1997). *Would-Be Worlds*. John Wiley & Sons: New York.

[Chaib 94] B. Chaib-draa (1994). "Distributed Artificial Intelligence : An overview". In A. Ken, J. G. Williams, C. M. Hall, and R. Kent, editors, *Encyclopedia Of Computer Science And Technology*, vol. 31, pp. 215-243. Marcel Dekker, Inc.

[Chaib 96] B. Chaib-draa (1996). "Industrial applications of distributed AI". *Communications of the ACM*, vol. 38(11) pp. 49-53.

[Chaib et al., 01] B. Chaib-draa, I. Jarras, B. Moulin (2001), « Systèmes multiagents : Principes généraux et applications » Article paru dans : J. P. Briot et Y. Demazeau « *Agent et systèmes multiagents* » chez Hermès.

[Chalupsky et al., 01] H. Chalupsky, Y. Gil, C. Knoblock, K. Lerman, J. Oh, D. Pynadath, T. Russ, M. Tambe, (2001) « Electric Elves: Applying Agent Technology to Support

Human Organizations» International Conference on *Innovative Applications of AI (IAAI'01)*.

[**Chandler et al., 91**] C. Chandler, P. Cheney, P. Thomas, L. Trabaud, D. Williams, (1991). « Fire and Forestry. » vol. 2. *Forest Management and Organizations*. Wiley, New York.

[**Chapman et Agre 86**] D. Chapman, P. Agre (1986). “Abstract reasoning as emergent from concrete activity”. In M. P. George- and A. L. Lansky, editors, *Reasoning About Actions & Plans* - Proceedings of the 1986 Workshop, Morgan Kaufmann Publishers : San Mateo, CA, pp. 411-424.

[**Charade 97**] site Web : <http://sra.itc.it/projects/charade/>. Dernière mise à jour : 1997.

[**Chauhan 97**] D. Chauhan. (1997). “JAFMAS: A Java-Based Agent Framework for Multi-Agent Systems Development and Implementation.” PhD thesis, University of Cincinnati.

[**Chavez et Maes 96**] A. Chavez, P. Maes, (1996). “Kasbah: An Agent Marketplace for Buying and Selling Goods”, In *Proceedings the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM '96)*, London, pp. 75-90.

[**Chown et al., 01**] E. Chown, B. Booker, S. Kaplan (2001). “Perception, Action Planning, and Cognitive Maps.” Mitre technical paper. Site Web : www.mitre.org

[**Cohen 97**] A.G. Cohn, (1997). “Qualitative spatial representation and reasoning techniques.” In Brewka, G.; Habel, C.; and Nebel, B., eds., *KI-97 (LNAI 1303)*. Berlin: Springer, pp. 1-30.

[**Cohen et al., 89**] P.R. Cohen, M.L. Greenberg, D.M. Hart, A.E. Howe, (1989). “Trial by fire: Understanding the design requirements for agents in complex environments.” *AI Magazine*, vol. 10(3), pp. 34-48.

[**Cohen et Hazarika 02**] H A.G. Cohn, S.M. Hazarika, (2002). “Qualitative spatial representation and reasoning: An overview.” *Fundamenta Informaticae*, vol. 46(1-2) pp. 1-29.

[Conry et al., 89] S.E. Conry, R.A. Meyer, R.P. Pope (1989). "Mechanisms for assessing the non local impact of local decisions in distributed planning". In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence*. vol. 2, pp. 245-258. Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA.

[Conry et al., 91] S.E. Conry, K. Kuwabara, V.R. Lesser, R.A. Meyer (1991) "Multistage negotiation in distributed constraint satisfaction". In *IEEE Transactions on Systems, Man, and Cybernetics*.

[Coogaar 05] Cougaar architecture document. (2005). Sur le site Web du BBN Technologies <http://www.cougaar.org>. dernière mise à jour : 2005.

[Couclelis et Gale 86] H. Couclelis, N. Gale (1986) « Space and Spaces. » *Geografiska Annaler* vol. 68(B) pp. 1-12.

[Cova et Church 97] T.V. Cova, R.L. Church, (1997). « Modelling community vulnerability using GIS » *International Journal of Geographical Information Systems*, vol. 11, pp. 763-784.

[CPBWTF 99] Wildfire Task Force Central Pine Barrens Fire Management Plan. Avril, 1999.

[Craik 43] K.J.W. Craik, (1943). *The nature of explanation*. London: Cambridge University Press.

[Crowley 84] J.L. Crowley, (1984). "Dynamic World Modeling for an Intelligent Mobile Robot." In proceedings of IEEE Seventh International Conference on *Pattern Recognition*, Montreal, Canada, pp. 207-210.

[Dautenhahn 95] K. Dautenhahn, (1995). "Getting to know each other -- artificial social intelligence for autonomous robots." *Robotics and Autonomous Systems* vol. 16, pp. 333-356.

[de la Asunción et al., 04] M. de la Asunción, L. Castillo, J. Fdez.-Olivares, O. García-Pérez, A. González, F. Palao, (2004). "SIADEx: an interactive artificial intelligence

planner for decision support in forest fire fighting.” Workshop on *Binding Environmental Sciences and Artificial Intelligence (BESAI 2004)*, European Conf. on *Artificial Intelligence*.

[De Silva et al., 93] F.N. de Silva, M. Pidd, R.W. Eglese, (1993). “Spatial decision support systems for emergency planning: an operational research/geographical information systems approach to evacuation planning” in Sullivan, J. D. ed., *Proceedings of the Simulation multiconference on the International emergency management and engineering conference*, San Diego, Society for Computer Simulation, 130-3.

[Dean et al., 90] T. Dean, K. Basye, R. Chekaluk, S. Hyun, M. Lejter, M. Randazza (1990). “Coping with uncertainty in a control system for navigation and exploration”. In *proceedings of the 8th National Conference on Artificial Intelligence (AAAI-90)* Boston, MA (Cambridge, MA : MIT Press) pp. 1010-1015.

[Dean et Kanazawa 87] T.L. Dean, K. Kanazawa (1987) « persistence and probabilistic inference ». Technical report CS-87-23, Brown University department of Computer Science.

[Delin et al., 99] K.A. Delin, S.P. Jackson, R.R. Some, (1999) « Sensor webs », *NASA tech Briefs*, 23, 80.

[Denis 89] M. Denis (1989), « Image et Cognition », *Presses Universitaires de France*, Paris.

[Dennett 78] D.C. Dennett (1978). "Why the Law of Effect Will Not Go Away" from *Brainstorms*. Bradford.

[Desjardins et al., 99] M.E. Desjardins, E.H. Durfee, C.L. Ortiz, M.J. Wolverton, (1999). “A survey of research in distributed continual planning.” *AI Magazine* vol. 4, pp. 13-22.

[DPI 03] Department of Primary Industries / Department of Sustainability and Environment, Victoria, Australia, Site web : <http://www.nre.vic.gov.au/> Dernière mise à jour 2003.

[Drogoul 95] A. Drogoul, (1995). “When Ants Play Chess (or Can Strategies Emerge from Tactical Behaviors?)” in *From reaction to cognition*. C. Castelfranchi & J. P. Müller, eds. Berlin-Heidelberg, Springer-Verlag. vol. 957, pp. 13-27.

[Drogoul et al., 02] A. Drogoul, D. Vanbergue, T. Meurisse, (2002). “Multi-Agent Based Simulation: Where are the Agents ?” Proceedings of *MABS'02 (Multi-Agent Based Simulation)*, Bologna, Italy, July 2002), LNCS, Springer-Verlag. Vol. 2581, pp. 1-15.

[Dukes et Walter 76] R.L. Dukes, S.J. Waller, (1976). “Toward a general evaluation model for simulation games: GEM.” *Simulation & Games*, vol. 7, pp. 75-96.

[Durfee 99] E. H. Durfee (1999) « Distributed Continual Planning for Unmanned Ground Vehicle Teams » , *AI Magazine*, vol. 4, pp. 13-22.

[Egenhofer et Mark 95] M. Egenhofer, D. Mark (1995), "Naive Geography," Spatial Information Theory---A Theoretical Basis for GIS, International Conference *COSIT '95*, Semmering, Austria, A. Frank and W. Kuhn, eds., LNCS, Springer-Verlag, vol. 988, pp. 1-15.

[Egenhofer Golledge 94] M. Egenhofer, R. Golledge, (1994) “Time in Geographic Space” Report on the Specialist Meeting of Research Initiative 10. National Center for Geographic Information and Analysis, Santa Barbara, CA, Technical Report 94-9.

[EIIFC 03] Eastern Idaho Interagency Fire Center, site Web <http://www.id.blm.gov/eiifc/index.html>. Dernière mise à jour 2003.

[Emmanuel et al., 94] T. Emmanuel, L. Fagegaltier, A. Liegeois, (1994). “Motion Planning for a Patrol of Outdoor Mobile Robots.” In Proceedings of *European Robotics and Intelligent Systems Conference (EURISCON'94)*, Malaga, Spain, pp. 103-139.

[Eng 02] P. Eng. (2002) « Tiny Fire Marshals : Dust-Sized Sensors Could Provide Early Warnings of Forest Fires », article paru le 13 septembre 2002 dans *abcnews.com*.

[Farsite 05] site Web : <http://farsite.org/> Dernière mise à jour : 2005.

[**Ferber 95**] J. Ferber (1995). *Les systèmes multi-agents, vers une intelligence collective*. InterEditions.

[**Feinstein et Cannon 02**] A.H. Feinstein, H.M. Cannon, (2002). "Constructs of Simulation Evaluation." *Simulation Gaming*. vol. 33 pp. 425-440.

[**Ferrand 96**] N. Ferrand, (1996). "Modelling and supporting multi-actor spatial planning using multi-agents systems". Proceedings of the 3rd NCGIA conference on *GIS and Environmental Modelling*, Santa Fe, Santa Barbara : National Center for Geographical Information and Analysis.

[**FICCDC, 88**] FICCDC, (1988). "The proposed standard for digital cartographic data", *The American Cartographer*, vol. 15(1), pp. 9-140.

[**Fikes et al 72**] R.E. Fikes, P.E. Hart, N.J. Nilsson (1972). « Learning and Executing Generalized Robot Plans », *Artificial Intelligence* vol. 3(4) pp. 251-288.

[**Filks et Nilsson 71**] RE.. Fikes, N. Nilsson (1971). "strips : A new approach to the application of theorem proving to problem solving". *Artificial Intelligence*, vol. 2(3/4) pp. 189-208.

[**Firby 87**] R.J. Firby (1987). « An investigation into reactive planning in complex domains. » In *AAAI-87*, pp. 202-206.

[**FireM3 05**] site Web: http://cwffis.cfs.nrcan.gc.ca/en/current/cc_m3_e.php dernière mise à jour sep. 2005.

[**Firewise 05**] site Web: http://www.firewise.org/pubs/wildfire_control/ dernière mise à jour: sep. 2005

[**Fishwick 92**] P.A. Fishwick (1992). « An Integrated Approach to System Modelling using a Synthesis of Artificial Intelligence, Software Engineering and Simulation Methodologies. » *ACM Transactions on Modeling and Computer Simulation*.

[**Fishwick 94**] P.A. Fishwick (1994). *Simulation Model Design and Execution : Building Digital Worlds*. Prentice Hall.

[**Fishwick 95**] P.A. Fishwick (1995), *Simulation Model Design and Execution: Building Digital Worlds* Prentice-Hall. Englewood Cliffs, New Jersey.

[**Fishwick et Zeigler 92**] P.A. Fishwick, B.P. Zeigler, (1992). « A Multimodel Methodology for Qualitative Model Engineering. » *ACM Transactions on Modeling and Computer Simulation*, vol. 2(1) pp. 52-81

[**Forbus 81**] K.D. Forbus, (1981). "A Study of Qualitative and Geometric Knowledge in Reasoning about Motion," Massachusetts Institute of Technology Artificial Intelligence Laboratory, AI-TR-615.

[**Forbus 95**] K. Forbus, (1995). « Qualitative Spatial Reasoning: Framework and Frontiers. » In J. Glasgow, N. Narayanan, and B. Chandrasekaran. *Diagrammatic Reasoning: Cognitive and Computational Perspectives*. MIT Press, pp. 183-202.

[**Forbus et al., 01**] K. Forbus, J.V. Mahoney, K. Dill, (2001). "How qualitative spatial reasoning can improve strategy game AIs: A preliminary report." 15th International workshop on *Qualitative Reasoning (QR01)*, San Antonio, Texas.

[**Forbus et al., 91**] K. Forbus, P. Nielsen, B. Faltings, (1991). "Qualitative Spatial Reasoning: The CLOCK Project", *Artificial Intelligence*, vol. 51(1-3), pp. 417-471.

[**Fox et al., 97**] D. Fox, W. Burgard, S. Thrun, (1997). "The dynamic window approach to collision avoidance." in *IEEE Robotics & Automation Magazine*, vol. 4(1).

[**FRAMES 03**] Fire Research And Management Exchange System. site Web : <http://www.frames.gov/tools/>

[**Frank 00**] A. Frank, (2000). "Geographic Information Science: New methods and technology." *Journal of Geographical Systems*, vol.2, pp. 99-105.

[**Fuenekes et Methven 88**] U. Fuenekes, I.R. Methven, (1988). « Fire weather/behavior analysis systems », In Proc. 5th Scient. And Tech. Regional Fire Weather Comm. Atmos. Env. Serv., Winnipeg, Man. pp.15-23.

[Funtowicz et Ravetz 93] S. Funtowicz, J. Ravetz, (1993). "Science for the Post-Normal Age." *Futures* vol. 25(7) pp. 735-755.

[Gahegan 95] M. Gahegan (1995), "Proximity Operators for Qualitative Spatial Reasoning", Proceedings of the International Conference *COSIT'95*, Austria.

[Gat 98] E. Gat, (1998). "On three-layer architectures." In *Artificial Intelligence and Mobile Robots*, Chap. 8, edited by D. Kortenkamp, P. Bonnasso and R. Murphy (AAAI Press).

[Georgeff 83] M.P. Georgeff (1983). "Communication and interaction in multi-agent planning". In Proceedings of the Third National Conference on *Artificial Intelligence (AAAI-83)*, Washington, D.C..

[Georgeff et Ingrand 89] M.P. Georgeff, F.F. Ingrand (1989). "Decision-making in an embedded reasoning system". In proceedings of the 11th International Joint Conference on *Artificial Intelligence (IJCAI-89)*. Detroit, MI (San Mateo, CA: Morgan Kaufmann) pp. 972-978.

[Georgeff et Lansky 86] M.P. Georgeff, A.L. Lansky (1986). « Procedural knowledge ». Proceedings of the *IEEE Special issue on Knowledge Representation*, vol. 74(10) pp. 1383-1398.

[Goel 02] V. Goel, (2002). "Cognitive & Neural Basis of Planning." *Encyclopedia of Cognitive Science*. Macmillan

[Golden 97] K. Golden, (1997). "Planning and Knowledge Representation for Softbots". Ph.D. Dissertation, University of Washington.

[Golden et al., 94] K. Golden, O. Etzioni, D. Weld (1994). "Omnipotence without omniscience: Efficient sensor management for planning". In Proceedings of the Twelfth National Conference on *Artificial Intelligence*, AAAI Press. pp. 1048-1054.

[Golden et Weld 96] K. Golden, D. Weld (1996). "Representing sensing actions : The middle ground revisited". In Proceedings of 5th Int. Conf. On *Knowledge Representation and Reasoning*.

[Golledge 78] R. Golledge, (1978) « Learning about Urban Environments. » in: T. Carlstein, D. Parkes, and N. Thrift (Eds.), *Timing Space and Spacing Time*. London: Edward Arnold.

[Graham and al., 03] R. Graham, H. McCabe, S. Sheridan, (2003). "Pathfinding in Computer Games." *ITB Journal*.

[Gross et Strand 00] D. Gross, R. Strand, (2000). "Can Agent-Based Models Assist Decisions on Large-Scale Practical Problems? A Philosophical Analysis", *Complexity*, vol. 5(6), pp. 26-33.

[Grosz et al., 99] B.J. Grosz, L. Hunsberger, et S. Kraus, (1999). « Planning and Acting Together ». *AI Magazine*, vol. 20(4) pp. 23-34.

[Grosz et Kraus 96] B. J. Grosz, S. Kraus. (1996). "Collaborative Plans for Complex Group Action." *Artificial Intelligence* vol. 86(2) pp. 269–357.

[Grosz et Kraus 99] B.J. Grosz, S. Kraus. (1999). "The Evolution of SHAREDPLANS." In *Foundations of Rational Agency*, eds. A. Rao and M. Wooldridge, Dordrecht, The Netherlands: Kluwer Academic. pp. 227–262.

[Hamilton et al., 91] M.P. Hamilton, L.A. Salazar et K.E. Palmer, (1989). « Geographic information systems: providing information for wildland fire planning ». *Fire Technology* vol. 25 pp.5-23.

[Hammond 89] K. Hammond (1989), « Cased-based planning », in perspectives in *Artificial Intelligent*, Academic Press. vol. 1.

[Hanks et al., 93] S. Hanks, M.E. Pollack, P.R. Cohen, (1993) "Benchmarks, test beds, controlled experimentation, and the design of agent architectures," *AI Mag.*, pp. 17-42.

[Harding 99] A. Harding, (1999). "Modeling techniques for examining the impact of population ageing on social expenditure." In Conference on the *Policy implications of the Ageing of Australia's Population*, Melbourne, NATSEM.

[Hellstrom 02] T. Hellstrom, (2002). "Autonomous navigation for Forest Machines." A project Pre-study in the Department of Computer Science, Umea University, Sweden.

[Hempel 66] C. Hempel, (1966). *Philosophy of Natural Science*. Prentice-Hall: Engelwood Cliffs, NJ.

[Hendler et al., 90] J. Hendler, A. Tate, M. Drummond, "AI planning: systems and techniques." *AI Magazine*, vol. 11(2), pp. 61-77.

[Herath 01] S. Herath (2001), « Geographical Information Systems in Disaster Reduction. » *Information Technology For Disaster Management* pp. 25-31. No.1, site web : <http://www.adrc.or.jp/>

[Hernandez 94] D. Hernandez, (1994). *Qualitative Representation of Spatial Knowledge*. Springer Verlag LNAI.

[Herskovitz 91] P.J. Herskovitz, (1991). "A theoretical framework for simulation validation: Popper's falsification." *Int. J. Model. Simul.* vol. 11(2), pp. 56-58.

[Hirtle et Jonides 85] S. Hirtle, J. Jonides (1985) "Evidence of Hierarchies in Cognitive Maps." *Memory and Cognition* vol. 13(3) pp. 208-217.

[Hoc 87] J.M. Hoc (1987). *Cognitive psychology of planning*. Book in Academic press.

[Hoff et al., 95] B. Hoff, M.D. Howard, D.Y. Tseng, (1995). "Path Planning with Terrain Utilization in ModSAF." In Proceeding of the Fifth Conference on *Computer Generated Forces and Behavioral Representation*, Orlando, Florida, pp. 255-263.

[Horn et Baxter 00] G. Horn, J. Baxter (2000). « An Interactive Planner for Tank Squadron Assaults » on the first *Planning and Scheduling Special Interest Group* workshop, Milton Keynes, UK.

- [**Howden et al., 01**] N. Howden, R. Ronnquist, A. Hodgson, A. Lucas. (2001). “Jack summary of an agent infrastructure.” In 5th International Conference on *Autonomous Agents*.
- [**Hu et al., 00**] J. Hu, M. Prandini, S. Sastry, (2000). “Optimal maneuver for multiple aircraft conflict resolution: A braid point of view,” in Proc. IEEE Conference on *Decision and Control*, (Sydney).
- [**Hu et al., 93**] T. C. Hu, A. B. Kahng, G. Robins, (1993). "Optimal Robust Path Planning in General Environments" *IEEE Trans. on Robotics and Automation* vol. 9(6), pp. 775-784.
- [**Huberman et Glance 93**] B. Huberman, N. Glance, (1993). “Evolutionary games and computer simulations.” In: Proceedings of the National Academy of Science USA. pp. 7716-7718
- [**Hugues et Drogoul 00**] L. Hugues, A. Drogoul, (2000). “Grounded Representations for a Robot Team.” Proceedings of IROS 2000, IEEE/RSJ intern. Conference on *Intelligent Robots and Systems*, Japan.
- [**Jennings 00a**] N.R. Jennings, (2000) «On agent-based software engineering!», *AI Journal*, vol. 117, pp. 277–296.
- [**Jennings 00b**] N.R. Jennings (2000). “Agent methodology for software engineering.” *Communication of ACM*.
- [**Jennings et al., 98**] N. R. Jennings, K. Sycara. M. Wooldridge (1998). “A roadmap of agent research and development.” *Journal of Autonomous Agents and Multi-Agent Systems* vol. 1 pp. 275–306.
- [**Jensen and Veloso 98**] R.M. Jensen, M.M. Veloso, (1998). “Interleaving Deliberative and Reactive Planning in Dynamic Multi-Agent Domains.” In Proceedings of the AAAI Fall Symposium on *Integrated Planning for Autonomous Agent Architectures*, AAAI Press.

[Jiang et al., 04] X. Jiang, N.Y. Chen, J.I. Hong, K. Wang, L. Takayama, J.A. Landay, (2004). “Siren: Context-aware Computing for Firefighting”, in *Pervasive 2004: The Second International Conference on Pervasive Computing*, Linz /Vienna, Austria, pp. 19-23.

[Johanson 94] A.A. Johnson, (1994), “Digital flood insurance rate maps: Standards for shared data”, *Proceedings of the Eighteenth Annual Conference of the Association of Floodplain Managers*, Tulsa, Oklahoma, pp. 280-285.

[Jonsson et Backstrom 95] P. Jonsson, C. Backstrom (1995). “Incremental Planning”. In *New directions in AI Planning: Proc 3rd Eur. WS. Planning (EWSP 95)* IOS Press pp. 72-90.

[Kahneman et Tversky 82] D. Kahneman, A. Tversky, (1982). “The simulation heuristic.” In D. Kahneman, P. Slovic, & Tversky (Eds.) *Judgement under uncertainty: Heuristics and biases*. New York: Cambridge University Press.

[Karacapilidis et al., 95] N. Karacapilidis, D. Papadias, M. Egenhofer, (1995). « Collaborative spatial decision making with qualitative constraints.” In P. Bergougnoux, K. Makki, and N. Pissinou, editors, *Proc 3^{rs} ACM International Workshop on Advances in Geographic Information Systems*, Baltimore,. ACM, ACM Press, pp. 53-59.

[Kavraki et al., 96] L.E. Kavraki, P. Svestka, J.C. Latombe, M. Overmars, (1996). “Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces.” *IEEE Transactions on Robotics and Automation*, vol. 12(4) pp. 566-580.

[Keeny et Raiffa 93] R.L. Keeny, H. Raiffa (1993). “Decisions with multiple objectives preferences and value trade-offs”. New York, Cambridge University Press.

[Kenge 99] site web: <http://www.nr.usu.edu/swarm/> dernière mise à jour juin : 1999.

[Kerst et al., 87] S.M. Kerst, J.H. Howard, L.J. Gugerty, (1987). “Judgment accuracy in pair-distance estimation and map sketching.” *Bulletin of the Psychonomic Society*, vol. 25(3), pp. 185-188.

[Kettani 99] D. Kettani (1999), « Conception et implantation d'un modèle spatial qualitatif qui s'inspire du raisonnement spatial de l'être humain », Ph.D. Thesis., Computer Science Department, Laval University.

[Kettler et Pemberton 98] B. Kettler J.C. Pemberton (1998). « A General Sentinel Framework: Towards a Consensus for Plan Monitoring Agents ». In *Virtual Proceedings of the Workshop Integrating Planning, Scheduling and Execution in Dynamic and Uncertain Environments AAAI Technical Report WS-98-02*, AAAI Press.

[Khatib 86] O. Khatib, (1986). "A unified approach for motion and force control of robot manipulators: the operational space formulation." *IEEE Journal of Robotics and Automation*, vol. 3(1), pp. 43-54.

[Kitchin 94] R.M. Kitchin (1994). "Cognitive Maps: What Are They and Why Study Them?" *Journal of Environmental Psychology*, vol. 14 pp. 1-19.

[Knoblock 95] B. Knoblock (1995). "Planning, executing, sensing, and re-planning for information gathering". In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, pp. 1686-1693.

[Koditschek 87] D. Koditschek, (1987). "Exact robot navigation by means of potential functions: Some topological considerations," in *Proc. IEEE Robotics and Automation*, pp. 1-6.

[Koditschek et Rimon 88] D. Koditschek, E. Rimon, (1988). "Exact robot navigation using cost functions: the case of distinct spherical boundaries in en," in *Proc. IEEE Robotics and Automation*, vol. 3, pp. 1791–1796.

[Koenig et Likhachev 02] S. Koenig, M. Likhachev, (2002). "D* Lite". In *Proceedings of AAAI/IAAI (Innovative Applications of Artificial Intelligence)*, Edmonton, Canada, pp. 476-483.

[Koestler 67] A. Koestler, (1967). *The Ghost in the machine*. Hutchinson.

[Kourtz et O'Regan 77] P. Kourtz, W. O'Regan, (1971). "A model for a small forest fire." *For. Sci.* vol. 17(2) pp. 163-169.

[Kozierok et Maes 93] R. Kozierok, P. Maes, (1993). "A Learning Interface Agent for Scheduling Meetings", *Proceedings of the ACM-SIGCHI International Workshop on Intelligent User Interfaces*, Florida, pp. 81-93.

[Kray 01] C. Kray, (2001). "The benefits of multi-agent systems in spatial reasoning." Special Track on Spatio-temporal Reasoning. *Proceedings of Flairs '01*, AAAI Press, Menlo Park, CA, pp. 552-556.

[Kremens et al., 01] R. Kremens, A. Seema, A. Fordham, D. Luisi, B. Nordgren, S. VanGorden, A. Vodacek (2001). « Networked, autonomous field-deployable fire sensors. » *Proceedings of the 2001 International Wildland Fire Safety Summit*

[Kuipers 78] B. Kuipers, (1978) "Modeling Spatial Knowledge." *Cognitive Science*, vol. 2 pp. 129-153.

[Kuipers 82] B. Kuipers, (1982) "The Map in the Head Metaphor", *Environment and Behaviour*, vol. 14, pp. 202-220.

[Kuipers et Levitt 88] B. Kuipers, T. Levitt (1988) "Navigation and Mapping in Large-Scale Space." *AI Magazine* vol. 9(2) pp. 25-46.

[LaMothe 99] A. LaMothe, (1999). "Tricks of the Windows Game Programming Gurus." New York, Sams Publishing.

[Lanza et Siccardi 95] L. Lanza, F. Siccardi (1995), "The role of GIS as a tool for the assessment of flood hazard at the regional scale", In: *Geographical Information Systems in Assessing Natural Hazards*, ed A. Carrara and F. Guzzetti, Kluwer Academic, Netherlands, pp. 199-217.

[Latombe 91] J.C. Latombe, (1991). *Robot Motion Planning*, Kluwer Academic Publishers.

[**Lee et Fishwick 94**] J.J. Lee, P.A. Fishwick (1994), "Real-Time Simulation-Based Planning for Computer Generated Force Simulation", *Simulation*, vol. 63(5), pp. 299-315

[**Lee et Fishwick 95**] J.J. Lee, P.A. Fishwick, (1995). "Simulation-Based Real-Time Decision Making for Route Planning." In 1995 Winter *Simulation* Conference, December, Crystal City, VA, pp. 1087-1095.

[**Lee et Fishwick 97**] J.J. Lee, P.A. Fishwick (1997) Simulation-based Planning for Multi-Agent Environments. Winter *Simulation* Conference, pp. 405-412

[**Lewis et al., 01**] M. Lewis, T. Lenox, T. Payne, K. Sycara, (2001). "Spatial planning in teams of human and machine agents." Proceedings of the 2nd Conference on *Information Technology and Spatial Planning*, Isole Tremiti, It.

[**Liebermann 95**] H. Lieberman, (1995). "Letizia: An Agent that Assists Web Browsing", In *Proceedings of IJCAI 95*, AAAI Press.

[**Lloyd et Helvly 87**] R. Lloyd, C. Heivly, (1987). « Systematic distortions in urban cognitive maps. » *Annals of the Association of American Geographers*, vol. 77, pp. 191-207.

[**Luneburg 47**] R.K. Luneburg, (1947). *Mathematical Analysis of Binocular Vision*. Princeton: Princeton University.

[**Lynch 60**] K. Lynch, (1960). *The Image of a City*. Cambridge, MA: MIT Press.

[**Maamar et al., 02**] Z. Maamar, W. Mansoor, Q. H. Mahmoud (2002). « Software agents to support mobile services. » Proceedings of the first international joint conference on *Autonomous agents and multiagent systems*, Bologna, Italy

[**Maaß et al., 95**] W. Maaß, J. Baus, J. Paul, (1995). "Visual grounding of route descriptions in dynamic environments." In *AAAI Fall Symposium on Computational Models for Integrating Language and Vision*. MIT, Cambridge.

[**MAGS 03**] B. Moulin, J. Perron, J. Hogan, F. Bellafkir, W. Chaker, P. (2003) Tremblay. Manuel du développeur MAGS.

[**Maes 95**] P. Maes, (1995b). “Artificial Intelligence meets Entertainment: Lifelike Autonomous Agents”, *Communications of the ACM* vol. 38 (11), pp. 108-114.

[**MAML 00**] site Web : <http://www.maml.hu/maml/initiative/> dernière mise à jour : Oct. 2000.

[**Marchant et al., 97**] J.A. Marchant, T. Hague, N.D. Tillet, (1997). “Row-following accuracy of an autonomous vision-guided agricultural vehicle,” *Computers and Electronics in Agriculture*, vol. 16(2), pp. 165-175.

[**Mather 93**] M. Mather, (1993), “Maps in action for protecting Trinidad and Tobago from disasters”, In *Natural Disasters: Protecting Vulnerable Communities*, ed P. A. Merriman and C. W. A. Browitt, Thomas Telford, London, pp. 365-373.

[**McNamara et al., 89**] T. McNamara, J. Hardy, S. Hirtle (1989) “Subjective Hierarchies in Spatial Memory”, *Journal of Environmental Psychology: Learning, Memory, and Cognition* vol. 15(2) pp. 211-227.

[**Merrill et Alexander 87**] D.F. Merrill, M.E. Alexander, editors. (1987). « Glossary of forest fire management terms. » 4th ed. National Research Council of Canada, Canadian Committee on Forest Fire Management, Ottawa. Publication NRCC No. 26516.

[**Michel et al., 02**] F. Michel, P. Bommel, J. Ferber, (2002). “Simulation interactive distribuée de SMA par des SMA.” proceedings of *JFIADSMMA 2002* (French workshop on Multi-Agent Systems), Hermès, Paris, France.

[**Minsky 86**] M. Minsky, (1986). *The Society of Mind, Touchstone*, Simon & Schuster, Inc., New York.

[**Montello 88**] D.R. Montello (1988) “Route information and travel time as bases for the perception and cognition of environmental distance.” Unpublished doctoral dissertation, Arizona State University, Tempe, AZ.

[Moulin 96] B. Moulin, B. Chaib-draa (1996). “An overview of distributed artificial intelligence”. In G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed AI*, John Wiley & Sons : Chichester, England. pp. 3-54.

[Moulin et al., 01] B. Moulin, N. Sahli, Z. Maamar (2001), « Utilisation d'agents stationnaires et d'agents mobiles pour la planification d'itinéraires dans un environnement dynamique. » *JFIADSMA'01* Montreal, Hermes (France) pp. 163-175.

[Moulin et al., 03] B. Moulin, W. Chaker, J. Perron, P. Pelletier, J. Hogan (2003). « MAGS Project: Multi-Agent GeoSimulation and Crowd Simulation ». *COSIT'2003*.

[Moulin et Sahli 02] B. Moulin, N. Sahli (2002). « Software agents and Distributed Continual Planning: towards geospatial awareness. » *Agentcities* workshop in *AAMAS 2002*, Bologna, Italie.

[Mulvehill et al., 04] A. Mulvehill, B. Benyo, D. Rager, E. DePalma, (2004). “ACT - The Automated Clearance Tool: Improving the Diplomatic Clearance Process for AMC”. *2004 Command and Control Research Technology Symposium*.

[Myers 99] K.L. Myers (1999). « A Continuous Planning and Execution Framework ». *AI Magazine*, vol. 20(4), pp. 63-69.

[Nagel 61] E. Nagel, (1961). “*The Structure of Science. Problems in the Logic of Scientific Explanation.*” Routledge & Kegan Paul: London.

[Neisser 76] U. Neisser. (1976). *Cognition and reality- Principles and Implications of Cognitive Psychology*, Freeman, New York.

[NewKirk 93] R.T. Newkirk, (1993), “Extending geographic information systems for risk analysis and management”, *Journal of Contingencies and Crisis Management*, vol. 1(4), pp. 203-206.

[Newsom et Mitrani 93] D.E. Newsom, J.E. Mitrani (1993), “Geographic information system applications in emergency management”, *Journal of Contingencies and Crisis Management*, vol. 1(4) pp. 198-202.

- [**NFPA 89**] National Fire Protection Association. « Case study : Black Tiger Fire » 1989.
- [**Nickerson et Lu 04**] B.G. Nickerson, J. Lu, (2004). “A language for wireless Sensor Webs.” 2nd Annual Conference on *Communication Networks and Services Research CNSR*.
- [**Nicolas et Beebe 99**] M.V.J. Nicolas, G.S. Beebe (1999). “The Training of Forest Firefighters in Indonesia.” Report of Government of Indonesia – Ministry and Estate crops, European Union-European Commission, and GTZ- German Agency for Technical Cooperation.
- [**Niederberger et Gross 04**] C. Niederberger, M.H. Gross, (2004). “Generic Path Planning for Real-Time Applications.” in Proceedings of *Computer Graphics International*, Heraklion, Crete, Greece.
- [**Normn et Rumelhart 75**] D. Norman, D. Rumelhart (1975). « Memory and Knowledge. » In D. Norman & D. Rumelhart (Eds.) *Exploration in Cognition*. San Francisco :Freeman.
- [**Nourbakhsh 97**] I. Nourbakhsh (1997). “Interleaving planning and Execution for Autonomous Robots”. (Dordrecht, Netherlands : Kluwer Academic). PhD thesis.
- [**Nwana et al., 98**] H. Nwana, D. Ndumu, L. Lee. (1998). “Zeus: An advanced tool-kit for engineering distributed multi-agent systems.” In Proceedings of *PAAM98*, London U.K. pp. 377-391.
- [**O’Connor et al., 96**] M. O’Connor, T. Bell, G. Elkaim, B. Parkinson, (1996). “Automatic Steering of Farm Vehicles Using GPS,” Proceedings of the 3rd *International Conference on Precision Agriculture*.
- [**Olawsky et Gini 90**] D. Olawsky, M. Gini. (1990). « Deferred planning and sensor use ». In Proceedings *DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control*. DARPA Workshop, pp 166-174.
- [**OMG 99**] Object Management Group, Framingham Corporate Center MA USA « Agent Technology » Green paper, OMG document ec/99-08-06 Version 0.8 August 1999

[Onder et Pollack 99] N. Onder, M.E. Pollack (1999). « Conditional, probabilistic planning: A unifying algorithm and effective search control mechanisms ». In Proceedings of the Sixteenth National Conference on *Artificial Intelligence (AAAI-99)*, Orlando, FL, pp. 577-584.

[Orcutt 57] G.H. Orcutt, (1957). “A new type of socio-economic system.” *Review of Economics and Statistics* vol. 39 pp. 116-123

[Ören et al., 00] T.I Ören, S.K. Numrich, A.M. Uhrmacher, L.F. Wilson, E. Gelenbe. (2000). “Agent-Directed Simulation - Challenges to Meet Defense and Civilian Requirements.” Proc. of the Winter *Simulation* Conf. Vol. 2, pp. 1757-1762,.

[O'Sullivan et Torrens 00] D. O'Sullivan, P. Torrens, (2000). “Cellular Models of Urban Systems” in Bandini, S & Worsch, T (eds) 2000, Theoretical and Practical Issues on Cellular Automata, Proceedings of the Fourth International Conference on *Cellular Automata for Research and Industry (ACRI 2000)*, Karlsruhe, Germany, pp. 108-116.

[Paolucci et al., 00] M. Paolucci O. Shehory, K. Sycara (1998). « Interleaving Planning and Execution in a Multiagent Team Planning Environment ». tech. report CMU-RI-TR-00-01, Robotics Institute, Carnegie Mellon University.

[Paramics 05] Site Web : <http://www.sias.com/sias/s-paramics/overview/overview.html>. dernière mise à jour 2005.

[Pegden et al., 95] C.D. Pegden, R.E. Shannon, R.P. Sadowski, (1995). “*Introduction to simulation using SIMAN*” (2nd ed.). Hightstown, NJ: McGraw-Hill.

[Peot et Smith 92] M. Peot et D. Smith (1992). « Conditional nonlinear planning ». In proceedings of 1st Int. Conf. On *Principles of Knowledge Representation and Reasoning*.

[Petrov et Stoen 00] P.V. Petrov, A.D. Stoyen, (2000). “An Intelligent-Agent Based Decision Support System for a Complex Command and Control Application”. In Proceedings of *Sixth IEEE International Conference on Complex Computer Systems (ICECCS'00)*, pp. 94-104.

[PFMT 03] Private Forest Management Team, site Web: <http://www.pfmt.org/fire/>
Dernière mise à jour: mai 2003.

[Phoenix 94] site Web : <http://babs.cs.umass.edu/research/phoenix.html>. Dernière mise à jour: mai 1994.

[Piaget et Inhelder 67] J. Piaget, B. Inhelder (1967). *The Child's Conception of Space*. New York: Norton.

[Pietrula et al., 98] M. Pietrula, K. Carley, L. Gasser, (1998). "Simulating Organizations." M.I.T. Press.

[PNWCG 02] Pacific Northwest Wildfire Coordinating Group 2002. « Living with fire (a guide for the homeowner) » <http://www.or.blm.gov/nwfire/docs/Livingwithfire.pdf>

[Pollack et Horty 99] M.E. Pollack, J.F. Horty (1999). "There's more to life than making plans". *AI Magazine*, vol. 4, pp.13-22.

[Pollack et McCarthy 99] M. Pollack, C. McCarthy. (1999). « Applying rationale-based monitoring to causal-link planning ». Technical Report 99-09, Univ. of Pittsburgh Dept. of Computer Science, Pittsburgh, PA.

[Popper 59] K. Popper, (1959). *The Logic of Scientific Discovery*. Hutchinson, London.

[PP 99] Partners in Protection, (1999). "FireSmart: protecting your community from wildfire." Partners in Protection, Edmonton, Ata.

[Prometheus 05] Site web: <http://www.firegrowthmodel.com/index.cfm>. dernière mise à jour: 2005.

[Rao et Georgeff 95] A.S. Rao, M.P. Georgeff, (1995). « BDI agents: from theory to practice ». In Victor Lesser, editor, Proceedings of the First International Conference on *Multi-Agent Systems*, San Francisco, CA, MIT Press. pp. 312-319.

[Rathbun et al., 02] D. Rathbun, S. Kragelund, A. Pongpunwattana, B. Capozzi, (2002). "An evolution based path planning algorithm for autonomous motion of a UAV through

uncertain environments,” in *Proc. IEEE Digital Avionics Systems Conference*, vol. 2, pp. 8D2-1 – 8D2-12.

[Raubal 01] M. Raubal, (2001). “Agent-based Simulation of Human Wayfinding: A Perceptual Model for Unfamiliar Buildings.” Ph.D. Thesis, Vienna University of Technology, Vienna.

[Reese et Stout 99] B. Reese, B. Stout, (1999). “Finding a Pathfinder.” In Proceedings of the *AAAI 99 Spring Symposium on Artificial Intelligence and Computer Games*.

[Resnick et Turtles 95] M. Resnick, (1995). “Turtles, Termites and Traffic Jams”, *MIT Press*, Cambridge, US.

[Richards 90] G.D. Richards, (1990). “An elliptical growth model of forest fire fronts and its numerical solution.” *Int. J. Num. Methods Eng.* vol. 30 pp. 1163-1179.

[Richards 95] G.D. Richards, (1995). “A general mathematical framework for modelling two-dimensional wildland fire spread.” *Int. J. Wildland Fire* vol. 5(2) pp. 63-72.

[RNC 01] Ressources naturelles du Canada 2001, « Les feux de forêt. » Direction générale du Programme Scientifique, Service canadien des forêts, Ottawa.

[RNC 02] Ressources naturelles Canada septembre 2002, site Web : http://www.nrcan-nrcan.gc.ca/cfs-scf/science/prodserv/firereport/firereport_e.html

[Rodrigues et Raper 97] A. Rodrigues, J. Raper, (1997), *Defining Spatial Agents, In Spatial Multimedia and Virtual Reality Research Monograph*, published by Taylor and Francis.

[Rodriguez et al., 05] S. Rodriguez, V. Hilaire, A. Koukam, (2005). “Holonc Modelling of Environments for Situated Multi-Agent Systems.” The Second International Workshop on *Environments for Multiagent Systems*. In *AAMAS'05*, Utrecht, Holland.

[Rogers et al., 99] S. Rogers, C.N. Fiechter, P. Langeley, (1999). “An adaptive interactive agent for route advice.” In Proceedings of *Autonomous Agents '99*, pp. 198-205. New York, New York: ACM Press.

[Rogers et al., 99] T. Rogers, R. Ross, V. Subrahmanian. (1999). “Impact: A system for building agent applications.” *Journal of Intelligent Information Systems*. vol. 14, pp. 95-113.

[Rothkegel et al., 98] R. Rothkegel, K. F. Wender, S. Schumacher, (1998). « Judging spatial relations from memory. » In C. Freksa, C. Habel, & K. F. Wender (Eds.), *Spatial Cognition - An interdisciplinary approach to representation and processing of spatial knowledge* Berlin: Springer. pp. 79-105.

[Russel et Norvig 95] S.J. Russel, P. Norvig, (1995). *Artificial Intelligence, a Modern Approach*, Prentice Hall.

[Sadeh-Konicpol et al. 03] N. Sadeh-Konicpol, D. Hildum, D. Kjenstad, (2003). “Agent-based e-Supply Chain Decision Support”. *Journal of Organizational Computing and Electronic Commerce*, Vol. 13 (3).

[Sahli 01] N. Sahli (2001), « élaboration d'une approche à base d'agents mobiles pour une interoperabilité dans un environnement dynamique et application à l'évacuation aéromédicale. » Master thesis. University Laval.

[Sahli et Moulin 03] N. Sahli, B. Moulin, (2003). “La planification agent dans un environnement dynamique et distribuée: entre la réalité et la simulation.” *Revue des sciences et technologies de l'information*. édition Lavoisier., Hammamet, Tunisie, pp. 116-127.

[Sahli et Moulin 05a] N. Sahli, B. Moulin, (2005). “Real-World Pathfinding using Agent-Based Simulation”. *Agent-Directed Simulation Symposium Part of the 2005 Spring Simulation Multiconference*.

[Sahli et Moulin 05b] N. Sahli, B. Moulin, (2005). “Agent-based geo-simulation to support human planning and spatial cognition.” In workshop *Multi-Agent-Based Simulation (MABS)* in International Conference *AAMAS'05*. Utrecht, Holland.

[**Salisbury et Tallis 93**] M. Salisbury et H. Tallis (1993). « Automated Planning and Replanning for Battlefield Simulation ». In Proceedings of the Third Conference on *Computer Generated Forces and Behavioral Representation*, Orlando, FL, pp. 243-254.

[**Sauvagnargues et al., 00**] S. Sauvagnargues, B. L'Héritier, T. Boussardon (2000) « Implementation of a GIS Application for French Fire-fighters in Mediterranean Area. » in 2nd International Workshop on *Telegeoprocessing*, Nice-Sophia Antipolis, France.

[**SCF 03**] Service canadien des forêts 2003, site Web http://www.nofc.forestry.ca/index_f.html

[**Schoppers 87**] M. Schoppers (1987). « Universal Plans for reactive robots in unpredictable domains » In International Joint Conference on *Artificial Intelligent*.

[**Schruben 90**] L.W. Schruben, (1990). "Establishing the credibility of simulations." *Simulation* vol. 34(3), pp. 101-105.

[**Schweitzer et Zimmermann 01**] F. Schweitzer, J. Zimmermann, (2001). "Communication and Self-Organization in Complex Systems: A Basic Approach", in: *Knowledge, Complexity and Innovation Systems* (Eds. M. M. Fischer, J. Fröhlich), Springer, Berlin. pp. 275–296

[**Serban 02**] G. Şerban, (2002). "Real Time Learning in Agent Based Systems." In the 4th International Workshop on *Symbolic and Numeric Algorithms for Scientific Computing*, Timisoara, Romania.

[**Servat et al., 98**] D. Servat, E. Perrier J-P. Treuil, A. Drogoul, (1998). "When Agents Emerge from Agents: Introducing Multi-Scale Viewpoints." in Multi-Agent Simulations. Proceedings of *MABS'98*, LNAI, Springer-Verlag. vol. 1534, pp. 183-198

[**Servat et Drogoul 02**] D. Servat, A. Drogoul, (2002). "Combining amorphous computing and reactive agent based systems: a paradigm for pervasive intelligence?", proceedings of the First International Joint Conference on *Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, Bologna, Italy.

[**Shannon 75**] R.E. Shannon, (1975). *Systems Simulation: The Art and the Science*. Prentice Hall, Englewood Cliffs, NJ.

[**Sheth et Maes 93**] B. Sheth, P. Maes, (1993). “Evolving Agents for Personalised Information Filtering”, In *Proceedings of the 9th IEEE Conference on Artificial Intelligence for Applications*.

[**Shoham et Tennenholtz 92**] Y. Shoham et M. Tennenholtz (1992) “On the synthesis of useful social laws for artificial agent societies”. In proceeding of *AAAI-92*, pp. 276-281.

[**Sigurd et How 03**] K. Sigurd, J. How, (2003). “UAV trajectory design using total field collision avoidance,” in Proc. *AIAA Guidance, Navigation, and Control Conference*.

[**Sloman et Poli 96**] A. Sloman, R. Poli. (1996). “SIM AGENT: A toolkit for exploring agent designs.” In M. Wooldridge, J. Mueller, and M. Tambe, editors, *Intelligent Agents Vol II (ATAL-95)*, Springer-Verlag, pp. 392–407.

[**sopfeu 02**] site Web: http://www.sopfeu.qc.ca/francais/index_statistiques.html données parues en 2002.

[**SPQ 03**] Site Web de la Sécurité Publique Québec 2003. « Guide des opérations à l'intention des services de sécurité incendie », dernière mise à jour : janvier 2003. http://www.msp.gouv.qc.ca/incendie/incendie.asp?txtSection=chefs&txtCategorie=&txtSousCategorie=&txtNomAutreFichier=mot_ministre.htm&txtAutreFichier=2

[**Stam 00**] J. Stam, (2000). “Interacting with Smoke and Fire in Real Time.” *Communications of the ACM*, vol. 43(7), pp. 76-83.

[**Stejo 01**] site web : <http://www-und.ida.liu.se/~stejo127/> document : path.doc, dernière mise à jour: nov. 2001.

[**Stentz 95**] A. Stentz, (1995). “The Focussed D* Algorithm for Real-Time Re-planning.” In Proceedings of the International Joint Conference on *Artificial Intelligence*.

[**Stevens et Coupe 78**] A. Stevens, P. Coupe (1978) “Distortions in Judged Spatial Relations.” *Cognitive Psychology* vol. 10 pp. 422-437.

[**Stolzenburg et al., 99**] F. Stolzenburg, O. Obst, J. Murray, B. Bremer, (1999). *Spatial agents implemented in a logical expressible language*. In Manuela M. Veloso, editor, Proceedings of the 3rd International Workshop on RoboCup in Conjunction with 16th Joint International Conference on *Artificial Intelligence*, Stockholm. IJCAI press, pp. 205-210.

[**Stone et Veloso 99**] P. Stone, M. Veloso (1999). "Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork." *Artificial Intelligence*. 110(2) pp. 241–273.

[**Stout 96**] B. Stout, (1996). "Moves: Intelligent Pathfinding." In *Game Developer Magazine* October. pp. 28-35.

[**Surdu et Hill 00**] J. R. Surdu, J.M.D. Hill, (2000). "Anticipatory Planning Support System." In Proceedings of the *Winter Simulation* Conference (Orlando, Florida, December 10-13). pp. 950-957

[**Sutton et Barto 81**] R.S. Sutton, A.G. Barto, (1981) « An adaptive network that constructs and uses an internal model of its environment. » Proceedings of the Seventh International Conference on *Machine Learning*, pp. 216-224.

[**Swarm 05**] Site Web de: The Swarm Development Group. http://www.swarm.org/wiki/Main_Page dernière mise à jour 2005.

[**Sycara et al., 03**] K. Sycara, M. Paolucci, M. Van-Velsen, J. Giampapa. (2003). "The RETSINA-MAS infrastructure." special joint issue of *Autonomous Agents and MAS* pp. 29-48.

[**Tambe 97**] M. Tambe (1997). « Towards flexible teamwork.» *Journal of Artificial Intelligence Research*, vol. 7 pp. 83–124.

[**Tang et al., 01**] P. Tang, Y. Yang, X. Li, (2001). "Dynamic obstacle avoidance based on fuzzy inference and transposition principle for soccer robots," in *IEEE*, 10th International Conference on *Fuzzy Systems*, vol. 3. IEEE, pp. 1062 – 1064.

[**Terral 96**] S. Terral, (1996). « Fighting talk. » *GIS Europe*, vol. 5(6) pp. 46-48.

[Tomlin et al., 00] C. Tomlin, J. Lygeros, S.S. Sastry, (2000). “A game theoretic approach to controller design for hybrid systems,” in Proc. *IEEE*, vol. 88, pp. 949 – 970.

[Troitzsch 97] K.G. Troitzsch, (1997). “Social science simulation - origins, prospects, purposes”. *Simulating Social Phenomena* vol. 456 pp. 41-54

[Turing 50] A.M. Turing, (1950). “Computing machinery and intelligence.” *Mind* 59 (Oct.), pp. 443-460.

[Turmel 01] C. Turmel (2001), cours simulation, Ecole polytechnique Montréal. Site Web : http://www.cours.polymtl.ca/ind6202/ind6202_seance_2.pdf

[Turner et Penn 02] A. Turner, A. Penn, (2002). “Encoding natural movement as an agent-based system: an investigation to human pedestrian behaviour in the built environment.” *Environment and Planning B: Planning and Design*.

[Tversky 81] B. Tversky, (1981). « Distortions in memory for maps. » *Cognitive Psychology*, vol. 13, pp. 407-433

[Valavanis et al., 00] K. Valavanis, Hebert, Tsourveloudis, (2000). “Mobile robot navigation in 2-d dynamic environments using an electrostatic potential field,” in *Trans. IEEE Systems, Man and Cybernetics Part A*, vol. 30, pp. 187–196.

[Vanbergue et al., 00] D. Vanbergue, J-P. Treuil, A. Drogoul, (2000). “Modelling urban phenomena with cellular automata.” In *Applications of Simulation to Social Science*. G. Ballot & G. Weisbuch, eds. Hermes, Paris, France.

[Velooso et al., 95] M. Veloso, J. Carbonell, M.A. P'erez, D. Borrajo, E. Fink, J. Blythe. (1995). « Integrating planning and learning: The Prodigy architecture ». *Journal of Experimental and Theoretical Artificial Intelligence* vol. 7(1) pp. 81-120.

[Vermeire et Watson 94] J.C. Vermeiren, C.C. Watson (1994), “New technology for improved storm risk assessment in the Caribbean”, *Disaster Management*, vol. 6(4) pp. 191-196.

[Von Martial 90] F. von Martial (1990). « Interactions among autonomous planning agents ». In Y. Demazeau and J.-P. Müller, editors, *Decentralized AI - Proceedings of the First European Workshop on Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW-89)*. Elsevier Science Publishers B.V. Amsterdam, The Netherlands.

[Von Martial 92] F.V. Martial (1992), *Coordinating Plans of Autonomous Agents*. Springer-Verlag : Heidelberg, Germany.

[Wass de Czege 99] H. Wass de Czege, (1999). « Anticipatory Planning ». personal communication To John Surdu.

[Weld 94] D.S. Weld (1994). “An Introduction to Least Commitment Planning”. *AI Magazine* vol. 15(4) pp. 27-61.

[Wender et al., 97] K.F. Wender, M. Wagener-Wender, R. Rothkegel (1997). “Measures of spatial memory and routes of learning.” *Psychological Research*, vol. 59, pp. 269-278.

[Weynes et al., 05] D. Weynes K. Schelfthout, T. Holvoet. (2005). “Exploiting a virtual environment in a real-world application.” In workshop *Environment for Multi_agent Systems (E4MAS)* in International Conference *AAMAS'05*. Utrecht, Holland.

[Wolfe et Jackson 89] J. Wolfe, R. Jackson, (1989). “An investigation of the need for algorithmic validity.” *Simulation & Games*, vol. 20, pp. 272-291.

[Wooldridge 99] M. Wooldridge, (1999). “Intelligent Agents.” In *Multiagent Systems--A Modern Approach to Distributed Artificial Intelligence*, edited by G. Weiss (Cambridge, MA: MIT Press), pp. 27-77.

[Xian Wen et al., 97] Z. XianWen, J. Ping, S. XiaoBing, G. ShiZhong, (1997). « Prevention and control of forest fire using satellite data. » Actes, *XIe congrès forestier*

[Yahya et al., 98] A. Yahja, A. Stentz, S. Singh, B. Brummit, (1998). “Framed-Quadtree Path Planning for Mobile Robots Operating in Sparse Environments.” In Proceedings, IEEE Conference on *Robotics and Automation*, (ICRA), Leuven, Belgium.

[Youngblood 93] S.M. Youngblood, (1993). “Literature review and commentary on the verification, validation, and accreditation of models and simulations.” *SCSC*, pp. 10–17.

[Zerger et Smith 03] A. Zerger, D.I. Smith (2003). « Impediments to using GIS for real-time disaster decision support. » *Computers, Environment and Urban Systems*, vol. 27(2), pp. 123-141.

[Zhu et al., 98] Z. Zhu, X. Lin, D. Shi, G. Xu, (1998). “WA Single Camera Stereo System for Obstacle Detection.” World Multiconference on *Systemics, Cybernetics and Informatics (SCI'98)* 4th International Conference on *Information Systems Analysis and Synthesis (ISAS'98)*, Orlando, U.S.A, vol. 3, pp. 230-237.

[Zilberstein 93] S. Zilberstein, (1993). “Operational Rationality through Compilation of Anytime Algorithms.” Ph.D. Dissertation, Computer Science Division, University of California at Berkeley.

ANNEXES

Annexe A : Les Feux de forêts

1. Généralités sur les feux de forêts

1.1. Ampleur

1.1.1. Premier constat

Chaque année des millions d'hectares de terres boisées brûlent, faisant dépenser des milliards de dollars pour leur extinction et occasionnant d'immenses pertes en bois, biens et espaces de récréation et, dans certains cas, vies humaines. Le phénomène est encore plus présent en Amérique du Nord. En 2002, le Canada a connu 7242 feux de forêt [RNC 02] qui ont ravagé plus de 2600000ha.

1.1.2. Le double rôle du feu

Les feux de forêts sont-ils donc aussi néfastes?

D'après [RNC 01], le feu de forêt représente à la fois un agent de renouvellement pour la forêt et une menace pour la vie et les biens matériels de la population. Du point de vue industriel, le feu entraîne la destruction de précieuses ressources en bois et représente une menace pour les installations et les structures souvent assez coûteuses. Il peut entraîner également la fermeture de secteurs touristiques. Du point de vue environnemental, les opinions varient. Il peut améliorer les habitats de certains gibiers et créer des clairières pour les humains. Cependant, il reste une menace importante pour la vie et pour les biens.

Le feu a aussi une importance écologique bien connue des scientifiques. Les feux qui ont dévasté le parc national Yellowstone au Wyoming en 1988 l'ont prouvé. En effet, en raison des pratiques de suppression des incendies, le combustible s'était accumulé dans le parc depuis des années ce qui explique l'ampleur de ces feux. Jusqu'aux années 80, plusieurs

considéraient encore les feux de forêts comme néfastes en toutes circonstances et voulaient absolument les supprimer tout le temps. Le grand public est maintenant plus conscient du rôle des feux de forêts et soutient de plus en plus une approche plus éclairée pour gérer les feux, qui pèse le pour et le contre des feux et agit en conséquence de façon optimale.

1.2. Causes des feux

On peut attribuer ce phénomène principalement aux conditions météorologiques, aux relations forêt-homme et à la présence d'associations de végétaux (combustibles) [Canakcoglu et Ozkazanc 97]

Environ 60% des feux au Canada sont attribuables à des activités humaines : brûlage de déchets, feux de camp, cigarette, étincelles produites par le frottement des roues de train sur les rails... La plupart de ces feux sont allumés dans des zones peuplées et près des routes. La détection de ces feux est donc rapide et la facilité d'accès fait que le feu est souvent maîtrisable D'après [RNC 01].

Le Tableau A-1 [sopfeu 02] montre les différentes causes et les aires brûlées correspondantes dans la province du Québec dans la période 1998-2002. Il est clair d'après le diagramme que des causes naturelles (la foudre) font le plus de ravage dans les forêts québécoises. En effet, les feux déclenchés par la foudre surviennent souvent sous des conditions météorologiques favorables à la propagation du feu (chaleur, humidité, sécheresse...) et à des endroits reculés. Lorsque les efforts d'extinction débutent, compte tenu du temps requis pour acheminer les ressources nécessaires (personnel et équipement) à ces endroits éloignés, les feux ont, dans bien des cas, dépassé le stade où ils peuvent être supprimés rapidement.

Tableau A-1. Statistiques sur les incendies de forêt de 1998 – 2002 au Québec

ANNÉE	INCENDIES ET SUPERFICIES AFFECTÉES	FOUDRE	CHEMIN DE FER	OPÉRATIONS FORESTIÈRES	OPÉRATIONS INDUSTRIELLES	INCENDIAIRES	RÉSIDENTS	RÉCRÉATION	DIVERSES	TOTAL
2002	Incendies	249	13	34	52	39	160	250	6	803
	Hectares	231,605.1	5.3	32.9	120.8	23.3	127.0	456.6	72.0	232,443.0
2001	Incendies	248	58	60	60	44	232	267	1	970
	Hectares	615.5	53.9	15.6	81.0	56.9	298.7	233.4	3.0	1,358.0
2000	Incendies	74	10	49	32	33	135	141	1	475
	Hectares	488.4	1.1	23.3	22.3	14.5	81.8	160.5	2.0	793.9
1999	Incendies	237	11	81	52	45	229	350	0	1005
	Hectares	24,688.8	9.2	1,317.8	64.3	35.2	795.2	896.3	0.0	27,806.8
1998	Incendies	160	7	52	55	48	202	268	5	797
	Hectares	686.0	15.1	8.9	26.2	9.3	5,014.5	6,085.3	0.0	11,845.3
Total	Incendies	968	99	276	251	209	958	1276	13	4050
	Hectares	258,083.8	84.6	1,398.5	314.6	139.2	6,317.2	7,832.1	77.0	274,247.0

2.4. Le système de commandement

Pour gérer efficacement une intervention, il est important de s'appuyer sur un système de commandement clairement défini et qui s'applique à tous les types d'intervention du service d'incendie. Ceci permet d'établir une structure et une coordination de la gestion des opérations d'urgence.

Cette partie précise les éléments de la structure de commandement à mettre en place selon l'ampleur ou le type d'intervention ainsi que le processus de commandement. Elle traite également de l'organisation géographique des lieux lors de différentes interventions. La plupart des nos informations (sauf mention) proviennent de La Sécurité Publique du gouvernement de Québec [SPQ 03].

2.4.1. La structure de commandement

2.4.1.1. Le responsable des opérations

Le responsable des opérations (PC) gère l'intervention et la sécurité du personnel. Son rôle consiste entre autres à :

- Analyser continuellement la situation;
- Déterminer les priorités (sauver des vies, circonscrire l'incendie, sauver les biens, etc.);
- Déterminer les stratégies d'intervention (offensive / défensive / non-intervention);

- Déterminer les tactiques en fonction des priorités et de la stratégie retenues;
- Faire un plan d'opération
- Vérifier l'atteinte des résultats.

Un bon responsable des opérations doit avoir [Nicolas et Beebe 99]:

- Une connaissance complète sur la prévention et le contrôle du feu.
- L'habilité à prendre des décisions rapides et efficaces basées sur sa connaissance et son expérience.
- La faculté de superviser une variété d'activités opérationnelles et d'investigation.
- Une connaissance suffisante pour utiliser les ordinateurs, les radios, les instruments météorologiques et les cartes.
- La capacité de calculer et d'interpréter les indices de feu et de rédiger de façon précise des rapports sur la situation.

C'est le PC qui décide de la structure de commandement appropriée à l'intervention. Cette structure peut évoluer selon l'ampleur de l'intervention ou le nombre d'intervenants nécessaires.

2.4.1.2 Les niveaux de commandement

Dans toutes structure de commandement, on trouve trois niveaux de commandement (Figure A-2) : stratégie, tactiques et tâches.

Stratégie

Plus haut niveau de la structure de commandement, le niveau « stratégie » est assumé par le responsable des opérations. C'est à ce niveau que le PC doit établir les grands objectifs stratégiques, déterminer les priorités, assigner les ressources, prévoir les résultats, choisir la stratégie d'intervention et fixer les objectifs du niveau « tactiques ».

Tactiques

À ce niveau, le commandement est assumé par les responsables de secteur. Ces derniers dirigent leur équipe pour atteindre des objectifs tactiques spécifiques.

Tâches

Les « tâches » désignent les activités et manœuvres effectuées par le personnel et les officiers pour réaliser les tactiques préconisées.

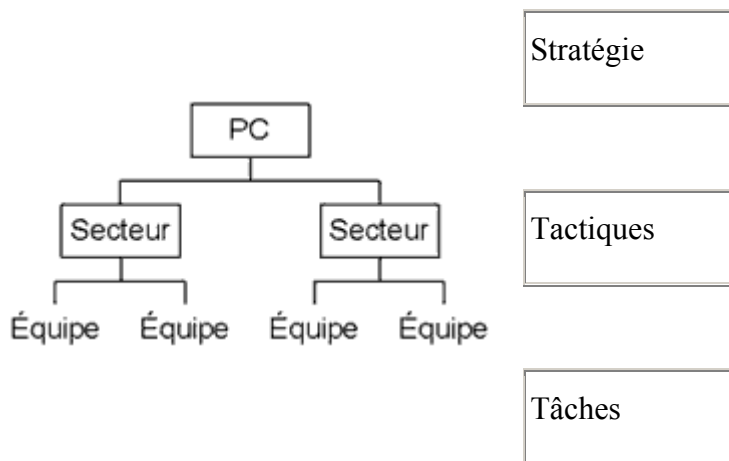


Figure A-2. Niveaux de commandement

Il arrive qu'un officier cumule plus d'un niveau de commandement. Lors d'une intervention mineure ou au début d'une intervention, le PC peut cumuler les niveaux « stratégie » et « tactiques ». Dans certaines interventions, un responsable de secteur, par exemple, peut assumer les niveaux « tactiques » et « tâches ». La structure de commandement doit être souple pour s'adapter à toute intervention selon l'ampleur, l'intensité et la complexité de celle-ci. Chaque fois que le besoin s'en fait sentir, le PC peut modifier la structure de commandement et déléguer les responsabilités des niveaux de commandement.

2.4.1.3. Les responsables de secteur

Le PC peut faire évoluer la structure de commandement lorsque la situation devient plus complexe. Il regroupe alors les équipes de travail sous la direction de responsables de secteur, généralement des officiers. Le nombre d'interlocuteurs qui communiquent directement avec le PC ne devrait pas dépasser cinq. Le responsable de secteur dirige les équipes de pompiers dans le secteur qui lui est assigné et transmet des comptes rendus périodiques au PC.

2.4.1.4. Les responsables de division

Le PC peut, au besoin, continuer de faire évoluer la structure de commandement. Il peut regrouper les responsables de secteur par division afin de ne pas dépasser le nombre de cinq interlocuteurs qui communiquent directement avec lui. Le responsable d'une division dirige les responsables de secteurs et transmet des comptes rendus périodiques au PC.

2.4.2. Le processus de commandement

Objectif : Préciser le processus de commandement qui s'applique lors d'une intervention

2.4.2.1. La prise de commandement

Un officier ou le premier pompier qualifié arrivé sur les lieux prend le commandement des opérations dès son arrivée. Il assume le commandement tant qu'il n'a pas été relevé par un officier de grade supérieur.

Si l'officier ou le premier pompier qualifié arrivé sur les lieux ne peut pas prendre le commandement des opérations en raison d'une situation critique, comme la nécessité de participer à un sauvetage, il y a un « PC requis ». Un autre officier doit alors assumer la fonction de responsable des opérations dès son arrivée. Cette procédure est exceptionnelle. Elle ne peut être utilisée qu'une seule fois lors d'une même intervention. Le message de « PC requis » doit parvenir au personnel et, s'il y a lieu, au centre de répartition.

2.4.2.2 Les modes de commandement

Le responsable des opérations (PC) peut choisir un des trois modes de commandement : le mode stationnaire, le mode reconnaissance ou le mode attaque. Généralement, le PC favorise le mode stationnaire. Les modes reconnaissance et attaque exigent des conditions particulières.

Le mode stationnaire

En mode stationnaire, le PC assume le commandement de l'intervention à partir d'un poste de commandement fixe. Le choix de son emplacement peut varier selon la catégorie de l'intervention.

Le mode reconnaissance

En mode reconnaissance, le PC assume le commandement de l'intervention tout en effectuant une reconnaissance des lieux avec son équipe.

Le mode attaque

En mode attaque, le PC assume le commandement de l'intervention tout en participant, avec son personnel, à l'attaque de l'incendie. Ce mode s'applique surtout lorsque le PC juge que la situation peut être rapidement maîtrisée et que sa participation peut faire la différence. Dans le doute, le PC devrait opter pour un mode stationnaire.

2.4.2.3. Le transfert de commandement

Lorsqu'un officier relève le PC et qu'il assume lui-même le commandement des opérations, il y a transfert de commandement.

Comment faire le transfert de commandement?

Le transfert de commandement se fait, de préférence, de vive voix entre l'officier et le PC à relever. Le nouveau PC doit aviser le personnel de l'intervention et, s'il y a lieu, le centre de répartition.

Le compte rendu du PC relevé

Le PC relevé doit donner un compte rendu le plus complet possible de la situation à l'officier qui prend le commandement. Le compte rendu doit comprendre les informations suivantes :

- La situation à l'arrivée et la situation actuelle;
- La stratégie établie;
- Les tactiques mises en œuvre;
- La structure de commandement déployée, l'organisation géographique des lieux de l'intervention et la localisation des équipes;
- Les risques établis;
- Les ressources appelées en renfort.

La notion de responsabilité

L'officier le plus haut gradé qui se trouve sur les lieux de l'intervention demeure responsable de l'intervention même s'il n'en assume pas le commandement.

3.2. Technologies utilisées

3.2.1. La communication

Les gestionnaires des incendies utilisent une série de tours et de sites éloignés alimentés par pile solaire pour créer un réseau permettant de communiquer verbalement à l'aide de radios portatives, que ce soit sur le terrain ou du terrain jusqu'aux stations de base et aux postes de commande [ALCIF 02]. Les sites alimentés par pile solaire sont généralement situés sur des tours d'observation des incendies qui ne sont plus utilisées. Tous les sites radio sont reliés par un réseau de tours hertziennes et de lignes terrestres.

Ce réseau extensif fournit les capacités de communication nécessaires aux pompiers dans la brousse, aux postes d'attaque initiale, aux postes de commande et aux centres régionaux de lutte contre les incendies. Ce réseau garantit aussi des communications à l'échelle régionale avec tous les aéronefs contrôlés par le programme de gestion des incendies.

Ces radios portatives multi-canaux utilisées par le personnel sur la ligne de feu fonctionnent sur une bande basse VHF-FM [ALCIF 02]. Elles peuvent faire un balayage de toutes les tours et même avoir accès au réseau téléphonique après les heures normales et en cas d'urgence, lorsque le système radio de la station de base n'est pas en marche.

Les campements mobiles se servent d'un appareil transmetteur-récepteur sur une antenne locale situé à la station de base pour permettre une vaste gamme de communications avec les pompiers sur le terrain, les postes de commande et les aéronefs qui fournissent des services de lutte.

Il y a d'autres outils de communication utilisés dont le satellite mobile, des téléphones portatifs GSN et des téléphones cellulaires [ALCIF 02]. Le réseau informatique permet de se relier par téléphone au courriel, à l'Internet et à l'Intranet. Ce raccordement peut se faire avec une ligne téléphonique ordinaire, une liaison cellulaire ou satellite. Le réseau de communications radio permet également d'avoir accès à des stations météorologiques

automatisées qui fournissent des données sur la température cruciales pour planifier les activités de gestion des incendies de forêt.

3.2.2. Le système de positionnement universel

GPS, système de localisation reconnu pour sa précision. Développé pour les besoins de l'armée américaine à la fin des années 70, il est maintenant utilisé par l'aviation civile, les excursionnistes, les travailleurs forestiers, les chasseurs, etc. Il se compose d'une constellation de 24 satellites. À tout instant, partout sur le globe, les signaux d'au moins quatre satellites peuvent être captés. En évaluant le temps qui sépare la réception des signaux émis par les satellites, le récepteur peut obtenir la distance entre les satellites et, par triangulation, sa propre position.

L'introduction de récepteurs du système de positionnement universel (GPS) dans le programme de gestion des incendies améliore l'utilisation du système de géomatique.

Ces récepteurs GPS permettent d'identifier plus efficacement l'emplacement de feux non maîtrisés et de documenter d'autres caractéristiques et installations au sol ou en effectuant un survol du périmètre dans les airs [ALCIF 02].

Ces données cartographiques peuvent ensuite être entreposées dans une base de données et utilisées dans des systèmes de géomatique pour mettre à jour et analyser des cartes.

Le programme de gestion des incendies se sert de ces données pour effectuer une analyse spatiale et créer des outils de prise de décisions qui permettront d'améliorer la planification tactique et stratégique ainsi que le processus de prise de décisions. Les systèmes de géomatique et de positionnement universel sont devenus des éléments intégraux des opérations du programme de gestion des incendies.

3.2.3. La technologie du système géomatique

Les chefs des opérations ont toujours besoin de cartes pour pouvoir planifier les opérations de lutte [Nicolas et Beebe 99]. Généralement, deux types de cartes sont nécessaires :

Cartes régionales à l'échelle 1 :250 000 pour localiser le feu et acheminer les ressources au bon endroit.

Cartes locales à l'échelle 1 :50 000 pour aider les CP à prendre des décisions puisqu'elles contiennent des données détaillées sur les villages, les routes d'accès, les ressources d'eau, les types de végétation, etc.

Les deux types de cartes doivent être échelonnés selon le système *Grid* (utilisé par les services de l'armée) relié aux lignes de latitudes et de longitudes et utilisant la grille de *Universal Transverse Mercator* (UTM) [Nicolas et Beebe 99].

Certains pays dont le Canada ont intégré les technologies du système de géomatique (SG) dans leurs systèmes de gestion quotidienne des données et dans leurs outils de planification stratégique [ALCIF 02].

Le programme de gestion des incendies utilise un vaste ensemble de données spatiales se rapportant à la gestion des ressources. On retrouve dans ces ensembles de données les cartes, les inventaires des ressources forestières et terrestres et le système de données intégrées sur la nature et la géographie des régions.

Au Canada par exemple, un système de données intégrées sur la nature et la géographie a été conçu sur la base d'un système géomatique pour gérer l'entreposage des renseignements numériques sur les terres du Ministère des ressources Naturelles MRN [ALCIF 02]. Il permet d'entreposer, de préserver et d'accéder à des renseignements portant sur plus de 690 différents types de caractéristiques géographiques dont :

- routes
- lacs
- aires de frai du doré jaune
- aires de nidification du faucon pèlerin
- aires d'élevage des orignaux
- lignes de piégeage

Ces inventaires fournissent l'information nécessaire pour évaluer les ressources menacées lorsqu'un incendie est signalé.

3.2.4. La télédétection

Seulement 2 à 3% des ces incendies au Canada touchent plus de 200 ha. Mais ce sont eux qui sont responsables de près de 97% des surfaces brûlées chaque année [RNC 02]. Cette situation justifie la surveillance des feux à l'aide de détecteurs à faible et à haute résolution dans le but de déterminer l'emplacement, le comportement et la sévérité du feu ainsi que la surface brûlée. Compte tenu de la taille du Canada, nous considérons que seules les mesures par télédétection peuvent permettre d'examiner en temps quasi réel et de façon économique l'ensemble des feux de forêt sur le territoire canadien.

La télédétection au Canada

Le centre canadien de télédétection et du service canadien des forêts a développé un système de surveillance, de cartographie et de modélisation des feux, appelé Fire M3 [FireM3 05].

Le type de données fourni par le système Fire M3 est indispensable à la planification stratégique nationale menée par des organismes tels que le Centre inter-services des feux de forêt du Canada (CIFFC) et Parcs Canada. Les agences provinciales et territoriales de gestion des incendies de forêts ont aussi besoin de ce genre d'information pour l'élaboration de plans stratégiques, ainsi que pour les opérations tactiques de lutte contre les incendies de forêts. La surveillance des feux et l'établissement de cartes répertoriant les incendies à chaque fin de saison servent également à la mise à jour des inventaires nationaux et régionaux de la couverture boisée, à la préparation des rapports annuels faisant état des statistiques concernant les feux, à l'établissement des critères et indicateurs pour la gestion durable de la forêt et à la recherche sur les incendies et les changements géoclimatiques planétaires.

Le système Fire M3 [FireM3 05]

Le système Fire M3 détecte, surveille et cartographie automatiquement et quotidiennement tous les gros incendies de forêt actifs au moyen de données satellitaires. Ces données sont traitées quotidiennement par un système qui met en jeu des procédures automatiques d'analyse des images permettant la détection des feux de forêt en temps quasi réel. Le système prépare également les données pour leur introduction dans les modèles SIG de comportement des feux. Le Système canadien d'information sur les feux de végétation (SCIFV) produit ainsi des données relatives aux conditions météorologiques propices aux feux et au comportement de ces derniers, données qui, une fois intégrées, permettent de prédire de façon quantitative le comportement des feux actifs. Le système produit quotidiennement des cartes et des rapports sommaires illustrant la quantité de combustible consommé, l'intensité du front et le type d'incendie mis en cause.

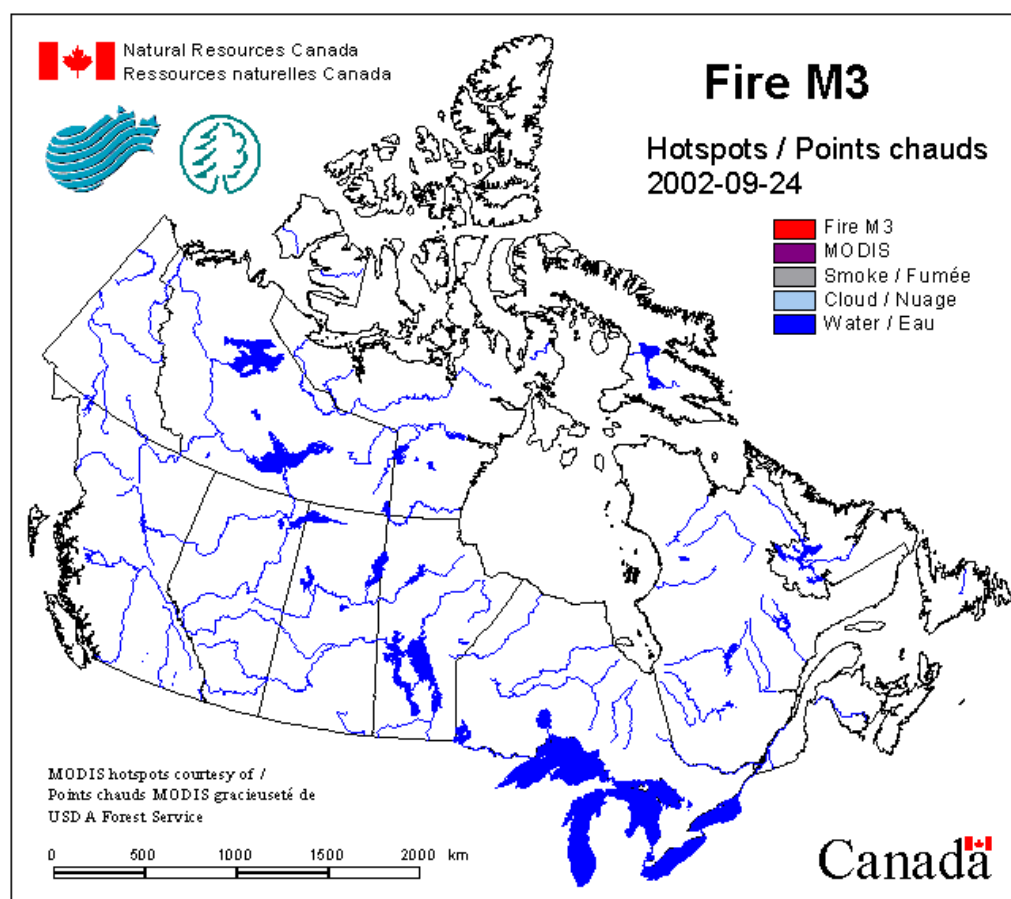


Figure A-3. Exemple de sortie du Fire M3

Les limites de la télédétection

Qu'est-ce qui pourrait alors empêcher l'utilisation de la télédétection comme seul moyen de suivi pour les feux de forêts?

Le coût

Les images satellites sont coûteuses. Le tableau II-5 suivant donne une idée sur un ordre des prix en 2001 :

Tableau A-2. Prix des images satellites

Satellite	Coût par image
Landsat	4.200\$US
Spot	2.500\$US
IRS	3.200\$US
Radarsat	200\$US
ERS	1.500\$US

Le cycle de visite *Revisit Cycle*

Un satellite se déplace selon une orbite prédéterminée qui le renvoie sur le même point terrestre une fois par période de quelques jours appelée *Revisit Cycle*. Cependant, les satellites sont équipés de senseurs avec des angles de vue ajustables qui sont capable de collecter plusieurs images d'un même point terrestre dans une seule période de *revisit cycle*. En effet, les senseurs peuvent collecter des images d'une région à des centaines de kilomètres du point de passage; cette région pourrait ne pas passer dans le *revisit cycle* par le satellite avant plusieurs jours. Cette capacité des senseurs est critique pour certaines applications qui nécessitent une surveillance fréquente d'une situation qui évolue rapidement comme les feux de forêt ou les inondations. Toutefois, le nombre d'images reste insuffisant ainsi que la résolution. En effet, les senseurs en allant chercher des images qui ne sont pas dans la région balayée, augmentent l'angle de capture ce qui entraîne une

baisse de résolution. A titre d'exemple, télédétection Canada collecte des images satellites dans le cadre de la surveillance des feux à raison de 4 fois par jour avec une résolution de 1.1km² (un pixel correspond à 1.1 km).

Pour avoir des images à haute résolution (cf. Figure A-4), il faut attendre une certaine période de temps qui peut être relativement longue.

Tableau A-3. Les cycles de visite des principaux satellites de télédétection

Satellite	Revisit cycle
Landsat	16 jours
Spot	1-26 jours
IRS	5-24 jours
Radarsat	44 jours
ERS	35 jours

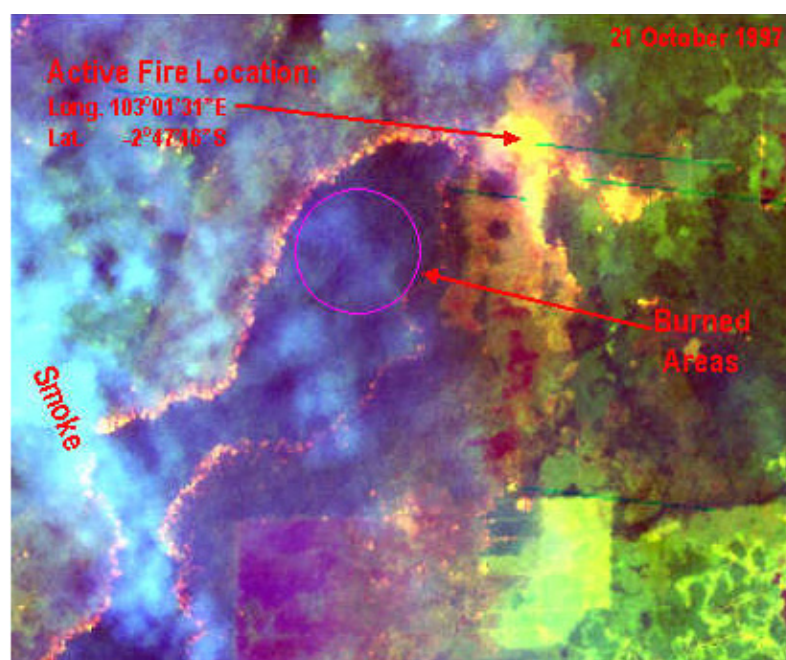


Figure A-4. Zoom du satellite Landsat lors d'un feu en Indonésie

3.2.5. Les senseurs web : nouvelles technologie prometteuse

Un détecteur autonome de feu (ou *autonomous fire detector* AFD) [Kremens et al., 01] est un dispositif (cf. Figure A-5) qui a la capacité de rapporter sa localisation et si le feu se situe à proximité. Il peut également être équipé pour enregistrer la température locale, le taux d'humidité et la vitesse du vent. Ces données sont ensuite transmises par radio aux pompiers ou à la centrale. Actuellement, les pompiers sont quasi-aveugles pendant la lutte contre le feu. Seuls les avions survolant la région peuvent leur fournir des informations sur la progression des feux. Les AFDs, par contre, peuvent leur fournir des données précises en temps réel et de façon directe.

Les AFDs seront lâchés à partir d'un avion ou placés manuellement par des équipes de pompiers dans les régions menacées par des incendies.

Les AFDs rapportent périodiquement leurs positions (via leur GPS interne) ainsi que les alarmes aux autres AFDs et à la centrale.

Plusieurs configurations sont possibles pour mettre en œuvre un système de surveillance basé sur les AFDs.

Dans [Eng 02] l'auteur appelle ces dispositifs les « *Tiny Fire Marshals* ». Il insiste sur le fait que bientôt cette technologie sera très utilisée.

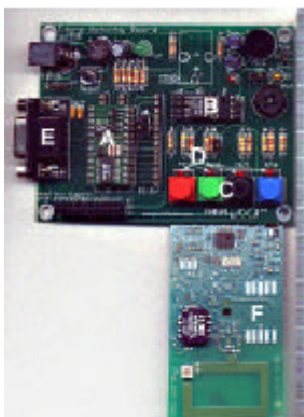


Figure A-5. Prototype de l'AFD avec microprocesseur et radio émetteur

3.3.2.1. Modèles de propagation

Plusieurs modèles de propagation existent. Nous allons encore une fois présenter ce qui est disponible au Canada. Le Modèle canadien de propagation des feux de végétation (MCPFV) [SCF 03], qui se base sur l'inventaire des complexes combustibles du Canada, est en cours d'élaboration. Ce modèle utilise les paramètres régissant le comportement des feux tels que définis dans la Méthode canadienne d'évaluation des dangers d'incendie de forêt (MCEDIF). Il incorporera de plus les dernières équations mises au point pour la modélisation de la propagation elliptique des incendies, les connaissances les plus récentes en science du comportement des incendies et les techniques les plus avancées de modélisation des conditions météorologiques diurnes. Le modèle sera également convivial, interactif et l'utilisateur pourra facilement le relier à un système d'information géographique (SIG) ou à un système de gestion de l'information.

Les objectifs du projet de MCPFV sont les suivants: Développer un modèle de propagation des incendies, s'inspirant de la propagation des ondes, explicite du point de vue spatial, qui pourra simuler la propagation d'un feu dans un paysage par incréments horaires ou journaliers. Appliquer le Modèle de propagation des incendies (MPI) à la prédiction en temps réel de la propagation d'un feu de végétation ou d'un feu dirigé. Appliquer le MPI de manière stratégique pour déterminer les risques d'incendies simples ou multiples qui menacent certaines valeurs exposées et évaluer l'efficacité de diverses stratégies pour l'atténuation des risques (p.ex. gestion du combustible). Faire du MPI un outil autonome convivial facile à intégrer à d'autres applications (p.ex. modélisation du paysage)

La Méthode canadienne de prévision du comportement des incendies de forêt (PCI) [SCF 03] fournit des estimations quantitatives de la vitesse de propagation de la tête des incendies, de la consommation de combustible et de l'intensité des incendies, ainsi qu'une description des incendies.

Au moyen d'un modèle elliptique de la propagation du feu, on produit des estimations de la superficie et du périmètre des zones touchées et de la vitesse d'extension du périmètre, de même qu'une indication du comportement du feu sur les flancs et l'arrière.

Le diagramme de la Figure A-6 présente les composantes de la Méthode canadienne de l'indice forêt-météo (**IFM**). Les calculs sont basés sur des observations quotidiennes consécutives de la température, de l'humidité relative, de la vitesse du vent et des hauteurs de pluie au cours des 24 dernières heures. Les six composantes standard fournissent des évaluations numériques du potentiel relatif d'incendie de végétation.

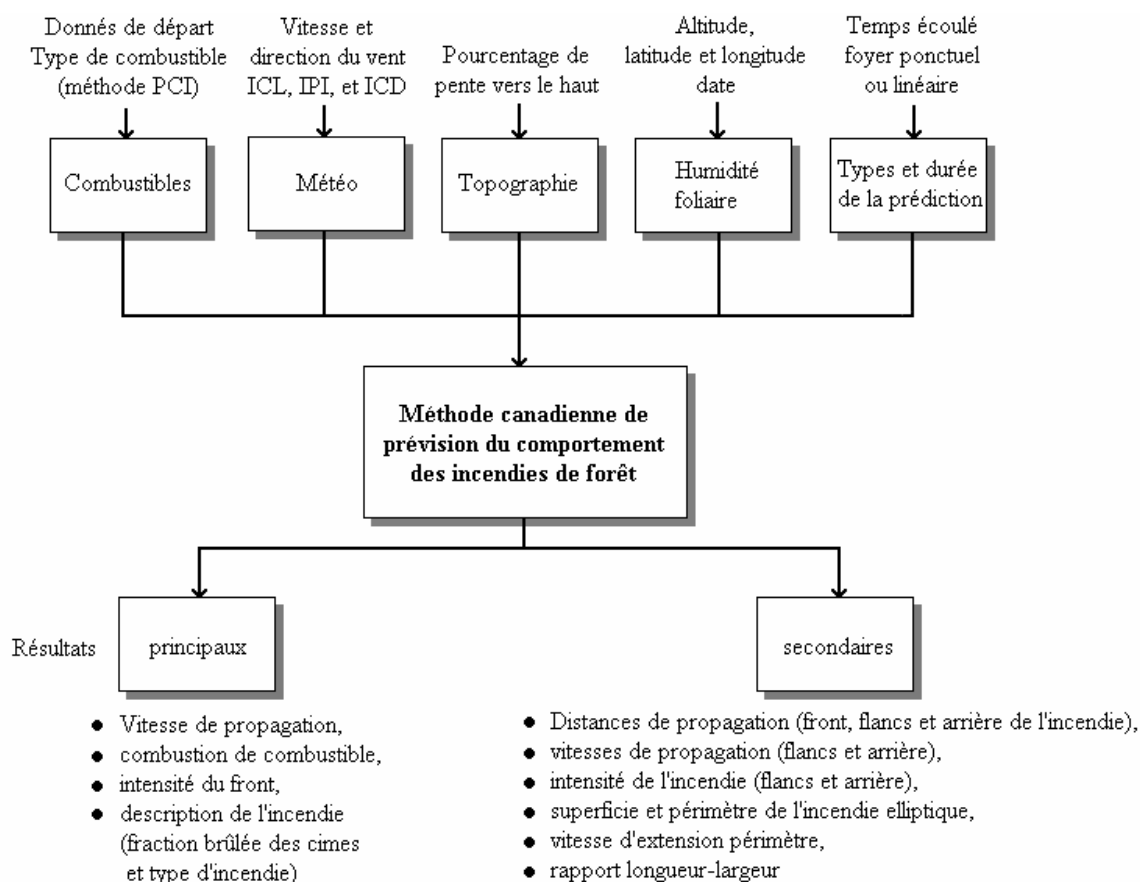


Figure A-6. Méthode canadienne de l'indice forêt-météo

Légende [SCF 03]:

Indice du combustible léger

L'indice du combustible léger (**ICL**) est une évaluation numérique de la teneur en eau de la litière et d'autres combustibles légers. Il donne une indication de l'inflammabilité du combustible léger.

Indice d'humidité de l'humus

L'indice d'humidité de l'humus (**IH**) est une évaluation numérique de la teneur moyenne en eau des couches organiques peu tassées de moyenne épaisseur. Il donne une indication de la combustion du combustible dans les couches organiques de moyenne épaisseur et les matières ligneuses de taille moyenne.

Indice de sécheresse

L'indice de sécheresse (**IS**) est une évaluation numérique de la teneur moyenne en eau des épaisses couches organiques compactes. C'est un indicateur utile des effets saisonniers des sécheresses sur les combustibles forestiers et du degré de latence du feu dans les épaisses couches organiques et les grosses billes.

Indice de propagation initiale

L'indice de propagation initiale (**IPI**) est une évaluation numérique du taux prévu de propagation du feu. Il combine les effets du vent et de l'indice du combustible léger sur le taux de propagation sans l'influence de quantités variables de combustible.

Indice du combustible disponible

L'indice du combustible disponible (**ICD**) est une évaluation numérique de la quantité totale de combustible disponible qui combine l'indice d'humidité de l'humus et l'indice de sécheresse.

Indice forêt-météo

L'indice forêt-météo (**IFM**) est une évaluation numérique d'intensité du feu qui combine l'indice de propagation initiale et l'indice du combustible disponible. Il constitue un indice général du danger d'incendie dans l'ensemble des régions forestières de Canada

3.3.2.2. Systèmes de simulation à base de modèles de propagation

Parmi les systèmes de simulation de la propagation des feux les plus connus et les plus utilisés, qui implémente un modèle de propagation :Farsite [Farsite 05] et Prometheus [Prometheus 05].

3.3.3.1. Le projet Phoenix

C'est un planificateur adaptatif en temps réel, basé sur une architecture agent capable d'intégrer plusieurs méthodes de planification IA. Phoenix est un simulateur des feux de forêt du parc Yellowstone, basé sur des données satellite et sur des agents autonomes ayant pour mission de contrôler ces données. Les agents sont organisés de façon hiérarchique et adaptent leurs plans selon les contraintes en temps réel. Il s'agit d'une planification adaptable incluant le *least-commitment planning* ainsi qu'une combinaison de planification réactive et délibérative. Le produit a été finalisé en 1994 et a été développé en LISP. D'autres détails sur le projet dans le Chapitre 2.

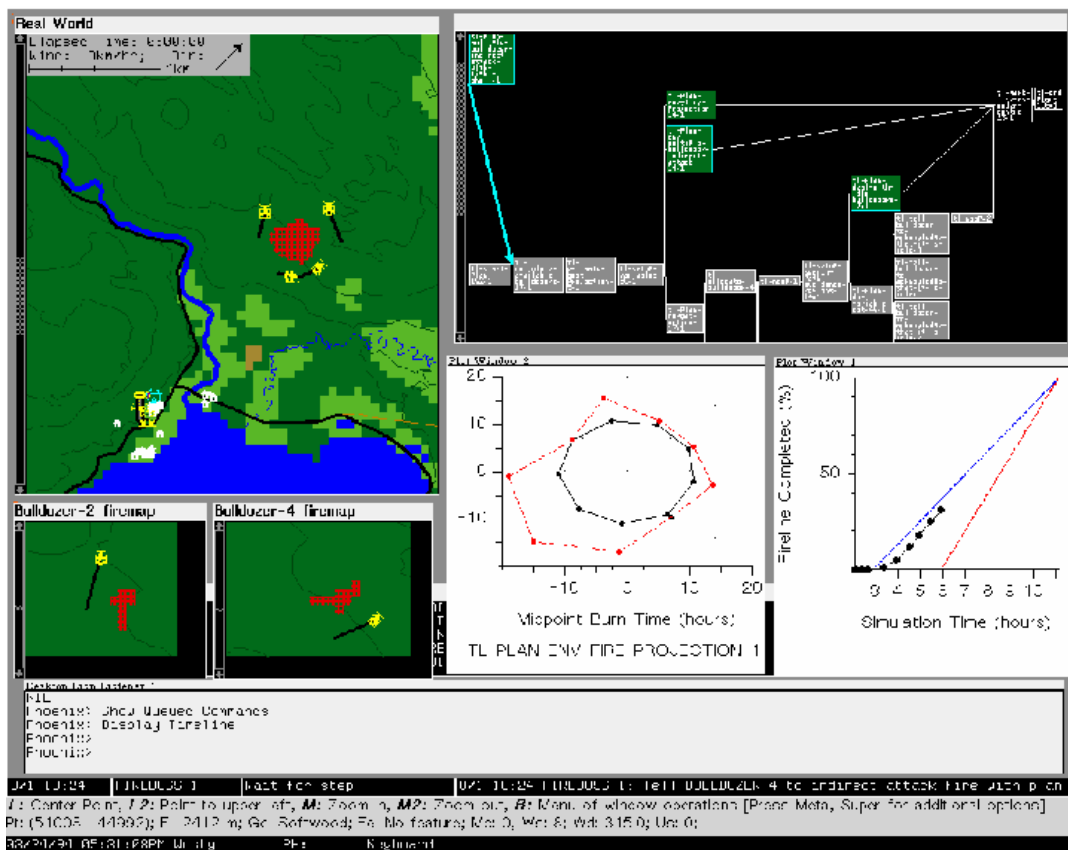


Figure A-7. Aperçu de l'environnement d'exécution de Phoenix

3.3.3.2. *Le système CHARADE*

Combining Human Assessment and Reasoning Aids for Decision-making in environmental Emergencies, CHARADE [Avesani et al., 00] est un projet mené par IRST-SRA (Italie). Il est actuellement utilisé par Centre Valabre (France), COAU, (Sardegna- Italie) et par des pompiers consultants en France et en Italie. Il a été finalisé en juillet 1995. L'objectif du projet était de réaliser une évaluation rapide de la situation et une meilleure exploitation des ressources lors d'un feu de forêt en utilisant des techniques de planification et de scheduling.

CHARADE est composé des modules suivants :

Le simulateur de l'environnement externe : il fournit des données de simulation (données prises à partir des capteurs)

Le module cartographique : il affiche les informations géographiques (routes, chemins de fer, villes, points d'eau, les bois...) ainsi que la propagation prévue des feux. Il permet aussi à l'utilisateur de décomposer la zone en feu en des secteurs d'intervention.

Le module de l'évaluation de la situation qui fournit les informations nécessaires pour une meilleure évaluation de la situation. Il fournit une description détaillée de la situation courante ainsi que la situation future prévue. Il renferme plusieurs composantes :

- Une base de données géographique thématique (végétation, ressources en eau, routes...)
- Un gestionnaire des données géographiques qui permet une exploitation facile de la base
- Un modèle de propagation du feu qui prend en compte les conditions météo, la végétation et la topographie (les pentes)
- Un gestionnaire des ressources qui trace et contrôle les ressources
- Un module cartographique qui montre la situation d'une façon orientée carte
- Un module d'évaluation qui a pour objectif d'élaborer des conclusions sur les paramètres physiques du feu, l'accessibilité, les points dangereux...

- Un module de scénario global qui gère les changements qui peuvent survenir dans l'environnement (nouveaux feux, changements météo...)

Le gestionnaire des ressources qui fournit des informations mises à jour sur les ressources déployées.

Le planificateur interactif qui a pour tâche de trouver un plan d'intervention à partir de cas antérieurs similaires. Il gère aussi les interactions de l'utilisateur pour éditer les plans. Le planificateur interactif est basé sur une architecture hybride basée sur les cas et basée sur les contraintes. Il est composé de trois parties :

Planificateur basé sur les cas : Il retrouve des plans partiels dans sa mémoire en utilisant une métrique de similarité qui prend en compte les paramètres du scénario (feu, météo, paramètres géographiques...). Les plans sont hiérarchiques (plan global et plans partiels).

Constraint Reasoner : il permet aux utilisateurs de mettre à jour de façon incrémentale le plan en cours (ajouter et supprimer des actions et des contraintes). Il vérifie et propage les changements dus à l'interaction avec les usagers.

Scheduler de ressources : Il affecte les ressources (humains et moyens) aux différentes actions à entreprendre en tenant compte des contraintes (capacité, temps...). L'utilisateur a toujours le moyen de changer cette affectation.

Id	Name	Commune	UTM_E	UTM_N	Danger	Status	Time
109	Bultei	unnamed	501.47	4480.85	4	part-plani	13 48
110	Bultei	unnamed	507.88	4485.29	2	asses-done	13 54
111	Esportat	unnamed	494.21	4472.72	1	new-fire	14 07

The screenshot displays the CHARADE software interface. At the top, a table lists hazard data. Below the table is a map showing a terrain with several red-outlined hazard zones and various icons. To the right of the map is a control panel with several sections: 'Editors' (Distance, Area), 'Sector' (Create sector, Delete sector, Move sector), 'Work Area' (Update WorkArea, Move WorkArea), 'Global Window' (Show checkbox), 'General Infos' (Position X:512146 Y:4479244), and navigation controls (Scale: 24300, Zoom, Pan).

Figure A-8. Aperçu de l'environnement d'exécution de CHARADE

Annexe B : Prometheus

1. Recherche dans le domaine des feux de forêts

Plusieurs produits ont été développés ces dernières années pour aider la lutte contre les feux de forêts. Ces produits peuvent être classés en quatre types selon leurs fonctions :

- Produits pour « le comportement du feu » : INFLAME, SALTUS, EFAISTOS, MEFISTO, Prometheus (Canada) et Farsite.
- Produits pour « la gestion des feux et détection » : PROMETHEUS (Europe), FIRE TORH, RAPSODI, FARSITE.
- Produits de « détection par satellite » : synthèse sur MEGAFIRES, INFLAME et CONTROL-FIRE_SAT, FIERS, FUEGO.
- Produits de « lutte et impact sur l'écosystème » : ACRE

Nous avons besoin d'un outil pour prédire la progression des feux dans la forêt. Les deux logiciels les plus connus et les plus avancés auxquels nous avons accès sont FARSITE [Farsite 05] et Prometheus [Prometheus 05]. FARSITE est basé sur le modèle américain alors que Prometheus est basé sur le modèle canadien. Étant donné notre partenariat avec Sustainable Resource Development (ASRD) (où Prometheus a été conçu), nous avons opté pour l'outil Prometheus. De plus, les concepteurs du logiciel nous ont fourni l'accès à l'interface COM de Prometheus, ce qui nous a permis d'utiliser directement (par code C++) le moteur du simulateur du feu à partir de notre application. La Section 2 donne un aperçu général du simulateur Prometheus et la Section 3 présente l'interface COM que nous avons utilisé pour accéder aux fonctionnalités du logiciel à partir de notre code source.

2. Le simulateur Prometheus

Comprendre le comportement, la propagation et les effets des feux majeurs est essentiel pour protéger les forêts et les habitations. L'une des meilleures méthodes pour comprendre

ces facteurs est la modélisation de la progression des feux. Les modèles de progression du feu sont utilisés pour simuler l'avancée des feux dans une forêt à essences et à topographie hétérogènes, basée sur des données météorologiques périodiques (toutes les heures) ou journalières. Le modèle de Prometheus (*Prometheus Canadian Wildland Fire Growth Model (CWFGM)*) a été développé pour cet objectif.

La recherche et le développement des modèles de propagation des feux ont subi plusieurs évolutions les dernières décennies. La plupart des modèles développés au Canada ont tenté d'utiliser l'approche de propagation des ondes. Le modèle Prometheus a été développé pour utiliser et améliorer les caractéristiques clefs des anciens modèles afin de créer un genre « d'État de l'art ». Le modèle est né du système *Canadian Forest Fire Behavior Prediction (FBP)* ainsi qu'à partir de la dernière vague d'algorithmes de propagation des feux développés à l'université de Brandon.

Le FBP est un système complexe et semi-empirique qui exprime de façon mathématique plusieurs types d'essences, climats et des facteurs topographiques influençant le comportement du feu. SBP est utilisé à travers le Canada et plusieurs autres pays pour prédire le comportement du feu de façon quantitative et structurée. Il est basé sur plus de 500 observations de feu et sur des feux bien documentés. Il peut produire en sortie des données sur les caractéristiques physiques du feu (p.ex. vitesse de propagation, consommation de l'essence, intensité du feu, etc.).

Les premiers modèles de propagation des feux ont été développés dans les années 70 (p.ex. [Kourtz et O'Regan 71]). Cependant, les dernières avancées en matière de SIG et d'autres technologies informatiques, combinées avec le développement de nouveaux modèles et algorithmes de propagation (p.ex. [Richards 90; Richards 95]), ont permis de créer au Canada un modèle de propagation avancé et orienté utilisateur. Des composantes de plusieurs modèles courants ont été combinées pour produire le CWFGM.

CWFGM a été développé en tant que modèle orienté processus, qui peut servir comme premier « moteur » pour plusieurs applications opérationnelles ou stratégiques dans le domaine des feux et la gestion forestière. Ces applications incluent :

- La prédiction (périodiquement) de la progression des feux.
- L'évaluation du risque potentiel que les grands feux peuvent causer.
- L'évaluation de l'efficacité de différentes stratégies de gestion dans le but de réduire le risque d'élargissement des feux.
- L'évaluation du risque relatif aux pertes causées par les feux (ou au cas où il y aura un feu) dans la zone sinistrée pour différentes stratégies de gestion forestière.

CWFGM a été conçu dans le but d'aider à développer, à réadapter et à implémenter un modèle déterministe basé sur la physique et qui permet de faire des évaluations opérationnelles et stratégiques sur le comportement potentiel des feux dans l'espace géographique. Le comité du projet a identifié les objectifs suivants :

- Développer un modèle spatialement explicite et selon l'approche de la propagation des ondes afin de simuler l'avancée des feux à travers l'espace géographique et ce à raison d'une période bien déterminée (tout les heures ou tous les jours, etc.).
- Démontrer l'applicabilité du CWFGM dans les prédictions temps réel et opérationnelles.
- Appliquer le CWFGM de façon stratégique pour déterminer le risque potentiel qu'un feu ou plusieurs feux pourraient représenter.
- Développer CWFGM comme une application à part qui est facile à utiliser et qui peut s'intégrer avec d'autres applications.

Étant donné que nous avons notre propre plate-forme de simulation (MAGS), nous avons besoin des données sur les prédictions des feux (résultats fournis par Prometheus) directement dans notre plate-forme. En d'autres termes, nous avons besoin de lancer le moteur de simulation de Prometheus par du code C++, récupérer les résultats (les différents périmètres du feu) et les manipuler dans le prototype en code C++. Heureusement, les concepteurs de Prometheus ont prévu une interface COM pour leur logiciel qui permet ce genre de facilité.

3. Le COM de Prometheus

Le *Component Object Model* (COM) est une architecture logicielle qui permet aux applications et aux systèmes d'être construits (ou d'utiliser) des composantes fournies par de différents fournisseurs de logiciels.

COM fournit des mécanismes pour :

- Communiquer entre les composantes, même à travers des processus ou des réseaux.
- Partager une mémoire de gestion entre les composantes.
- Reporter les erreurs et les conditions sur les états.
- Charger les composantes dynamiquement.

COM offre également plusieurs avantages par rapport aux précédentes méthodologies de conception telles que les DLL.

L'architecture COM a été donc appliquée sur certains modules dans CWFGM. Il y a en tout cinq interfaces COM dans la conception du CWFGM. Ces interfaces sont accessibles aux utilisateurs (autorisés) mais peuvent aussi être utilisées par les concepteurs pour transférer la technologie CWFGM à d'autres applications et modèles. Les interfaces COM sont :

- Fuel COM (pour manipuler les essences)
- FireEngine COM (pour simuler le feu)
- FWI COM
- Grid COM (pour manipuler les données sur le terrain telles que l'élévation et la pente)
- Weather COM

Afin de faciliter l'usage de ces cinq interfaces, un COM séparé haut niveau (*Prometheus COM*) a été développé afin d'offrir aux programmeurs une interface unique plus facile d'utilisation. C'est en fait l'interface que nous utilisons pour notre application.

La Figure B-1 donne la liste des fonctions de préparation des données d'un scénario de feu. Ces fonctions viennent de l'interface *Prometheus COM* et son donc utilisable par un code externe (C++ dans notre cas). La Figure B-2 présente la liste de fonctions COM qui permettent de lancer les simulations et récupérer les résultats en C++.



Figure B-1. Les fonctions (d'établissement du scénario) de Prometheus COM

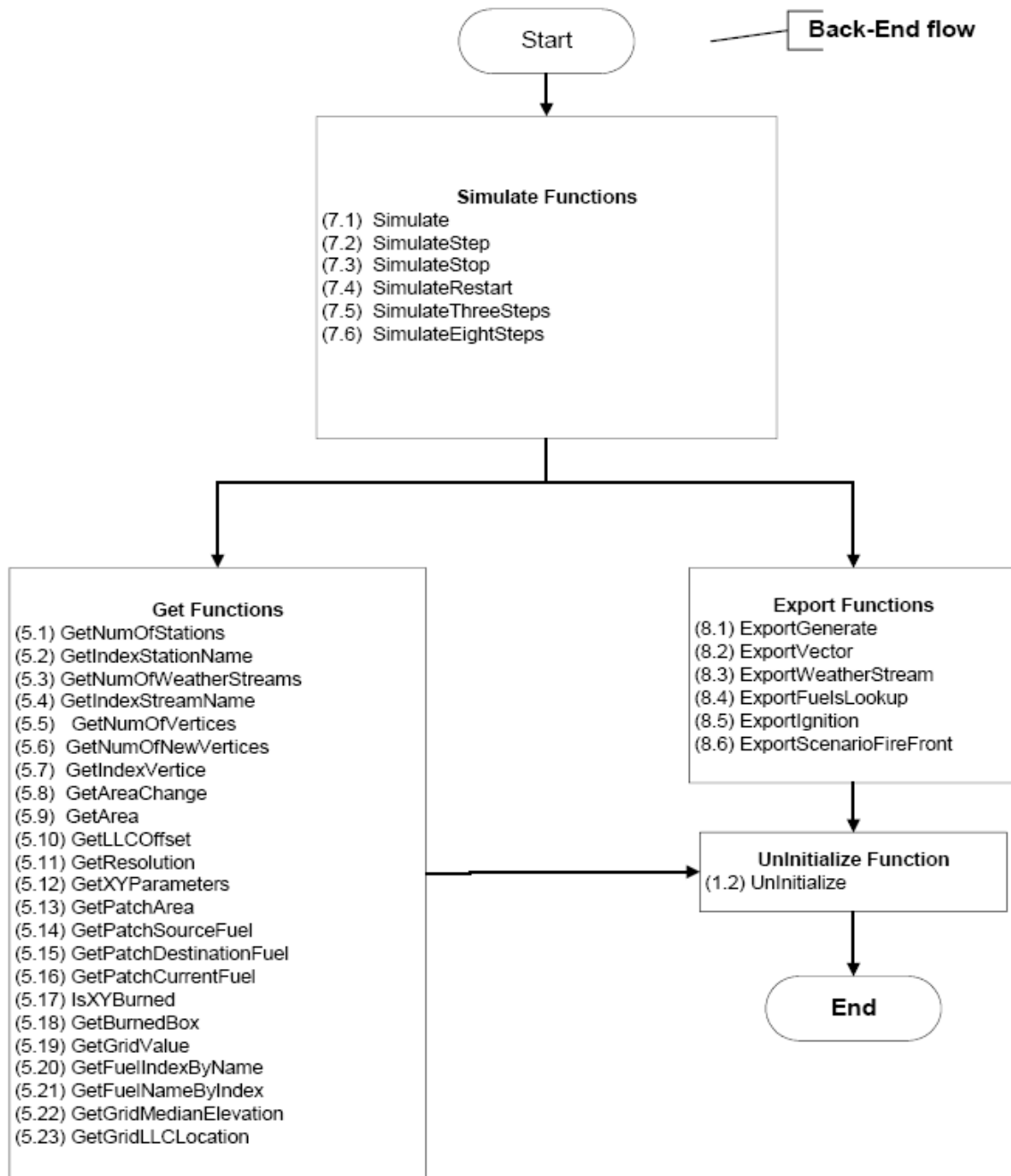


Figure B-2. Les fonctions (de simulation) de Prometheus COM

Le code suivant donne une idée sur l'utilisation de ces fonctions dans un code C++.

```

/* le suivant code est pour établir le scénario et les différentes données relatives au feu */
PROMETHEUSCOMLib::IPromeAppPtr pApp;
CoInitialize(NULL);
pApp.CreateInstance(CLSID_PromeApp);
HRESULT hr=pApp->Initialize(51.7, -115.4, 3, 23);
  
```

```

if(FAILED(hr))
return;
CString csScenarioStart("20/06/2000:10:00:00"),
csScenarioEnd("20/06/2000:20:00:00");
BSTR bScenarioStart, bScenarioEnd;
BSTR bPrjPath, bGridPath;
bScenarioStart=csScenarioStart.AllocSysString();
bScenarioEnd=csScenarioEnd.AllocSysString();
bPrjPath=m_csPrjPath.AllocSysString();
bGridPath=m_csGridPath.AllocSysString();
pApp->ImportProjection(bPrjPath);
pApp->ImportFuelGrid("Test",bGridPath);
pApp->CreateWeatherStation("First", 52, -115, 500);
pApp->CreateWeatherStream("First", "FirstStream", 85, 15, 25, 200, "20/06/2000:00:00:00");
pApp->SetInitialFFMC("First", "FirstStream", 85, 15);
pApp->SetStreamFFMCMethod("First", "FirstStream", 0);
pApp->AddDailyConditionToStream("First", "FirstStream", 15, 25, 15, 25, 20, 20, 90);
pApp->CreateScenario("FirstScenario");
pApp->SetScenarioGrid("FirstScenario", "Test");
pApp->SetScenarioTime("FirstScenario", bScenarioStart,bScenarioEnd);
pApp->CreateIgnitionPoint("FirstFire", "20/06/2000:9:00:00", 2485.36, 2900);
pApp->SetScenarioFire("FirstScenario", "FirstFire");
pApp->SetScenarioStream("FirstScenario", "First", "FirstStream");
CString first,second, third;
first=pathHead+"1.shp";

```

/* le code suivant est pour lancer la simulation du feu établi plus haut et envoyer les résultats (les différents périmètres de feu générés) dans des fichiers externes */

```

second=pathHead+"2.shp";
third=pathHead+"3.shp";
BSTR bFirst,bSecond,bThird;
bFirst=first.AllocSysString();
bSecond=second.AllocSysString();
bThird=third.AllocSysString();
long stepResult;
stepResult=pApp->SimulateRestart("FirstScenario");
pApp->SimulateStep("FirstScenario", &stepResult);
pApp->ExportScenarioFireFront("FirstScenario", bFirst);
pApp->SimulateStep("FirstScenario", &stepResult);
pApp->ExportScenarioFireFront("FirstScenario", bSecond);
pApp->SimulateStep("FirstScenario", &stepResult);
pApp->ExportScenarioFireFront("FirstScenario", bThird);
MessageBox("Done");
pApp->UnInitialize();
pApp.Release();
CoUninitialize();
::SysFreeString(bFirst);
::SysFreeString(bSecond);
::SysFreeString(bThird);
SysFreeString(bScenarioStart);
SysFreeString(bScenarioEnd);
SysFreeString(bPrjPath);
SysFreeString(bGridPath);

```

Annexe C : Application 2 : « Assistance sur le terrain »

1. Principe

Cette deuxième application (non implémentée) vise à faire profiter les acteurs, qui sont dans le monde réel, de la connaissance qui est disponible dans le monde de simulation. Ce principe découle directement de la raison d'être d'ENCASMA et peut être appliqué à n'importe quelle situation analogue (qui respecte les contraintes décrites dans Chapitre 4 Section 4.1.1).

Nous allons expliquer le principe sur un exemple particulier : Si un pompier demande à son *Agent de Simulation APS* de lui indiquer la ressource d'eau la plus proche de sa position courante. L'*Agent de Simulation*, informé de la position réelle de l'agent et ayant à sa disposition l'information géospatiale (ou autre) nécessaire, lui communique alors l'information requise. La Figure C.1 résume ce principe dans le cas général.

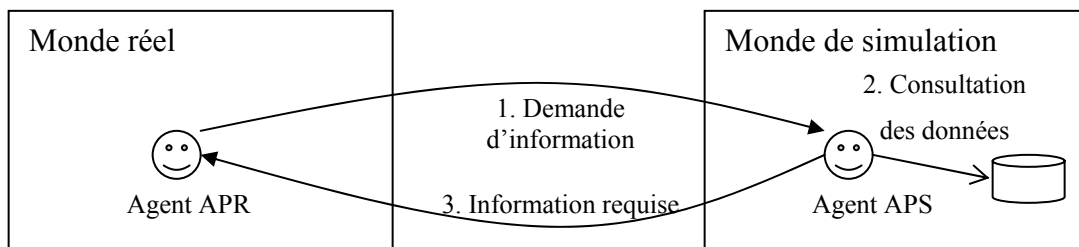


Figure C.1. Principe de demande d'information entre APR et APS.

Nous allons expliquer maintenant comment l'agent *APS* peut chercher la connaissance que demande l'*APR*.

Étant donné qu'il s'agit d'un système multiagent, l'agent *APS* peut adopter l'une des deux approches classiques de recherche d'information :

- À partir d'autres agents qui possèdent déjà la connaissance requise.

- Par lui-même si la première approche est impossible.

2. Approche impliquant l'APS

Nous présentons tout d'abord la deuxième approche. Nous supposons que l'*APR* a demandé au *APS* de lui trouver un espace rasé afin de pouvoir s'y réfugier. La solution que nous proposons est adaptée à la plate-forme MAGS. Dans MAGS, l'information est disponible sur les bitmaps. L'agent *APS* devrait donc naviguer dans ses alentours afin de consulter les informations encapsulées dans les bitmaps. Ceci pourrait par contre prendre du temps et surtout faire perdre la cohérence entre le monde réel et le monde de simulation. En effet, si le pompier n'a pas bougé sur le terrain, son agent *APS* ne devrait pas non plus se déplacer sur l'environnement de simulation (ceci pourrait donner une fausse image à l'utilisateur de ce qui se passe). L'*APS* par contre, pourrait créer quelques agents légers (*Agents Explorateurs AE*) qui sont munis des mécanismes de navigation et de perception de MAGS mais n'ont pas forcément les capacités que possède l'*APS* (surtout les capacités de planification, voir plus loin). Ces agents *AEs* explorent l'espace qui entoure l'*APS* selon des directions différentes (pour couvrir le plus d'espace possible) et selon un rayon limité (spécifié par l'*APS*). L'espace d'exploration (rayon d'exploration) va dépendre bien entendu de la requête de l'agent *APR*. Si ce dernier désire par exemple un espace rasé dans un rayon de 3km, les *AEs* explorent l'espace qui entoure l'*APS* dans un rayon de 3km. Si par contre le rayon n'est pas spécifié (p.ex. « l'espace rasé le plus proche »), l'*APS* peut choisir un rayon fixe qui soit crédible. La Figure C.2 illustre le principe d'exploration par les agents *AEs*. Il est clair que plusieurs contraintes peuvent s'ajouter sur le rayon et les directions d'exploration selon les besoins des pompiers, la propagation des feux, le sens du vent, etc.

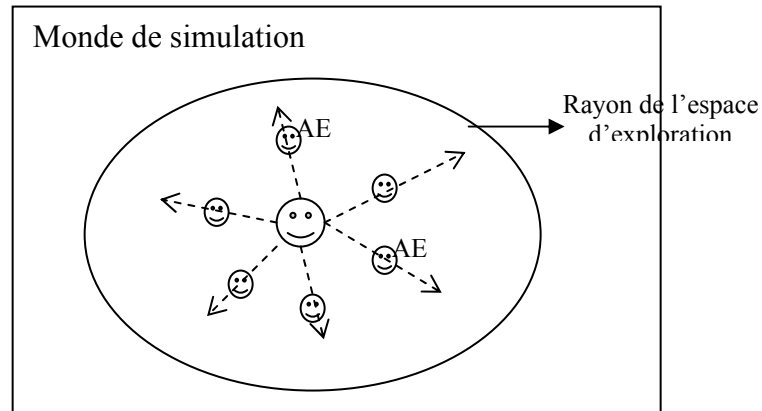


Figure C.2. Principe d'exploration par agents AEs

Pour gérer les résultats collectés par les différents *AEs*, l'agent *APS* peut adopter plusieurs stratégies selon les circonstances :

- *Premier trouvé premier adopté* : Cette stratégie consiste à prendre le premier résultat trouvé par l'un des *AEs*. Elle pourrait être adoptée si le temps est très critique. L'*APS* se contente de la première solution même si elle n'est pas la meilleure. L'*APS* ayant reçu une première solution, détruit tous les agents *AEs* et renvoie le résultat à l'*APR*.
- *Meilleure solution* : Cette stratégie consiste à chercher le plus de solutions possibles et à en choisir ensuite la meilleure. Cette approche donne de meilleurs résultats mais requiert beaucoup plus de temps. Elle ne pourrait donc être adoptée que si l'*APS* dispose de suffisamment de temps pour répondre à la requête du *APR*.
- *Stratégie hybride* : C'est une combinaison des deux premières. L'*APS* continue de recevoir des solutions jusqu'à un certain temps limité, qui une fois atteint, l'*APS* choisit la meilleure solution parmi celles déjà trouvées. Cette stratégie est efficace si l'*APS* connaît la marge temporelle dont il dispose pour répondre.

Remarques

- Il est donc important que l'*APS* tienne compte du temps dont il dispose pour répondre à la requête du *APR* afin de pouvoir choisir la stratégie qui convient.

- Il est à noter également qu'une fois qu'une solution est trouvée, les *AEs* doivent être détruits.
- L'*APS* peut gérer plusieurs requêtes en même temps en lançant des *AEs* qui ont plusieurs buts (p.ex. chercher une source d'eau ou un terrain complètement rasé).
- Les agents *AEs* communiquent avec l'*APS* (dans MAGS, via des boîtes à lettres que les agents consultent fréquemment).

3. Approche impliquant la collaboration des autres agents

Il est parfois inutile pour l'*APS* d'essayer de trouver l'information (demandée par son *APR*) par ses propres moyens (et ses agents *AEs*) si d'autres agents disposent déjà de cette information. Étant donné que dans le monde de simulation, tout peut être agent, nous disposons d'un vaste système multiagent où chaque agent détient une partie de la connaissance : Les agents *APSs* détiennent les informations sur les pompiers (localisation réelle, plan d'action, etc.), des agents représentant les ressources à protéger (habitation, hôpitaux, etc.) renferment des informations sur ces ressources (surtout leur emplacement), etc. Certains objets pertinents de l'environnement peuvent également être représentés par des agents. Par exemple, un lac peut être représenté par un *Agent Lac* qui renferme une certaine connaissance utile sur ce lac (p.ex. localisation, profondeur, etc.).

Il suffit donc de communiquer avec l'agent qui détient la connaissance recherchée. Mais comment un *APS* peut-il savoir quel agent détient l'information qu'il cherche ? En fait, il s'agit ici de collaboration classique entre agents dans un SMA. L'important dans tout ceci est que ENCASMA a fourni un SMA dans un environnement spatial complexe en mode simulation. Il est ensuite facile de tirer avantage de ce SMA pour implémenter diverses applications basées sur la collaboration multiagent [Grosz et Kraus 96; 99].

4. Ontologie des requêtes

Lorsqu'un *APS* recherche une source d'eau par exemple, dans son entourage, que ce soit par l'intermédiaire de ses agents *AEs* ou en collaborant avec d'autres agents, il a besoin

d'une ontologie qui lui indique ce que pourrait être une ressource d'eau (un lac, une rivière, un réservoir d'eau, etc.). Cette ontologie doit évidemment être la même pour tous les agents présents dans l'environnement de simulation afin que la collaboration soit possible. Chaque agent dispose donc d'une ontologie sur laquelle il se base pour formuler ses requêtes et comprendre les requêtes des autres agents.

Cette ontologie dépend fortement du domaine d'application. Les arbres sont par exemple considérés comme une essence pour le feu alors que dans un contexte de désertification, ils peuvent être considérés comme des barrières naturelles. Au sein d'une même application, un objet peut être vu de plusieurs façons. En effet, si nous considérons une forêt lors d'un incendie, un espace rasé peut être vu comme un lieu possible d'évacuation comme il peut très bien être vu comme une partie d'une ligne d'arrêt des incendies (dans une stratégie de lutte indirecte). Une ontologie précise et organisée est donc nécessaire.

Nous n'avons pas vraiment développé une ontologie pour cette application car tout simplement nous n'avons pas implémenté de prototype. Nous avons cependant mis plus d'effort sur la deuxième application qui présente plus de défis. L'application 'Assistance sur le terrain' n'a été évoquée dans ce chapitre que pour illustrer l'intérêt d'utiliser des agents logiciels sensibles à l'espace pour mieux appréhender un environnement géographique réel.