

Jacques Ferber

LES SYSTÈMES MULTI-AGENTS

Vers une intelligence collective



INFORMATIQUE
ia
INTELLIGENCE ARTIFICIELLE

InterEditions

Les Systèmes Multi Agents: vers une intelligence
collective

Jacques Ferber

1995

Contents

Préface	xiii
Avant-propos	1
1 Principes des systèmes multi-agents	5
1.1 Pour une intelligence collective	5
1.1.1 De la machine pensante...	5
1.1.2 ... à l'organisation artificielle	7
1.2 Agent et société	12
1.2.1 Quelques définitions	12
1.2.2 Niveaux d'organisation	17
1.2.3 Social ou biologique?	19
1.2.4 Architecture et comportement	24
1.2.5 Langages, communications et représentations	25
1.3 Un peu d'histoire	28
1.3.1 Les premiers âges	28
1.3.2 L'âge classique	29
1.3.3 L'influence de la vie artificielle	31
1.3.4 Les temps modernes	32
1.4 Domaines d'application	34
1.4.1 La résolution de problèmes	34
1.4.2 La simulation multi-agent	38
1.4.3 La construction de mondes synthétiques	43
1.4.4 La robotique distribuée	50
1.4.5 La conception kénétique de programmes	52
1.5 Les grands thèmes de la kénétique	54
1.5.1 La problématique de l'action	54
1.5.2 L'individu et sa relation au monde	54
1.5.3 L'interaction	55
1.5.4 L'adaptation	56
1.5.5 La réalisation et l'implémentation des SMA	56
1.6 Domaines voisins des systèmes multi-agents	56
1.6.1 L'intelligence artificielle	57
1.6.2 La systémique	57
1.6.3 Les systèmes distribués	58

1.6.4	La robotique	60
1.6.5	Ce qui ne relève pas de la kénétique	60
2	Interactions et coopération	63
2.1	Situations d'interaction	63
2.2	Composantes des interactions	65
2.2.1	Buts compatibles et incompatibles	65
2.2.2	Relations aux ressources	67
2.2.3	Capacités des agents par rapport aux tâches	68
2.3	Types d'interaction	69
2.3.1	Indépendance	70
2.3.2	Collaboration simple	70
2.3.3	Encombrement	70
2.3.4	Collaboration coordonnée	70
2.3.5	Compétition individuelle pure	71
2.3.6	Compétition collective pure	71
2.3.7	Conflit individuel pour des ressources	71
2.3.8	Conflits collectifs pour des ressources	71
2.3.9	Niveau d'analyse des situations d'interaction	72
2.4	Formes de coopération	74
2.4.1	La coopération comme attitude intentionnelle	74
2.4.2	La coopération du point de vue de l'observateur	75
2.4.3	L'amélioration de la survie	78
2.4.4	L'accroissement des performances	79
2.4.5	La résolution de conflit	81
2.5	Les méthodes de coopération	82
2.5.1	Le regroupement et la multiplication	82
2.5.2	La communication	83
2.5.3	La spécialisation	84
2.5.4	La collaboration par partage de tâches et de ressources	84
2.5.5	La coordination d'actions	85
2.5.6	La résolution de conflit par arbitrage et négociation	85
2.6	Organisations et coopération	86
2.6.1	Le système des activités de coopération	86
2.6.2	Avantages	88
2.6.3	Contraintes sociales et émergences de structures	89
3	Organisations multi-agents	91
3.1	Qu'est-ce qu'une organisation?	91
3.1.1	Structures organisationnelles et organisations concrètes	92
3.1.2	Les niveaux d'organisation	93
3.1.3	Comment étudier une organisation?	94
3.2	L'analyse fonctionnelle	95
3.2.1	Les fonctions d'une organisation	95
3.2.2	Les dimensions d'analyse	99

3.2.3	Analyse dimensionnelle d'une organisation	100
3.2.4	Grille d'analyse fonctionnelle des organisations	106
3.3	L'analyse structurale	110
3.3.1	Agents et tâches	110
3.3.2	Relations abstraites	115
3.3.3	Les modes de couplage	116
3.3.4	Structures de subordination et prises de décision	117
3.3.5	Types de constitution des structures organisationnelles	118
3.4	Les paramètres de concrétisation	119
3.5	Analyse d'une organisation concrète	122
3.5.1	L'exemple des robots explorateurs	123
3.5.2	Organisations à structure fixe-hiérarchique-prédéfinie	125
3.5.3	Organisations à structure variable-égalitaire-émergente	126
3.5.4	Organisations à structure variable-égalitaire-prédéfinie	127
3.5.5	Organisations à structure évolutive	127
3.5.6	Autres travaux sur les organisations	128
3.6	Organisations individuelles	129
3.6.1	Tableau des principales architectures	130
3.6.2	Architecture modulaire horizontale	131
3.6.3	Les architectures à base de tableaux noirs	132
3.6.4	L'architecture de subsomption	135
3.6.5	Les tâches compétitives	136
3.6.6	Les systèmes de production	137
3.6.7	Les systèmes à base de classifieurs	139
3.6.8	Les architectures connexionnistes	140
3.6.9	Les architectures à base de système dynamique	142
3.6.10	Architectures multi-agents et acteurs	144
4	Action et comportement	147
4.1	La modélisation	147
4.1.1	Des modèles...	147
4.1.2	... et de leur utilité pour les SMA	148
4.1.3	Que modéliser?	149
4.1.4	Agents et actions: des notions faussement élémentaires	149
4.1.5	Modéliser l'action	151
4.2	L'action comme transformation d'un état global	152
4.2.1	Une représentation fonctionnelle de l'action	152
4.2.2	Les opérateurs de type STRIPS	153
4.2.3	Planifier avec des opérateurs de type STRIPS	155
4.2.4	Quelques catégories de plans	158
4.2.5	Limites des planificateurs de type STRIPS	160
4.2.6	Limites des représentations classiques de l'action	161
4.3	L'action comme réponse à des influences	164
4.3.1	Présentation générale	164
4.3.2	Les états	165

4.3.3	Actions et réactions	166
4.3.4	Intérêt du modèle influences/réactions pour les SMA	168
4.4	L'action comme processus informatique	169
4.4.1	Représentation de processus par automates à états finis	171
4.4.2	Automates à registres	174
4.4.3	Représentation de processus par réseaux de Petri	176
4.4.4	Autres modèles événementiels	182
4.5	L'action comme déplacement physique	182
4.5.1	Déplacements dans un champ de potentiel	184
4.5.2	Intérêt de cette conception de l'action	186
4.6	L'action comme modification locale	187
4.6.1	Les automates cellulaires	188
4.6.2	Représentation d'un automate cellulaire	189
4.6.3	Automates cellulaires et systèmes multi-agents	190
4.7	L'action comme commande	191
4.8	Agents tropiques et hystérétiques	192
4.9	Agents tropiques	194
4.9.1	Approche formelle	194
4.9.2	Un système multi-agent tropique	194
4.9.3	Agents tropiques et actions situées	195
4.9.4	Souplesse des actions situées	200
4.9.5	Les buts sont dans l'environnement	202
4.10	Agents hystérétiques	206
4.10.1	Approche formelle	206
4.10.2	Un système multi-agent hystérétique	207
4.10.3	Modélisation d'agents hystérétiques par automates	207
4.11	Modélisation de SMA en BRIC	209
4.11.1	Décrire des SMA à l'aide de composants	209
4.11.2	Modélisation de SMA purement communicants	211
4.11.3	Modélisation des environnements	212
4.11.4	Modélisation d'un SMA situé	218
4.11.5	Modélisation d'un SMA complet	220
4.11.6	Un exemple: les agents transporteurs	221
5	L'agent dans tous ses états	225
5.1	Etats mentaux et intentionnalité	225
5.1.1	Introduction	225
5.1.2	La notion de cogniton	227
5.1.3	Une nomenclature des cognitons	228
5.2	Le système interactionnel	231
5.3	Le système représentationnel	233
5.3.1	Qu'est-ce que la connaissance?	233
5.3.2	Représenter les connaissances et les croyances	240
5.3.3	Logiques des savoirs et des croyances	246
5.3.4	Adéquation et révision des croyances	255

5.4	Que croire? Les contenus des représentations	263
5.4.1	Croyances environnementales (χ)	264
5.4.2	Croyances sociales (σ)	265
5.4.3	Croyances relationnelles (α)	265
5.4.4	Croyances personnelles (ω)	268
5.5	Le système conatif	268
5.5.1	Rationalité et survie	269
5.5.2	Un modèle du système conatif	271
5.6	Les motivations: les sources de l'action	273
5.6.1	Les motivations personnelles: plaisir et contraintes	274
5.6.2	Les motivations environnementales: le désir de l'objet	275
5.6.3	Les motivations sociales: le poids de la société	276
5.6.4	Les motivations relationnelles: la raison c'est les autres	278
5.6.5	Les engagements: des motivations et contraintes relationnelles et sociales	279
5.7	Le passage à l'acte réactif	284
5.7.1	Actes consommatoires et comportements appétitifs	285
5.7.2	Modes de sélection et de contrôle des actions	286
5.7.3	Sélection d'action ou combinaison dynamique	294
5.8	Passages à l'acte intentionnels	298
5.8.1	Théories logiques de l'intention	299
5.8.2	La théorie de l'action rationnelle de Cohen et Levesque	301
6	Communications	309
6.1	Aspects de la communication	309
6.1.1	Signes, indices et signaux	309
6.1.2	Définition et modèles de communication	310
6.1.3	Catégories de communication	312
6.1.4	A quoi sert la communication?	317
6.2	Les actes de langage	320
6.2.1	Dire, c'est faire	320
6.2.2	Actes locutoires, illocutoires et perlocutoires	322
6.2.3	Succès et satisfaction	323
6.2.4	Composantes des actes illocutoires	325
6.3	Les conversations	326
6.3.1	Conversations et automates à états finis	328
6.3.2	Conversations et réseaux de Petri	329
6.3.3	Une classification des actes de langage pour des structures conversationnelles multi-agents	330
6.4	KQML	338
7	Collaboration et répartition des tâches	343
7.1	Modes d'allocation des tâches	343
7.1.1	Critères de décomposition des tâches	344
7.1.2	Rôles	344

7.1.3	Formes d'allocation	345
7.2	Allocation centralisée des tâches par médiateur	346
7.3	Allocation distribuée des tâches	350
7.3.1	Allocation par réseau d'accointances	350
7.3.2	Allocation par appel d'offre: le réseau contractuel	361
7.3.3	Variantes et allocations hybrides	376
7.3.4	Contrats et engagements	378
7.4	Intégrer les tâches et les états mentaux	379
7.4.1	Le système SAM	379
7.4.2	La hiérarchie des architectures	380
7.4.3	Les résultats	381
7.4.4	L'implémentation des architectures	383
7.4.5	Le niveau 1	383
7.4.6	Le niveau 2	384
7.4.7	Le niveau 3	385
7.5	Allocation émergente	388
7.6	Un exemple: le système Manta	391
7.6.1	Description générale	391
7.6.2	L'architecture du système	391
7.6.3	Les expérimentations	394
7.6.4	Des fourmis aux robots fourmis	397
8	Coordination d'actions	399
8.1	Qu'est-ce que la coordination d'actions?	399
8.1.1	Définitions	400
8.1.2	La coordination comme résolution de problème	402
8.1.3	Caractéristiques des systèmes de coordination	405
8.1.4	Formes de coordination d'action	409
8.2	Synchronisation d'actions	411
8.2.1	Synchronisation de mouvements	411
8.2.2	Synchronisation d'accès à une ressource	411
8.3	Coordination d'actions par planification	413
8.3.1	Planification multi-agent	415
8.3.2	Planification centralisée pour agents multiples	419
8.3.3	Coordination centralisée pour plans partiels	421
8.3.4	Coordination distribuée pour plans partiels	427
8.4	Coordination réactive	432
8.4.1	Coordination par actions situées	432
8.4.2	Du comportement de meute aux systèmes anti-collisions	433
8.4.3	Marquage de l'environnement	438
8.4.4	Actions de coordination	444
8.5	Résolution par coordination : l'éco-résolution	446
8.5.1	Principes de l'éco-résolution	447
8.5.2	Les éco-agents	448
8.5.3	Exemples simples d'éco-problèmes	450

8.5.4	Univers évolutifs	459
8.5.5	Formalisation	461
8.5.6	Résolution de contraintes par éco-résolution	462
9	Conclusion	465
A	BRIC	469
A.1	Les composants	469
A.2	Composants composites	470
A.3	Constitution des composants élémentaires	470
A.4	Liens de communication	472
A.5	Conventions de notations et équivalences	472
A.6	Traduction sous forme de réseaux de Petri	473
A.7	Exemple	473
	Références bibliographiques	475
	Index	493

Préface

Ce livre répond à un besoin: celui de donner un contenu informatique précis à l'expression "système multi-agents". Quelles notions, quelles tendances, quelles possibilités nouvelles se cachent sous ce terme à la mode? C'est avant tout une question de modélisation. Les langages actuels obligent les programmeurs d'applications à décrire les phénomènes qu'ils traitent d'une manière strictement mécaniste et hiérarchisée. On arrive tout au plus à bâtir une application à partir de modules construits indépendamment – comme un immeuble moderne fait d'éléments préfabriqués au lieu de pierres taillées sur place. Le processus est parfait pour un programme traitant du fonctionnement d'un immeuble. Mais quid de la vie de ses habitants? La belle ordonnance qui explique, par exemple, la structure complexe d'un étage comme une composition de pièces et de parois réputées plus simples n'a manifestement plus cours. Il faut prendre en compte l'indépendance opérationnelle des individus (et non plus seulement leur indépendance conceptuelle), les phénomènes de communication, de libre arbitre, de croyance, de concurrence, de consensus — et aussi de discorde. Cela n'est pas chose facile. Rien cependant n'échappe à l'appétit des ordinateurs! La représentation en machine de bâtiments, d'immeubles, voire de cités entières étant à peu près devenue chose courante (demandez aux architectes), il s'agit maintenant de s'attaquer aux être humains et aux sociétés qui habitent ces lieux. Comment va-t-on procéder? Les systèmes multi-agents bien sûr...

Plusieurs forces sont sans doute à l'œuvre derrière cette évolution. Je crois que la montée du libéralisme et de l'individualisme à l'américaine nous poussent plus fortement que jamais à vouloir modéliser toute situation un tant soit peu complexe en termes économiques de concurrence et de marché. Ce faisant, nous avons tendance à privilégier les modèles informatiques mettant en jeu les mêmes concepts. On peut lire le même mouvement d'idées d'un œil plus optimiste, parler d'une salutaire tendance à la décentralisation, d'un utile correctif au jacobinisme ou invoquer les richesses de la création collective. Dans un sens comme dans l'autre, notre époque se veut plurielle et il n'est pas surprenant que cette idéologie cherche à se refléter jusque dans les modèles informatiques de la réalité. Les machines évoluent dans le même sens que les idées: la montée des architectures parallèles et des réseaux a détrôné le processeur unique, image fidèle du cerveau unique et de l'administration centralisée. Là aussi, place au pluralisme et à la concurrence!

Comme toujours, il y a loin de la coupe aux lèvres. La tâche des spécialistes de systèmes multi-agents et d'intelligence artificielle distribuée (deux manières de désigner la même communauté) est d'une part de trouver les "bonnes" manières de maîtriser la fabuleuse puissance de calcul de nos machines (en définissant ce

qu'en jargon on appelle des architectures – logicielles et matérielles) et d'autre part d'expliquer aux utilisateurs ce qu'ils peuvent en attendre. La difficulté est alors de dire les choses de manière suffisamment claire et évocatrice sans masquer pour autant les limitations et les insuffisances des systèmes proposés, qui sont souvent énormes: après tout, ces constructions ne reposent que sur des machines! De ce point de vue, l'informatique est tributaire des sciences cognitives, puisqu'il s'agit de mettre en correspondance les processus intellectuels de l'utilisateur avec ceux (sic) de la machine. En retour, cette confrontation apporte à l'étude de la cognition des éclairages nouveaux. C'est ce que fait Jacques Ferber dans son livre, armé de pensée systémique et de psychologie cognitive. Ce n'est pas un ouvrage sur les techniques de programmation, bien que l'auteur soit parfaitement conscient des possibilités et des limitations des moyens informatiques actuels. C'est en fonction de ces moyens, mais de manière implicite, qu'il analyse les nombreux facteurs qui entrent dans les systèmes multi-agents que l'on sait actuellement construire. Certains exemples sur lesquels il fonde son argumentation ont d'ailleurs été réalisés dans son équipe et ont fait l'objet de thèses de doctorat.

Comme on le verra, la modélisation complète d'une société humaine par un système multi-agent est encore assez loin de nous – pour des raisons qui ne tiennent pas toutes à la stupidité des machines. La simulation de *La vie devant soi* n'est pas pour demain. En revanche, nous savons modéliser certaines activités des collectivités humaines, par exemple le régime des pêches dans le delta central du Niger, où entrent en concurrence divers facteurs climatiques, culturels, économiques et halieutiques. Nous savons aussi représenter les caractéristiques de certaines populations de fourmis, chaque insecte étant traité individuellement comme un agent très simple et l'interaction à grande échelle de ces agents multiples faisant émerger des comportements globaux fort proches de ceux qu'observent les entomologistes dans leurs laboratoires d'éthologie – ce qui ouvre des voies de recherche nouvelles. Et, pour revenir à des préoccupations d'ingénierie, de systèmes qui servent à résoudre des problèmes, Jacques Ferber vous proposera tout un éventail de techniques de résolution de problèmes qui s'avèrent prometteuses.

Ce livre est un commencement, c'est vous, lecteur, qui écrirez la suite!

Jean-François Perrot
*Professeur à l'université
Pierre et Marie Curie*

Avant-propos

Depuis quelques années, les systèmes multi-agents ont pris une place de plus en plus importante en informatique, que ce soit dans le domaine de l'intelligence artificielle, dans ceux des systèmes distribués, de la robotique, ou même dans ce champ disciplinaire nouveau qu'est la "vie artificielle", en introduisant la problématique de l'intelligence collective et de l'émergence de structures par interactions. En situant la question au niveau de l'autonomie des individus, que l'on appelle agents, et des interactions qui les lient, les systèmes multi-agents soulèvent de nombreuses questions. Quels sont en effet les concepts fondateurs de ce domaine? Comment se différencie-t-il des autres disciplines et en particulier de l'intelligence artificielle, des systèmes distribués et de la robotique? Quels sont ses apports aux sciences cognitives et à la réflexion philosophique en général?

Science intégratrice plus qu'analytique, la recherche sur les systèmes multi-agents suscite un certain nombre d'interrogations: qu'est-ce qu'un agent qui interagit avec d'autres agents? Comment peuvent-ils coopérer? Quels sont les modes de communication nécessaires pour qu'ils se répartissent des tâches et coordonnent leurs actions? Quelle architecture peut-on leur donner pour qu'ils satisfassent leurs buts? Ces questions sont posées avec une vigueur particulière car l'enjeu porte sur la réalisation de systèmes disposant de caractéristiques particulièrement intéressantes: souplesse, adaptation aux changements, intégration de programmes hétérogènes, rapidité d'obtention de résultats, etc.

Les recherches dans le domaine des systèmes multi-agents poursuivent deux objectifs majeurs: le premier concerne l'analyse théorique et expérimentale des mécanismes d'auto-organisation qui ont lieu lorsque plusieurs entités autonomes interagissent; le second s'intéresse à la réalisation d'artefacts distribués capables d'accomplir des tâches complexes par coopération et interaction. Leur position est donc double: d'un côté elles se placent au sein des sciences cognitives et sociales (psychologie, éthologie, sociologie, philosophie...) et naturelles (écologie, biologie...) pour à la fois modéliser, expliquer et simuler des phénomènes naturels, et susciter des modèles d'auto-organisation; de l'autre, elles se présentent comme une pratique, une technique tendue vers la réalisation de systèmes informatiques complexes à partir des concepts d'agents, de communication, de coopération et de coordination d'actions.

Ce livre est issu de l'enseignement des systèmes multi-agents (ou SMA) que j'effectue dans le DEA IARFA à l'université Pierre et Marie Curie (Paris 6) depuis plusieurs années. Il n'existe, à ma connaissance, aucun manuel (textbook) sur les SMA. Les seuls ouvrages disponibles sont des thèses publiées ou des compilations d'articles écrits par des spécialistes pour d'autres spécialistes du domaine, ce qui

ne permet pas à un étudiant ou à un non-spécialiste d'avoir un aperçu global et synthétique de la question. L'éventail des recherches menées depuis bientôt vingt ans sur les systèmes multi-agents est extrêmement vaste, et il n'existe pas (encore) de fondements suffisamment simples et précis de cette discipline pour qu'il soit facile de présenter de manière progressive et didactique les travaux dans ce domaine. Il fallait donc commencer par unifier les positions et établir un cadre conceptuel pour le développement de théories futures. C'est à cet objectif que cet ouvrage est consacré, celui de synthétiser les connaissances majeures dans ce domaine en commençant à poser les fondations d'une science de l'interaction, que j'ai appelée la *kénétiq*ue, d'après le terme grec *koïnon*, ce qui est commun.

Pour l'instant cette tentative est encore timide et s'appuie essentiellement sur la définition de cadres conceptuels, de quelques classifications et formalisations et sur la présentation d'une méthode modulaire de construction de systèmes multi-agents. L'intérêt d'une telle synthèse est double: elle permet de s'affranchir, à terme, des différences de notations et des méandres légitimes effectués par les chercheurs dans l'exploration de leur domaine, et de montrer un visage uni autour d'une problématique claire pouvant s'énoncer précisément et simplement. Nous n'en sommes pas encore là, mais j'espère que cet ouvrage contribuera à la fois à faire comprendre au lecteur non spécialiste les résultats et les enjeux des systèmes multi-agents sous la forme d'un langage unifié et à donner à l'étudiant ou au chercheur des bases solides pour l'aider à élaborer de nouvelles connaissances et à apporter sa propre pierre à ces fondations.

Contenu de l'ouvrage

On trouvera dans cet ouvrage:

1. un état de l'art sur la question,
2. un point de vue sur les systèmes multi-agents présentés comme une nouvelle systémique qui met en avant la problématique de l'interaction et de ses conséquences,
3. une analyse conceptuelle du domaine à partir d'une classification des principaux problèmes et résultats et, surtout, une analyse fonctionnelle et structurale des organisations servant de cadre à l'ensemble des travaux menés dans ce domaine,
4. un formalisme, BRIC, permettant la conception modulaire et incrémentale de SMA, par la modélisation des comportements, des structures de communication et des principales formes d'interactions que sont l'allocation de tâches et la coordination d'actions,
5. une tentative de formalisation des SMA et de l'interaction, à partir d'un modèle de l'action fondé sur un couple influence/réaction.

A qui s'adresse-t-il?

Cet ouvrage s'adresse tout d'abord à des professionnels de l'informatique non spécialistes de la question qui désirent avoir un point de vue synthétique sur ce domaine. Il s'adresse aussi à des non-informaticiens spécialistes des sciences sociales ou des sciences de la nature qui veulent utiliser les systèmes multi-agents pour modéliser des comportements naturels et étudier l'émergence de phénomènes complexes. Il s'adresse aussi aux non spécialistes de l'informatique souhaitant avoir une connaissance des concepts essentiels permettant de comprendre ce qu'est une "intelligence collective" et avoir une vue d'ensemble de la problématique multi-agent. Il s'adresse enfin aux étudiants en deuxième et troisième cycles d'informatique qui voudraient se spécialiser dans ce domaine et pour lesquels il fournit un support de cours.

Organisation de l'ouvrage

L'organisation de cet ouvrage suit une démarche volontairement constructive et ascendante: après une brève présentation du domaine et la présentation d'un cadre général d'analyse des organisations multi-agents, les concepts et mécanismes mis en œuvre dans les systèmes multi-agents sont progressivement étudiés et analysés. Les notions de base sont présentées dans une première partie. Le chapitre 1 donne un aperçu général du domaine, les thèmes principaux et leurs relations avec d'autres disciplines. Le chapitre 2 introduit le concept de situation d'interaction et propose un cadre général pour appréhender les différentes composantes de la coopération. Le chapitre 3 propose une analyse fonctionnelle et structurale des organisations ainsi que les différentes architectures utilisées généralement pour les concevoir. Le chapitre 4 propose une transition entre les généralités de la partie précédente et les descriptions plus détaillées des parties suivantes. Il aborde la problématique de la formalisation de l'action et du comportement ainsi que celle de la modélisation des systèmes multi-agents et introduit la plupart des notations utilisées dans le reste de l'ouvrage. On y montre que les conceptions classiques de l'action ne sont pas suffisantes pour bien comprendre les interactions, ce qui aboutit au développement d'une théorie de l'action considérée comme le résultat d'un ensemble d'influences produites par les agents. Ce chapitre présente aussi un formalisme de modélisation des agents et des systèmes multi-agents, BRIC, qui associe des réseaux de Petri à une structure modulaire. Le chapitre 5 fait le point sur la notion d'état mental d'un agent (croyances, intentions, engagements, etc.) et propose une représentation de sa dynamique mentale sous la forme d'une architecture modulaire décrite en termes de composants BRIC. La dernière partie est consacrée aux différentes méthodes et outils conceptuels utilisés pour construire des organisations coopératives. Le chapitre 6 porte sur les communications et fournit une modélisation des principales structures de communications à l'aide de réseaux de Petri. Le chapitre 7 est dédié à l'étude des organisations de collaboration dans lesquelles le travail est réparti dynamiquement entre les agents. Le chapitre 8 présente enfin les principaux modèles de coordination

d'action qui gèrent l'articulation dynamique de la planification et de l'exécution des tâches et tentent d'éviter les conflits. On y montre aussi qu'il est possible de résoudre des problèmes par interaction entre agents simples.

Remerciements

Ce livre n'aurait pu voir le jour sans l'aide d'un grand nombre de personnes qui m'ont apporté leurs remarques, leurs critiques et surtout leur soutien. Je pense en particulier à Jacqueline Zizi qui, au delà de son amitié, a relu mon livre de fond en comble. Son exigence et sa rigueur m'ont beaucoup aidé, sa diplomatie et ses encouragements m'ont permis de garder confiance dans ce projet. Qu'elle reçoive ici le témoignage de ma reconnaissance. Je pense également à Philippe Laublet, Alexis Drogoul et Anne Collinot qui ont su apporter les commentaires et critiques constructives à ce qui était encore un brouillon. Leurs remarques m'ont été précieuses.

Je remercie les membres de l'équipe MIRIAD, Stephane Bura, Thierry Bouron, Patrice Carle, Christophe Cambier, Eric Jacopin, Karim Zeghal et tous les autres. Ils ont su apporter leurs compétences et leur dynamisme à la constitution de ce groupe de recherche, parfait exemple de système multi-agent. Ils ont été un stimulant permanent durant l'élaboration de cet ouvrage.

Je remercie également les étudiants du DEA IARFA, pour avoir supporté mes explications — parfois confuses j'en suis certain — sur l'intérêt des SMA. Ils ont ainsi favorisé la mise au point de certaines théories et synthèses qui sont présentées dans cet ouvrage.

Yves Demazeau, Jean Erceau, Les Gasser, Charles Lenay, Jean-Pierre Müller, Jean-François Perrot m'ont apporté leur amitié, leur aide et leurs encouragements. Je tiens à leur exprimer toute ma gratitude.

Je tiens aussi à faire savoir à Pierre Azema, Jean-Paul Barthès, Paul Bourguine, Jean-Pierre Briot, Christian Brassac, Christiano Castelfranchi, Brahim Chaib-Braa, Bruno Corbara, Pascal Estrailier, Dominique Fresneau, France Guérin, Alain Pavé, Joël Quinqueton, Mario Tokoro, Dominique Lestel, Christian Mullon, Gérard Sabah, Lena Sanders, Luc Steels, Jean-Pierre Treuil — et à bien d'autres que la place m'empêche de citer tous — combien leur gentillesse et les discussions scientifiques particulièrement fécondes que nous avons pu avoir ensemble m'ont été d'une grande aide.

Je tiens enfin à remercier tous ceux qui m'ont accompagné, famille et amis, pour leur compréhension, leur soutien inconditionnel et leur affection alors que ces dernières années, la phrase "Je suis en train de finir mon livre" était presque devenue un leitmotiv.

Chapter 1

Principes des systèmes multi-agents

1.1 Pour une intelligence collective

1.1.1 De la machine pensante...

Longtemps, l'informatique en général et l'intelligence artificielle (IA) en particulier ont considéré les programmes comme des entités individualisées capables de rivaliser avec l'être humain dans des domaines précis. Au départ simple machine à calculer, l'ordinateur s'est ainsi vu consacrer à des tâches qui relevaient de domaines de plus en plus complexes tels que la gestion d'une entreprise, la surveillance et le contrôle de processus industriels, l'aide au diagnostic médical ou la conception de nouvelles machines. Cette compétition entre l'être humain et la machine s'est accompagnée d'une identification de la machine à l'humain, un programme représentant directement un "expert" capable de résoudre un problème par lui-même.

Il est possible de constater cette individualisation des programmes d'IA, c'est-à-dire cette mise en rapport des capacités cognitives de l'être humain et des ordinateurs, dès le début des développements de l'intelligence artificielle. En effet, depuis les premiers balbutiements des programmes expérimentaux de Newell, Shaw et Simon (Newell et al. 1957) sur la démonstration automatique de théorèmes, le terme "intelligence artificielle" a été utilisé pour désigner un projet de recherche consistant à concevoir une machine intelligente, c'est-à-dire capable de réussir aussi bien qu'un être humain dans des tâches jugées complexes. Ce projet souleva un grand nombre de questions portant non seulement sur la nature d'une possible "intelligence artificielle" mais aussi sur un problème crucial pour l'humanité tout entière: est-ce qu'une machine pense? Est-ce qu'elle peut être intelligente? Doit-on lui donner un statut juridique semblable à celui d'un être humain? Toutes ces questions ont alimenté un débat qui trouve aujourd'hui son expression dans un certain nombre de critiques contre les possibilités de l'IA.

Pour caractériser cette notion de machine intelligente, Turing inventa un test consistant à dire qu'une machine est intelligente si on ne peut la distinguer d'un être humain lors d'une conversation. L'expérience consiste à mettre un expérimentateur

devant un terminal informatique et à le relier soit à un autre opérateur humain soit à un ordinateur. L'opérateur et l'ordinateur prétendent tous les deux qu'ils sont des humains. Au travers d'un dialogue, l'expérimentateur doit décider si son interlocuteur est un humain ou une machine. S'il ne peut faire cette distinction ou s'il n'indique la bonne réponse que dans la moitié des cas, on peut dire que la machine a passé le test avec succès et on la qualifiera d'intelligente. Il faut noter que, pour une machine, être considérée comme intelligente revient à imiter le comportement d'un humain et ne repose pas sur des critères intrinsèques. L'intelligence serait donc le fait des individus isolés et non des groupes, puisqu'en aucun cas on ne demande à une machine (ou à un groupe de machines) de se mettre à la place de plusieurs personnes.

Cette façon de concevoir les programmes informatiques comme des sortes de "penseurs" repliés sur eux-mêmes se retrouve directement dans la notion de *système expert*, c'est-à-dire des programmes informatiques capables de remplacer l'être humain dans ses tâches réputées les plus complexes et qui réclament de l'expérience, du savoir-faire et une certaine forme de raisonnement.

Cette pensée est à la fois la racine et la conséquence du développement de systèmes centralisés et séquentiels dans lesquels le contrôle¹, c'est-à-dire ce qui décide ce qu'il faut faire, est défini comme le pilote des différentes activités exécutées l'une derrière l'autre. Cependant cette conception centralisatrice et séquentielle se heurte à plusieurs obstacles d'ordres aussi bien théorique que pratique. Sur le plan théorique, l'intelligence artificielle s'était engagée sur une voie à la fois trop optimiste et en même temps trop réductrice. L'intelligence, comme la science (Latour 1989) (Lestel 1986), n'est pas une caractéristique individuelle que l'on pourrait séparer du contexte social dans lequel elle s'exprime. Un être humain ne peut se développer convenablement s'il ne se trouve pas entouré d'autres êtres de son espèce. Sans un entourage adéquat, son développement cognitif est très limité, et le simple apprentissage d'une langue articulée lui devient proprement impossible s'il n'a pas été plongé dans une culture humaine dès sa première enfance. En d'autres termes, les autres sont indispensables à notre développement cognitif et ce que nous appelons "intelligence" est autant dû aux bases génétiques qui définissent notre structure neuronale générale qu'aux interactions que nous pouvons avoir avec le monde qui nous entoure et, en particulier, avec la société humaine.

On peut donc dire que, dès le départ, le projet de l'intelligence artificielle pouvait être considéré comme inaccessible, et ce, non pas à cause des polémiques supposant une nature transcendantale de l'intelligence humaine et de la conscience impossible à reproduire dans un système informatique, mais à cause du manque d'interaction des programmes avec l'extérieur et dans l'absence d'insertion dans une communauté d'autres entités intelligentes de même nature. Les critiques de Dreyfus (Dreyfus 1979) et de Searle (Searle 1991), notamment, portent essentiellement sur le manque d'*incorporation* (embodiment) des programmes et donc de leur incapacité à interagir

¹Nous utiliserons dans l'ensemble de l'ouvrage le terme "contrôle" dans son acception anglo-saxonne de *maîtrise*, et non dans celle de *vérification* qui correspond à l'usage français. Ceci dans un souci de cohérence avec cette dérive terminologique qui est maintenant devenue classique en informatique.

de manière signifiante avec leur environnement et avec leurs semblables.

Cette difficulté théorique s'est vue confirmée par des problèmes pratiques dus à la complexité des questions à résoudre. Les systèmes informatiques deviennent de plus en plus complexes, et il est nécessaire de les décomposer en modules "faiblement couplés", en unités indépendantes dont les interactions sont limitées et parfaitement contrôlées. Ainsi, au lieu de se trouver en présence d'une "machine" — une entité bien localisée par sa structure et son architecture — on se trouve, comme le développement des langages objets l'ont montré, devant un ensemble d'entités en interactions, chaque entité étant définie de manière locale sans vision globale détaillée de toutes les actions du système. Cette façon d'envisager des programmes introduisit de nouvelles méthodes de conception en génie logiciel et un changement de perspective; on passa de la notion de programme à celle d'organisation, notion que nous aurons l'occasion de développer tout au long de cet ouvrage.

Pour des raisons différentes, les concepteurs de systèmes experts industriels sont arrivés à des conclusions semblables à partir des difficultés soulevées par le développement de bases de connaissance. La plupart d'entre elles proviennent de ce que, dans une application complexe, l'expertise que l'on cherche à intégrer dans la base de connaissance, c'est-à-dire le savoir-faire, les compétences et les connaissances diverses, est détenue par des individus différents qui, au sein d'un groupe, communiquent, échangent leurs connaissances et collaborent à la réalisation d'une tâche commune. Dans ce cas, la connaissance du groupe n'est pas égale à la somme des connaissances des individus: chaque savoir-faire est lié à un point de vue particulier et ces différentes visions, parfois contradictoires, ne sont pas nécessairement cohérentes entre elles. La réalisation d'une tâche commune nécessitera alors des discussions, des mises au point, voire même des négociations pour résoudre les conflits éventuels.

Enfin, les problèmes sont parfois naturellement posés de manière distribuée. Que l'on doive concevoir un ensemble de robots explorateurs ou que l'on soit amené à définir un système de contrôle de trafic aérien, le problème suppose qu'il existe un certain nombre d'entités capables d'agir et d'interagir. Dans certains cas, comme dans celui des robots explorateurs, les interactions peuvent être positives et il est possible aux agents de s'aider les uns les autres, alors que dans d'autres, tels que le contrôle de trafic aérien, elles peuvent être négatives, les avions se gênant mutuellement, et la difficulté consiste à faire en sorte qu'ils n'entrent pas en collision. A partir de toutes ces remarques, n'est-il pas possible de "dés-individualiser" l'informatique en général et l'IA en particulier? N'est-il pas possible de prendre le problème à l'envers et de considérer justement les phénomènes collectifs comme à la fois premiers et porteurs de cette intelligence que l'on n'arrive toujours pas ni à définir, ni à véritablement comprendre ni même à reproduire effectivement?

1.1.2 ... à l'organisation artificielle

L'approche développée ici trouve en effet son fondement dans une critique radicale de l'informatique séquentielle et de l'IA classique, en considérant que les activités simples ou complexes, telles que la résolution de problèmes, l'établissement d'un

diagnostic, la coordination d'action ou la construction de systèmes sont le fruit d'une interaction entre entités relativement autonomes et indépendantes, appelées *agents*, qui travaillent au sein de communautés selon des modes parfois complexes de coopération, de conflit et de concurrence, pour survivre et se perpétuer. De ces interactions émergent des structures organisées qui, en retour, contraignent et modifient les comportements de ces agents.

Ce projet scientifique, que j'ai appelé *kénétique* (Ferber 1994), consiste à pouvoir étudier, concevoir et réaliser des univers ou des organisations d'agents artificiels (électroniques ou informatiques) capables d'agir, de collaborer à des tâches communes, de communiquer, de s'adapter, de se reproduire, de se représenter l'environnement dans lequel ils évoluent et de planifier leurs actions, pour répondre soit à des objectifs définis extrinsèquement (par un programmeur humain par exemple), soit intrinsèquement à partir d'un objectif général de survie.

La kénétique se veut à la fois la science et la technique des *organisations artificielles*², que l'on appelle ces organisations populations, sociétés, groupes, mondes ou univers. On dira aussi que la kénétique procède par la construction de *systèmes multi-agents*, c'est-à-dire par la réalisation de modèles électroniques ou informatiques composés d'entités artificielles qui communiquent entre elles et agissent dans un environnement. Plus précisément, la kénétique se propose de:

1. Définir une discipline scientifique qui prenne en compte l'interaction entre agents comme fondement pour comprendre le fonctionnement et l'évolution des systèmes.
2. Définir les différentes formes d'interactions, telles que la coopération, la compétition, la collaboration, l'encombrement, etc., et les relier à la problématique de l'auto-organisation ou à celles de la performance ou de la survie du système.
3. Dégager les grands mécanismes donnant lieu à l'auto-organisation, tels que le regroupement, la spécialisation, la répartition des tâches et des ressources, la coordination d'actions, la résolution de conflits, etc.
4. Définir des modèles opérationnels de ces interactions en décrivant le fonctionnement des agents et des systèmes multi-agents.

C'est ainsi, en mettant l'accent sur les interactions et plus exactement en analysant les systèmes d'interactions qui existent entre les agents, que les systèmes multi-agents (ou SMA) se distinguent des approches systémiques plus classiques en prenant le parti de l'émergence et en considérant que l'action et l'interaction sont les éléments moteurs de la structuration d'un système dans son ensemble.

La méthode utilisée relève à la fois des sciences théoriques et expérimentales. C'est par une analyse fine des différents types de systèmes multi-agents que l'on sera à même de classer les différentes formes d'interaction et les différents comportements

²On emploiera le terme système artificiel uniquement dans le sens "dispositif construit par l'être humain et qui est lui-même composé d'éléments artificiels". Il s'agit donc essentiellement de systèmes électroniques ou informatiques qui sont visés par de tels vocables, et non des entreprises ou des écosystèmes naturels aménagés par l'homme.

distribués qui peuvent être mis en œuvre pour résoudre des problèmes de coopération ou de compétition. C'est aussi en réalisant des modèles mathématiques et informatiques que l'on sera capable de théoriser ce champ et d'en faire un domaine scientifique à part entière. C'est surtout à partir d'expérimentations, c'est-à-dire à partir de la construction d'artefacts collectivement organisés que l'on sera en mesure de mieux analyser les composants nécessaires à la description, l'auto-organisation et l'évolution des sociétés d'agents. Mais les formes que prend l'expérimentation sont bien différentes de celles des sciences sociales ou de la nature. Les univers que crée la kénétique sont purement artificiels et tous leurs paramètres peuvent être contrôlés directement par le concepteur. Le caractère totalement construit de ces organisations autorise chacun à déterminer son point de vue et à choisir les paramètres qui l'intéressent.

L'organisation artificielle résultante ne dépend alors plus que de ces choix, et les propriétés des structures ne sont pas affectées par des facteurs extérieurs non maîtrisés. En effet, les entités d'un univers multi-agent sont entièrement le fruit de conceptions humaines, d'une création délibérée, dans laquelle les caractéristiques sont étroitement contrôlées. En particulier, et à la différence des études portant sur les systèmes sociaux humains ou animaliers, tout ce qui se passe dans la "tête" de ces agents artificiels est entièrement défini. Le concepteur a accès à tous les recoins de leur "psychisme", puisque, par définition, il est l'auteur de ses créatures.

Cependant en être l'auteur ne signifie pas que toutes les évolutions de l'univers puissent être prédites. Elles ne sont pas soumises à un déterminisme simple qui permettrait de prévoir toutes les configurations futures à partir d'une situation initiale. Au contraire, les systèmes multi-agents obéissent aux principes des phénomènes chaotiques (Gleick 1989). Toute modification des conditions initiales, aussi menue soit-elle, toute introduction de variables aléatoires, aussi limitée soit-elle, est amplifiée par les interactions entre les agents et il n'est pas possible de connaître à l'avance l'état ultérieur précis des agents au bout de quelque temps. Des phénomènes connus sous le nom d'*effets papillons* (un battement d'aile d'un papillon en Asie peut conduire, par amplifications progressives, à une tornade en Amérique) perturbent alors toutes les prédictions fines, et la plupart des résultats.

Ces systèmes ne sont pas pour autant soumis à aucune loi. Il est possible au contraire d'observer des effets d'auto-organisation et d'apparition de structures émergentes dont les évolutions sont parfaitement maîtrisées. De fait, il est possible, avec un peu d'expérience, de construire des systèmes qui manifestent des comportements voulus dans une situation stable, tout en se montrant capable de s'adapter à des perturbations non prévues initialement.

L'intérêt des systèmes multi-agents tient à ce que l'on peut expérimenter directement sur des modèles réduits de sociétés les théories que l'on désire mettre à jour. Toutes les théories concernant l'évolution sociale ou biologique ont droit de cité dans de tels univers, puisqu'elles n'ont plus à être validées par une expérience sur le terrain: seul, le fait que cette théorie soit ou non "calculatoire", c'est-à-dire réalisable dans le cadre d'un programme informatique, détermine si oui ou non une théorie est valable. Une théorie irrecevable dans le cadre de la nature peut très bien trouver son compte dans le cadre de la kénétique. Par exemple, il est possible de créer

des univers dans lesquels les individus se reproduisent de manière “lamarkienne” en supposant que les acquis comportementaux sont directement transmis aux gènes des descendants, ce qui est en contradiction avec nos connaissances de l’évolution des espèces animales.

Cependant, cette absence de contraintes d’expérimentation sur un univers réel entraîne inversement une pression extrême: celle de devoir expliciter chacun des rouages des mécanismes implémentés. Il n’est plus possible de laisser des aspects dans l’obscurité, ni même de définir vaguement un concept par un discours tortueux. Tous les éléments, tous les composants, tous les mécanismes définissant un système doivent être minutieusement pensés et aucune marge d’imprécision n’est autorisée, l’ordinateur étant le censeur ultime.

Pourquoi distribuer l’intelligence?

Le développement de ces techniques pose un problème fondamental: pourquoi cherche-t-on à créer des intelligences collectives? Pourquoi vouloir à tout prix prendre un point de vue local? Pourquoi tout simplement vouloir distribuer l’intelligence?

On peut répondre à ces questions de plusieurs manières. La première revient à constater que les systèmes informatiques industriels sont de plus en plus compliqués et font intervenir un grand nombre de sous-systèmes de nature fort diverse, comprenant de nombreuses fonctionnalités, interagissant avec plusieurs spécialistes humains (opérateurs, experts, techniciens, etc.), qui se trouvent souvent répartis naturellement dans l’espace physique.

Dans les grosses applications industrielles, telles que la surveillance de réseaux étendus (télécommunications, transport et distribution d’énergie, de matières premières, de produits finis) les sous-systèmes de surveillance et de gestion sont naturellement distribués aux nœuds du réseau et font appel à un grand nombre de techniciens et de programmes divers qui doivent travailler ensemble pour accomplir l’objectif global: faire en sorte que le système fonctionne. Chaque intervenant n’a qu’une vue partielle de ce qu’il doit faire, mais l’ensemble réagit convenablement si les actions de chacun sont efficacement coordonnées. De ce fait, on peut expliquer la nécessité d’une distribution de l’activité et de l’intelligence par les raisons suivantes:

- *Les problèmes sont physiquement distribués.* En premier lieu, les problèmes complexes sont souvent physiquement distribués: un réseau de transport est par nature réparti sur un espace important, mais un système de contrôle industriel, tels que ceux qui sont à l’œuvre dans des raffineries ou dans des chaînes de montage, suppose lui aussi une certaine distribution dans l’espace des activités. Il en est de même pour la gestion de trafic de véhicules: chaque voiture ou chaque avion est une entité autonome qui doit parvenir à sa destination dans le meilleur état possible.
- *Les problèmes sont fonctionnellement très distribués et hétérogènes.* Concevoir un produit industriel aussi élaboré qu’une voiture de course, qu’un avion de ligne ou qu’un lanceur de satellites réclame l’intervention d’un grand nombre de spécialistes, qui ne possèdent qu’une vision locale de l’ensemble des problèmes

posés par la réalisation du système: nul en effet n'est suffisamment savant ou qualifié pour produire une telle réalisation à lui tout seul. L'ensemble des problèmes est trop vaste pour un seul individu. Une voiture de formule 1, par exemple, fait intervenir un grand nombre d'experts pour sa mise au point: il y a le spécialiste des moteurs, celui des châssis, celui des pneumatiques, l'ingénieur en chef et le pilote. Toutes ces personnes mettent leurs connaissances en commun pour essayer de faire la meilleure voiture possible.

- *Les réseaux imposent une vision distribuée.* A l'heure des réseaux interplanétaires, où toute l'information et la puissance de traitement est répartie sur un nombre considérable de sites (le réseau Internet comprend d'ores et déjà plusieurs dizaines de milliers de sites de par le monde), le génie logiciel doit cesser de "penser global". Comme l'indique C. Hewitt (Hewitt 1985), il faut penser en termes de systèmes ouverts, c'est-à-dire en termes d'interopérabilité radicale des systèmes informatiques. La téléinformatique mobile en particulier nécessite une refonte complète de nos présupposés fondés sur la notion de machine de Von Neumann, séquentielle, fixe et qui gère de façon individuelle toute l'information dont elle a besoin. La proposition de M. Tokoro (Tokoro 1993) concernant la réalisation d'un "champ calculatoire" (computing field) s'inscrit dans ce courant. Il faut considérer que l'espace informatique n'est qu'une gigantesque toile d'araignée (la *World Wide Web* du réseau Internet en constitue une première approche) dans laquelle tout ordinateur, qu'il soit fixe, portable ou mobile, est connecté à l'ensemble des autres ordinateurs du monde entier. Les SMA se présentent ainsi comme des candidats sérieux pour la construction d'architectures ouvertes, distribuées, hétérogènes et souples, capables d'offrir une grande qualité de service dans un travail collectif, sans imposer une structure a priori.
- *La complexité des problèmes impose une vision locale.* Lorsque les problèmes sont trop vastes pour être analysés globalement, les solutions fondées sur des approches locales permettent souvent de les résoudre plus rapidement. Par exemple, la régulation de trafic aérien est un problème global complexe, difficile à résoudre du fait du grand nombre de paramètres qui sont mis en jeu et de l'ensemble des contraintes qui doivent être satisfaites. Comme nous le verrons au chapitre 8, des approches locales permettent de résoudre élégamment et efficacement ce type de problème. Evidemment, il est plus difficile (voire souvent impossible) de vérifier théoriquement la validité de la solution. Mais les résultats empiriques montrent que les solutions sont bonnes et paraissent souvent très "naturelles". La pensée locale induite par la kénétique constitue ainsi une approche prometteuse pour la résolution de grands problèmes complexes, car elle permet à la fois de simplifier l'obtention des résultats en évitant les difficultés liées à l'inefficacité des algorithmes fonctionnant sur des espaces globaux, tout en offrant des résultats comparables qui "émergent" des interactions locales.
- *Les systèmes doivent pouvoir s'adapter à des modifications de structure ou*

d'environnement. Savoir concevoir des systèmes informatiques efficaces, fiables et corrects ne suffit plus. Devant les défis de la complexité, il faut aussi penser à l'adaptabilité d'un logiciel à des modifications du contexte de travail (changement de système d'exploitation, de gestionnaire de bases de données, d'interfaces graphiques, ajouts d'autres logiciels, etc.), et sa nécessaire capacité d'évolution face à des besoins sans cesse accrus (ajouts de fonctionnalités, modifications de l'utilisation, intégration à d'autres logiciels, etc.). Dans ce cadre, les SMA, de par leur nature distribuée, parce qu'ils supposent toujours un raisonnement local, qu'ils permettent l'intégration et l'apparition ou la disparition d'agents en cours même de fonctionnement, constituent des architectures particulièrement aptes à prendre en compte l'évolutivité et l'adaptation nécessaires au fonctionnement du système.

- *Le génie logiciel va dans le sens d'une conception en termes d'unités autonomes en interactions.* L'histoire du développement des logiciels montre que la réalisation de programmes informatique suit une démarche visant à la réalisation de systèmes conçus comme des ensembles d'entités de plus en plus distribuées, mettant en jeu des composants davantage individualisés et autonomes. Le récent développement des langages à objets dans tous les secteurs de l'informatique est là pour en témoigner: le génie logiciel passe par la réalisation de modules autonomes capables d'interagir les uns avec les autres, même, et surtout, lorsque ceux-ci sont conçus par des personnes, des équipes ou des entreprises différentes. Il faut donc associer la fluidité des calculs, la distribution des traitements et l'hétérogénéité des réalisations. Les SMA ont ici un rôle essentiel à jouer en s'inscrivant comme les possibles successeurs des systèmes à objets, en ajoutant à la localité des comportements l'autonomie et la répartition des prises de décision. On peut ainsi déjà parier que le génie logiciel de demain sera "orienté agent", comme celui d'aujourd'hui commence à être "orienté objet".

1.2 Agent et société

Toute la problématique de la kénétique se situe au carrefour des notions d'agents et de sociétés, ou si l'on préfère, de la relation entre les comportements individuels et les phénomènes observés au niveau global. C'est ainsi que, dans la dialectique de l'individu et du groupe, les notions de coopération, conflit, collaboration et coordination d'actions prennent tout leur sens. Mais avant d'entrer plus avant dans cette problématique, nous devons essayer de circonscrire ce que l'on entend par les termes d'agent et de système multi-agent.

1.2.1 Quelques définitions

Qu'est ce qu'un agent? Comme dans tous les domaines porteurs, le terme *agent* est utilisé de manière assez vague. Cependant on peut dégager une définition minimale commune qui est approximativement la suivante:

Définition: *On appelle agent une entité physique ou virtuelle*

- a. qui est capable d'agir dans un environnement,*
- b. qui peut communiquer directement avec d'autres agents,*
- c. qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),*
- d. qui possède des ressources propres,*
- e. qui est capable de percevoir (mais de manière limitée) son environnement,*
- f. qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),*
- g. qui possède des compétences et offre des services,*
- h. qui peut éventuellement se reproduire,*
- i. dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.*

Chacun des termes de cette définition est important. Une *entité physique* est quelque chose qui agit dans le monde réel: un robot, un avion ou une voiture sont des exemples d'entités physiques. En revanche, un composant logiciel, un module informatique sont des *entités virtuelles*, car elles n'existent pas physiquement.

Les agents sont *capables d'agir*, et non pas seulement de raisonner comme dans les systèmes d'IA classique. L'action, qui est un concept fondamental pour les systèmes multi-agents, repose sur le fait que les agents accomplissent des actions qui vont modifier l'environnement des agents et donc leurs prises de décision futures. Ils peuvent aussi *communiquer* entre eux, et c'est d'ailleurs là l'un des modes principaux d'interaction existant entre les agents. Ils agissent dans un environnement, sauf, comme nous le verrons, pour les agents purement communicants pour lesquels toutes les actions se résument à des communications.

Les agents sont doués d'*autonomie*. Cela signifie qu'ils ne sont pas dirigés par des commandes venant de l'utilisateur (ou d'un autre agent), mais par un ensemble de *tendances* (cf. chap. 5) qui peuvent prendre la forme de buts individuels à satisfaire ou de fonctions de satisfaction ou de survie que l'agent cherche à optimiser. On pourrait dire ainsi que le moteur d'un agent, c'est lui-même. C'est lui qui est actif. Il a la possibilité de répondre par l'affirmative ou le refus à des requêtes provenant des autres agents. Il dispose donc d'une certaine liberté de manœuvre, ce qui le différencie de tous les concepts semblables, qu'ils s'appellent "objets", "modules logiciels" ou "processus". Mais l'autonomie n'est pas seulement comportementale, elle porte aussi sur les ressources. Pour agir, l'agent a besoin d'un certain nombre de ressources: énergie, CPU, quantité de mémoire, accès à certaines sources d'informations, etc. Ces ressources sont à la fois ce qui rend l'agent non seulement dépendant de son environnement, mais aussi, en étant capable de gérer ces

ressources, ce qui lui donne une certaine indépendance vis-à-vis de lui. L'agent est ainsi à la fois un système ouvert (il a besoin d'éléments qui lui sont extérieurs pour survivre) et un système fermé (car les échanges qu'il a avec l'extérieur sont très étroitement réglementés).

Les agents n'ont qu'une *représentation partielle* de leur environnement, c'est-à-dire qu'ils n'ont pas de vision globale de tout ce qui se passe. C'est d'ailleurs ce qui se passe dans les réalisations humaines d'envergure (la fabrication d'un Airbus par exemple) dans lesquelles personne ne connaît tous les détails de la réalisation, chaque spécialiste n'ayant qu'une vue partielle correspondant à son domaine de compétence.

L'agent est ainsi une sorte "d'organisme vivant" dont le *comportement*, qui se résume à communiquer, à agir et, éventuellement, à se reproduire, vise à la satisfaction de ses besoins et de ses objectifs à partir de tous les autres éléments (perceptions, représentations, actions, communications et ressources) dont il dispose.

Définition: *On appelle système multi-agent (ou SMA), un système composé des éléments suivants:*

1. *Un environnement E , c'est-à-dire un espace disposant généralement d'une métrique.*
2. *Un ensemble d'objets O . Ces objets sont situés, c'est-à-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans E . Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.*
3. *Un ensemble A d'agents, qui sont des objets particuliers ($A \subseteq O$), lesquels représentent les entités actives du système.*
4. *Un ensemble de relations R qui unissent des objets (et donc des agents) entre eux.*
5. *Un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O .*
6. *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(cf. chap. 4).*

La figure 1.1 donne une illustration de la notion de système multi-agents.

Il existe un cas particulier de systèmes dans lequel $A = O$, et E est égal à l'ensemble vide. Dans ce cas, les relations R définissent un réseau: chaque agent est lié directement à un ensemble d'autres agents, que l'on appelle ses *accointances* (chap. 5). Ces systèmes, que l'on peut appeler *SMA purement communicants*³, sont

³Nous utiliserons le terme *système multi-agent communicant* ou *SMA communicant*, comme notation réduite de ce que nous devrions appeler *système multi-agent dont les agents communiquent directement entre eux par le biais de messages*. Lorsqu'il n'y a pas d'environnement et que les agents ne font que communiquer, on parlera de *SMA purement communicant*. Nous effectuerons le même raccourci avec le terme *système multi-agent situé* ou *SMA situé*, pour signifier qu'il s'agit d'un *système multi-agent dont les agents sont positionnés dans un environnement*, et *SMA purement situé* pour désigner un SMA situé dans lequel les agents ne communiquent pas par envois de messages, mais seulement par propagation de signaux (cf. chap. 6).

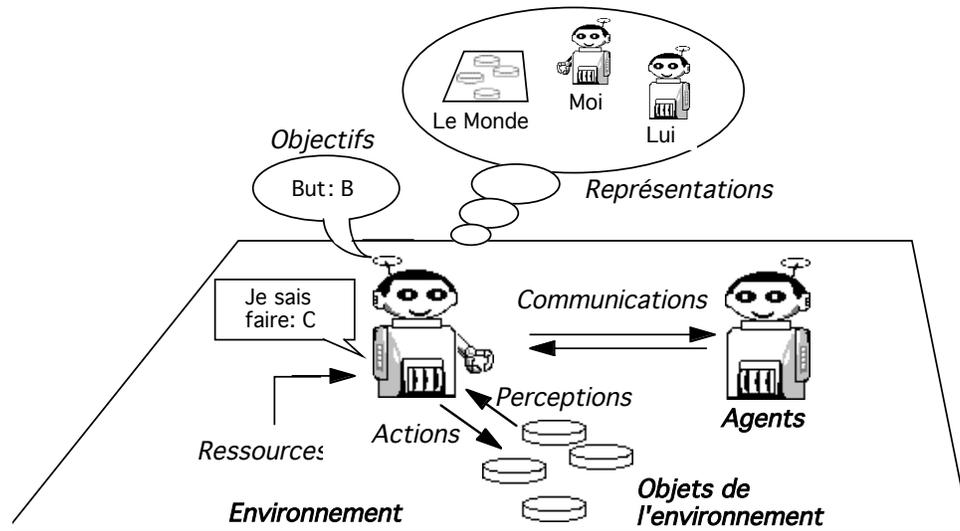


Figure 1.1: Représentation imagée d'un agent en interaction avec son environnement et les autres agents

très courants en intelligence artificielle distribuée. Leur domaine de prédilection est la coopération de modules logiciels dont la fonction est de résoudre un problème ou d'élaborer une expertise (interprétation de signaux ou conception d'un produit par exemple) à partir de modules spécialisés, comme dans le cas d'un système de contrôle distribué, où E est défini par la structure du réseau sous-jacent. Ces systèmes se caractérisent par le fait que les interactions sont essentiellement des communications intentionnelles (cf. chap. 6) et que le mode de travail ressemble à celui d'un organisme social (groupe de travail, entreprise, administration, etc.).

Lorsque les agents sont *situés*, E est généralement un espace métrique, et les agents sont capables de percevoir leur environnement, c'est-à-dire de reconnaître les objets situés dans l'environnement en fonction de leurs capacités perceptives, et d'agir, c'est-à-dire de transformer l'état du système en modifiant les positions et les relations existant entre les objets.

Nous verrons que la plupart des systèmes multi-agents réactifs considèrent que la notion d'environnement est fondamentale pour la coordination des actions entre plusieurs agents. Par exemple, dans un univers de robots, les agents A sont les robots, E est l'espace géométrique euclidien dans lequel se meuvent les robots et O se compose évidemment des agents, mais aussi de l'ensemble des objets physiques placés ici et là, et que les robots doivent éviter, prendre ou manipuler. Les opérations Op sont les actions que les robots peuvent faire en se déplaçant, en bougeant les autres objets ou en communiquant, et R est l'ensemble des relations qui unissent certains agents à d'autres, telles que des relations d'acointances (certains agents en connaissent d'autres) et les relations de communicabilité (les agents peuvent communiquer avec certains agents mais pas nécessairement avec tous).

De ce fait, la dualité agent/environnement est au cœur des systèmes multi-agents situés. En agissant à partir de ses perceptions de l'espace physique et des communications directes qu'il reçoit, l'agent se définit comme l'image duale de son en-

vironnement, c'est-à-dire par ce qui le distingue de ce qui l'entoure. Inversement, l'environnement d'un agent est caractérisé par tout ce qui n'est pas lui. Il est donc impossible de définir la notion d'agent situé indépendamment de celle d'environnement, les deux étant intrinsèquement liées et constituant deux aspects complémentaires d'un univers multi-agent. La réalisation d'un système multi-agent situé passe ainsi par la définition simultanée de la structure des agents et de celle de leur environnement, les actions de ceux-là devant s'exécuter au sein de celui-ci. Lorsque les agents sont purement communicants ou purement situés, la définition que nous avons donné des systèmes multi-agents peut être adaptée.

Définition: *Par comparaison avec la définition générale d'un agent donnée précédemment, on appelle agent purement communicant (ou agent logiciel) une entité informatique qui:*

- a. se trouve dans un système informatique ouvert (ensemble d'applications, de réseaux et de systèmes hétérogènes)*
- b. peut communiquer avec d'autres agents,*
- c. est mue par un ensemble d'objectifs propres,*
- d. possède des ressources propres,*
- f. ne dispose que d'une représentation partielle des autres agents,*
- g. possède des compétences (services) qu'elle peut offrir aux autres agents,*
- i. a un comportement tendant à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose et en fonction de ses représentations et des communications qu'elle reçoit.*

Un agent purement communicant se distingue donc de la notion d'agent en général par le fait qu'il ne possède pas de perception des autres agents (la mention (e) n'existe plus), que ses tendances prennent l'aspect d'objectifs (ou intentions, cf. chap. 5), qu'il n'agit pas dans un environnement⁴ et que son contexte d'évolution est naturellement celui des réseaux informatiques.

Définition: *On appelle agent purement situé une entité physique (ou éventuellement informatique si on la simule) qui:*

- a. se trouve située dans un environnement,*
- c. est mue par une fonction de survie,*
- d. possède des ressources propres, sous la forme d'énergie et d'outils,*

⁴Au sens où nous l'entendons dans cet ouvrage, c'est-à-dire d'un espace muni d'une métrique et dans lequel des agents sont positionnés. En revanche, les agents purement communicants vivent dans ce que l'on appelle souvent un *environnement informatique*, c'est-à-dire un ensemble de processeurs, de logiciels et de ressources qu'ils peuvent utiliser et avec lesquels ils interagissent. Pour éviter une confusion entre ces deux termes, nous utiliserons le terme *contexte* pour décrire le cadre dans lequel évoluent les agents.

- e. est capable de percevoir (mais de manière limitée) son environnement,
- f. ne possède pratiquement aucune représentation de son environnement,
- g. possède des compétences,
- h. peut éventuellement se reproduire,
- i. a un comportement tendant à satisfaire sa fonction de survie, en tenant compte des ressources, des perceptions et des compétences dont elle dispose.

Les agents purement situés sont donc à l’opposé des agents logiciels en ce qui concerne les capacités de représentations (quasiment nulles) et le fait que les communications ne s’effectuent généralement pas directement, mais indirectement par le biais des perceptions et de leurs actions dans l’environnement (cf. chap. 6).

1.2.2 Niveaux d’organisation

En reprenant la classification proposée par G. Gurvitch (Gurvitch 1963), maintenant traditionnelle en sociologie (Rocher 1968), on peut distinguer trois niveaux d’organisation dans les systèmes multi-agents:

1. Le niveau *micro-social*, où l’on s’intéresse essentiellement aux interactions entre agents et aux différentes formes de liaison qui existent entre deux ou un petit nombre d’agents. C’est à ce niveau que la plupart des études ont été généralement entreprises en intelligence artificielle distribuée.
2. Le niveau des *groupes*, où l’on s’intéresse aux structures intermédiaires qui interviennent dans la composition d’une organisation plus complète. A ce niveau, on étudie les différenciations des rôles et des activités des agents, l’émergence de structures organisatrices entre agents et le problème général de l’agrégation des agents lors de la constitution d’organisations.
3. Le niveau des *sociétés globales* (ou *populations*) où l’intérêt se porte surtout sur la dynamique d’un grand nombre d’agents, ainsi que sur la structure générale du système et son évolution. Les recherches se situant dans le cadre de la vie artificielle se situent assez souvent à ce niveau.

Evidemment, à l’instar de la plupart des classifications, celle-ci ne doit pas être considérée comme une norme fixe d’étude, mais plutôt comme un repère destiné à guider notre jugement et notre analyse. Elle suggère notamment qu’il est possible d’analyser et de concevoir des organisations artificielles soit à partir d’une spécification globale de la société que l’on veut obtenir et des propriétés recherchées, soit au contraire à partir de la définition fine des agents, la société finale étant la conséquence — recherchée ou non — des interactions entre les agents.

La première approche est descendante et caractéristique d’une démarche d’ingénieur cherchant à obtenir un système qui réponde à un besoin. La seconde est plus

expérimentale: elle s'accommode mieux d'un travail exploratoire dans lequel on tente d'obtenir des propriétés émergentes.

Le travail portant sur les organisations artificielles se situe donc au centre d'une dualité irréductible agent/organisation: toute organisation est le résultat d'une interaction entre agents, et le comportement des agents est contraint par l'ensemble des structures organisatrices, comme le montre la figure 1.2.

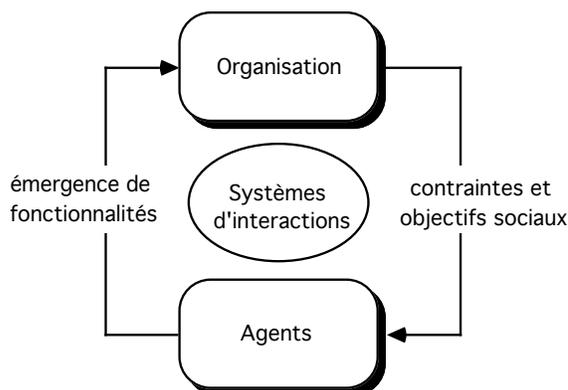


Figure 1.2: La relation micro-macro dans les systèmes multi-agents

Par exemple, si la différenciation comportementale et la constitution de spécialistes peuvent être vues comme des fonctionnalités émergentes, l'attribution d'un rôle à un agent réduit d'autant la marge de manœuvre des autres, ce qui impose des contraintes à chacun des agents quant à l'ensemble de ses comportements possibles (cf. chap. 7). De même, si des phénomènes émergents tels que des mouvements giratoires peuvent apparaître dans un système anti-collision fortement contraint (cf. chap. 8), tout agent venant s'insérer dans ce mouvement doit nécessairement tourner dans le sens giratoire imposé par le groupe.

De ce fait, une société n'est pas nécessairement une entité "donnée", c'est-à-dire un objet préexistant à l'action de ses membres. Si des organisations telles que des institutions publiques ou des entreprises possèdent une individualité propre qui se distingue de l'ensemble de celles de ses membres, il n'en est pas nécessairement ainsi de structures collectives plus simples tels que des groupes de travail ou des hordes d'animaux. Même des sociétés réputées complexes, telles que des colonies d'abeilles ou de fourmis, ne doivent pas nécessairement être considérées comme des individus à part entière si l'on veut en comprendre l'organisation et les phénomènes de régulation et d'évolution qui y règnent. En termes de systèmes multi-agents, cela signifie qu'une organisation peut *émerger* de la rencontre d'actions individuelles, sans qu'il soit nécessaire de définir un objet particulier (un élément de l'ensemble O) qui le représente.

A l'inverse, cela ne signifie pas qu'il est impossible ou inutile, ou même impensable, de représenter les sociétés comme des entités à part entière. On peut très naturellement concevoir un agent sous la forme d'un système multi-agents, et donc considérer les SMA comme des poupées russes, c'est-à-dire comme des

emboîtements d'agents eux-mêmes représentés sous la forme de SMA. Le processus de construction est alors récursif et se continue jusqu'à la réalisation d'agents primitifs indécomposables.

Le rapport qui existe entre agent et organisation est donc une réalité mouvante. Sauf dans les cas où l'organisation est décrite intégralement par le concepteur du système, la répartition du travail, la coordination des actions et l'assignation des rôles résultent des activités de ses membres. Une organisation n'est donc pas seulement une structure statique telle qu'on peut l'analyser si l'on considère un système multi-agent à un moment donné, mais un processus organisateur dont la structure n'est qu'un effet résultant. Cette conception rejoint celle de M. Crozier et E. Friedberg sur les organisations humaines (Crozier et Friedberg 1977), où ces dernières sont considérées comme procédant des systèmes d'actions des acteurs de ces organisations, c'est-à-dire des stratégies de prises de pouvoir, de repli et de négociations définissant des "ordres locaux" de régulation (Friedberg 1993).

Entre les individus qui en sont les acteurs et les organisations qui en résultent, il existe donc des processus essentiels par lesquels cette dualité s'exprime. Il s'agit des phénomènes d'interaction dont la logique dépend à la fois des capacités des agents mais aussi des caractéristiques et des contraintes du système dans son ensemble. C'est d'ailleurs en mettant l'accent sur les interactions et, plus exactement, en analysant les systèmes d'interactions qui existent entre les agents que les systèmes multi-agents se distinguent des considérations systémiques ou sociologiques plus classiques dont nous reparlerons plus loin. En effet, alors que nombre d'approches considèrent les systèmes d'un point de vue global, holiste même, et dont l'analyse procède de manière descendante, les systèmes multi-agents prennent le parti de l'émergence, c'est-à-dire de l'action et de l'interaction, comme élément moteur de la structuration de la société dans son ensemble. Les systèmes d'interactions constituent ainsi l'objet d'étude essentiel des systèmes multi-agents et servent à articuler le rapport qui existe entre l'agent d'une part et l'organisation dans sa globalité d'autre part.

1.2.3 Social ou biologique?

Faut-il concevoir les agents comme des entités déjà "intelligentes", c'est-à-dire capable de résoudre certains problèmes par eux-mêmes, ou bien faut-il les assimiler à des êtres très simples réagissant directement aux modifications de l'environnement?

Agents cognitifs et réactifs

Ces deux conceptions ont donné lieu à deux écoles de pensée. La première, l'école "cognitive", est la plus représentée dans le domaine que l'on appelle *intelligence artificielle distribuée* (IAD) car elle trouve son origine dans la volonté de faire communiquer et coopérer des systèmes experts classiques. Dans ce cadre, un système multi-agent est composé d'un petit nombre d'agents "intelligents". Chaque agent dispose d'une base de connaissance comprenant l'ensemble des informations et des savoir-faire nécessaires à la réalisation de sa tâche et à la gestion des interactions

avec les autres agents et avec son environnement. On dit aussi que les agents sont “intentionnels”, c’est-à-dire qu’ils possèdent des buts et des plans explicites leur permettant d’accomplir leurs buts. Dans ce cadre, les problèmes de coopération ressemblent étonnamment à ceux de petits groupes d’individus, qui doivent coordonner leur activité et sont parfois amenés à négocier entre eux pour résoudre leurs conflits (Bond et Gasser 1988) (Demazeau et Müller 1991) (Chaib-Draa et al. 1992). Les analogies sont alors sociales et nombre de chercheurs dans ce domaine s’appuient sur les travaux de sociologie et en particulier sur la sociologie des organisations et des petits groupes.

L’autre tendance, l’école “réactive”, prétend au contraire qu’il n’est pas nécessaire que les agents soient intelligents individuellement pour que le système ait un comportement global intelligent (Deneubourg et al. 1991) (Steels 1989) (Ferber et Dro-goul 1992). Des mécanismes de réaction aux événements, ne prenant en compte ni une explicitation des buts, ni des mécanismes de planification, peuvent alors résoudre des problèmes qualifiés de complexes. L’exemple le plus manifeste d’organisation émergente est celle de la fourmilière (Corbara et al. 1993): alors que toutes les fourmis se situent sur un plan d’égalité et qu’aucune d’entre elles ne possède de pouvoir d’autorité stricte sur les autres, les actions des fourmis se coordonnent de manière que la colonie survive et fasse donc face à des problèmes complexes tels que ceux posés par la recherche de nourriture, les soins à donner aux œufs et aux larves, la construction de nids, la reproduction, etc.

Intentions, pulsions et réflexes

En réalité cette division cognitif/réactif est parfois trop simpliste. Il nous faut l’affiner en dégageant deux axes:

1. La conduite des agents est régie par l’opposition entre des comportements *téléonomiques* dirigés vers des buts explicites et des comportements *réflexes* régis par les perceptions. Nous l’avons dit, tous les agents sont par définition mus par des tendances. Mais, comme nous le verrons au chapitre 5, certaines tendances peuvent provenir de l’environnement, alors que d’autres sont explicitement exprimées dans les agents. On parlera de comportements “réflexes” dans le premier cas et de comportements “téléonomiques” dans le second.
2. La relation de l’agent à son environnement pose la problématique classique du couple sujet/objet de savoir si l’agent dispose d’une représentation symbolique et explicite du monde à partir de laquelle il peut raisonner, ou si sa représentation se situe à un niveau sub-symbolique, c’est-à-dire intégré dans ses capacités sensori-motrices (Piaget et Inhelder 1966). Dans le premier cas, on parlera d’*agents cognitifs* et, dans le second, d’*agents réactifs*.

Ces deux axes peuvent se combiner pour distinguer quatre types d’agents opposés par leur manière d’agir et de concevoir le monde, comme le montre le tableau 1.1. La plupart des agents cognitifs sont *intentionnels*, cela signifie qu’ils possèdent des buts explicites qui motivent leurs actions. On les dit aussi *rationnels*, c’est-à-dire que les

Relation au monde	Agents cognitifs	Agents réactifs
Conduites		
Téléonomiques	Agents intentionnels	Agents pulsionnels
Réflexes	Agents "modules"	Agents tropiques

Table 1.1: Les différents types d'agent en fonction de leurs capacités représentationnelles et de leurs modes de conduite

actions qu'ils mettent en œuvre suivent un principe de rationalité par rapport aux buts qui les dirigent. Les *agents "modules"* (cognitifs réflexes) se situent à la limite de la notion d'agent. Ils sont parfois utilisés comme agents auxiliaires capables de répondre à des questions ou d'accomplir des tâches qui leur sont commandées par d'autres agents, sans que ces demandes viennent s'inscrire comme des buts explicites de l'agent. On peut penser à des bases de données ou à des agents calculateurs, c'est-à-dire à des agents qui répondent directement aux requêtes qui leur sont adressées.

Les agents réactifs peuvent être dirigés par des mécanismes de motivation les poussant à accomplir une tâche, tels que la satisfaction d'un besoin interne (par exemple maintenir leur niveau énergétique) ou l'accomplissement d'un but défini par le constructeur, auquel cas on parlera d'agents *pulsionnels*. On peut aussi ne les faire répondre qu'à des stimuli de l'environnement, leur comportement étant guidé intégralement par l'état local du monde dans lequel ils se trouvent plongés. On les qualifiera alors d'agents *tropiques*. Les systèmes multi-agents ne comprenant que des agents tropiques seront appelés SMA tropiques. La différence entre ces deux formes d'action est due à la disparité de provenance des sources de motivation. Dans le premier cas, celles-ci se trouvent à l'intérieur de l'agent alors que, dans le second, elles se situent dans l'environnement. En réalité, cette distinction est parfois difficile à préserver. Les comportements de ces agents sont souvent déclenchés par des combinaisons de pulsions internes et de stimulations externes, et le passage du réflexe au pulsionnel est relativement arbitraire. De ce fait, par la suite, sauf lorsque les besoins s'en feront sentir, nous parlerons essentiellement d'agents réactifs, sans distinguer entre pulsionnels et tropiques.

Les agents cognitifs, en raison de leur sophistication et de leur capacité à raisonner sur le monde, peuvent travailler de manière relativement indépendante. Les tâches qu'ils accomplissent sont complexes au regard des facultés plus élémentaires des agents réactifs. Ils peuvent ainsi résoudre des problèmes compliqués de manière relativement individuelle. Leur représentation interne et les mécanismes d'inférence dont ils disposent leur permettent de fonctionner indépendamment des autres agents et leur offrent une grande souplesse dans l'expression de leur comportement.

Inversement, la structure plus frustrée des agents réactifs leur impose des comportements plus rigides. De ce fait, les agents réactifs ne sont pas très puissants puisque réduits à leur propre moyens. Individuellement ils sont très faibles. Mais leur force vient de leur capacité à se mettre en groupe, c'est-à-dire de constituer des colonies capables de s'adapter à leur environnement. Ainsi, ce n'est pas au niveau de

l'individu que les agents réactifs sont intéressants, mais au niveau de la population et des capacités d'adaptation et d'évolution qui émergent des interactions entre ses membres. Les agents réactifs n'ont pas ou peu d'individualité: ils se fondent dans la masse, mais de par leur nombre et la redondance qui en découle, ils peuvent faire face à des tâches complexes et ainsi rivaliser en termes de performances avec des agents plus sophistiqués mais moins nombreux.

Anticipations et réaction

Enfin, et c'est peut-être ce qui constitue la distinction essentielle, l'opposition cognitif/réactif recouvre une autre dualité: *la capacité ou non d'anticiper sur les événements futurs et de s'y préparer*. Les agents réactifs, par le fait même qu'ils n'ont pas de représentation de leur environnement et des autres agents, sont incapables de prévoir ce qui va se passer et donc d'anticiper en planifiant les actions à accomplir.

Au contraire, les agents cognitifs, par leur capacité de raisonner sur des représentations du monde, sont capables à la fois de mémoriser des situations, de les analyser, de prévoir des réactions possibles à leurs actions, d'en tirer des conduites pour les événements futurs et donc de planifier leur propre comportement (toute planification suppose, en effet, que l'on soit capable de traiter des actions en dehors de leur exécution, à partir de description de l'environnement et de ces actions). Et c'est donc grâce à leurs capacités cognitives qui leur permettent de se construire un monde virtuel qu'ils peuvent manipuler que les agents cognitifs sont à même de produire des plans d'actions.

Cette capacité d'anticipation et de planification permet à certains agents d'optimiser leur comportement et de n'effectuer ainsi que les actions véritablement nécessaires. Par exemple, supposons qu'un robot veuille franchir une porte et que celle-ci soit fermée à clef. S'il s'agit d'un agent cognitif, il pourra construire un plan "dans sa tête" tel que:

```
Plan ouvrirPorte
  aller jusqu'à l'endroit où se trouve la clef
  prendre la clef
  aller jusqu'à la porte
  ouvrir la porte avec la clef
```

En exécutant ce plan, l'agent cognitif ira directement au lieu où se trouve la clef pour la prendre, puis il se dirigera vers la porte pour l'ouvrir à l'aide de la clef.

Au contraire, un agent réactif, du fait qu'il ne dispose pas de représentation de l'univers dans lequel il évolue, ne peut pas effectuer ce type de raisonnement a priori. Il ne réagit que face à la situation. Pour résoudre ce problème, on pourra construire un agent réactif qui dispose du comportement suivant:

```
R1: si je suis devant la porte et que j'ai une clef,
    alors l'ouvrir
R2: si je suis devant la porte et sans clef,
    alors essayer de l'ouvrir
```

```
R3: si la porte ne s'ouvre pas
    et que je n'ai pas la clef,
    alors aller chercher la clef
R4: si je cherche une clef
    et qu'il y a une clef devant moi,
    alors prendre la clef et aller vers la porte
```

Ces quatre règles suffisent pour régler le comportement d'un robot réactif: si l'agent se trouve devant une porte fermée à clef, il essaiera de chercher la clef, puis il reviendra pour ouvrir la porte⁵.

La différence entre les deux comportements est caractéristique. D'abord, dans le premier cas, le plan est construit, alors que dans le second les règles sont données par le concepteur de l'agent. Ensuite, la palme de l'efficacité, en termes de nombre d'actions effectuées, revient évidemment à l'agent cognitif qui optimise le nombre de ses déplacements puisqu'il peut prévoir la suite des actions à entreprendre, alors que l'agent réactif est contraint d'aller d'abord vers la porte avant de se rendre compte que la clef n'est pas là et qu'il faut aller la chercher. En revanche, l'agent réactif est plus souple. Si la porte est ouverte, il l'ouvre directement sans aller chercher la clef au préalable.

On peut aussi dire que l'agent réactif contient une connaissance compilée des actions à effectuer: il n'a pas besoin de construire une représentation mentale de son monde car il lui suffit de simplement réagir aux situations qui se présentent. Mais la simplicité de ses comportements témoigne plus de l'intelligence des concepteurs que de la sienne propre.

Cette opposition entre cognitif et réactif, si elle permet de bien situer les problèmes, ne doit cependant pas être prise au pied de la lettre. En effet, où placer la limite entre cognitif et réactif? Où se situe la ligne de démarcation entre les agents qui disposent de modèles explicites et ceux qui ne réagissent qu'aux stimuli de leur environnement? Il existe en effet toute une graduation entre l'agent réactif pur, qui ne réagit qu'aux stimuli, et l'agent cognitif total qui possède un modèle symbolique du monde qu'il met continuellement à jour et à partir duquel il planifie toutes ses actions. Entre ces deux extrêmes, tout un éventail de possibilités est offert, de l'agent qui s'adapte en mémorisant certaines informations sous forme de paramètres numériques à celui qui possède une "carte mentale", c'est-à-dire une représentation analogique de son environnement lui permettant ainsi de déterminer des trajectoires lors de son déplacement. Cette distinction, quoiqu'utile dans un premier temps ne représente donc que les extrémités d'un segment de droite, et non une opposition catégorique (fig. 1.3). En effet, l'enjeu actuel est d'une part d'essayer de construire des agents cognitifs à partir d'organisations réactives et d'autre part de réaliser des agents qui disposent à la fois de capacités cognitives et réactives.

Nous ferons par la suite souvent mention de ce contraste cognitif/réactif, mais il faudra toujours pondérer cette dichotomie et comprendre qu'il ne s'agit que d'une

⁵Nous verrons au chapitre 4 qu'il existe plusieurs manières de définir le comportement d'un agent réactif, les règles données ici étant caractéristiques d'une technique fondée sur les "actions situées".

classification de travail, qui doit pouvoir être dépassée, par une synthèse des deux termes, chaque fois que cela s'avère nécessaire.

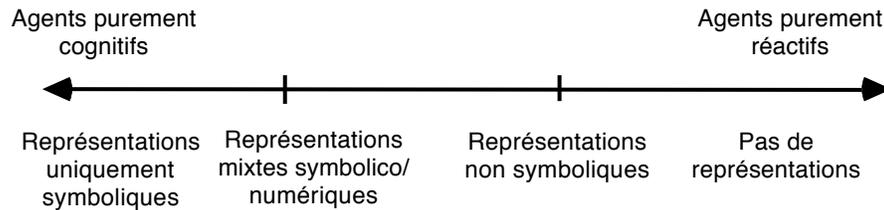


Figure 1.3: La distinction cognitif/réactif définit un axe pratique d'évaluation de la capacité des agents à accomplir individuellement des tâches complexes et à planifier leurs actions.

1.2.4 Architecture et comportement

Un agent se caractérise essentiellement par la manière dont il est conçu et par ses actions, en d'autres termes par son *architecture* et par son *comportement*. L'architecture correspond à un point de vue de concepteur, qui peut se résumer ainsi: comment assembler les différentes parties d'un agent de manière qu'il accomplisse les actions que l'on attend de lui? Certaines questions deviennent alors particulièrement pertinentes: les agents disposent-ils d'une représentation de leur environnement ou bien ne font-ils que réagir à des stimuli qu'ils perçoivent? Les agents sont-ils capables de raisonner à partir de symboles ou bien n'appliquent-ils que des actions pré-établies? Les réponses à ces questions se trouvent à "l'intérieur" des agents, dans leurs "entrailles". Il faut "soulever le capot" pour savoir quels sont les choix d'implémentation qui ont été effectués. L'architecture d'un agent caractérise ainsi sa structure interne, c'est-à-dire le principe d'organisation qui sous-tend l'agencement de ses différents composants. Nous verrons au chapitre 3 qu'il existe un grand nombre d'architectures envisageables. Certaines s'avèrent plus efficaces en temps de calcul, d'autres sont plus souples et permettent de coder un vaste éventail d'actions, d'autres encore présentent l'avantage de la simplicité.

En revanche, le comportement est analysable sans connaître les détails d'implémentation. Il s'agit d'un phénomène qui peut être appréhendé par un observateur extérieur, qui, au regard des actions qu'entreprend l'agent, décrit la relation qui existe entre l'agent, son environnement et les autres agents. Par exemple, le fait pour un avion d'éviter un autre avion est une caractéristique de son comportement, quelle que soit la manière dont il est implémenté. Le comportement caractérise ainsi l'ensemble des propriétés que l'agent manifeste dans son environnement, c'est-à-dire ce que l'on nomme en biologie sa *fonction*. Ce n'est plus en "soulevant le capot" que l'on peut comprendre le comportement d'un agent, mais en regardant sa manière d'évoluer et de répondre aux sollicitations de son environnement. Cette approche conduit à la réalisation de modèles comportementaux indépendants de l'architecture, même si, comme nous

le verrons, certaines architectures se prêtent mieux que d'autres à la réalisation de certains comportements. L'opposition entre les points de vue du concepteur et de l'observateur recoupe le débat bien connu en biologie entre la structure et la fonction. Pour le concepteur, le problème consiste à trouver une architecture qui lui permette d'implémenter des robots capables de manifester un certain comportement. Ainsi, pour le concepteur, le comportement de l'agent est la manifestation de la structure de l'agent, c'est-à-dire de son architecture et de son programme. Le comportement apparaît alors comme une spécification externe de l'agent, l'architecture définissant les relations internes permettant d'aboutir à cette spécification.

Inversement, l'observateur perçoit un comportement à partir duquel il peut induire ou spécifier ce qu'une architecture est censée produire. Si l'agent est naturel, alors les architectures que l'on peut proposer ne sont que des modèles par rapport à la réalité. En revanche, si l'agent est artificiel, les modèles comportementaux agissent comme des spécifications, c'est-à-dire comme des contraintes de conception lors de la réalisation de la structure interne d'un agent.

1.2.5 Langages, communications et représentations

L'informatique est avant tout affaire de langage et de description. On ne peut concevoir de programmes informatiques sans l'emploi d'un ensemble de langages destinés à représenter, décrire et implémenter des structures informatiques et des processus. Les concepteurs de systèmes multi-agents font ainsi usage d'un grand nombre de langages et de formalismes afin d'en cerner tous leurs aspects. On peut les classer en quatre grands types, selon qu'on s'intéresse à l'implémentation ou à la formalisation de SMA, à la représentation des connaissances des agents, à la définition de leur comportement ou à leurs communications (fig. 1.4).

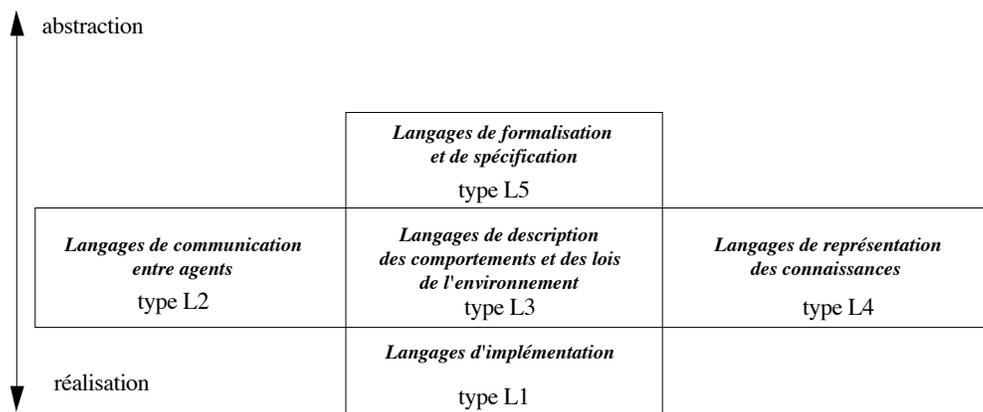


Figure 1.4: Les différents langages et formalismes intervenant dans la réalisation de SMA

Type L1: Langages d'implémentation

Ils sont utilisés pour programmer le système multi-agents. Ceci recouvre à la fois les structures informatiques utilisées pour les agents et l'environnement (s'il est simulé), les mécanismes informatiques pour permettre le parallélisme inter et intra-agent, l'implémentation effective des comportements, les activités d'envois et de réception de messages, la perception d'objets et l'ensemble des outils nécessaires pour mettre au point un SMA. Les langages les plus utilisés à ce niveau sont le plus souvent des langages de programmation classiques tels que LISP, C/C++, PROLOG ou SMALLTALK, des langages supportant déjà des mécanismes d'exécution parallèles tels que les langages d'acteurs (cf. chap. 3), voire des systèmes à base de règles.

Type L2: Langages de communication

Ils assurent les interactions entre agents (communicants) par le biais de transmissions d'informations et de demandes mutuelles de renseignements et de services. De ce fait, les langages de communication constituent la "colle" permettant à des agents éventuellement hétérogènes de coordonner leurs actions, de coopérer à un but commun et de s'intégrer à un groupe d'agents.

Type L3: Langages de description des comportements et des lois de l'environnement

Décrire directement les comportements des agents ou les lois de l'univers de l'environnement dans des langages d'implémentation ajoute un certain nombre de détails inutiles pour la compréhension du système et masque les principes essentiels. C'est pourquoi, des formalismes adaptés peuvent être utilisés pour décrire ces comportements ou ces lois à un niveau plus abstrait, de manière à bénéficier de représentations indépendantes de l'implémentation et de la nature des agents. Néanmoins, ces formalismes restent à un niveau opératoire, c'est-à-dire qu'ils décrivent effectivement *ce qui se passe* de manière abstraite. On pourra utiliser des langages à base de règles de production, ou, comme nous le verrons au chapitre 3, bénéficier des capacités d'abstraction fournies par des formalismes issus de l'automatisme ou des systèmes distribués tels que les réseaux de Petri. Nous présenterons au chapitre 4 le modèle BRIC pour décrire les comportements des agents de manière modulaire.

Type L4: Langages de représentation des connaissances

Les langages de représentation des connaissances sont utilisés par les agents cognitifs pour décrire des modèles internes du monde dans lequel ils évoluent et leur permettre de raisonner et de faire des prévisions sur l'avenir à partir des informations dont ils disposent. Ces langages sont le plus souvent associés à des systèmes logiques, c'est-à-dire à des systèmes formels disposant d'une syntaxe et d'une sémantique définie rigoureusement pour expliciter leurs inférences. Ils servent non seulement à exprimer les états mentaux des agents, et notamment à représenter le contenu de leurs buts et de leurs croyances, mais aussi à décrire le comportement des agents.

Lorsque les agents sont simples et réactifs, on peut éventuellement se passer de cette couche de langage. Mais pour les agents cognitifs ou des agents un tant soit peu complexes, l'usage de langages de ce type est presque impératif. Dans cette famille, on trouve essentiellement des langages d'IA, tels que des langages à base de règles ou de tableaux noirs, et des langages de représentation structurée des connaissances tels que les réseaux sémantiques (Lehmann 1992) ou les "frames" (Masini et al. 1989).

Type L5: Langages de formalisation et de spécification

Au niveau le plus abstrait se trouvent des langages qui servent d'une part à formaliser ce que l'on entend par systèmes multi-agents, par la notion d'interaction, par le concept d'intention, etc., et d'autre part à spécifier les conditions qui doivent être respectées lors de la modélisation et de l'implémentation de tels systèmes. Les langages de ce type, qui sont issus des mathématiques, ont donc une perspective "méta" par rapport aux autres familles de langages. On verra notamment au chapitre 4 une formalisation de la notion d'action et d'interaction dans les SMA et au chapitre 5 comment certains formalismes issus des logiques modales permettent de spécifier les conditions qui doivent être remplies pour des agents cognitifs intentionnels.

Tous ces langages sont évidemment liés: un protocole de négociations, par exemple, utilisera un langage de type L2, c'est-à-dire un langage de communication, comme support des conversations et un langage de type L3 pour décrire les mécanismes d'interprétation des messages et les fonctions d'évaluation des propositions évoquées de part et d'autre, le second interprétant les énoncés du premier. Les composants de la cognition ainsi que le mode de raisonnement des agents pourront être représentés dans des langages de type L1, l'interprète de ce langage étant généralement écrit en un langage de type L3. Enfin, le fonctionnement général du système ainsi que la structure conversationnelle employée devront respecter les spécifications écrites en L4.

Ces langages peuvent avoir des formes dissemblables: le langage de type L3 peut être C++; le langage de type 1 un langage à base de règles; le langage de type L2 peut posséder une syntaxe très particulière et celui de type L4 être un langage modal issu de la logique modale. Mais ils peuvent aussi se ressembler: si le langage de type L3 est Prolog, ceux de type L1 et L4 des langages issus de la logique des prédicats du premier ordre et celui de type L2 un langage ayant une syntaxe prédicative, l'impression générale sera de manipuler toujours le même langage. Cependant cette impression ne sera que très superficielle car toutes ces formes d'expression se distinguent très naturellement par leur sémantique, c'est-à-dire par ce qu'elles désignent. En effet, alors qu'un langage de programmation décrit les séquences d'instructions permettant aux agents d'exécuter leur comportement, un langage de représentation se réfère aux choses que l'agent peut connaître et un langage de communication désigne des actes de langage comme nous le verrons ci-dessous. Enfin, un langage de spécification représente les conditions que doivent respecter les entités présentes dans l'univers multi-agent.

1.3 Un peu d'histoire

L'approche multi-agent se situe au carrefour de nombreuses disciplines. Les deux plus importantes sont l'*intelligence artificielle distribuée* (IAD) qui a pour objet de réaliser des organisations de systèmes capables de résoudre des problèmes par le biais d'un raisonnement le plus généralement fondé sur une manipulation de symboles, et la *vie artificielle* (ou VA) qui cherche à comprendre et à modéliser des systèmes doués de vie, c'est-à-dire capable de survivre, de s'adapter et de se reproduire dans un milieu parfois hostile.

1.3.1 Les premiers âges

L'IAD a pris naissance aux Etats-Unis sous des formes multiples. Les premiers systèmes ont exploré essentiellement la relation existant entre architecture et mode de raisonnement, en dégageant deux types de contrôle:

1. Le premier s'est organisé autour d'une communication par partage d'informations et a donné lieu à l'élaboration du modèle de système à tableau noir. Les travaux de F. Hayes-Roth, L. Erman et V. Lesser (Erman et al. 1980) avec HEARSAY II ont été à la base de toutes les recherches ultérieures portant sur ce type de système. En partant d'une application de compréhension automatique de la parole, ils ont développé la métaphore du tableau noir fondée sur l'idée que la résolution de problèmes peut résulter de l'activation opportuniste de spécialistes, les KS (pour Knowledge Sources), sans qu'il soit nécessaire de définir une structure de contrôle a priori. L'activité des KS consiste à déposer, à modifier et à retirer des objets sur une zone de travail commune, le tableau noir, qui structure la modélisation du domaine d'application comme l'espace des hypothèses et des solutions. Un dispositif de contrôle gère les conflits d'accès entre les agents et organise leur fonctionnement. Nous décrivons un peu plus longuement ces architectures au chapitre 3.

Dans le même ordre d'idées, les travaux de Kornfeld et Hewitt sur ETHER (Kornfeld 1979) ont porté sur la résolution de problèmes en partant de la métaphore des communautés scientifiques. Des lutins (sprites), qui présentent beaucoup de points communs avec les KS des systèmes précédents, déposent des faits, des hypothèses et des démonstrations sur une zone commune semblable à celle d'un tableau noir. La différence résulte essentiellement de la décomposition en deux catégories de lutins, les défenseurs et les sceptiques, les premiers tentant de démontrer qu'une hypothèse est vraie, les seconds cherchant à montrer leur fausseté en exhibant des contre-exemples. Enfin, à partir d'une idée très américaine, des sponsors réglaient l'allocation de temps à allouer à chaque lutin.

Bien que l'idée générale soit très séduisante, les moyens mis en œuvre ne permettaient pas de répondre aux objectifs généraux, car il régnait une certaine confusion entre architecture et technique de coordination. On ne définit pas, en effet, une organisation capable de démontrer ou de réfuter une hypothèse à partir de considérations sur les modèles architecturaux et les modes de communication. Depuis, les travaux en IAD portant sur la coopération, la coordination d'action et la négociation se sont de plus en plus dégagés des contraintes architecturales. Ces

limites ne doivent cependant pas masquer l'influence considérable de ces systèmes sur les recherches ultérieures dans ce domaine.

2. Les travaux sur le second sont dus à Lenat et à Hewitt. Dans les années 75, D. Lenat, le futur auteur des systèmes AM et EURISKO (Lenat et Brown 1984) et l'instigateur du projet CYC (Lenat et Guha 1990), développa un système appelé PUP6 qui comprenait une foule d'idées dont la plupart furent développées ultérieurement par d'autres systèmes (Lenat 1975). PUP6 implémentait, pour peut-être la première fois, l'idée de résolution par une communauté de spécialistes, appelés "êtres" (beings), travaillant à la synthèse d'un spécialiste spécifique, CONCEPT-FORMATION, capable de résoudre une tâche par lui-même. L'utilisateur, via un spécialiste, répond aux questions qui ne peuvent être résolues par l'assemblée. Ces spécialistes ne sont pas des KS, car ils sont perpétuellement modifiés pendant l'exécution. Chaque spécialiste est composé d'un ensemble de couples attribut/valeur, semblables à ceux des "frames" et peut demander des informations aux autres spécialistes sans avoir à les connaître. Lenat n'a en fait réalisé qu'une maquette du système général BEINGS, l'application PUP6 se bornant à être capable de reconnaître une arche en utilisant le programme, à l'époque particulièrement fameux, de P.H. Winston.

Les travaux sur les acteurs de Hewitt, bien qu'initialement effectués au sein de l'intelligence artificielle, ont, comme beaucoup d'autres, rejoint celui de la programmation. Au début des années 70, après avoir conçu un système, PLANNER, dédié à la démonstration de théorèmes, il se rendit compte que la problématique du contrôle était centrale pour comprendre le déroulement d'un raisonnement. Plutôt que de considérer ce contrôle comme une suite séquentielle de choix, il s'orienta vers une vision distribuée en considérant les structures de contrôles comme des "patterns" d'envois de messages (Hewitt 1977) entre entités actives appelées *acteurs*. Il eut ainsi l'idée de considérer la réalisation d'un problème comme l'activité d'une assemblée d'experts, en considérant le développement d'un raisonnement comme une confrontation de points de vue. De ces idées naquirent toute une série de langages d'acteurs, caractérisés par leurs mode de communication par messages asynchrones bufferisés et par le passage de la continuation du calcul. Ces langages sont encore considérés comme de bonnes bases pour la réalisation de plates-formes multi-agents. Mais l'influence de ces travaux s'étendit bien au-delà de l'IAD. Par le développement des acteurs comme modèles à part entière de programmation parallèle et par ses travaux de visionnaire dans le domaine des *systèmes ouverts* (open systems) (Hewitt 1991), C. Hewitt est l'une des figures les plus brillantes du paysage de l'informatique en général et de l'intelligence artificielle en particulier.

1.3.2 L'âge classique

Dans la phase qui suivit, que l'on pourrait qualifier d'*âge classique de l'IAD*, trois systèmes eurent un impact considérable. Le premier, DVMT (Distributed Vehicule Monitoring Test), qui fut développé par l'équipe de V. Lesser à l'université du Massachusetts (Lesser et Corkill 1983), est un grand projet de recherche portant sur la perception et la reconnaissance de situations distribuées. Des capteurs envoient

des informations à des agents de traitements, implémentés sous la forme de tableaux noirs. Le problème consiste pour les agents à avoir une vue cohérente d'une situation de trafic routier et donc à identifier et à suivre des véhicules à partir des informations — redondantes, contradictoires et bruitées — qui leur arrive des capteurs. L'intérêt de cette étude porte en particulier sur le fait d'avoir examiné un grand nombre de configurations entre les capteurs et les agents de traitement, d'avoir analysé la problématique de la planification multi-agent à l'aide de plans partiels, d'avoir défini les bases des mécanismes de coopération et de négociation et d'une manière générale d'avoir concouru à la définition de l'IAD en tant que domaine d'étude à part entière. Cependant, si DVMT influença considérablement la communauté américaine, il eut moins de retentissements en Europe que MACE.

Autre “monstre sacré” de l'IAD, le système MACE développé par L. Gasser eut un impact considérable sur l'ensemble des recherches ultérieures en IAD (Gasser et al. 1987). Pour la première fois, il était expliqué clairement comment réaliser un système d'IAD et quels étaient les composants essentiels d'une plate-forme générique pour le développement de systèmes d'IAD. L. Gasser, en reliant ses travaux à ceux de Hewitt sur les acteurs, montrait non seulement qu'il était possible de réaliser un SMA à partir de la notion d'envoi de message, mais aussi que cela n'était pas suffisant, une organisation sociale ne pouvant se ramener à un simple mécanisme de communication. Il faut en plus introduire des notions telles que les représentations d'autrui et faire en sorte qu'un agent puisse raisonner sur ses compétences et ses croyances. En outre, on doit distinguer, comme le faisait MACE, la compétence effective, le “savoir-faire” directement applicable, de la connaissance qu'un agent peut avoir de sa propre compétence. On peut dire que, peu ou prou, toutes les plates-formes actuelles de développement de SMA sont des descendants directs ou indirects de MACE.

Au début des années 80, R. Smith développa un modèle de distribution des tâches et des problèmes appelé *réseau contractuel* (contract net) dont le retentissement est encore actuel. Il s'agit d'un système opportuniste d'allocation de tâches fondé sur le protocole des appels d'offre des marchés publics. En permettant à chaque agent d'être un décideur ou un exécutant et en diffusant les demandes le long d'un réseau, R. Smith résolvait le problème de la répartition des tâches sans utiliser une zone de mémoire commune, comme dans les tableaux noirs, et sans avoir la nécessité de bien identifier le receveur du message comme dans les systèmes à base d'acteurs (Smith 1979). Aujourd'hui le réseau contractuel fait partie du paysage traditionnel des études en IAD; son fonctionnement est examiné au chapitre 7.

A côté de ces travaux, on peut citer aussi ceux réalisés par Steeb, Cammarata et F. Hayes-Roth au sein de la Rand Corporation (Cammarata et al. 1983), où ils étudièrent l'impact de nombreuses organisations sur la résolution de problèmes de coopération, tels que le contrôle de trafic aérien, et surtout la foisonnante moisson de formalisation de l'action coopérative au travers des notions d'intentions de croyances et d'engagements dont nous reparlerons plus loin.

On trouvera une très bonne présentation des recherches menées en IAD jusqu'en 1988 dans (Bond et Gasser 1988).

1.3.3 L'influence de la vie artificielle

L'histoire de la vie artificielle est à la fois plus longue et plus récente. Plus longue parce que ses soubassements plongent dans l'histoire de l'informatique elle-même avec les travaux de Von Neumann sur les automates cellulaires, ceux de Grey Walter sur les robots réactifs, ceux de Warren McCulloch sur la réalisation de neurones formels ou ceux de Ross Ashby sur l'homéostasie (Dupuy 1994), mais plus récente parce que sa constitution en tant que champ d'étude ne date que de quelques années. On ne décrira donc que brièvement quelques points clés de la vie artificielle, laissant au lecteur le soin de se référer aux ouvrages suivants (Langton 1988; Langton et al. 1990; Langton 1994) (Varela et Bourguine 1992) (Meyer et Guillot 1989; Meyer et al. 1993; Meyer et Wilson 1991), s'il désire en savoir plus long sur la question, et à (Heudin 1994) pour une bonne introduction du domaine. La problématique de la vie artificielle a été lancée par C. Langton, comme "l'étude de la vie telle qu'elle pourrait être, et non de la vie telle qu'elle est", cette dernière étant réservée à la biologie classique. Il s'agit donc "d'abstraire les principes sous-jacents à l'organisation du vivant et de les implémenter dans un ordinateur afin de pouvoir les étudier et les tester" (Langton et al. 1990). Aujourd'hui, le domaine de la vie artificielle s'étend le long de plusieurs grands thèmes de recherches, dont on peut distinguer les principaux:

- L'analyse de la dynamique des phénomènes complexes à l'aide d'automates cellulaires ou d'équations différentielles non linéaires, menées par Wolfram, Kaufman et Langton.
- L'évolution de populations par utilisation d'algorithmes génétiques, dont les principaux leaders sont Wilson et Koza.
- La réalisation d'animats, c'est-à-dire de créatures autonomes capables d'agir et donc de survivre dans un environnement non entièrement spécifié, dont les représentants principaux sont Brooks, Meyer et Husbands.
- L'étude des phénomènes collectifs à partir de l'interaction d'un ensemble d'agents réactifs, avec essentiellement Steels, Deneubourg, Mataric, Ferber et Drogoul.

Ce quatrième thème est évidemment le plus en rapport avec la problématique kénétique et nous le traiterons abondamment dans cet ouvrage, mais de manière générale tous les autres contiennent des travaux dans lesquels l'idée de "collectif" est présente.

La vie artificielle, contrairement aux travaux d'IAD qui suivent le paradigme cognitiviste et considèrent que l'intelligence procède d'une manipulation de symbole, met l'accent sur le comportement, sur l'autonomie et surtout sur la problématique de la viabilité. J.P. Aubin, P. Bourguine et F. Varela ont développé l'idée de viabilité. Pour qu'un être reste vivant, il faut que son état demeure à l'intérieur de son domaine de viabilité K au cours de son évolution, c'est-à-dire, si f est sa fonction d'évolution, que:

$$s(t + 1) = f(s(t)) \text{ avec } \forall s, f(s) \subseteq K$$

Ceci ne peut se faire que pour l'une des deux raisons suivantes: soit l'entité vivante ne se trouve jamais dans une configuration pour laquelle elle n'a pas été créée, soit il lui est possible de s'adapter au fur et à mesure de son évolution et donc d'envisager des comportements viables en "devinant" l'état suivant, c'est-à-dire en étant capable d'avoir un comportement d'*abduction* (Varela et Bourguine 1992).

A la différence des approches cognitives, caractéristiques de l'IAD, celles des tenants de la vie artificielle sont minimalistes et *sub-cognitives*: elles tentent de comprendre les mécanismes de coopération et de coordination d'actions à partir de critères ne faisant pas intervenir de symboles. Les agents ont un comportement de type stimulus/réponse, leurs communications s'effectuent à l'aide de simples propagations de signaux sans signification intrinsèque et ils ne disposent généralement pas de représentations. Ceci montre qu'il était possible d'obtenir un comportement collectif complexe sans que les agents eux-mêmes ne disposent de facultés évoluées. Les travaux dans ce domaine ont ainsi donné une nouvelle vigueur à la notion de système multi-agent en permettant un dialogue stimulant entre les tenants "cognitivistes" et les partisans du "réactif", et ainsi de mieux comprendre ce qui revient de droit à chacun. Nous en ferons bien évidemment part dans cet ouvrage, ces deux approches étant traitées sur un même pied.

A ces deux domaines forts, il faut ajouter l'influence des systèmes distribués pour tout ce qui touche aux communications et à la modélisation du parallélisme et à celle de l'automatisme et de la robotique, pour tout ce qui a trait à la régulation des actions dans un univers réel.

1.3.4 Les temps modernes

Il est difficile de faire un panorama exact des tendances actuelles sur les SMA, car ce domaine se déploie dans plusieurs directions avec des options fondamentales parfois très différentes. Il ne s'agit pas ici de donner une liste exhaustive des recherches menées sur ce sujet, mais seulement de présenter quelques points de repère afin de comprendre les tendances actuelles de la recherche ainsi que certaines "écoles" de pensée qui tendent à se dessiner.

L'école IAD américaine. L'école dominante en IAD est certainement l'école américaine groupée toujours autour de Victor Lesser (Lesser et Corkill 1983) et de ses "disciples", Ed Durfee (Durfee et al. 1987) et Susan Conry (Conry et al. 1988) notamment. A côté de ce groupe, quelques francs tireurs particulièrement importants sont L. Gasser (Gasser 1991), M. Huhns (Huhns 1987), et K. Sycara (Sycara 1989). Plutôt tournée vers l'analyse de SMA cognitifs composés d'un petit nombre d'agents à tendance très "Intelligence Artificielle", l'école IAD américaine a très peu de contacts avec les chercheurs spécialisés en agents réactifs et notamment avec la communauté "vie artificielle", pourtant particulièrement féconde dans ce pays (cf. ci-dessus). En revanche, des relations très fortes se sont nouées entre l'IAD et les sciences de l'organisation, sous l'impulsion notamment de K. Sycara et L. Gasser. On trouvera une très bonne présentation de cette école de pensée (et aussi d'autres approches) et des recherches menées dans ce domaine en IAD jusqu'en 1988 dans (Bond et Gasser 1988).

La formalisation logique. A côté de cette école IAD, il existe une école très importante consacrée essentiellement à la formalisation des agents rationnels autonomes, que ces agents travaillent collectivement ou non. Ces recherches suivent le courant théorique initié par Cohen et Levesque qui formalise les intentions et les croyances à partir de logiques modales. Les tenants les plus importants actuellement, Y. Shoham (Shoham 1993) et Georgeff (Rao et Georgeff 1992), ajoutent à leur démarche théorique une dimension pratique qui les conduit à réaliser des systèmes qui implémentent leurs théories. En Europe, cette école est particulièrement illustrée par J. Galliers (Galliers 1988), E. Werner (Werner 1989), C. Castelfranchi et R. Conte (Conte et al. 1991), Wooldridge et Jennings (Wooldridge et Jennings 1994), Coelho (Corrêa et Coelho 1993), et, au Québec, par Chaib-Draa (Chaib-Draa 1989).

Actes de langages et SMA. Les liens qui existent entre les actes de langages (cf. chap. 6) et les communications dans les systèmes multi-agents sont très étroits. Si des travaux théoriques concernant les liens entre ces deux domaines se poursuivent (Moulin et Cloutier 1994) (Brassac 1992), une tendance très nette consiste à définir effectivement des protocoles de communication à l'aide de la théorie des actes de langage, comme le montre l'ensemble des travaux portant sur le standard KQML (Finin et al. 1994).

IAD et théorie des jeux. Uniquement représentée par J. Rosenschein (Zlotkin et Rosenschein 1992) et des chercheurs issus de son enseignement tels que Zlotkin, cette école, essentiellement israélienne, analyse les SMA à partir d'un point de vue de théorie des jeux. Une synthèse de leur approche peut être trouvée dans (Rosenschein et Zlotkin 1994).

Réseaux de Petri et SMA. Essentiellement française, ce courant de recherche vise à utiliser les outils théoriques des systèmes distribués et en particulier les réseaux de Petri, pour l'analyse, la conception et la validation des SMA. Elle comprend notamment les équipes de P. Estrailier (Bachatène et Estrailier 1992), J. Ferber (Ferber et Magnin 1994), S. Pinson et S. Haddad (El Fallah-Seghrouchni et Haddad 1994) et P. Azema (Vernadat et Azemat 1993).

Langages d'acteurs et SMA. Dans la même veine que les recherches précédentes, ce courant, particulièrement illustré au Japon par M. Tokoro (Maruichi et al. 1990), T. Ishida (Ishida 1989) et A. Yonezawa (Yonezawa 1990), à la Réunion par S. Giroux (Giroux 1993) (initialement au Québec) et en France par P. Carle et J. Ferber (Ferber et al. 1993), tente d'intégrer les recherches portant sur le parallélisme en général et les langages d'acteurs en particulier, avec les concepts et les objectifs des SMA.

Agents réactifs. De R. Brooks à L. Steels, en passant par J.-L. Deneubourg, J. Ferber et A. Drogoul, Y. Demazeau, P. Bourguine et J.A. Meyer, cette approche s'inscrit dans la mouvance "vie artificielle" développée ci-dessus.

Les activités scientifiques sont regroupées au sein de conférences et "workshops" internationaux: le DAI Workshop aux Etats-Unis, MAAMAW (Modeling Autonomous Agents in Multi-Agent Worlds) en Europe et MACC (Multi-Agent and Concurrent Computing) au Japon. Ces trois conférences se sont réunies pour créer une conférence internationale, ICMAS (International Conference on Multi-Agent Systems) dont la

première session s'est tenue en juin 1995 à San Francisco. Les présentations des MAAMAW ont été intégralement publiées dans des ouvrages facilement disponibles (Castelfranchi et Werner 1994; Demazeau et Müller 1990; Demazeau et Müller 1991; Werner et Demazeau 1992), ce qui n'est pas totalement le cas des conférences américaines dont deux seulement ont fait l'objet de publications (Gasser et Huhns 1989; Huhns 1987). Il existe aussi des conférences nationales dans la plupart des pays européens. Dans les pays francophones, une conférence, intitulée Journées Francophones sur l'Intelligence Artificielle Distribuée et les systèmes Multi-Agents (JFI-ADSMA), regroupe chaque année la communauté scientifique de langue française, ce qui inclut, outre la France, la Belgique, la Suisse, le Québec, des chercheurs individuels provenant d'Europe occidentale (Italie, Espagne, Portugal), de pays d'Afrique de l'Ouest (Sénégal, Cameroun, Bénin, Mali, Côte d'Ivoire) et d'Europe de l'Est (Pologne notamment) parlant français.

1.4 Domaines d'application

Les domaines d'application des systèmes multi-agents sont particulièrement riches. Nous en citerons seulement les principales directions, toute recherche d'exhaustivité aboutissant à une sclérose a priori d'un domaine de recherche en pleine évolution. On peut considérer qu'il existe cinq grandes catégories d'applications des systèmes multi-agents: la résolution de problèmes au sens large, la robotique distribuée, la simulation multi-agent, la construction de mondes hypothétiques et la conception génétique de programmes (fig. 1.5).

1.4.1 La résolution de problèmes

La résolution de problèmes au sens large concerne en fait toutes les situations dans lesquelles des agents logiciels accomplissent des tâches utiles aux êtres humains. Cette catégorie s'oppose aux applications de robotique distribuée en ce sens que les agents sont purement informatiques et n'ont pas de structure physique réelle.

La terminologie anglo-saxonne en appelant ces techniques "distributed problem solving" reste assez ambiguë. Il existe en effet deux manières de traduire cette appellation: soit par "résolution distribuée de problèmes", soit par "résolution de problèmes distribués". Ces deux traductions se rapportent à deux grandes classes d'applications. Dans la première, l'expertise globale est distribuée parmi l'ensemble des agents, chacun n'ayant qu'une compétence restreinte par rapport à l'ensemble du problème à résoudre. On appelle aussi cette forme "coopération de spécialistes". Dans la seconde au contraire, c'est le problème qui est lui-même distribué, les agents pouvant avoir des compétences semblables. Enfin, il existe une autre classe de problèmes dans laquelle on utilise des agents en interaction pour résoudre des problèmes classiques, et je les présenterai sous l'appellation de résolution par coordination (cf. chap. 8).

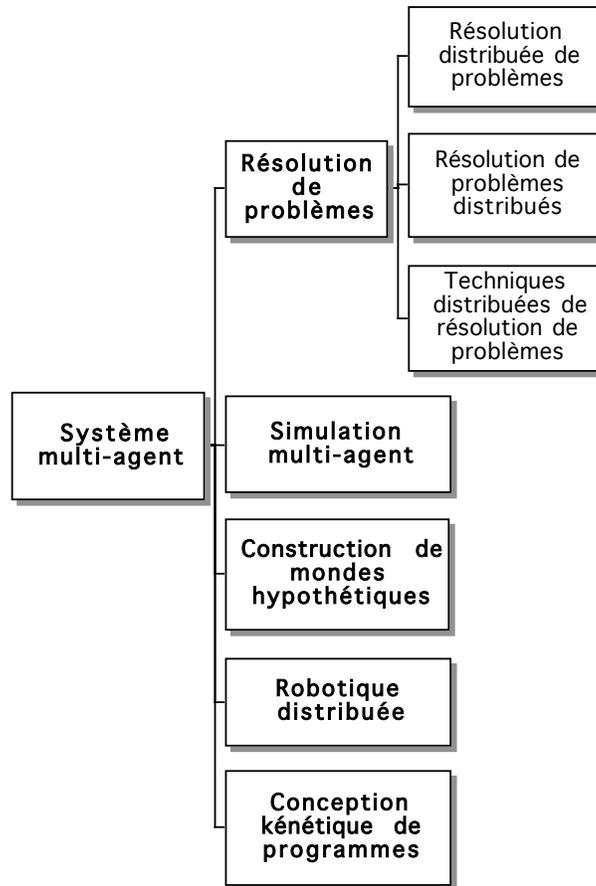


Figure 1.5: Une classification des différents types d'application des systèmes multi-agents

La résolution distribuée de problèmes

Toutes les applications relevant de la résolution distribuée de problèmes supposent qu'il est possible d'effectuer une tâche complexe en faisant appel à un ensemble de spécialistes disposant de compétences complémentaires et donc, lorsque c'est l'expertise ou le mode de résolution qui sont distribués, sans que le domaine le soit. Lorsque l'expertise est en effet si vaste et complexe qu'elle ne peut appartenir qu'à une seule personne, il faut faire appel à plusieurs spécialistes qui doivent travailler ensemble à la poursuite d'un objectif commun.

Comme nous l'avons dit précédemment, la construction d'une voiture de course nécessite plusieurs spécialistes: l'un connaîtra mieux les moteurs, un autre s'occupera du choix et du test des pneumatiques, un troisième s'intéressera au châssis et aux suspensions, un quatrième saura trouver de bons carburants et un dernier se chargera de la gestion de la course et de la définition des interfaces avec le pilote. Ces spécialistes coopèrent entre eux pour résoudre un problème général, tel qu'un diagnostic médical, la conception d'un produit industriel (Iffenecker et Ferber 1992), l'acquisition de connaissances, le diagnostic de réseaux (Jennings et al. 1995), la reconnaissance de formes (Demazeau et al. 1994), la compréhension de

langage naturel (Sabah 1990), le système de contrôle et de surveillance d'un réseau de télécommunications (Weihmayer et Brandau 1990), etc.

Le système CONDOR conçu par C. Iffenecker à la société Télémécanique pour l'aide à la conception de produits électromécaniques constitue un bon exemple de SMA tourné vers la coopération de spécialistes (Iffenecker 1992). Des spécialistes en cahier des charges, en conception, en assemblage, en matériaux, en planification, en marketing et bien d'autres, qui prennent part à la conception d'un produit industriel, sont représentés sous la forme d'un ensemble d'agents dans le système CONDOR. Ces derniers, qui sont bien entendu cognitifs, sont conçus autour d'une architecture de tableau noir (cf. chap. 3) comprenant trois tableaux: celui des connaissances du domaine, celui du contrôle et celui des communications.

Tous ces agents possèdent leur propre expertise et interviennent à différents moments lors de la réalisation du produit. La particularité de ce système réside dans son organisation générale qui prend elle-même la forme d'une architecture de tableau noir. On y représente ainsi des groupes de travail, tels que le groupe "qualification", le groupe "décisionnel" ou le groupe "bureau d'étude", qui travaillent à partir de leurs propres protocoles d'automatisation de flux d'information, à la manière des systèmes de "work flow" actuels.

Le système KBS-SHIP, issu d'un programme ESPRIT, est conçu comme une plateforme intégrant plusieurs experts travaillant en commun. Ce système est dédié à la surveillance et à la maintenance des équipements d'un navire de commerce. Il intègre plusieurs systèmes experts distincts (pilotage, chargement du fret, maintenance des équipements électroniques, diagnostic de pannes, etc.) qui travaillent au sein d'une architecture multi-agent, sous le contrôle d'un expert chargé de la gestion des communications via un réseau Ethernet et de la résolution des conflits entre agents.

Dans le cadre du contrôle de processus industriel, le système Flavors Paint Shop, utilisé pour peindre des camions, montre assez simplement l'intérêt des systèmes multi-agents pour l'industrie (Morley 1993) (l'intérêt de ce système a été décrit dans (Parunak 1993)) et leur compétitivité par rapport à des approches centralisées. A la sortie d'une chaîne de montage, des camions doivent être peints d'une couleur particulière en fonction des désirs des clients. Le nombre des postes de peinture est inférieur au nombre des couleurs disponibles. De ce fait, il est parfois nécessaire de changer la couleur d'un poste de peinture, ce qui consomme à la fois beaucoup de temps et de peinture.

Initialement, la gestion de l'optimisation des couleurs utilisait un programme classique dont la mise au point et la maintenance coûtaient très cher. De plus, il nécessitait une planification rigoureuse et avait du mal à prendre en compte les pannes éventuelles des postes. Le système a alors été réécrit sous la forme d'un système multi-agent dans lequel chaque poste de peinture a été représenté sous la forme d'un agent. Lorsqu'un poste est libre, il accepte un nouveau camion à partir d'une file d'attente contenant une centaine de camions en attente d'être peints. Cette sélection s'effectue à partir des trois règles suivantes:

1. Prendre le premier camion de la file qui réclame la même couleur que celle qui est disponible actuellement.

2. S'il n'y a pas de camion de cette couleur, prendre le camion le plus prioritaire et affecter au poste la couleur du camion.
3. S'il n'y a pas de camion prioritaire, prendre le camion suivant dans la file et affecter au poste la couleur du camion.

Malgré la simplicité de ce système, il a réduit de moitié les opérations de changement de peinture par rapport au système centralisé, ce qui a conduit à une économie de plus d'un million de dollar par an. De plus, le système peut prendre en compte sans problème les pannes de postes de peinture.

Résolution distribuée de problèmes distribués

Si le domaine est lui aussi distribué, on parle alors de *résolution (distribuée) de problèmes distribués*. Il s'agit essentiellement d'applications telles que l'analyse, l'identification, le diagnostic et la commande de systèmes physiquement répartis, pour lesquelles il est difficile d'avoir une vision totalement centralisée. Par exemple, s'il s'agit de contrôler un réseau de communication ou d'énergie, le domaine, qui est représenté par le réseau lui-même, constitue un système réparti qu'il s'agit de surveiller, voire de superviser, en décentralisant au maximum les tâches de surveillance au sein des nœuds du réseau. La perception distribuée, telle qu'elle a été initiée par DVMT, ou la surveillance de réseaux d'énergie ou de télécommunications dans lesquels la supervision est répartie sur chacun des nœuds constitue un bon exemple de résolution de problèmes distribués.

Le système IDEAL, conçu par l'Onera et Alcatel-Alsthom, est un exemple caractéristique de SMA dédié à la surveillance et au diagnostic d'un réseau de télécommunication. Il comprend trois sortes d'agents: des superviseurs, chargés de localiser et de diagnostiquer des pannes, des agents de suivi, qui doivent maintenir la cohérence entre l'état réel du réseau et la vision qu'en ont les agents, et des opérateurs de maintenance chargés d'exécuter des tests et de réparer des éléments du réseau. Les agents, qui présentent une architecture de tableau noir (chap. 3), comportent cinq modules: un module de communication qui gère les protocoles de communication, un module expert qui contient les connaissances concernant la supervision du réseau, un module de coopération qui gère les tables d'acointances (chap. 5), la représentation de soi et des modèles de dialogue (chap. 6), un module pour l'interface graphique permettant à un utilisateur de suivre le fonctionnement de l'agent et d'intervenir si besoin, et un module de contrôle qui gère l'ensemble des activités de l'agent.

Résolution par coordination

Les agents peuvent aussi servir d'une manière beaucoup plus élémentaire à résoudre des problèmes au sens classique du terme, c'est-à-dire à tenter de trouver une solution à quelque chose dont l'énoncé est bien posé et dont l'ensemble des informations est entièrement disponible, comme, par exemple, trouver une affectation de tâches pour une machine-outil ou définir un emploi du temps pour un collège, donner la suite des actions à accomplir pour sortir d'un labyrinthe ou pour faire partir une

fusée, assembler des cubes ou des composants mécaniques, résoudre un casse-tête ou démontrer un théorème.

Dans ce cas, le domaine n'est pas distribué et l'expertise ne l'est pas non plus. Et pourtant l'approche multi-agent peut apporter un mode de raisonnement nouveau en décomposant le problème de manière totalement différente. Par exemple, s'il s'agit d'empiler des cubes ou d'assembler des pièces mécaniques, on peut considérer que les cubes ou les pièces sont des agents qui doivent satisfaire des buts précis donnés par le plan imposé par le concepteur et que les liaisons sont des contraintes que les agents doivent respecter. C'est exactement le type de problème que sait résoudre de manière très efficace l'éco-résolution décrite au chapitre 8.

De même, s'il s'agit de résoudre un problème à base de contraintes, on peut considérer, comme l'ont fait K. Ghedira (Ghedira 1993) et K. Sycara (Liu et Sycara 1993), que les tâches et les ressources sont des agents qui tentent de satisfaire leurs objectifs propres: une tâche désire être placée et une ressource souhaite être utilisée au mieux, sans avoir trop ni trop peu de travail. En adoptant une technique semblable à celle de l'éco-résolution, et en lui associant le principe du recuit simulé, K. Ghedira est capable de résoudre des problèmes de contraintes très rapidement et en limitant les modifications lors d'un changement de donnée.

Ces approches abordent les problèmes à traiter avec un tel changement d'attitude que même des spécialistes en résolution de problèmes sont initialement totalement dérouterés par elles. De plus, elles sont souvent peu formalisées et il est donc difficile de connaître a priori leur validité: sont-elles décidables, semi-décidables, incomplètes? Même si l'on dispose de résultats partiels, dans bien des cas, il est difficile de trancher. Leur avantage fondamental réside dans leur rapidité d'exécution. Fondées sur des considérations très différentes des approches classiques par exploration d'espace d'état, elles utilisent beaucoup mieux certaines caractéristiques structurelles des problèmes, ce qui les rend souvent très efficaces. Elles ont été utilisées notamment dans le domaine de la robotique distribuée (cf. section 1.4.4). Nous développerons certaines de ces techniques au chapitre 8.

1.4.2 La simulation multi-agent

La simulation est une branche très active de l'informatique qui consiste à analyser les propriétés de modèles théoriques du monde environnant. La physique, la chimie, la biologie, l'écologie, la géographie et les sciences sociales notamment font un grand usage des simulations pour essayer d'expliquer et de prévoir les phénomènes naturels. Pour cela, les chercheurs de ces différentes disciplines construisent des modèles de la réalité, puis testent leur validité en les faisant "tourner" sur des ordinateurs. Généralement, ces modèles sont donnés sous la forme de relations mathématiques entre des variables représentant des grandeurs physiques mesurables dans la réalité. Les modèles les plus utilisés sont les équations différentielles, les matrices de transitions, etc. Elles reposent sur la définition d'une relation de cause à effet entre des variables d'entrées et des variables de sorties. L'un des exemples fameux, utilisé en écologie, est celui du modèle mathématique de dynamique des populations introduit par Lotka et Volterra (Volterra 1926) qui exprime le taux de croissance de populations de proies

et de prédateurs occupant un même territoire.

$$\frac{dN_1}{dt} = r_1 N_1 - P N_1 N_2 \quad \frac{dN_2}{dt} = a P N_1 N_2 - d_2 N_2$$

où P est le coefficient de prédation, N_1 et N_2 sont les effectifs des proies et des prédateurs, a est l'efficacité avec laquelle les prédateurs convertissent la nourriture en descendants, r_1 détermine la fécondité des proies et d_2 le taux de mortalité des prédateurs. Bien qu'ayant permis de nombreuses avancées sur le plan théorique et pratique (Pavé 1994), ces modèles et les techniques de simulation numérique associées présentent néanmoins certains problèmes, dont on peut donner ici les principaux :

- *Etanchéité du niveau d'analyse.* Les modèles mathématiques relient des paramètres qui se situent tous au même niveau d'analyse. Par exemple, il n'est pas possible de relier la taille d'un effectif aux prises de décision effectuées par les individus. On peut dire que ces niveaux d'analyse sont "étanches" puisqu'il n'est pas possible de faire correspondre des comportements effectués au niveau "micro" à des variables globales mesurées au niveau "macro".
- *Complexité et réalisme des paramètres.* Pour être utilisables et correspondre à la réalité, ces équations comportent souvent un grand nombre de paramètres difficiles à estimer et manquent surtout de réalisme. Par exemple, dans le modèle Proie-Prédateur de l'équation précédente, le coefficient a qui indique l'efficacité avec laquelle les prédateurs transforment la nourriture en descendance semble bien appauvrissant quand on pense à l'ensemble des comportements complexes (hiérarchies et dominances, stratégies sexuelles, utilisation du territoire et construction d'abris) qui peuvent avoir un impact direct sur leur fécondité.
- *Difficulté à modéliser l'action.* La critique peut-être majeure que l'on pourrait faire aux modèles mathématiques actuels porte sur leur difficulté (voire leur impossibilité) à prendre en compte les actions des individus, et donc les modifications effectives de l'environnement qui découlent de leur comportement. La plupart des phénomènes collectifs (que l'on pense par exemple aux phénomènes boursiers) sont le résultat d'un ensemble de prises de décisions individuelles qui tiennent compte des comportements des autres acteurs du système. En ne considérant les actions que par leurs conséquences mesurables au niveau global ou par leur probabilité d'apparition, il s'avère difficile d'expliquer les phénomènes émergents dus à l'interaction de ces comportements individuels, en particulier tous ceux portant sur la coopération intra et inter-spécifique.
- *La carence qualitative.* Par leur nature même, les simulations numériques ne peuvent considérer que des paramètres quantitatifs et apparaissent désarmées devant la multitude d'informations qualitatives recueillies par les chercheurs de terrain, les naturalistes notamment. Devant l'intrication des jeux de régulation qui s'établissent entre les habitudes comportementales des espèces et leurs

mécanismes de reproduction et d'occupation de l'espace, les modèles mathématiques abandonnent et laissent ainsi de nombreux domaines de recherche dans l'incapacité de bénéficier des avantages que procure la simulation.

Les systèmes multi-agents apportent une solution radicalement nouvelle au concept même de modèle et de simulation dans les sciences de l'environnement, en offrant la possibilité de représenter directement les individus, leurs comportements et leurs interactions. La simulation multi-agent est fondée sur l'idée qu'il est possible de représenter sous forme informatique le comportement des entités qui agissent dans le monde et qu'il est ainsi possible de représenter un phénomène comme le fruit des interactions d'un ensemble d'agents disposant de leur propre autonomie opératoire. Par exemple, dans un modèle multi-agent de population, on représentera directement les individus sous la forme d'agents et la quantité d'individus d'une espèce donnée sera le résultat des confrontations (coopération, lutte, reproduction) des comportements de tous les individus représentés dans le système.

L'intérêt de ces simulations est de pouvoir considérer aussi bien des paramètres quantitatifs (c'est-à-dire des paramètres numériques) que qualitatifs (des comportements individuels faisant éventuellement appel à des raisonnements stratégiques). On utilise aussi parfois le terme de *modèles micro-analytiques* (Collins et Jefferson 1991) ou de *simulation individu-centrée* pour bien indiquer que les représentations se situent au niveau de l'individu, ces derniers interagissant aussi bien entre eux qu'avec l'environnement simulé. Ce sont des constructions de "micro-mondes artificiels", dont il est possible de contrôler toutes les caractéristiques et de reproduire des séries d'expérimentations comme s'il s'agissait de situations de laboratoire. On se trouve donc en présence d'une modélisation analogique, semblable aux maquettes de format réduit utilisées en aéronautique ou en construction navale, la différence essentielle provenant du fait que le modèle réduit est exprimé sous la forme d'une entité informatique et non d'une structure physique.

De ce fait, l'utilisateur d'un tel simulateur a un rôle actif. Il emploie un système multi-agent comme s'il s'agissait d'un laboratoire miniature, déplaçant des individus, changeant leur comportement et modifiant les conditions environnementales. Chaque agent est évidemment "marqué" comme pourrait l'être un être naturel, mais ce marquage est évidemment plus facile, puisque l'individu peut être suivi à tout moment dans son évolution et avec le degré de finesse désiré. On exploite alors les capacités des ordinateurs pour traiter les données obtenues, les agréger et les exploiter à l'aide de techniques statistiques afin de vérifier les hypothèses émises. Ainsi, à la différence des approches classiques, la simulation multi-agent ne se réduit pas à l'implémentation d'un modèle puis à l'analyse de la réponse de ce modèle en fonction des paramètres d'entrées, mais participe au processus de recherche de modèles.

Les principales qualités des modélisations multi-agents sont leur capacité d'intégration et leur flexibilité. En effet, il est possible d'intégrer dans la même modélisation des variables quantitatives, des équations différentielles, et des comportements fondés sur des règles symboliques. Il est aussi très facile d'intégrer des modifications, chaque enrichissement du modèle étant réalisé par l'adjonction de règles comportementales qui agissent au niveau de l'individu. De plus, les individus étant toujours distingués

les uns des autres, il est possible d'ajouter de nouveaux types d'agents disposant de leur propre modèle de comportement, qui viennent interagir avec les agents déjà définis. Par exemple, dans une modélisation forestière, on peut introduire de nouvelles espèces animales ou végétales et analyser leurs interactions avec celles déjà modélisées.

Enfin, les systèmes multi-agents permettent la modélisation de situations complexes dont les structures globales émergent des interactions entre individus, c'est-à-dire de faire surgir des structures du niveau macro à partir de modélisations du niveau micro, brisant ainsi la barrière des niveaux, si criante dans les modélisations classiques. Nous présentons ici deux systèmes caractéristiques de simulation multi-agent qui sont le résultat de collaboration entre informaticiens et chercheurs écologues. Il existe bien évidemment un grand nombre d'autres applications de simulation comportementale, mais ces deux-ci, plus le système MANTA développé au chapitre 7, donnent un aperçu des domaines variés dans lesquels les SMA peuvent être employés.

SIMDELTA

Le simulateur SIMDELTA a été utilisé pour synthétiser les connaissances d'un ensemble de spécialistes (halieutes, écologistes, pédologues, biologistes, anthropologues, etc.) ayant travaillé pendant plusieurs années sur l'étude du système de pêche du delta central du Niger au Mali. Il s'agissait de pouvoir modéliser des informations tant quantitatives (concernant l'évolution des crues du Niger par exemple) que qualitatives (telles que les techniques de pêches).

Ce simulateur, réalisé par F. Bousquet et C. Cambier (Cambier et al. 1992), permet de simuler en même temps la dynamique de population des poissons, en tenant compte des nombreux facteurs biologiques et topologiques qui peuvent affecter son évolution, et la prise de décision des pêcheurs. Les agents sont ici les bancs de poissons et les pêcheurs. La technique employée met en présence trois types d'agents (fig. 1.6): les biotopes qui représentent des portions d'environnement, les poissons qui ont un comportement plutôt réactif et les pêcheurs qui se comportent comme des agents cognitifs. L'environnement est constitué de biotopes dont les connexions peuvent être modifiées dynamiquement lorsque le niveau de l'eau change. Pour chaque biotope, une fonction de ressource indique la quantité de nourriture disponible pour une population de poissons en fonction du temps. Les bancs de poissons sont des agents qui représentent des ensembles de poissons et dont les paramètres (taille et nombre d'œufs, processus de migration, régime, etc.) caractérisent leur stratégie d'adaptation.

Chaque pêcheur est représenté sous la forme d'un agent cognitif et son comportement est décrit par un système à base de connaissances composé d'une base de données qui contient ses croyances et sa mémoire de pêcheur et d'un système de règles qui représente sa stratégie cognitive pour exploiter les biotopes.

Deux séries d'expérimentations ont été réalisées. La première a porté sur l'étude de la dynamique de population des poissons en fonction d'un effort de pêche de plus en plus grand. La simulation de la dynamique de ces poissons est fondée

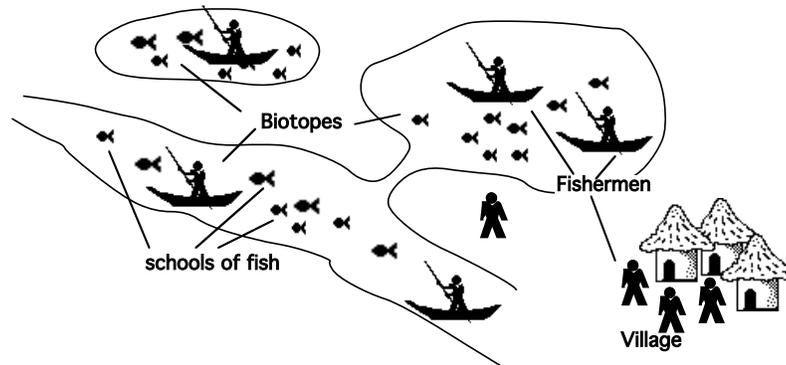


Figure 1.6: Le monde artificiel de SIMDELTA

sur le comportement des poissons d'eau douce sahéliens et, en particulier, sur les connaissances des biologistes concernant la reproduction, la croissance, la migration et la mortalité de ces poissons. Cette simulation a permis de reproduire la courbe en trois phases caractéristiques de l'évolution d'un système de pêche limité et surtout de faire apparaître des variations portant sur la composition spécifique et la taille des poissons.

La seconde a eu pour objet de modéliser des pêcheurs qui prennent des décisions et agissent sur la réserve renouvelable. Cette modélisation a montré l'importance du mécanisme de décision sur la dynamique des poissons. Des stratégies fondées sur des critères de rationalité économique sont, notamment, globalement moins performantes à terme que celles qui intègrent des règles d'accès à l'espace (Bousquet et al. 1993; Cambier 1994).

SIMPOP

Si les entités mobiles peuvent naturellement profiter des avantages des simulations multi-agents, il en va de même pour les agents fixes. L'exemple du système SIMPOP en constitue une bonne illustration (Bura et al. 1993). Il s'agit ici de modéliser la dynamique de l'évolution d'un système de villes et, plus particulièrement, la genèse, le développement et la concentration des fonctions urbaines à différents niveaux durant une longue période de temps (environ 2000 ans).

L'environnement est représenté par un ensemble de "places" de tailles et de formes variées (essentiellement des carrés et des hexagones). Elles sont caractérisées par leur nature (plaines, montagnes, mer, marais), leurs ressources naturelles (agriculture, pêche, minerai) ainsi que par des éléments divers tels que la présence d'une voie de communication (rivière, route, etc.). Les ressources correspondent au potentiel qu'une population peut exploiter, la productivité dépendant de facteurs tels que les possibilités techniques ou l'activité d'un peuplement voisin.

Chaque place comporte un agent "peuplement" que l'on appelle communément une ville (en fait ces "villes" peuvent aller d'un simple hameau à une grande métropole). Les villes sont caractérisées par la taille de leur population, leur richesse économique et les fonctions qu'elles possèdent (agriculture, économie, industrie, ad-

ministration). Le comportement d'une ville est donné par la somme des comportements de ses habitants, ces derniers étant représentés par des fonctions économiques correspondant aux principaux groupes sociaux. Par exemple, les hameaux voient la majeure partie de leur population associée à la fonction agricole.

Les villes sont donc les principaux agents du système et, du fait qu'elles ne bougent pas, les interactions s'effectuent essentiellement par des transferts de biens, de monnaies, de services et d'habitants qui s'expriment sous la forme de mécanismes d'offres et de demandes entre villes. De ce fait, certaines villes tendent à grossir et de nombreux phénomènes locaux viennent renforcer les différences apparaissant entre elles de manière à former ce que l'on nomme une "hiérarchie" de villes, portant sur leur taille et leur richesse, comme le montre la figure 1.7.

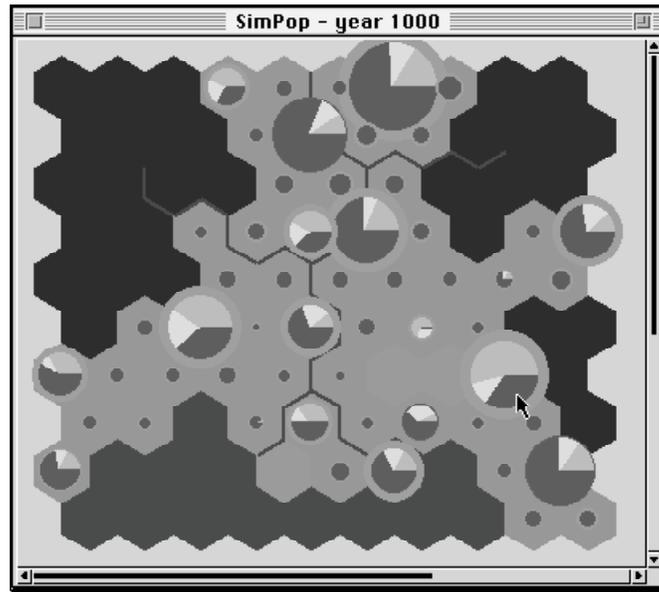


Figure 1.7: L'état d'un peuplement après mille années simulées dans SIMPOP

1.4.3 La construction de mondes synthétiques

Bien qu'il ne s'agisse pas d'applications à proprement parler, puisqu'elles ne permettent pas de résoudre un problème concret, n'utilisent pas d'agents physiques et ne simulent aucun monde réel, la construction de mondes synthétiques tient une large part dans les recherches portant sur les systèmes multi-agents parce qu'elle permet d'analyser certains mécanismes d'interaction de manière plus fine que ne pourrait le faire une véritable application. Par exemple, l'analyse de protocole de coopération ou la compréhension de l'influence du comportement sur la régulation d'une société sont souvent plus parlant sur des mondes "virtuels" que sur des applications immédiatement rentables.

De plus, du fait de la diversité des systèmes multi-agents et du nombre des domaines dans lesquels ils peuvent être utilisés, il s'avère très important de définir des situations montrant les problèmes traités et les difficultés rencontrées par une

approche kénétique. Certaines de ces situations sont des *situations-jouets*, c'est-à-dire des situations dépouillées à l'extrême de toutes les aspérités de la réalité mais qui, du fait de leur austérité, offrent les conditions idéales à une bonne compréhension des mécanismes mis en œuvre. D'autres correspondent à des situations réelles, même si leur expression est souvent simplifiée pour que l'on puisse saisir les enjeux et l'emploi des systèmes multi-agents dans des applications grandeur nature.

Un exemple caractéristique: la poursuite

L'un des exemples les plus connus est celui de la poursuite (que l'on appelle aussi le problème des proies et des prédateurs⁶). Ce problème vient d'un jeu lui-même tiré d'une histoire, *Octobre Rouge*, dans laquelle un sous-marin soviétique doit essayer d'échapper à des torpilleurs américains. Evidemment, la situation qui est analysée dans le jeu est un peu plus simple. Les agents, c'est-à-dire les proies (sous-marins ou herbivores) comme les prédateurs (torpilleurs ou carnassiers), se déplacent sur un espace représenté sous la forme d'une grille. Le but de ce jeu est de parvenir à ce que les prédateurs capturent les proies en l'entourant, comme le montre la figure 1.8.

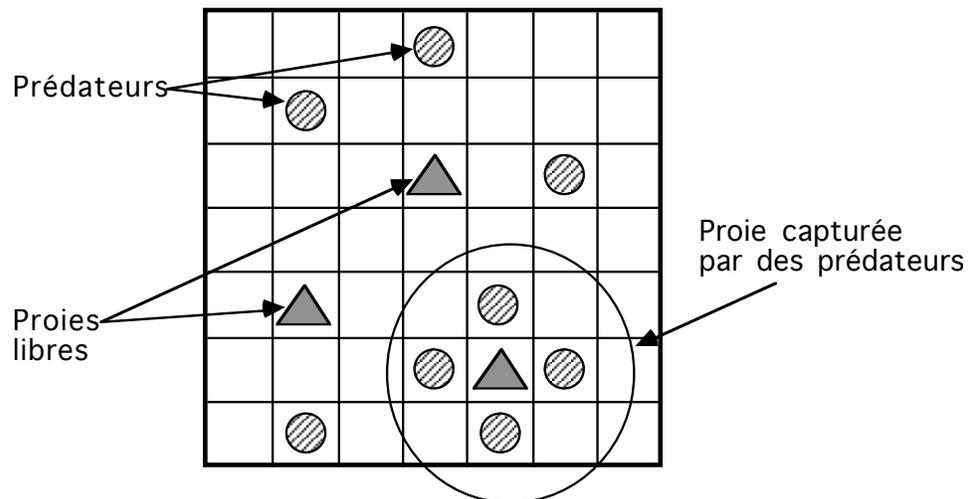


Figure 1.8: Le jeu de la poursuite: les prédateurs doivent entourer des proies pour les capturer.

On pose les hypothèses suivantes:

1. Les dimensions de l'environnement (c'est-à-dire de la grille) sont finies.
2. Les prédateurs et les proies se déplacent à des vitesses fixes, et généralement la même.

⁶Il existe un problème du même nom en écologie des populations, comme nous l'avons vu à la section précédente. Mais bien que son appellation soit semblable et que, dans les deux cas, il s'agisse de prédateurs et de proies, la problématique est totalement différente: en écologie théorique, on s'intéresse à l'évolution de la démographie des agents, alors que, dans la version de la poursuite, on étudie les mécanismes de coopération des prédateurs pour entourer les proies.

3. Les proies évoluent de manière aléatoire en effectuant ce que l'on appelle un mouvement brownien, c'est-à-dire qu'à chaque moment elles choisissent aléatoirement une position.
4. Les prédateurs peuvent utiliser les coins et les bords pour bloquer une proie.
5. Les prédateurs ont une perception limitée du monde qui les entoure, ce qui signifie qu'ils ne peuvent voir la proie que si elle se trouve dans des cases à une distance inférieure à leur champ de perception.

Evidemment, il existe de nombreuses variantes à ce jeu: il suffit de rendre le monde non borné, par exemple en repliant les bords de manière qu'un agent sortant du damier à l'est rentre par l'ouest, ou de faire en sorte que les vitesses de déplacement entre les proies et les prédateurs soient différentes, ou encore que la proie se dirige de manière moins aléatoire, en cherchant à rejoindre une case déterminée par exemple, pour que les stratégies des prédateurs soient totalement modifiées. Enfin, le nombre respectif des proies et des prédateurs et évidemment la qualité de perception des prédateurs sont aussi des paramètres importants pour le succès de la capture (Benda et al. 1986) (Gasser et al. 1989) (Bouron 1992).

Le problème consiste à coordonner les actions des prédateurs afin qu'ils puissent entourer les proies le plus rapidement possible, une proie entourée étant considérée comme morte. L'intérêt de ce problème est d'être à la fois très bien défini et de laisser cependant ouvert tous les modes de coopération possibles. En effet, de nombreuses stratégies peuvent être mises en œuvre et cet exemple illustre bien les différences d'approches qui existent entre systèmes cognitifs et systèmes réactifs.

Les stratégies cognitives procèdent généralement d'une analyse descendante en définissant les différentes fonctions que doit remplir le système: détection des proies, élaboration des équipes de chasse, allocation des rôles (prendre la proie par le nord, l'ouest, l'est ou le sud), réorganisation des équipes si le nombre des chasseurs est mal distribué (deux équipes de trois chasseurs par exemple), etc. Toutes ces fonctions sont bien identifiées et implémentées dans les agents sous forme de comportements adaptés nécessitant un système de communication permettant le dialogue et la prise de décision distribuée. On suppose alors que les agents ont des buts et qu'ils agissent rationnellement par rapport à ces buts en désignant, si nécessaire, un agent "leader" pour organiser la répartition du travail et coordonner les actions comme nous le verrons au chapitre 8.

En revanche, la réalisation d'un système réactif pour résoudre ce problème procède d'une tout autre démarche. On supposera que les proies émettent un signal dont l'intensité décroît proportionnellement à la distance et qui joue le rôle d'attracteur pour les chasseurs. De ce fait, plus un agent se trouve près d'une proie, plus le signal reçu sera fort. Les chasseurs se trouvant près d'une proie P auront donc plus tendance à se diriger vers P que vers toute autre proie dont ils pourraient percevoir les émanations et dont ils sont plus éloignés. De plus, afin que les chasseurs ne se retrouvent pas tous au même endroit et qu'ils entourent donc effectivement la proie, on fait émettre par chaque chasseur un signal servant de répulseur pour les autres chasseurs. Ainsi, chaque chasseur est à la fois attiré par les proies et

(faiblement) repoussé par les autres chasseurs (fig. 1.9). La capture des proies émerge ainsi de la combinaison de ces différentes réactions aux stimuli envoyés par les proies et les chasseurs. On peut dire que, si la coopération induite par les agents cognitifs est

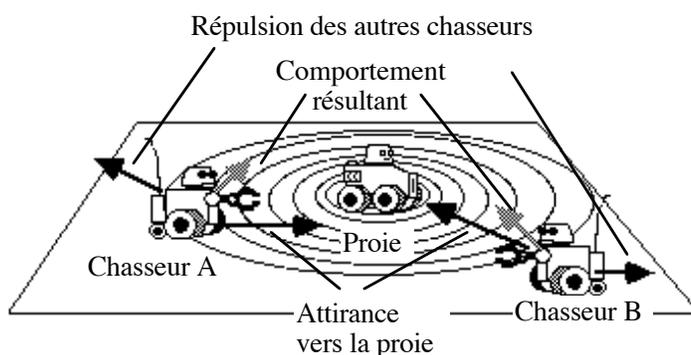


Figure 1.9: Les chasseurs sont attirés par les proies et repoussés par les autres chasseurs.

de type *intentionnelle*, puisqu'ils disposent de capacités de dialogue et d'allocation de tâches explicites, les agents réactifs introduisent un autre type de coopération que l'on pourra qualifier de *réactive* et qui n'apparaît que comme un simple effet de bord du comportement individualiste de chaque agent.

L'animat: moitié robot, moitié animal

Ces mondes hypothétiques reposent souvent sur deux métaphores quasiment interchangeables: celle du robot et celle de l'animal. Dans la première on suppose que les agents sont des robots qui accomplissent des actions dans un monde virtuel. Mais ces robots n'ont aucune réalité physique et l'appellation robot est donc trompeuse. De plus, on leur donne la possibilité d'accomplir des actions, telles que de se reproduire avec d'autres robots, que l'on serait bien en mal de demander à de véritables êtres de métal et d'électronique! D'un autre côté, appeler ces créatures des animaux n'est pas plus réaliste que d'appeler "humain" un personnage de jeu d'aventure, puisque ces prétendus animaux n'ont que des rapports lointains avec ceux que l'on rencontre dans la nature. C'est pourquoi, J.-A. Meyer et S. Wilson ont inventé le terme *animat* par contraction des mots "animal" et "artefact" (Meyer et Guillot 1989), pour bien indiquer à la fois que ces agents descendent métaphoriquement des animaux et qu'ils n'ont qu'une existence virtuelle, celle que veut bien leur donner l'imagination de leur concepteur (et de ses collègues) au regard des traces visuelles de quelques processus informatiques.

Il existe donc un grand nombre de situations qui font intervenir un ensemble d'animats dans un environnement représenté généralement sous la forme d'un ensemble de cases et qui n'ont comme tout objectif que de satisfaire leurs besoins: prendre de la nourriture, boire, se reproduire, comme le montre la figure 1.10 qui représente un instantané typique d'un tel monde. Nous reverrons, en particulier dans le chapitre 5, des exemples de comportements d'animats explorant leur espace

afin d'en tirer les éléments nutritifs nécessaires à leur survie tout en évitant les prédateurs.

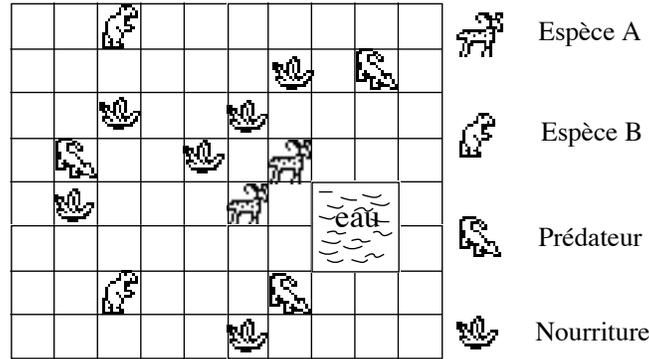


Figure 1.10: Un monde caractéristique de l'approche animat: un ensemble d'individus d'espèces différentes cherchent à survivre

Des robots explorateurs à l'assaut de Mars

Lorsque la métaphore met plus l'accent sur le "robot" que sur "l'animal", il est possible d'inventer toute une gamme de comportements qui diffèrent de la survie d'un animal, tel que l'action de rapporter du minéral.

Au début tout commença par une application réelle: lorsque la NASA décida d'investir dans la réalisation d'un projet spatial conduisant à l'exploration de la planète Mars, il fut clair que cette exploration ne pourrait se faire dans un premier temps qu'à l'aide de robots. Mais du fait de la distance de Mars à la Terre, les liaisons radios durent au moins 3 à 5 mn⁷, un délai bien trop grand pour pouvoir télécommander les robots depuis la Terre. Les robots devaient donc être autonomes, c'est-à-dire capables de se débrouiller par leurs propres moyens dans un univers peut-être hostile. La NASA décida alors de financer des recherches permettant de concevoir et de construire de tels robots autonomes.

C'est là que Brooks entra en jeu et lança l'idée d'une invasion de Mars par un grand nombre de petits robots très simples dans leurs capacités et bon marchés (Brooks 1990). Dans ce cas, la coordination des actions entre les robots devient cruciale et peut donc être envisagée selon une perspective multi-agent. J.-L. Deneubourg (Deneubourg et al. 1986) et L. Steels (Steels 1989) avancèrent alors, chacun de son côté, l'idée qu'il était possible de réaliser efficacement des tâches avec des robots dont le comportement ressemblait à celui des fourmis. En les simulant d'abord, puis en construisant de véritables robots, ils montrèrent qu'il était effectivement possible de réaliser des groupes de robots réactifs dont le comportement était calqué sur celui du comportement animal. La collaboration entre L. Steels et D. McFarland, éthologue de renom (McFarland 1990), comme celle que nous avons entreprise

⁷Cela ne prend en compte que le trajet dans un sens. Pour réagir à une situation périlleuse depuis la Terre, il faut au moins le double de temps entre le moment où la situation est perçue par le robot et celui où il reçoit ses commandes.

A. Drogoul et moi-même avec des biologistes spécialistes du comportement animal (Drogoul et al. 1992), est caractéristique de cette fusion qui peut exister entre les points de vue des spécialistes des sciences de la vie et ceux des sciences de l'artificiel, informatique et robotique notamment. Depuis, de nombreux laboratoires relevant de ces deux dernières sciences ont relevé le défi et s'emploient à la définition de sociétés de robots autonomes efficaces dans des conditions semblables à celles d'une exploration martienne.

Le problème de la récupération d'échantillons de minerai a particulièrement capté l'attention des chercheurs en définissant une situation-jouet immédiatement compréhensible et sur laquelle il est possible de comparer différentes organisations coopératives. Le problème général peut être décrit ainsi: on suppose qu'il existe une base fixe (ou très peu mobile) à partir de laquelle plusieurs robots mobiles doivent explorer un espace a priori inconnu (on suppose qu'il n'existe pas de cartes du terrain) pour trouver, récupérer et transporter le minerai découvert jusqu'à la base. Le problème consiste donc à définir la structure et le comportement des robots ainsi que l'organisation de la société de manière à obtenir les meilleurs résultats possibles, en termes de temps mis par l'ensemble des robots pour récupérer le plus de minerai possible (fig. 1.11).

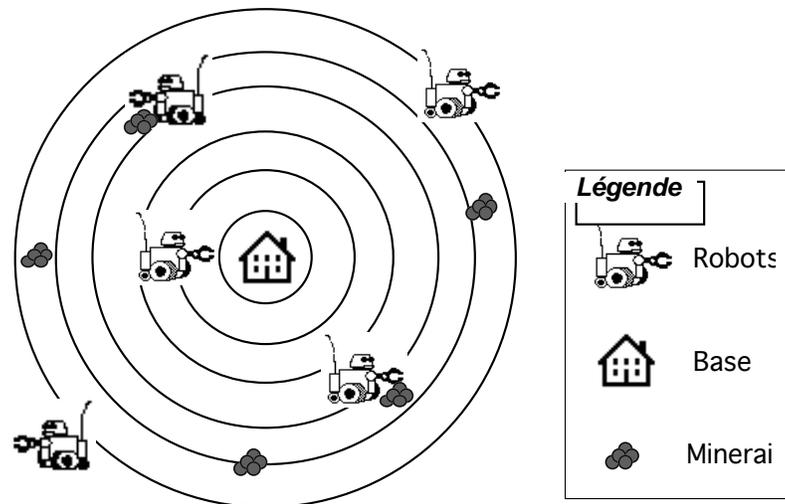


Figure 1.11: Les robots explorateurs

Pour résoudre un tel problème, il faut passer par les étapes suivantes:

1. Définir les différents types de robot à construire: doit-on supposer qu'ils sont tous identiques ou au contraire qu'ils sont spécialisés? Il est peut-être préférable de définir des robots spécialistes en transport de minerai dont les capacités de détection sont plus faibles mais qui peuvent transporter des charges plus importantes que des robots généralistes.
2. Déterminer les capacités cognitives des agents: peuvent-ils mémoriser le terrain qu'ils traversent et construire des cartes? Sont-ils purement réactifs et capables

uniquement de perceptions locales ou peuvent-ils communiquer entre eux par messages sur de longues distances?

3. Décrire les structures collectives de travail: travaillent-ils en groupe ou de manière solitaire? Les équipes sont-elles fixes ou dynamiques? Peuvent-ils recruter des aides et, si oui, comment, ou doivent-ils s'en remettre uniquement à leurs ressources propres?
4. Définir les mécanismes de coopération et les protocoles d'interactions permettant à ces agents d'accomplir collectivement leur travail en réduisant le nombre de conflits potentiels. Par exemple, que se passe-t-il lorsque deux robots découvrent en même temps un tas de minerai?
5. Etre capable d'évaluer les choix retenus et de les comparer à d'autres organisations.

L'intérêt de cette "application" est donc de soulever, dans le cadre d'une situation simple à appréhender et à évaluer, l'ensemble des problèmes que l'on peut se poser lors de la réalisation d'un SMA. Nous reprendrons cet exemple au chapitre 3 en analysant quelques organisations possibles pour résoudre ce problème. A côté des problèmes de poursuite et d'exploration, il existe de nombreuses situations qui mettent en œuvre ces animaux-robots que sont les animats et qui peuvent servir de point de départ pour l'analyse et la compréhension des systèmes multi-agents. Nous en citerons trois: les robots déménageurs, l'atelier de production, les robots constructeurs. Tous ces exemples, hormis le dernier, relèvent pour l'instant plus de la science fiction que d'une réalité industrielle, mais ils offrent l'avantage de présenter des situations relativement complexes qui mettent en jeu des problèmes de coopération et de coordination d'actions.

Des robots déménageurs

On suppose que vous vouliez déménager de votre maison et aller vous installer ailleurs. Vous avez à votre disposition une équipe de robots capables de porter des meubles et de transporter des objets fragiles, ainsi qu'un certain nombre de véhicules autonomes. Comment programmer ces robots de manière que votre déménagement s'effectue dans de bonnes conditions, que vos déménageurs ne laissent pas tomber votre piano à queue et protègent bien vos vases de chine? Le type de comportement requis est très semblable à celui de l'exploration de Mars. Il s'y ajoute des contraintes sur la nature des objets (certains meubles peuvent être trop lourds pour qu'un robot puisse les porter seul) et sur la topologie de l'environnement (les couloirs sont étroits et il est parfois impossible à deux robots de se croiser; la configuration des pièces des maisons et celle du réseau routier peuvent être complexes, etc.), même si le problème de l'exploration du terrain peut être supprimé en supposant connues la localisation et la topographie des deux maisons.

Un atelier de production autonome

Construire une cellule autonome de production est un exemple particulièrement intéressant pour tester des hypothèses concernant les différentes manières de concevoir des systèmes multi-agents. Le problème peut être défini ainsi: supposons que l'on doive construire des produits manufacturés, A_1, \dots, A_k à partir d'un ensemble de robots machines M_1, \dots, M_n et de matière première, P_1, \dots, P_j et que l'on dispose d'un ensemble de robots transporteurs T_1, \dots, T_m pour porter les produits intermédiaires d'une machine à une autre. Sachant que la matière première arrive à un bout de l'atelier de production et que les produits manufacturés partent d'un autre bout, comment organiser cette unité de production, de manière qu'elle puisse réagir à toute modification de la demande? Même si ce problème ressemble beaucoup à un problème de recherche opérationnelle, il en diffère néanmoins par le fait que l'on ne demande pas seulement de prévoir la meilleure manière d'arranger les différentes unités de l'atelier, mais aussi de définir les programmes de chacune de ces unités de façon qu'elles puissent travailler ensemble et donc coordonnent leur travail en réagissant à des pannes éventuelles tout en pénalisant le moins possible l'ensemble de la production.

1.4.4 La robotique distribuée

La robotique distribuée porte sur la réalisation non pas d'un seul robot, mais d'un ensemble de robots qui coopèrent pour accomplir une mission. A la différence du domaine d'application précédent (la construction de mondes hypothétiques), la robotique distribuée utilise des agents concrets qui se déplacent dans un environnement réel. Le domaine de la robotique distribuée recouvre en fait deux types de robotique bien distincte, la robotique cellulaire et la robotique mobile:

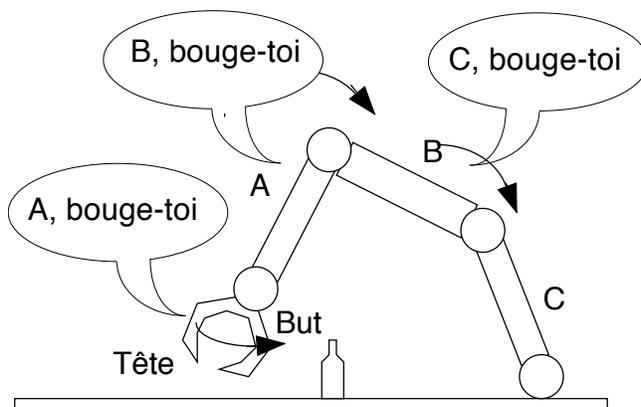


Figure 1.12: Un bras manipulateur considéré comme un système multi-agents

1. La robotique "cellulaire" s'intéresse à la constitution modulaire de robots. Dans ce cadre, un robot sera considéré comme un système multi-agent et chacun de ses composants sera regardé comme un agent. La réalisation d'un mouvement sera alors le résultat de la coordination d'un ensemble d'agents. Des techniques semblables

à celle que nous verrons avec l'éco-résolution (chap. 8) permettent d'accomplir des gestes complexes avec un minimum de calcul. A partir d'idées semblables, les recherches menées parallèlement par l'équipe de J. Perram au Danemark (Overgaard et al. 1994) et de D. Duhaut en France (Regnier et Duhaut 1995) ont produit des modèles de déplacement de bras manipulateur dans lesquels chaque élément du bras est considéré comme un agent, les articulations décrivant des contraintes pour l'ensemble des mouvements acceptables (fig. 1.12). L'agent de tête cherche à satisfaire le but qui lui est donné, par exemple souder deux pièces entre elles ou prendre un objet sur une table. S'il peut le faire lui-même, il effectue le déplacement et le système s'arrête. Sinon, il entraîne l'agent qui le suit (l'agent A sur la figure) en lui donnant des buts pour que l'agent de tête se rapproche de son objectif. Le processus se répète récursivement, chaque agent cherchant à satisfaire les buts qu'on lui propose en transmettant ses desiderata à l'agent suivant.

Les calculs s'expriment simplement et s'exécutent très rapidement. De plus, à l'encontre des techniques classiques utilisées en robotique, l'augmentation du nombre de degrés de libertés (c'est-à-dire du nombre d'articulations) améliore la vitesse de résolution du problème. Ce type de modèle est suffisamment souple pour être effectivement utilisé dans des environnements industriels. Le système de J. Perram, par exemple, est utilisé pour programmer des robots soudeurs au chantier naval d'Odense.

2. La *robotique mobile* fait intervenir au moins deux robots qui doivent coordonner leurs déplacements et coopérer à l'accomplissement de tâches telles que nettoyer le plancher, surveiller des bâtiments, intervenir pour secourir des personnes, réparer des canalisations ou explorer des espaces lointains ou dangereux. Pour l'instant, tous ces projets sont plutôt à l'état de recherches, mais de nombreuses contributions ont été avancées dans ce domaine, notamment par M. Mataric (Mataric 1994), L. Steels (Steels 1994) et J.-L. Deneubourg (Beckers et al. 1994).

La coordination de véhicules, qu'il s'agisse d'avions, de voitures ou de bateaux, fait aussi partie de ce domaine d'application. Plusieurs véhicules, en se déplaçant, risquent de se gêner mutuellement. Par exemple, deux avions qui évoluent dans des secteurs voisins, ou deux voitures qui sont sur la même route, doivent faire attention de ne pas se rencontrer. Les risques de collision sont d'autant plus importants que les véhicules sont proches les uns des autres et que leurs trajectoires les amènent à se croiser. Il s'agit alors de coordonner leurs mouvements de manière à ce que chacun d'entre eux puisse aller là où il désire aller sans entrer en collision (fig. 1.13).

Dans une variante de ce problème, on demande à certains véhicules d'avancer en formation à la manière des oies sauvages ou des escadrilles aériennes. Le problème est alors plus complexe puisqu'il s'agit de faire évoluer ensemble des véhicules de manière que la structure de la formation reste constante. Les premières tentatives de résolution ont été effectuées dans le domaine du contrôle de trafic aérien et mirent en œuvre des agents cognitifs (Cammarata et al. 1983; Chaib-Draa 1989) mais avec des résultats très limités. Les approches plus actuelles (Zeghal et al. 1993) utilisent des agents réactifs et des mécanismes de champs de force, comme nous le présenterons au chapitre 8.

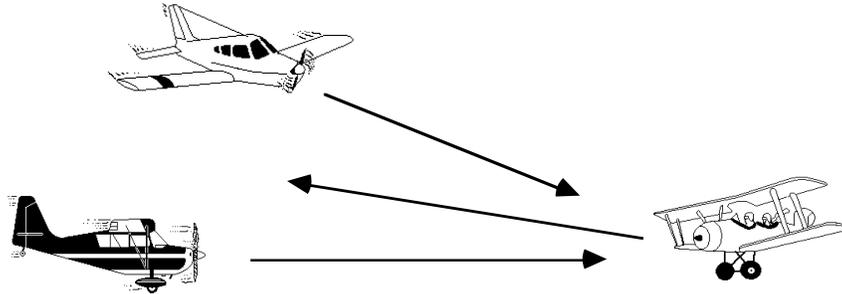


Figure 1.13: Le problème de l'évitement de collision. Comment ces avions peuvent-ils se croiser sans se télescoper tout en conservant leurs objectifs initiaux?

1.4.5 La conception kénétique de programmes

La kénétique a aussi pour ambition de prôner un nouveau mode de conception de systèmes informatiques, cherchant à dépasser les techniques informatiques actuelles pour réaliser des logiciels distribués fonctionnant avec une grande souplesse et une grande adaptabilité à leur environnement. L'objectif alors de ce que j'appellerai la *conception kénétique de logiciels* est de donner naissance à des systèmes informatiques capables d'évoluer par interaction, adaptation et reproduction d'agents relativement autonomes et fonctionnant dans des univers physiquement distribués.

L'étendue de grands réseaux informatiques tels que Internet, l'extension des réseaux locaux et l'intégration des données aussi différentes que du texte, de l'image, du son ou de l'animation laissent encore de côté la question cruciale de l'utilisation de cette information de manière automatique. Il devient en effet très difficile de s'y retrouver dans cette gigantesque toile d'araignée (la bien nommée *World Wide Web*), dans laquelle tout nœud est un serveur d'informations et de ressources fonctionnant de manière relativement autonome par rapport aux autres. Nul ne gère globalement l'ensemble — qui est d'ailleurs ingérable —, mais chaque nœud est responsable de ses propres informations et des liens qu'il peut avoir avec d'autres nœuds.

La kénétique propose une nouvelle technologie de construction de logiciels à partir des concepts d'agent et d'interaction. Par exemple, chaque unité de programme peut prendre la forme d'un agent qui dispose de sa propre autonomie, de ses propres objectifs et "vit" sur le réseau comme un animal dans un écosystème naturel, coopère ou négocie avec d'autres unités de même nature. Les utilisateurs deviendraient alors des "bergers" donnant des consignes à leur troupeau d'agents pour recueillir ensuite les fruits de leur labour. Cette métaphore, qui donne encore une impression de science-fiction, a toutes ses chances de devenir bientôt une réalité, car les technologies de base nécessaires à cette évolution de la programmation sont déjà présentes. On verra donc apparaître bientôt ces "agents", d'abord simples et encore peu autonomes puis bientôt plus élaborés et complexes, aider les utilisateurs dans leurs tâches quotidiennes en récupérant des données, en gérant les rendez-vous et les tâches communes, en offrant et en demandant des services à d'autres agents, le tout de manière intégrée et naturelle.

La notion d'agent circulant dans un réseau ne constitue qu'une des facettes,

peut-être la plus visible, de la conception kénétique de programmes. L'autre aspect concerne l'architecture des programmes eux-mêmes. Si, pour l'instant, la conception par objets semble être la panacée universelle de l'écriture de programmes (Booch 1991; Ferber 1990), elle n'est en fait qu'une étape vers une conception encore plus "modulaire" de l'écriture de logiciels. Chaque composant d'un programme pourra être envisagé comme un agent disposant de ses propres compétences et de ses propres buts, l'ensemble cherchant à répondre aux besoins de l'utilisateur et aux extensions nouvelles qui apparaissent continuellement.

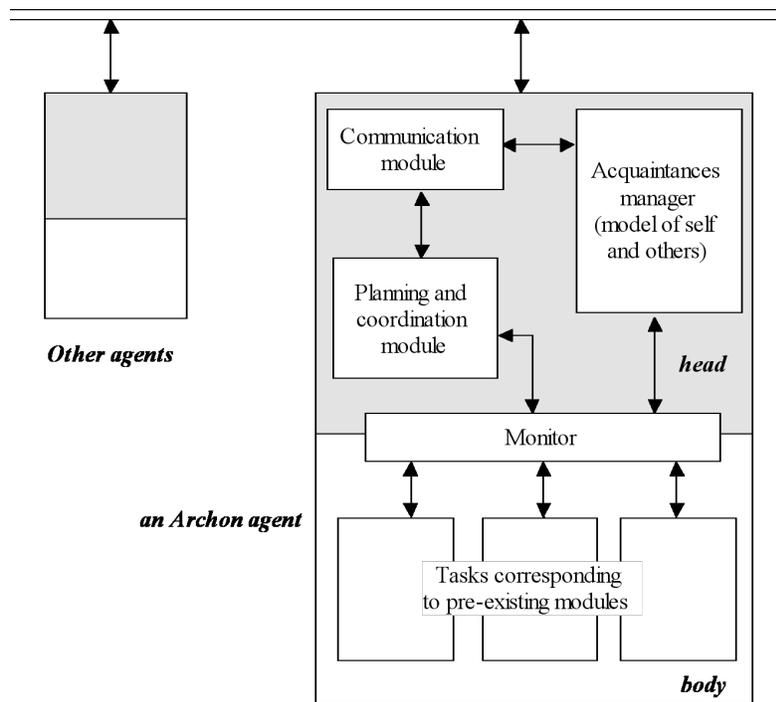


Figure 1.14: L'architecture générale du système ARCHON

L'un des systèmes les plus connus en Europe pour avoir tenté de donner un cadre opérationnel à ces idées est le système ARCHON (Architecture for Cooperating Heterogeneous On-line Systems), issu d'un projet ESPRIT qui propose une architecture générale de SMA pour intégrer différents programmes devant coopérer ensemble (Burg et Arlabosse 1994; Wittig 1992). L'architecture générale du système est présentée figure 1.14. La "tête" d'un agent comprend un gestionnaire de communications de haut niveau qui s'appuie sur les recommandations de la norme OSI, un modèle des compétences des autres et de soi sous la forme d'un gestionnaire d'accountances (cf. chap. 5) et un système de planification et de coordination. Un moniteur fait l'interface et la gestion entre la "tête" et le corps d'un agent composé d'un ensemble de tâches, chacune correspondant à un module logiciel préexistant. Il est ainsi possible d'encapsuler des programmes existants à l'aide d'agents ARCHON et faire ainsi coopérer des programmes hétérogènes, initialement non coopérants. Le système ARCHON a été testé dans de nombreuses configurations industrielles, en particulier dans des domaines de surveillance de réseaux de distribution d'énergie

électrique et dans celui de l'analyse de pannes dans un accélérateur de particules. Il s'agit maintenant d'un logiciel véritablement industriel commercialisé par la société française Framentec. L'ouvrage (Jennings 1994) donne une présentation détaillée du fonctionnement de ce système⁸.

1.5 Les grands thèmes de la kénétique

Concevoir un système multi-agent, c'est avant tout analyser et résoudre un très grand nombre de problèmes. La conception d'un système anti-collision, par exemple, soulève un grand nombre de questions qui recouvrent des domaines très divers. Comment les agents se représentent-ils leur environnement et les autres agents? Comment coopèrent-ils? Peuvent-ils assurer leur viabilité? Sont-ils capables d'adapter leur comportement à des modifications du milieu dans lequel ils évoluent? Sont-ils pris dans des rapports hiérarchiques ou d'autorités entre eux? Ces différentes questions peuvent être classées en cinq catégories principales: la problématique de l'action, l'agent et sa relation au monde, l'interaction, l'adaptation, la réalisation et l'implémentation de SMA.

1.5.1 La problématique de l'action

La problématique de l'action est souvent peu considérée, car il s'agit d'une question subtile qui donne l'impression d'être aisée à résoudre alors qu'elle pose de nombreuses difficultés. La question peut se poser ainsi: comment plusieurs agents peuvent-ils agir simultanément et quelles sont les conséquences de leurs actions dans l'environnement? La réponse a l'air évidente: l'état de l'environnement est la conséquence directe de la combinaison des actions des agents. Mais comment ces actions se combinent-elles? Comment se fait-il qu'un agent qui décide d'avancer et qui est bloqué par un obstacle ne puisse pas bouger? Comment deux agents peuvent-ils entrer en collision ou au contraire s'éviter? Comment les contraintes environnementales se combinent-elles à l'action des agents? Toutes ces questions soulèvent d'importants problèmes théoriques qui doivent être soulevés dès que l'on met en œuvre des agents situés, ou dès que l'on cherche à faire de la planification multi-agent.

1.5.2 L'individu et sa relation au monde

Les différents thèmes relatifs à l'individu portent, d'une part, sur son architecture et son organisation interne et, d'autre part, sur l'ensemble des moyens qu'il met en œuvre pour assurer sa viabilité et satisfaire ses objectifs. Il s'agit évidemment de thèmes essentiels car la réalisation d'un système multi-agent passe nécessairement pas la description de l'architecture des agents et des fonctionnalités dont ils disposent pour accomplir leurs tâches.

⁸Bien que le système présenté dans cet ouvrage ait pour nom GRATE, il s'agit bel et bien de la description d'ARCHON.

C'est à ce niveau que l'on s'intéresse aux éléments cognitifs dont l'organisation permet la constitution d'un comportement adapté. Sauf pour des agents réactifs simples, un agent se trouve à un certain moment dans un certain "état mental" qui résulte de sa propre histoire, de sa perception du monde et de ses interactions avec le monde et avec les autres agents. Ces états mentaux sont souvent très complexes et font intervenir un grand nombre d'éléments dont la combinaison explique le comportement d'un agent "de l'intérieur".

Ces éléments cognitifs, que nous appellerons *cognitons* au chapitre 5, sont aux états mentaux des agents ce que les corpuscules élémentaires sont aux corps physiques: des composants de base dont la combinaison permet d'exprimer l'état mental d'un agent et dont les lois d'interaction servent à décrire l'évolution future du comportement d'un agent et de ses états mentaux ultérieurs. Ils régissent l'ensemble des aspects de l'activité intérieure d'un agent: perception et exécution d'action, croyances, désirs et tendances, intentions, méthodes et plans, etc. C'est à ce niveau aussi que sont décrits les engagements. En s'engageant à accomplir une action auprès d'un tiers, l'agent contraint l'ensemble des actions qu'il peut réaliser et donne la possibilité au tiers de planifier son propre comportement en diminuant son incertitude sur l'ensemble des états futurs.

1.5.3 L'interaction

Pour un agent, interagir avec un autre constitue à la fois la source de sa puissance et l'origine de ses problèmes. C'est en effet parce qu'ils coopèrent que des agents peuvent accomplir plus que la somme de leurs actions, mais c'est aussi à cause de leur multitude qu'ils doivent coordonner leurs actions et résoudre des conflits. Pour un agent, l'autre est à la fois le pire et la meilleure des choses.

Traiter le problème de l'interaction, c'est se donner les moyens non seulement de décrire les mécanismes élémentaires permettant aux agents d'interagir, mais aussi d'analyser et de concevoir les différentes formes d'interaction que des agents peuvent pratiquer pour accomplir leurs tâches et satisfaire leurs buts. Tout d'abord, les agents doivent être capables, par le biais de la communication, de transmettre des informations, mais surtout d'induire chez l'autre un comportement spécifique. Communiquer est donc une forme d'action particulière qui, au lieu de s'appliquer à la transformation de l'environnement, tend à une modification de l'état mental du destinataire. Par exemple, demander à un autre d'exécuter une tâche tend à provoquer chez l'autre une intention d'accomplir cette tâche et constitue donc une manière de satisfaire un objectif sans réaliser la tâche soi-même.

Les différentes formes d'interaction que nous étudierons plus particulièrement sont la collaboration (cf. chap. 7) et la coordination d'actions (cf. chap. 8). La première s'intéresse à la manière de répartir le travail entre plusieurs agents, qu'il s'agisse de techniques centralisées ou distribuées, et la seconde analyse la manière dont les actions des différents agents doivent être organisées dans le temps et l'espace de manière à réaliser les objectifs. Enfin, lorsque des conflits apparaissent, il est important de pouvoir en limiter les effets. Les techniques de négociation servent ainsi à satisfaire les parties impliquées en établissant des compromis ou en dépassant la

nature du conflit.

La coopération est la forme générale d'interaction la plus étudiée dans les systèmes multi-agents (Ferber 1995). Nous verrons en particulier que, contrairement au sens commun qui considère que la coopération demeure l'apanage des êtres capables d'avoir un projet explicite, donc des agents cognitifs, il est possible de parler de coopération de manière aussi bien réactive que cognitive si l'on envisage uniquement le résultat des actions et non les intentions des agents. De manière simplifiée, le problème de la coopération peut se ramener à déterminer qui fait quoi, quand, où, avec quels moyens, de quelle manière et avec qui, c'est-à-dire en fait à résoudre les différents sous-problèmes que constituent la collaboration par répartition de tâches, la coordination d'actions et la résolution de conflits. Nous résumerons cela par la formule:

$$\begin{aligned} \text{Coopération} &= \text{collaboration} + \text{coordination d'actions} \\ &+ \text{résolution de conflits} \end{aligned}$$

1.5.4 L'adaptation

On peut voir le problème de l'adaptation structurelle et comportementale d'un ensemble d'agents de deux manières différentes: soit comme une caractéristique individuelle des agents, et on parle alors *d'apprentissage*, soit comme un processus collectif mettant en jeu des mécanismes reproductifs, ce que l'on appelle *évolution*. Bien que les deux approches soient généralement traitées de manière distincte, il est possible de les considérer globalement comme un processus à la fois individuel et collectif d'adaptation. Ces problématiques font l'objet de recherches importantes dans les domaines de l'intelligence artificielle distribuée et de la vie artificielle, le premier se consacrant plus naturellement à l'apprentissage, par des techniques symboliques ou neuronales, et le second à l'évolution, en utilisant des algorithmes génétiques. Bien que l'adaptation soit un thème particulièrement important dans l'étude des systèmes multi-agents (Lestel et al. 1994), nous ne le développerons que très partiellement, en ne l'abordant qu'au travers des comportements collectifs, l'adaptation résultant alors de mécanismes de régulation issus des actions conjointes d'un ensemble d'agents (cf. chap. 7).

1.5.5 La réalisation et l'implémentation des SMA

Enfin, il reste un grand domaine qui ne sera pas traité dans cet ouvrage et qui porte sur l'implémentation des systèmes multi-agents. Ce thème, qui englobe à la fois les langages d'implémentation (les langages de type L1 de la section 1.2.5), les architectures informatiques, les plates-formes de développement et les aspects méthodologiques de la mise en œuvre de ces systèmes, est si vaste qu'il mériterait un volume à lui tout seul. Il ne sera qu'effleuré ici.

1.6 Domaines voisins des systèmes multi-agents

La kénétique voisine avec un grand nombre d'autres domaines scientifiques sans pour cela être confondue avec eux. Il s'agit essentiellement de l'intelligence artificielle, de la systémique, des systèmes distribués et de la robotique.

1.6.1 L'intelligence artificielle

A la différence de l'intelligence artificielle et de la plupart des programmes informatiques, les systèmes multi-agents ne sont plus des "penseurs" renfermés sur leur propre raisonnement qui ignorent leur environnement, mais constituent de véritables sociétés d'êtres qui doivent se mouvoir, planifier, communiquer, percevoir, agir et réagir et, d'une manière générale, "vivre" et travailler dans un milieu dans lequel ils entrent parfois en conflit avec d'autres agents.

Par rapport à l'IA, la kénétique fait appel à un grand nombre de concepts nouveaux pour le domaine, tels que la coopération, la coordination d'action, la négociation, les conflits, la satisfaction, l'engagement, l'action et la réaction ou la perception. Ces concepts ne trouvent plus uniquement leur origine dans la psychologie cognitive individuelle, comme pour l'IA classique, mais également dans la sociologie et dans la biologie, voire dans l'éthologie.

Alors que pour l'IA, c'est l'individu qui est intelligent, pour la kénétique, c'est l'organisation qui présente des fonctionnalités que l'on peut caractériser comme intelligente. Concevoir des êtres intelligents n'est pas un but pour la kénétique, puisqu'elle s'intéresse essentiellement aux relations entre individus et aux conséquences de leurs interactions, mais seulement un moyen pour développer des organisations artificielles plus performantes en réduisant la redondance et en augmentant l'efficacité des actions effectuées.

Dans le domaine des systèmes multi-agents, une blague consiste d'ailleurs à dire que l'IA est une forme d'IAD mono-agent! Bien qu'il s'agisse d'une boutade, cela exprime bien que l'IA et la kénétique travaillent à des niveaux d'étude différents, un peu comme la psychologie et la sociologie.

1.6.2 La systémique

Le courant systémique, bien qu'ayant toujours considéré l'importance des interactions pour la définition des systèmes, a plus mis l'accent sur l'analyse des flux entrants et sortants que sur l'aspect comportemental des interactions. En effet, la partie proprement opératoire de l'analyse systémique, depuis ses débuts avec V. Bertalanffy (Bertalanffy 1968), N. Wiener (Wiener 1948) et surtout J.W. Forrester (Forrester 1980), s'est surtout consacrée à l'analyse des systèmes biologiques, économiques et artificiels sous l'angle de transfert de flux de matières, d'énergies, d'informations ou d'argent entre différents éléments de stockage, de commande ou de régulation. Mêmes les premiers travaux de J.-L. Le Moigne sur la théorie du système général (Le Moigne 1977) ou ceux de P. Delattre sur les structures organisées (Delattre 1971)

avaient repris cette idée devenue classique de considérer un système comme un ensemble de sous-systèmes interconnectés par des liens servant à transférer des flux ou à contrôler leur activité. Bien que très utile dans de nombreux domaines, tels que l'organisation d'entreprise ou la réalisation de dispositifs électroniques, cette approche, malgré son universalité proclamée, s'avère en fait limitante puisque les seules notions véritablement fécondes que l'on y trouve sont celles de régulation, de stabilisation et de fonction de transfert. Comme l'indique en effet un ouvrage récent sur l'analyse des systèmes (Lapierre 1992), "Le concept de systèmes tel que nous le concevons ici n'est pas un concept d'objet, c'est un concept d'outils. (...) Ces systèmes sont des entités abstraites, non des parties concrètes de cette totalité".

Dans cette approche, un système, qu'il s'agisse d'un être humain, d'une entreprise ou de n'importe quelle organisation, est décomposé en sous-systèmes, chaque sous-système répondant à une fonction particulière du système telle que la consommation, la production, la recherche, la conception (design), l'échange et la gestion (Charest 1980). Ces sous-systèmes résument les caractéristiques du système étudié au travers d'une approche analytique, pourtant récusée, qui ne permet pas de comprendre l'évolution du système ni de comprendre les facteurs internes d'une organisation. Le systémisme devient alors un concept "fourre-tout" qui se réduit parfois à du fonctionnalisme déguisé.

Il existe néanmoins un deuxième courant fort représenté actuellement, de E. Morin (Morin 1977) à F. Varela (Varela 1989), en passant par H. Atlan (Atlan 1979), I. Prigogine (Prigogine et Stengers 1979) ou E. Bernard-Weil (Bernard-Weil 1988), qui consiste à comprendre les systèmes comme des entités auto-organisatrices, dont le fonctionnement et l'évolution sont les produits du comportement d'un ensemble d'entités en interactions. Cependant, à part peut-être avec la notion d'autopoïèse de Maturana et Varela (Maturana et Varela 1987) qui postule qu'un système auto-organisateur n'a pas d'autre fonction que de maintenir sa structure organisée, l'accent est toujours mis sur les flux d'interconnexion et sur la modélisation des boucles de rétroaction.

Quelques travaux remarquables, tels que ceux de F. Le Gallou (Le Gallou 1992) ou d'E. Schwartz (Schwartz 1993), tentent d'unifier ces deux voies. Mais ces entreprises, qui cherchent plus à expliquer de manière qualitative les phénomènes qu'à en donner un modèle opérationnel, abordent les systèmes de manière holiste (un système est un tout) et ne tiennent compte que de manière marginale du comportement des entités qui les composent. Car, définir un système par ses relations globales n'explique pas la genèse de sa structure mais seulement les formes générales de son évolution à partir d'un point de vue macroscopique. Cette approche ignore ainsi l'importance des actions individuelles qui concourent à l'élaboration de la structure et donc à l'organisation du système en tant que tel.

Au contraire, la kénétique met en avant les interactions locales et les phénomènes d'émergence qui en résultent. Il s'agit donc d'une forme de systémique, plus exactement une *néo-système* fondée sur l'analyse des comportements d'agents interagissants, l'auto-organisation des systèmes étant alors le produit des actions transformatrices de ces agents.

1.6.3 Les systèmes distribués

Il existe de nombreux points communs entre la problématique soulevée par la kénétique et les questions que se posent les systèmes distribués. Mais ces deux champs présentent néanmoins des traits distincts. Les systèmes distribués cherchent à concevoir des systèmes informatiques capables de gérer l'exécution de tâches en employant au mieux des ressources physiquement réparties (mémoire, processeur, gestionnaires d'informations). Les techniques mises en œuvre (client/serveur, distribution des applications) relèvent plus spécifiquement de la pratique informatique ou, plus exactement, d'une opérationnalisation de systèmes informatiques que ne le fait la kénétique. Par exemple, les systèmes distribués cherchent à faire des systèmes d'exploitation répartis dans lesquels l'utilisateur n'a pas à spécifier sur quel processeur son programme va tourner, ni à vérifier que les ressources nécessaires à son exécution sont bien présentes sur sa machine, laissant au système le soin de gérer ces détails. Les problèmes qui se posent alors nécessitent de pouvoir concevoir, spécifier et valider des applications réparties, en définissant des protocoles permettant à plusieurs modules informatiques répartis d'utiliser mutuellement leurs services.

La problématique de la kénétique est beaucoup plus ouverte: elle ne se cantonne pas à la gestion d'applications informatiques réparties ou à la définition de systèmes d'exploitation transparents pour l'utilisateur. Les SMA tentent d'appréhender le problème de l'interaction entre entités individualisées de manière générale et abstraite, considérant que les systèmes distribués ne sont qu'une application ou qu'une réalisation possible (et particulièrement importante) de ce concept général. Le problème de l'autonomie, de la coopération (au sens vrai du terme et non dans le sens restreint de partage de tâches ou de ressources informatiques), et de la constitution d'organisations artificielles émergentes capables d'accomplir des travaux qu'une entité unique serait incapable (ou difficilement capable) d'accomplir constitue le fondement de cette approche. De ce fait, la perspective de type "résolution de problèmes", "coordination d'actions" ou "partage de connaissances" que soulèvent les systèmes multi-agents ne correspond pas du tout aux systèmes distribués.

Le rapport qui existe entre processus et agent est à cet égard particulièrement intéressant: alors que la notion de processus est relative à celle d'exécution, celle d'agent correspond à une individualisation des compétences et des objectifs, à une autonomie des capacités d'action et d'interaction. Ainsi, un agent peut-il s'implémenter sous la forme de plusieurs processus sans perdre de son individualité, le niveau conceptuel ne devant pas être confondu avec le niveau physique de son implémentation. Néanmoins, les liens qui unissent ces deux domaines sont très étroits: les SMA utilisent un grand nombre de techniques provenant des systèmes distribués, en se situant à un niveau plus conceptuel et méthodologique. Certains algorithmes qu'emploient les SMA, tels que les algorithmes d'allocation de tâches, et certains formalismes de représentation, tels que les réseaux de Petri, font partie du bagage naturel des systèmes distribués. De ce fait, les SMA et les systèmes distribués entretiennent plus un rapport synergique qu'une véritable opposition, les seconds apportant leur rigueur et leurs algorithmes, les premiers donnant une

conceptualisation et une approche plus générale, moins centrée sur les mécanismes d'exécution. Il n'est donc pas étonnant que des recherches en cours tendent vers une collaboration de plus en plus étroites entre les spécialistes des systèmes distribués et ceux des systèmes multi-agents.

1.6.4 La robotique

Les systèmes multi-agents entretiennent aussi de très nombreux rapports avec la robotique. Mais dans ce cas, les relations sont très claires: la robotique, et surtout la robotique distribuée, est une application possible des systèmes multi-agents, ces derniers décrivant ce que les robots doivent être capables d'accomplir pour coopérer à des tâches mettant en œuvre plusieurs robots.

En revanche, la robotique étudie tout un ensemble de problèmes mécaniques, électroniques, géométriques et, d'une manière générale, de commandes de mouvement qui ne font pas partie des intérêts de la kénétique. De ce fait, sauf pour de très rares applications qui consistent à décomposer un robot manipulateur en un ensemble d'agents (Overgaard et al. 1994) et (Regnier et Duhaut 1995), la kénétique ne se tourne vers la robotique que lorsqu'il s'agit de faire interagir un ensemble de robots mobiles ensemble. C'est dans ce sens que les problèmes abordés aux chapitres 7 et 8 peuvent concerner la robotique.

1.6.5 Ce qui ne relève pas de la kénétique

Afin d'éviter toute équivoque, voici quelques éléments permettant de dire ce qui ne relève pas de la kénétique. D'une manière générale, on peut dire que tout ce qui est de l'ordre d'une conception individualiste des systèmes n'appartient pas au domaine des SMA. D'autre part, tout système décomposé en un ensemble de sous-systèmes n'est pas du ressort de la kénétique. Pour qu'un système puisse être considéré comme un système multi-agent, il faut:

1. qu'il dispose d'agents autonomes fonctionnant en parallèle et cherchant à satisfaire un but ou une fonction de satisfaction (ce que nous appellerons des tendances au chapitre 5);
2. que ces agents possèdent un mécanisme d'interaction de haut niveau indépendant du problème à résoudre (protocoles de communications ou mécanismes d'interaction avec l'environnement).

On peut ainsi répondre à certaines questions:

— *Est-ce que les réseaux sont des systèmes multi-agents?*

— Non, puisque les nœuds du réseau ne disposent généralement ni de but, ni de fonction de satisfaction. Néanmoins, les réseaux sont bien placés pour évoluer vers une conception multi-agent, du fait de leur nature déjà distribuée et de leurs problèmes de gestion qui nécessitent une approche locale. C'est pourquoi de nombreuses applications multi-agents utilisent les réseaux comme support de leur activité.

— *Est-ce qu'une décomposition modulaire d'un programme constitue un système multi-agents?*

— Non. Pour les mêmes raisons que précédemment (les modules n'ont ni but ni fonction de satisfaction) et, d'autre part, parce que les mécanismes d'interaction que l'on implémente dans les modules sont de bas niveau (par exemple des appels de procédure ou des envois de message pour demander un service que les modules sont obligés de rendre au risque, sinon, de produire une erreur).

— *Est-ce qu'un programme d'IA qui s'exécute en parallèle est un système multi-agent?*

— Non, pour les mêmes raisons que précédemment.

On pose aussi souvent la question suivante:

— *Quelle est la différence entre objet, acteur et agent?*

— La réponse est simple. Dans le domaine de l'informatique, les termes objet et acteur désignent des entités informatiques caractérisées par leur structure et leurs mécanismes d'exécution. La notion d'*objet* est définie par trois concepts: la relation classe/instance, qui décrit la classe comme un modèle structural et comportemental et l'instance comme un représentant d'un modèle; l'héritage, qui permet de dériver une classe d'une autre et de faire bénéficier la première des caractéristiques de la seconde; et l'envoi de message, qui autorise la définition de procédures polymorphes, c'est-à-dire de procédures dont le code diffère en fonction du receveur du message (Ferber 1990). Les objets se situent donc, d'une part, à un niveau d'implémentation avec les langages à objets et, d'autre part, à un niveau logique, dans le cadre de l'analyse et de la conception par objet. Dans tous les cas, les objets ne sont pas des agents car ils ne remplissent pas les deux critères mentionnés ci-dessus: ils n'ont ni but ni recherche de satisfaction et le mécanisme d'envoi de message se résume à un appel de procédure. Les mécanismes d'interaction sont donc à la charge du programmeur. Les *acteurs* en informatique sont des entités qui s'exécutent en parallèle, qui communiquent par envois de messages asynchrones bufferisés et qui, généralement, n'attendent pas le retour d'un calcul mais demandent au receveur du message de renvoyer la réponse à un autre acteur appelé la continuation locale du calcul (Agha 1986). Bien qu'il s'agisse de modèles parallèles d'exécution, et pour les mêmes raisons que pour les objets, les acteurs ne peuvent être identifiés à des agents. La différence essentielle qui existe entre les objets et les agents (purement communicants) est illustrée à la figure 1.15. Un objet est défini par un certain nombre de services (ses méthodes) qu'il ne peut refuser d'exécuter si un autre objet le lui demande et les messages sont donc nécessairement des invocations de méthodes. Le développeur d'un logiciel écrit à l'aide d'objets doit donc vérifier que tous les objets recevront bien des ordres sensés qu'ils seront effectivement capables d'exécuter. Par rapport aux objets, les *agents* peuvent recevoir des messages qui ne sont pas uniquement des demandes d'exécution mais aussi des informations ou des demandes d'informations sur leurs capacités, etc. De ce fait, les services que peuvent rendre un objet sont donc filtrés par une couche logicielle qui découple les demandes et le fonctionnement interne de l'agent. Enfin, comme nous l'avons dit, les agents tentent de satisfaire des objectifs, ce qui leur procure une autonomie supplémentaire par rapport aux objets. En effet, l'agent, à l'encontre de l'objet,

peut refuser d'accepter d'effectuer un certain travail, ce refus pouvant s'expliquer par son manque de compétence (il ne possède pas le savoir-faire nécessaire) ou par sa trop grande occupation à une autre tâche ou par toute autre raison. La figure 1.15 représente la différence fondamentale qui existe entre un agent et un objet. Un agent encapsule les méthodes qu'un objet peut offrir sous la forme d'un ensemble de services qui ne sont accessibles que par un langage particulier connu de l'ensemble des agents et qui utilise généralement la théorie des actes de langages (cf. chap. 6). Mais le lien qui existe entre objets et agents ne doit pas non plus être sous-estimé.

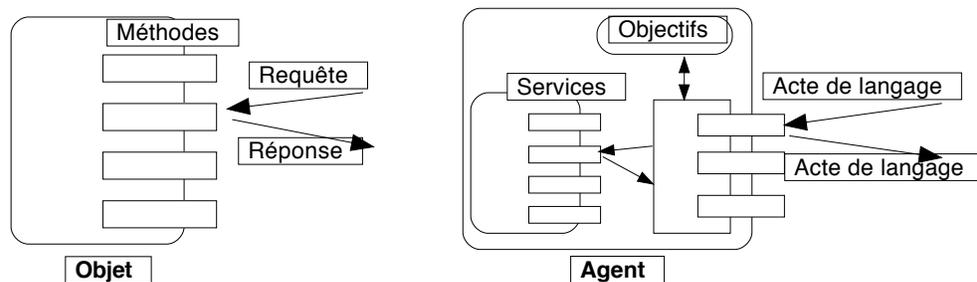


Figure 1.15: Un objet répond directement à des requêtes correspondant à ses méthodes, alors qu'un agent encapsule ses compétences (ou services) par des mécanismes supplémentaires qui "filtrent" les communications externes et gèrent les dialogues. De plus, les agents sont mus par des objectifs (ou tendances) personnels.

Si un agent purement communicant peut être considéré comme une sorte d'objet amélioré, inversement, un objet peut passer pour un agent dégénéré, c'est-à-dire un agent dont le langage d'expression se résume à l'emploi des mots clés correspondant à ses méthodes. Enfin, et c'est d'ailleurs ce qui augmente encore la confusion possible, on implémente souvent, pour des raisons pratiques, les agents sous la forme d'objets ou d'acteurs. Ceci n'est pourtant qu'une des implémentations possibles. On peut de la même manière implémenter des agents en FORTRAN, C ou LISP, qui ne sont pas naturellement des langages à objets, sans que cela modifie en quoi que ce soit leur caractère d'agent.

On peut donc dire, pour résumer, que les agents sont définis à un niveau conceptuel et les objets ou les acteurs aux niveaux d'implémentation et d'exécution. C'est d'ailleurs pour éviter cette confusion que l'on ne parlera pratiquement pas d'objets ni d'acteurs dans cet ouvrage.

Chapter 2

Interactions et coopération

2.1 Situations d'interaction

La notion d'interaction est au centre de la problématique des systèmes multi-agents et de la kénétique. Mais qu'est-ce qu'une interaction?

Une interaction est une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Les interactions s'expriment ainsi à partir d'une série d'actions dont les conséquences exercent en retour une influence sur le comportement futur des agents. Les agents interagissent le long d'une suite d'événements pendant lesquels les agents sont d'une certaine manière en contact les uns avec les autres, que ce contact soit direct ou qu'il s'effectue par l'intermédiaire d'un autre agent ou de l'environnement.

Les situations d'interactions sont nombreuses et diverses: l'aide d'un robot par un autre, l'échange de données entre serveurs informatiques, la mise en commun des compétences de plusieurs spécialistes pour porter un diagnostic, l'utilisation d'une imprimante par deux programmes simultanément, la répartition des charges sur plusieurs processeurs, la collision de deux véhicules sont autant d'exemples de situations d'interactions.

Les interactions sont non seulement la conséquence d'actions effectuées par plusieurs agents en même temps, mais aussi l'élément nécessaire à la constitution d'organisations sociales. C'est par les échanges qu'ils entretiennent, par les engagements qui les lient, par l'influence qu'ils exercent les uns sur les autres que les agents sont des entités sociales et que de nouvelles fonctionnalités peuvent émerger de ces systèmes d'actions mutuelles. Les groupes sont donc à la fois les résultats d'interactions et les lieux privilégiés dans lesquels s'accomplissent les interactions. C'est pourquoi il est généralement impossible d'analyser des organisations sociales sans tenir compte des interactions entre leurs membres.

Si l'on prend l'exemple des robots transporteurs, déplacer une masse de 300 kilos alors que chacun ne peut en porter que 100 suppose d'une part qu'il existe un nombre de robots suffisant pour porter cette charge, et d'autre part que leurs actions vont être coordonnées de telle manière que la charge soit effectivement déplacée et arrive à bon port. Il en est de même pour des véhicules qui se meuvent sur un réseau routier: leurs actions doivent être coordonnées pour qu'ils puissent parvenir

à leur destination sans entrer en collision. Plus précisément, la notion d'interaction suppose:

1. La présence d'agents capables d'agir et/ou de communiquer.
2. Des situations susceptibles de servir de point de rencontre entre agents: collaboration, déplacement de véhicules amenant à une collision, utilisation de ressources limitées, régulation de la cohésion d'un groupe.
3. Des éléments dynamiques permettant des relations locales et temporaires entre agents: communication, choc, champ attractif ou répulsif, etc.
4. Un certain "jeu" dans les relations entre les agents leur permettant à la fois d'être en relation, mais aussi de pouvoir se séparer de cette relation, c'est-à-dire de disposer d'une certaine autonomie. Si des agents sont totalement liés par un couplage fixe, leur interaction devient rigide et ils n'interagissent plus au sens plein du terme.

L'interaction est à la base de la constitution d'organisations (Morin 1977) et simultanément les interactions supposent la définition d'un espace et généralement d'une organisation préétablie dans lesquels ces interactions peuvent se produire. De ce fait, l'interaction est le composant de base de toute organisation, à la fois source et produit de la permanence de cette organisation, et la dissolution d'une organisation est concomitante de la disparition (ou en tout cas de la diminution) des interactions des individus présents dans cette organisation.

Enfin, l'interaction est la conséquence de l'aspect pluriel des sociétés multi-agents, en apportant une dimension supplémentaire à l'individu. Ce dernier n'est plus le centre de l'univers, comme dans les conceptions "ego-centrées" caractéristiques de l'intelligence artificielle classique dans laquelle l'intelligence peut être comprise du point de vue de l'individu seul, mais le nœud d'un ensemble d'échanges et d'interdépendances qui, en retour, le façonnent et lui donnent tout son sens: un agent sans interaction avec d'autres agents n'est plus qu'un corps isolé, qu'un système de traitement d'informations, dépourvu de caractéristiques adaptatives.

Cette prise en compte de l'interaction comme élément essentiel et fondateur des formes d'intelligence, bien qu'apparaissant comme novateur pour les sciences cognitives, n'est pourtant pas une nouveauté. L'éthologie, qui s'intéresse aux comportements animaux et humains, montre que l'apparition des formes cognitives est le produit d'une série de comportements complexes dans lesquels l'échange et l'interaction avec des individus de sa propre espèce jouent un rôle essentiel dans le développement d'un être vivant (McFarland 1990) (Lannoy et Feyereisen 1987) (Cyrułnik 1989). Lorsque cet échange n'existe pas, les individus ne peuvent s'accomplir totalement et restent dans un état "primitif".

Les interactions entre agents s'expriment sous diverses formes. Mais au-delà de la diversité superficielle, il est possible de dégager des différences et des points communs à un niveau plus profond. En effet, comment comparer des interactions de marins en train de manœuvrer une voile et des ouvriers qui construisent une maison? Des automobilistes qui se croisent sur la route et des robots qui recherchent et récupèrent

du minerais? La première étape consiste à considérer différentes situations dans lesquelles plusieurs agents sont en interaction. Afin de décrire ces activités, nous recourons à la notion de *situation d'interaction* qui permet de décrire des types d'interaction en les reliant aux éléments qui la composent.

Définition: *On appellera situation d'interaction un ensemble de comportements résultant du regroupement d'agents qui doivent agir pour satisfaire leurs objectifs en tenant compte des contraintes provenant des ressources plus ou moins limitées dont ils disposent et de leurs compétences individuelles.*

Par exemple, la construction d'une maison est une situation d'interaction (en l'occurrence une situation de coopération nécessitant une certaine coordination des actions) définie par l'ensemble des comportements des agents (les ouvriers, l'architecte, le propriétaire, le maître d'œuvre) caractérisés par leurs objectifs (la réalisation de la maison envisagée suivant les différents points de vue des agents) et leurs compétences (le savoir-faire de l'architecte et des différents corps de métier) en tenant compte des ressources disponibles (les matériaux de constructions, les finances, les outils, l'espace constructible, le temps).

Cette notion de situation d'interaction sert à définir des catégories abstraites indépendantes de leurs réalisations concrètes, en dégageant d'une part les principaux invariants que l'on retrouve partout et, d'autre part, les rapports de différence qu'ils entretiennent. A partir de cette définition des catégories, il est possible de les classer, c'est-à-dire de les ordonner les uns par rapport aux autres. Cependant, nous ne nous attacherons pas à considérer toutes les interactions possibles, mais seulement celles caractéristiques de la kénétique à l'heure actuelle. Par exemple, malgré leur importance dans les interactions humaines, nous ne prendrons pas en compte les phénomènes dus aux émotions et aux relations de tendresse ou d'amour.

2.2 Composantes des interactions

Quels sont les différents types d'interaction? Est-il possible de définir des paramètres d'interaction permettant de classer les situations? Dans les sections qui suivent nous allons décrire plusieurs critères par lesquels il sera possible de classer différents types d'interaction et ainsi parvenir à une meilleure compréhension à la fois des potentialités et des problèmes que présente cette notion. Les principales situations d'interaction peuvent être classées par rapport à trois critères: les objectifs ou intentions des agents, les relations que les agents entretiennent envers les ressources qu'ils possèdent ainsi que les moyens (ou compétences) dont ils disposent pour parvenir à leurs fins.

2.2.1 Buts compatibles et incompatibles

Les objectifs et interactions des agents sont évidemment au centre des problématiques de l'interaction: les agents ont-ils des buts concordants, c'est-à-dire allant dans le même sens, ou bien leurs objectifs sont-ils contradictoires, voire opposés?

Par exemple, lorsqu'un couple veut prendre des vacances et que l'un des conjoints veut aller à la montagne quand l'autre désire aller à la mer, les buts sont contradictoires. Il y a aussi contradiction dans les objectifs lorsque deux associés ne sont pas d'accord sur la stratégie à suivre pour développer leur entreprise, l'un voulant augmenter la productivité en licenciant du personnel et le second, au contraire, voulant embaucher parce qu'il estime que l'évolution du marché est en croissance. De même, la plupart des sports et des jeux sont fondés sur une situation de compétition, c'est-à-dire sur le fait que les buts des participants sont incompatibles: tous veulent gagner mais il n'y aura qu'un seul vainqueur. Autrement dit, il y a plusieurs objectifs locaux qui ne peuvent être fusionnés en un seul au niveau global.

D'une manière générale, on dira que des buts sont incompatibles si la satisfaction de l'un entraîne l'insatisfaction de l'autre, la réciproque étant automatiquement vraie.

Définition: *Le but d'un agent A est incompatible avec celui d'un agent B si les agents A et B ont comme buts respectifs d'atteindre les états décrits respectivement par p et q et que $p \Rightarrow \neg q$, c'est-à-dire que: $\text{satisfait}(\text{but}(A, p)) \Rightarrow \neg \text{satisfait}(\text{but}(B, q))$.*

Cette distinction entre buts compatibles et incompatibles est à la base d'une première classification: on dira que des agents sont dans une situation de coopération ou d'indifférence si leurs buts sont compatibles et dans une situation d'antagonisme dans le cas contraire.

Bien que la notion de but soit caractéristique des agents cognitifs, il est possible d'étendre cette notion de compatibilité aux agents réactifs. Lorsqu'il s'agit d'*agents pulsionnels* (cf. chap. 1), il est permis de distinguer des pulsions compatibles et incompatibles en définissant la possibilité de co-satisfaire ou non les désirs de ces agents. Dans le cas d'*agents tropiques*, on ne considérera que leur viabilité et l'on dira, par extension, que les objectifs des agents sont compatibles si le comportement de survie de l'un n'entraîne pas la mort de l'autre. De ce fait, les situations de coopération (ou d'indifférence) sont celles pour lesquelles la viabilité de l'un n'est pas affectée négativement par celle de l'autre. Dans les situations d'indifférence, la viabilité de l'un n'est pas affectée (ni négativement ni positivement) par le comportement de l'autre, alors que dans les situations de coopération, la viabilité de l'un est renforcée par la viabilité de celle de l'autre. Dans le règne animal et végétal, les cas de symbiose sont caractéristiques de coopération réactive. Par exemple, dans certains cas de symbiose animale, lorsqu'un organisme A nourrit un autre organisme B et que ce dernier défend A des prédateurs, on peut dire que ces deux animaux coopèrent puisque leur viabilité est renforcée par la présence et le comportement de l'autre. De ce fait, tout ce qui ressort de la survie d'un groupe est caractéristique de situations de coopération.

L'exemple du modèle Proie-Prédateur est une bonne illustration des mécanismes de coopération et d'antagonisme: les prédateurs, en formant un groupe, coopèrent pour chasser leurs proies, mais ils se situent dans un rapport antagonistes avec les proies: la satisfaction des prédateurs entraînant la mort des proies et la survie des proies l'échec des prédateurs.

2.2.2 Relations aux ressources

Une autre composante importante des interactions est relative aux ressources extérieures dont disposent les agents. On entend par ressources tous les éléments environnementaux et matériels utiles à la réalisation d'une action. Il peut s'agir de valeurs énergétiques, de possibilité financières, d'outils, de matières premières, mais aussi de la prise en compte de notions plus générales telles que l'espace et le temps disponibles pour l'accomplissement d'une tâche. Tout agent a besoin de ressources pour réaliser ses actions, lesquelles se déroulent dans un espace-temps quadridimensionnel, consomment de l'énergie (et de l'argent) et nécessitent des instruments comme catalyseur d'activité.

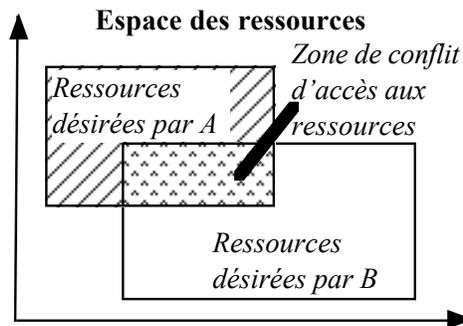


Figure 2.1: L'agent A désire des ressources qui recoupent celles qui sont désirées par B, ce qui engendre un conflit.

Cette quantité de ressources, nécessairement limitée, est à l'origine de conflits. Dans ce cas, les interactions sont perturbantes: du point de vue de chaque agent, l'autre est un gêneur quant à l'accomplissement de ses propres actions. Les conflits apparaissent essentiellement lorsque plusieurs agents ont besoin des mêmes ressources en même temps et au même endroit, c'est-à-dire lorsque les zones correspondant aux ressources dont ils ont besoin s'intersectent (fig. 2.1). Deux ouvriers ayant besoin du même outil, deux robots devant se réapprovisionner à la même prise de courant, deux personnes voulant imprimer en même temps ou deux programmes partageant le même processeur constituent autant de cas de conflits causés par manque de ressources.

Un autre exemple est donné par la circulation automobile, où la simple présence d'autrui constitue une gêne. Dans un embouteillage, c'est parce que l'autre se situe sur mon passage que je suis en conflit avec lui. S'il y avait plus de place ou si le nombre de véhicules était moindre, les conflits disparaîtraient. Le problème vient donc du nombre de véhicules et de la place disponible, c'est-à-dire des ressources mises à la disposition de chacun. Si le trafic est fluide, c'est-à-dire si la densité des voitures n'est pas trop importante, tout se passe bien, mais si cette densité vient à augmenter de manière trop brutale la circulation devient bloquée, les actions de chaque automobiliste étant contrariées par celles des autres.

Les situations conflictuelles peuvent être résolues d'une manière ou d'une autre par l'intermédiaire de mécanismes de coordination d'action et de résolution de

conflit. Certaines de ces méthodes sont particulièrement frustrées: par exemple la loi du plus fort est une technique de résolution de conflit qui définit un ordre de priorité en fonction de la “force” de l’agent, le plus fort dominant le plus faible pour l’accès aux ressources. D’autres méthodes sont plus élaborées. Les techniques de négociation par exemple sont utilisées pour sortir d’une situation de conflit par des compromis qui modèrent les satisfactions et les frustrations des uns et des autres.

Enfin les techniques de coordination d’actions sont utilisées pour anticiper les conflits à venir et faire en sorte qu’ils soient gérés avant même qu’ils n’apparaissent. Elles comportent un ensemble de dispositifs, de règlements et d’actions supplémentaires qu’il est nécessaire d’accomplir pour que les actions des différents agents soient possibles. Les feux de signalisation et le code de la route sont des exemples de dispositifs et de règlements permettant de réguler le trafic routier et de diminuer les risques de conflits entre automobilistes. Par exemple, un feu rouge à un carrefour sert à coordonner les actions des véhicules et à faire en sorte qu’il n’y ait pas de conflits de passage entre les véhicules. En l’absence de système de régulation de trafic, si chaque voiture cherche à passer sans se préoccuper de la gêne qu’elle peut causer aux autres voitures, des conflits apparaîtront, pouvant aller jusqu’à l’accident. Le constat amiable (négociation) ou le passage devant un tribunal (arbitrage) serviront alors à résoudre les conflits d’intérêts causés par l’accident.

Des conflits peuvent naître même si tous les agents sont dans les meilleures dispositions du monde et s’ils décident a priori de collaborer. Par exemple, lorsque plusieurs amis viennent vous donner un coup de main pour déménager et porter les meubles de votre salon, leurs actions ne vous gênent pas. Au contraire, elles sont les bienvenues! Cependant, du fait de l’exiguïté de votre appartement et du nombre de porteurs volontaires, les allées et venues de vos amis doivent être plus ou moins coordonnées pour que les meubles arrivent à bon port et en bon état et que cette assistance ne se transforme pas en désordre perturbateur. Nous discuterons plus en détail des problèmes de coordination au chapitre 8. Il suffit ici de comprendre que les limitations de ressources peuvent être causes de conflit que l’on peut éviter dans une certaine mesure en coordonnant les actions des agents.

2.2.3 Capacités des agents par rapport aux tâches

Le rapport entre les capacités des agents et les tâches à accomplir constitue la troisième composante fondamentale des interactions entre agents. Est-ce qu’un agent peut réaliser seul une tâche ou bien a-t-il besoin des autres pour parvenir à son but? Dans de nombreuses situations, un agent isolé s’avère capable de faire face seul à la réalisation de son but. Par exemple, aller chercher une bière dans le frigo ou du pain à la boulangerie sont des tâches simples qui peuvent être effectuées par une personne seule. D’autres plus complexes peuvent être effectuées par un individu unique, même si leur accomplissement est facilité par la présence d’autres personnes qui s’apportent ainsi mutuellement une aide. Traverser l’Atlantique à la voile, faire un diagnostic médical ou réparer une voiture sont des exemples de telles situations. Enfin, certaines requièrent impérativement les capacités de plusieurs personnes: l’édification d’un monument ou la construction d’une navette spatiale nécessitent les

efforts conjugués et coordonnés de plusieurs individus pour aboutir à la réalisation de l'œuvre commune, aucun n'étant capable de l'accomplir seul. Sans les autres, le maçon, l'électricien et le plombier dans un cas, l'ingénieur en propulsion et l'expert en aérodynamique dans l'autre, sont incapables de mener à bien l'objectif qu'ils se sont assigné.

Dans ces deux dernières situations, les interactions sont bénéfiques, les actions des uns venant aider celles des autres. Le système résultant dispose alors de propriétés nouvelles qui s'expriment parfois comme une fonctionnalité émergente. Une maison ce n'est pas simplement "construire des murs plus mettre des conduites d'eau plus mettre de l'électricité", un avion ne peut se résumer à la juxtaposition d'un fuselage et de moteurs. Dans tous ces cas, la chose produite est un plus par rapport à la simple somme des compétences de chacun des agents et cette somme apparaît dans l'interaction de chaque élément.

2.3 Types d'interaction

Ces trois composantes principales de l'interaction, à savoir la nature des buts, l'accès aux ressources et les compétences des agents, vont permettre de faire une première typologie des situations d'interactions, comme le montre le tableau 2.1, en envisageant tous les cas possibles.

Buts	Ressources	Compétences	Types de situation	Remarques
compatibles	suffisantes	suffisantes	<i>indépendance</i>	situation d'indifférence
compatibles	suffisantes	insuffisantes	<i>collaboration simple</i>	situations de
compatibles	insuffisantes	suffisantes	<i>emcombement</i>	coopérations
compatibles	insuffisantes	insuffisantes	<i>collaboration coordonnée</i>	
incompatibles	suffisantes	suffisantes	<i>compétition individuelle pure</i>	
incompatibles	suffisantes	insuffisantes	<i>compétition collective pure</i>	situations
incompatibles	insuffisantes	suffisantes	<i>conflits individuels pour des ressources</i>	d'antagonismes
incompatibles	insuffisantes	insuffisantes	<i>conflits collectifs pour des ressources</i>	

Table 2.1: Classification des situations d'interactions

2.3.1 Indépendance

Buts compatibles, ressources suffisantes, compétences suffisantes. La situation d'indépendance ne pose aucun problème du point de vue multi-agent et se résume à la simple juxtaposition des actions des agents pris indépendamment, sans qu'il y ait effectivement d'interaction. Par exemple, des personnes qui se croisent dans la rue sachant qu'il y a assez de place pour se croiser ou des ingénieurs qui ne travaillent pas sur le même projet sont dans une situation d'indépendance totale. Il s'agit d'une situation neutre qui ne réclame en fait aucune interaction spécifique du point de vue de la kénétique. Nous n'en parlerons donc pas.

2.3.2 Collaboration simple

Buts compatibles, ressources suffisantes, compétences insuffisantes. La collaboration simple consiste en une simple addition des compétences ne nécessitant pas d'actions supplémentaires de coordination entre les intervenants. Cette situation est caractéristique de systèmes communicants dans lesquels toute l'interaction s'exprime sous la forme d'allocation de tâches et de partage de connaissances. Les systèmes multi-spécialistes (cf. chap. 1), dont les agents ne font que partager des connaissances pour aboutir à la résolution d'un problème, ne font généralement intervenir que de la collaboration simple. Le chapitre 7 traite essentiellement de ces cas de figure.

2.3.3 Encombrement

Buts compatibles, ressources insuffisantes, compétences suffisantes. L'encombrement est caractéristique de toutes les situations dans lesquelles les agents se gênent mutuellement dans l'accomplissement de leurs tâches alors qu'ils n'ont pas besoin les uns des autres. L'exemple caractéristique est celui de la circulation de voitures, de la régulation du trafic aérien, de l'utilisation optimale des ressources telles que la définition d'emplois du temps, la gestion des stocks, le placement des tâches sur un processeurs, etc. La résolution de ce type de problème fait appel à des techniques spécifiques de coordination d'actions dont quelques-unes sont présentées au chapitre 8.

2.3.4 Collaboration coordonnée

Buts compatibles, ressources insuffisantes, compétences insuffisantes. La collaboration complexe suppose que les agents doivent coordonner leurs actions pour pouvoir disposer de la synergie de l'ensemble de leurs compétences. Presque toutes les activités industrielles nécessitant une approche distribuée, telles que le contrôle de réseaux, la conception et la fabrication de produits industriels, la régulation distribuée ou la réalisation de sociétés de robots autonomes un tant soit peu élaborées, se situent dans le cadre de la collaboration coordonnée. Il s'agit donc de la plus complexe des situations de coopération, puisque elle ajoute aux problèmes d'allocation de tâches des aspects de coordination dus aux ressources limitées. Le chapitre 8 en

abordant le problème des coordinations d'actions apportera quelques lumières sur la résolution de ce type de situation.

2.3.5 Compétition individuelle pure

Buts incompatibles, ressources suffisantes, compétences suffisantes. Quand les buts sont incompatibles, les agents doivent lutter ou négocier pour atteindre leurs buts. La compétition "sportive" suppose que les agents ont tous les mêmes ressources à leur disposition et qu'ils sont placés dans des situations initiales identiques. On parle de compétition pure lorsque les ressources ne sont pas limitées, plus exactement lorsque l'accès aux ressources ne constitue pas l'enjeu du conflit. La course à pied est un exemple de compétition pure: chaque joueur jouit d'un espace suffisant et similaire, seules ses caractéristiques sportives intrinsèques feront la différence. Il n'y a pas de problèmes spécifiques d'interaction liés à ce type de situation. Simplement, que le meilleur gagne.

2.3.6 Compétition collective pure

Buts incompatibles, ressources suffisantes, compétences insuffisantes. Lorsque les agents n'ont pas la compétence suffisante, ils doivent se regrouper au sein de coalitions ou d'associations pour parvenir à atteindre leurs objectifs. Ce regroupement est effectué le long d'un double mouvement: le premier tend à lier les individus au sein de groupes unis par des liens de collaboration coordonnée et le second à opposer les groupes entre eux. Un exemple caractéristique de ce type de situation est donné par la compétition par équipe, telle que la course de relais par équipe ou la coupe de l'America en voile, c'est-à-dire des oppositions dans lesquelles les équipes ne se gênent (normalement) pas.

2.3.7 Conflit individuel pour des ressources

Buts incompatibles, ressources insuffisantes, compétences suffisantes. Lorsque les ressources ne peuvent être partagées, on se trouve dans une situation caractéristique de conflit dont les ressources sont l'enjeu, chacun voulant les acquérir pour lui seul. L'objet du conflit peut être un territoire, une position financière dominante, etc. Il existe un grand nombre d'exemples de ce type de situation dans le monde animal ou humain, depuis la défense du territoire chez les animaux jusqu'à l'obtention d'un poste hiérarchiquement supérieur chez les humains.

2.3.8 Conflits collectifs pour des ressources

Buts incompatibles, ressources insuffisantes, compétences insuffisantes. Ce type de situation combine la compétition collective aux conflits individuels pour des ressources. Les coalitions luttent les unes contre les autres pour obtenir le monopole d'un bien, d'un territoire ou d'une position. Toutes les formes de guerres, de chasses, de compétitions industrielles et, d'une manière générale, de conflits collectifs ayant

pour objectif l’obtention d’un territoire ou d’une ressource en sont des exemples caractéristiques.

2.3.9 Niveau d’analyse des situations d’interaction

Evidemment, cette classification ne constitue qu’une première approche dans l’analyse des situations. Il serait possible de l’affiner en décrivant d’autres paramètres pour rendre compte plus en détail de certains cas. En particulier, on pourrait prendre en compte des caractéristiques telles que la reconnaissance cognitive de buts communs ou la volonté de modifier l’état cognitif des autres agents (exemple: chercher à les convaincre), l’existence d’une structure d’autorité et de mécanismes d’influence permettant à un agent de faire agir d’autres agents dans le sens qu’il désire, les types et modes de communication, les capacités cognitives des agents à anticiper les actions des autres, l’existence d’une récompense après l’action (ou au cours de celle-ci), la présence d’activités répétitives ou isolées, etc.

Les situations peuvent être analysées à différents niveaux. Une situation d’interaction complexe est composée de situations plus élémentaires. Par exemple, l’activité d’une entreprise résulte de tout un ensemble de situations plus simples dans lesquelles les employés de l’entreprise interagissent pour produire des biens et des services. On peut ainsi distinguer des *macro-situations*¹ d’interaction qui sont caractéristiques d’une analyse globale de l’activité d’un ensemble d’agents et les *micro-situations* d’interaction qui se situent à un niveau de détail plus fin. Les relations qui existent entre les micro-situations et les macro-situations s’expriment comme le rapport des parties au tout, suivant un point de vue systémique: la macro-situation, tout en étant la résultante des micro-situations qui la compose, introduit un ensemble de problèmes dont la résolution passe par la production d’une quantité de micro-situations elles-mêmes porteuses de problèmes, comme le montre la figure 2.2. Ainsi, une situation considérée au niveau macroscopique comme une situation de collaboration peut produire au niveau microscopique des situations locales de compétition ou d’encombrement. Et inversement une situation de conflit de territoire peut résulter de situations locales de collaboration.

En particulier, nous verrons au chapitre 8 que certaines méthodes utilisées pour gérer des situations de collaboration et d’encombrement passent par la définition de situations locales de compétition. Et c’est pourquoi, même si l’on n’est intéressé que par les situations ne comprenant que des objectifs compatibles, il ne faut pas négliger l’étude des situations dans lesquelles les agents possèdent des buts incompatibles. Il n’en est pas moins vrai que les situations de coopération constituent un ensemble majeur de situations d’interaction et nous ne traiterons pas dans cet ouvrage des problèmes liés aux situations faisant intervenir des buts antagonistes.

Par la suite, on supposera:

¹Les termes “macro” et “micro” ne renvoient pas à des échelles globales mais décrivent ici des positions relatives. Une situation peut donc apparaître comme “macro” d’un certain point de vue (par exemple l’entreprise est considérée comme “macro” par rapport aux interactions de ses employés) et “micro” d’un autre point de vue (une entreprise interagit avec d’autres entreprises pour constituer des structures économiques plus vastes).

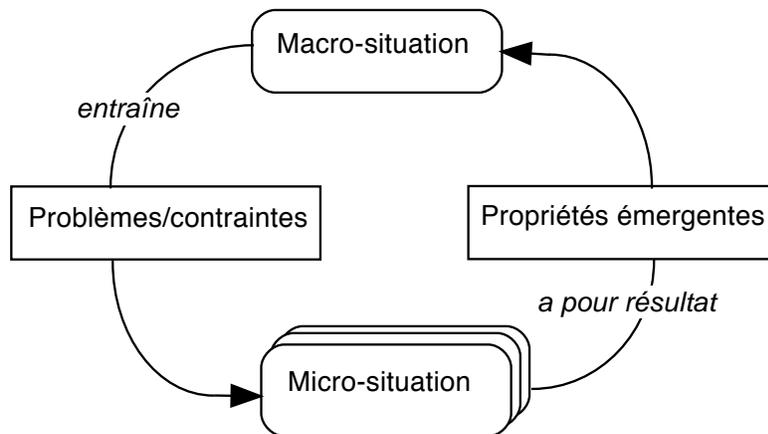


Figure 2.2: Les actions des agents qui se situent au niveau micro d'un système produisent des macro-situations relatives engendrant des contraintes au niveau micro.

1. que les buts des agents sont compatibles;
2. que les agents sont bienveillants et qu'ils cherchent à s'aider les uns les autres ou à parvenir à un compromis si des intérêts portant sur des ressources sont en jeu.

Bien que le problème de la coopération entre agents ait été soulevé depuis longtemps, peu de théories ont tenté, avant l'apparition des systèmes multi-agents, de décrire de manière opératoire les diverses formes de coopération ni de montrer la relation entre les individus, les tâches qu'ils accomplissent et l'intérêt global pour le groupe². La plupart des travaux traitant de la coopération portent sur l'intérêt qu'il y a ou non à coopérer avec autrui, comme dans le cadre du dilemme du prisonnier (Axelrod 1992), ou sur la manière de communiquer dans les groupes restreints en psychologie sociale (Stoetzel 1978). Mais alors que ces travaux mettent l'accent soit sur la décision et l'intérêt qu'il y a de coopérer pour un agent, soit sur les rapports d'autorité et sur les relations affectives qui amènent les individus à coopérer et à travailler ensemble, les systèmes multi-agents reformulent le problème en mettant en avant les caractéristiques cognitives et comportementales nécessaires à la mise en œuvre d'un travail collectif.

La coopération est souvent avancée comme l'un des concepts clés de l'intelligence artificielle distribuée (Demazeau et Müller 1991). Actuellement, les travaux les plus en pointe en ce qui concerne la coopération dans les systèmes multi-agents sont dus à Lesser et Durfee (Durfee et al. 1987), Galliers (Galliers 1991), Castelfranchi et Conte (Conte et al. 1991) et Bouron (Bouron 1992). Mais la plupart ne fournissent que des réponses partielles et ne s'intéressent de toute manière qu'à des agents cognitifs, c'est-à-dire à des agents capables d'avoir des représentations et de planifier leurs

²Il existe néanmoins tout un courant de travaux philosophiques portant sur la coopération et l'action collective dans une perspective cognitive. On pourra se référer à (Livet 1994) pour une présentation récente et en français de ces réflexions.

actions. Les travaux sur la coopération réactive, c'est-à-dire sur la coopération par des agents qui ne font que réagir aux stimuli provenant de leur environnement, n'en sont encore qu'à leurs débuts. Les travaux de Deneubourg (Deneubourg et al. 1993), Theraulaz (Theraulaz et al. 1991), Steels (Steels 1994), Drogoul (Drogoul 1993) et Ferber (Ferber 1995) témoignent cependant de l'importance du domaine.

2.4 Formes de coopération

Il existe plusieurs points de vue sur la coopération, selon que l'on considère que la coopération est une attitude des agents qui décident de travailler en commun ou que l'on se pose comme un observateur qui interprète a posteriori les comportements en les qualifiant de coopératifs ou non à partir de critères sociaux (ou physiques), tels que l'interdépendance des actions ou le nombre de communications effectuées.

2.4.1 La coopération comme attitude intentionnelle

Dans le premier cas, la coopération est caractéristique d'une *attitude* (posture) des agents. On dit que des agents coopèrent s'ils s'engagent dans une action commune après avoir identifié et adopté un but commun. La création d'une association, telle qu'une association de défense de l'environnement par exemple, correspond à ce schéma: les membres de l'association ont identifié un objectif commun, lutter contre les sources de pollution, et ils s'engagent à participer à une tâche commune, la défense de la nature. La constitution de l'association résulte de la prise de conscience de l'existence d'un but commun et de l'engagement de chacun de ses membres.

De même, lorsque des individus acceptent de travailler dans une entreprise, ils s'engagent pour participer à un travail en commun, en considérant non seulement que leurs buts sont compatibles avec ceux de l'entreprise, mais aussi que les employés comme l'organisme qui les emploie peuvent y trouver leur intérêt, l'entreprise en utilisant les compétences de ceux qu'elle engage et les employés en recevant une rémunération.

Le problème de l'engagement envers un but collectif est appelé *adoption de but* (goal adoption) par J. Galliers (Galliers 1991), C. Castelfranchi et R. Conte (Conte et al. 1991), laquelle est considérée comme un élément essentiel de l'activité sociale. En particulier pour J. Galliers, il y a coopération si les agents s'engagent dans une action et identifient un but commun, c'est-à-dire reconnaissent que les autres agents sont engagés dans le même but.

Cependant, toute forme de coopération est-elle le fruit d'une intention de coopérer, et toute intention de coopérer conduit-elle nécessairement à la coopération? Ainsi, dans l'histoire suivante:

Gaétan désire aider Ambroise qui construit un hangar pour sa voiture. Mais Gaétan n'est pas très adroit de ses mains, et il n'arrive pas à monter les moellons comme le lui demande Ambroise. De ce fait Ambroise doit passer derrière Gaétan pour refaire le travail de ce dernier et voit donc son efficacité réduite par rapport à ce qu'il faisait quand il était seul.

Peut-on dire que Gaétan et Ambroise sont dans une situation de coopération? D'après Galliers (et la plupart des chercheurs en IAD travaillant sur des agents cognitifs), Gaétan et Ambroise coopèrent parce qu'ils sont conscients que leurs actions sont engagées dans un but commun, ou une intention commune.

Cette conception présente deux inconvénients: d'une part, même si le résultat de la coopération est moins bon que la performance des agents pris individuellement, on prétend qu'il existe une coopération et, d'autre part, toute possibilité pour des agents réactifs (qui n'ont pas d'intention explicite et encore moins de modèles des autres) de pouvoir coopérer est supprimée. Pourtant on reconnaît souvent (en éthologie en particulier) que des insectes sociaux tels que des fourmis qui travaillent ensemble pour récupérer de la nourriture au dehors de leur nid coopèrent même si elles n'en ont pas conscience (Fresneau 1994). De manière plus générale, des agents réactifs qui parviennent à accomplir des tâches complexes à plusieurs peuvent-ils être coopératifs? Si l'on répond positivement, alors il faut supposer que la coopération ne résulte pas seulement d'une attitude, même si une attitude pour coopérer peut être un très bon catalyseur de coopération, mais qu'elle doit être comprise en termes de comportement et de résultat. Dans ce cas, la coopération n'est plus directement le produit d'une *intention de coopération* mais le *bénéfice positif* obtenu grâce aux interactions des agents.

2.4.2 La coopération du point de vue de l'observateur

C'est pourquoi, certains auteurs, (Durfee et al. 1989) et (Bouron 1992) notamment, considèrent la coopération comme une qualification de l'activité d'un ensemble d'agents par un observateur extérieur qui n'aurait pas accès aux états mentaux des agents. Par exemple, si l'on qualifie le comportement des fourmis de coopératif, c'est parce que, en tant qu'observateur, on observe un certain nombre de phénomènes que l'on utilise comme des indices d'une activité de coopération. L'idée d'*indice de coopération* (MIRIAD 1992) est particulièrement intéressante, car elle permet de s'affranchir des caractéristiques internes des agents et de ne s'occuper que de leur comportement observable. Par exemple, voici un ensemble d'indices proposés par T. Bouron pour qualifier les activités de coopérations:

1. La coordination d'actions, qui concerne l'ajustement de la direction des actions des agents dans le temps et l'espace.
2. Le degré de parallélisation, qui est fonction de la répartition des tâches et de leur résolution concurrente.
3. Le partage des ressources, qui concerne l'utilisation des ressources et des compétences.
4. La robustesse, qui concerne l'aptitude du système à suppléer la défaillance d'un agent.
5. La non-redondance des actions, qui caractérise le faible taux d'activités redondantes.

6. La non-persistence des conflits, qui témoigne du faible nombre de situations bloquantes.

La définition d'un tel ensemble repose en premier lieu sur des considérations d'observabilité et de quantifiabilité. Ces indices doivent pouvoir être examinés et mesurés pour que l'on puisse caractériser le degré de coopération qui existe entre différents agents. Mais l'application de ces contraintes n'est pas évidente: Quelle mesure donner à la coordination des actions? Comment quantifier la robustesse d'un système? Proposer des critères objectifs et mesurables est fondamentale si l'on veut donner un statut précis à la coopération. Autrement, on risque de spécifier ce concept en termes d'autres concepts tout aussi vagues, ce qui empêcherait toute définition de critères effectifs de coopération.

Cet ensemble d'indices doit de plus être cohérent: la mesure de la robustesse est plutôt une caractéristique de la capacité d'un système à s'adapter et la non-redondance des activités vient en opposition avec le degré de parallélisation des tâches: plus on parallélise, plus on tend à ce que les tâches soient dupliquées pour être exécutées plus rapidement et plus on introduit de redondance. Enfin, ces indices doivent être hiérarchisés: le degré de parallélisation ou la coordination d'actions sont des conséquences d'indices plus fondamentaux tels que le partage des ressources ou l'évitement de conflits.

Peut-on alors donner un ensemble minimal d'indices à partir desquels il serait possible de construire d'autres indices plus complexes? Nous pensons que cela est possible et, par la suite, nous ne retiendrons que deux indices qui semblent nécessaires et suffisants pour caractériser s'il y a ou non activité de coopération: l'efficacité du travail en groupe, d'une part, et l'existence de mécanismes de résolution de conflit d'accès à des ressources, d'autre part. Ces critères sont facilement mesurables et viennent s'intégrer naturellement à la typologie des situations d'interaction présentées précédemment.

Définition: *On dira que plusieurs agents coopèrent, ou encore qu'il y a une situation de coopération, si l'une des deux conditions est vérifiée:*

1. *L'ajout d'un nouvel agent permet d'accroître différentiellement les performances du groupe.*
2. *L'action des agents sert à éviter ou à résoudre des conflits potentiels ou actuels.*

Le premier indice recouvre un critère différentiel: si les agents coopèrent, un nouvel agent est considéré comme une aide pour l'ensemble du groupe. Lorsque ce critère est vérifié, on dit que les agents *collaborent* ou qu'ils sont dans une *situation de collaboration*. Accroître les performances peut avoir pour objectif de simplement améliorer un certain paramètre, tel que la quantité de minerai rapportée à la base pour des robots explorateurs. Mais, plus fondamentalement, le gain en performance sous-tend un objectif plus primordial qui est celui de la survie. La survie, qu'il s'agisse de la préservation d'une espèce, d'un écosystème, d'un groupe ou de chacun

de ses membres, constitue la finalité ultime des systèmes naturels où les problèmes d'adaptation et de reproduction sont en jeu et, en particulier, de tous les systèmes biologiques. Par exemple, les membres d'une horde coopèrent parce qu'il leur est plus facile de survivre et de se développer en commun qu'individuellement. Les éthologistes considèrent en effet que la spécialisation des individus d'une collectivité et la coopération qui lui est sous-jacente sont de bons indices de leur développement sur l'échelle de l'évolution. Par exemple, les colonies de fourmis les plus primitives vivent généralement dans des nids de quelques unités, alors que les fourmis les plus évoluées peuvent comporter plusieurs millions d'individus (Corbara 1991).

Si au contraire l'introduction d'un agent n'améliore pas les performances du groupe, cet agent sera considéré comme un gêneur pour les activités collectives. Dans ce cas, le second critère permet néanmoins de reconnaître qu'il y a coopération même s'il n'y a pas d'accroissement différentiel de performances. Les agents coopèrent alors pour faire en sorte que la diminution de performances ne soit pas trop radicale. C'est généralement le cas lorsque la quantité de ressources est limitée: l'introduction de nouveaux agents peut avoir des conséquences négatives sur le fonctionnement du groupe et les actions coopératives, telles que l'évitement, l'établissement de techniques de résolution de conflit, la constitution d'organisations de production et de transport de ressources ou, tout simplement, la formation de files d'attentes, sont alors des réponses collectives pour limiter la dégradation des performances individuelles et collectives. Dans ce cas, on dira que l'on se trouve dans une *situation de résolution de conflit*. Ce type de situation est caractéristique des situations d'encombrement, telles que la circulation routière ou aérienne, l'accès à un photocopieur par un ensemble de personnes, la recherche d'un taxi à un aéroport, l'accès à un point d'eau ou à un processeur par un ensemble de tâches, etc. Il existe de nombreuses manières de résoudre des conflits. Certaines reposent simplement sur la définition d'actions permettant d'éviter a priori l'apparition de conflits, tels que les comportements d'évitement, l'établissement de dominance ou la définition de règles de priorité, d'autres enfin tentent d'annuler les conflits qui apparaissent par l'arbitrage, la négociation, l'utilisation des relations de dominance ou par la destruction pure et simple des participants au conflit.

Néanmoins, bien que distinguées ici, les situations de collaboration et celles de résolution de conflit s'avèrent souvent très liées: s'il se pose des problèmes de survie, l'augmentation des performances permettra généralement de mieux survivre dans un univers difficile³. De même l'évitement de conflits permet soit de faire survivre les membres du groupe (par exemple en évitant des collisions ou des luttes), soit d'améliorer les performances par rapport à un système sans coordination. Par exemple, un ensemble de déménageurs forme a priori un collectif en situation de collaboration, puisque l'ajout d'un nouveau déménageur améliore généralement leurs performances. S'ils sont très nombreux, ils peuvent arriver à se gêner dans les escaliers et les passages étroits: ils se trouvent alors dans une situation de résolution de conflits et ils doivent coopérer pour résoudre des conflits d'accès à

³Dans certains cas, l'amélioration des performances individuelles peut avoir un effet négatif sur le groupe. Si, par exemple, des herbivores sont capables de mieux exploiter leur territoire, ils peuvent aboutir à l'épuisement de leur ressource et entraîner leur perte.

l'espace et faire en sorte que leurs performances globales ne soient pas trop affectées par ces problèmes. Cette définition de la coopération qui s'appuie totalement sur le point de vue de l'observateur, permet de caractériser de manière objective ce qu'est une situation de coopération en termes d'amélioration de performances, sans faire intervenir de considérations sur ce qui se passe dans l'esprit (mind) des agents. La coopération est alors ramenée à des critères simples et observables. De ce fait, si l'on peut définir la *coopération intentionnelle* comme une situation de coopération dans laquelle les agents ont l'intention de coopérer, ce qui suppose bien évidemment qu'il s'agisse d'agents cognitifs et capables d'intentions, on parlera de *coopération réactive* lorsque les agents ne sont pas doués d'intentions explicites, mais que leur comportement collectif répond au moins à l'un des deux critères mentionnés ci-dessus. On peut donc parler de coopération même si les agents n'ont aucune représentation du monde dans lequel ils évoluent et des autres agents et donc aucune conscience des conséquences de leurs actes, ni même bien souvent de la présence des autres agents. Par exemple, des termites construisent un nid en agissant simplement en réaction aux différents stimuli qu'ils reçoivent. Leur technique de construction repose simplement sur des phénomènes perceptifs: les ouvriers déposent des boulettes de terre et d'excrément préférentiellement là où se trouvent d'autres boulettes. De ce fait, ils développent des colonnes, puis en déposant des boulettes plutôt du côté où se trouve l'odeur d'autres termites, ils en viennent à former des arches (Wilson 1971). Dans ce cas, ils vaquent à leurs occupations sans qu'il y ait d'intention à coopérer et même sans qu'il y ait interaction directe entre les termites, la coopération intervenant comme un *effet secondaire* des actions de la population des termites dans sa totalité (on pourra se référer à (Theraulaz 1991) pour un développement semblable dans le cadre de la construction de nids d'abeilles).

Il est possible de caractériser trois familles d'indices correspondant aux trois grandes fonctions de la coopération que sont l'amélioration de la survie, l'accroissement de performances et la résolution de conflits.

2.4.3 L'amélioration de la survie

Les *indices de survie* reflètent la capacité d'un individu ou d'un groupe à maintenir son unité fonctionnelle alors qu'il est confronté à des forces qui tendent à détruire cette unité. On peut distinguer deux types de survie: la survie de l'individu, c'est-à-dire la capacité pour un agent à persister dans son être et la survie collective qui correspond au maintien du groupe. Si la première est relativement facile à identifier, la seconde apparaît comme étant plus difficile à vérifier et à définir. Entend-on par survie du collectif la survie de l'ensemble de ses membres ou la survie de l'organisation issue du rassemblement, peut-être temporaire, d'un ensemble d'agents? Dans le premier cas, la situation est simple car on peut se ramener à une analyse en termes de survies individuelles, alors que dans le second cas il s'avère parfois difficile de savoir à quel moment l'organisation collective a disparu. Néanmoins, on peut souvent se donner des critères arbitraires pour décréter qu'une organisation collective a cessé d'exister. On dira par exemple qu'une entreprise survit tant qu'elle n'a pas déposé son bilan ou qu'un groupe de robots existe tant qu'il en

comporte au moins deux.

Il est possible de définir la capacité individuelle de survie d'un agent de deux manières: soit en considérant sa probabilité de survie dans un environnement donné, soit en analysant son bilan énergétique, c'est-à-dire le rapport qui existe entre la quantité d'énergie qu'il absorbe et celle qu'il dépense par unité de temps: lorsque ce rapport est supérieur à 1, l'individu ou le groupe sont capables de survivre et même éventuellement d'investir leur surplus d'énergie à des tâches supplémentaires non rentables immédiatement, mais qui permettront par la suite d'améliorer encore ce bilan énergétique. La première est plus adaptée à l'analyse différentielle des comportements chez des agents très simples, alors que la seconde est plus générale mais aussi plus compliquée à manipuler puisqu'elle fait intervenir des facteurs économiques et suppose que l'on soit capable d'estimer les quantités "énergétiques" entrantes et sortantes.

Nous allons maintenant montrer sur un exemple d'agents chasseurs comment des mécanismes simples de coopération peuvent amener les agents à augmenter leur capacité de survie. Supposons que des agents prédateurs puissent survivre s'ils sont capables de trouver à chaque unité de temps une proie. La probabilité de trouver une proie est donnée par P_{ind} pour une espèce de prédateur et pour un environnement donné. Si les individus sont isolés, la probabilité de survie est alors directement égale à la probabilité de trouver une proie:

$$\sigma_{ind} = P_{chasse}$$

En se regroupant, les prédateurs augmentent leur probabilité de survie si les proies qu'ils ramènent peuvent nourrir un plus grand nombre d'agents que de chasseurs. Ainsi, la probabilité de survie pour un agent s'il y a regroupement est égale à

$$\sigma_{regroup} = P_{chasse} * k/n * c$$

où k représente le nombre d'agents qu'une proie peut nourrir, n est le nombre d'agents que contient le groupe et c est le nombre de chasseurs. La survie par regroupement est donc améliorée par rapport à la survie individuelle si $\sigma_{regroup} > \sigma_{ind}$, c'est-à-dire si $k * c > n$, ou autrement dit, si la quantité de nourriture est plus que suffisante pour le nombre d'agents concernés. Le regroupement, en permettant le partage des proies, a donc amélioré la capacité de survie des agents. Alors que tous les agents consomment, seuls les chasseurs sont des producteurs. Il est possible d'accroître les performances d'un tel système en augmentant la productivité des chasseurs, soit en attrapant des proies plus grosses, soit en améliorant la probabilité de trouver une proie par rapport à un agent isolé.

2.4.4 L'accroissement des performances

Les *indices de performance* traduisent la capacité d'un agent ou d'un groupe d'agents à accomplir des tâches définies par des tiers. L'observateur peut appartenir au groupe, par exemple être l'analyste financier d'une entreprise, ou être extérieur au groupe, s'il s'agit du programmeur de ses robots. Les indices de performance sont

variés, mais ils doivent traduire les caractéristiques intrinsèques d'un collectif. Par exemple, pour un groupe de robots récupérateurs de minerai, on pourra donner comme indice de performance la quantité de minerai retournée par unité de temps ou bien le nombre d'actions (mouvements) nécessaire pour retourner une quantité de minerai donnée. Pour un système de coordination d'actions permettant à des avions d'éviter toute collision, on pourra donner comme indice la déviation entre les trajectoires effectuées et les trajectoires idéales issues des plans de vols (Zeghal et al. 1993). Enfin, si cela est possible il s'avère toujours intéressant de revenir à des considérations énergétiques et d'interpréter les indices de performance comme des indices de rendements énergétiques.

Dans le cas de robots récupérateurs de minerai, le regroupement associé à la communication peut améliorer la productivité de ces robots par rapport à un même ensemble de robots non coopérants. Pour ramener du minerai, les robots doivent d'abord trouver des zones de minerai avant d'en rapporter des échantillons. Supposons que nous n'ayons initialement qu'un seul robot. Au départ, il cherche les zones de minerai au hasard, puis s'il en détecte une il se dirige vers elle, remplit sa benne de minerai et retourne à la base. On peut améliorer son efficacité en augmentant le nombre d'agents. Avec plusieurs robots non coopérants chaque robot ne fait qu'augmenter linéairement les performances du groupe: il n'y a pas d'augmentation de la productivité. Si au contraire, dès qu'un robot trouve du minerai, il en avertit les autres, soit en donnant directement la position, soit en laissant des marques sur le sol (cf. chap. 8), alors il est possible de mobiliser un plus grand nombre de robots à la tâche de transport de minerai et donc d'augmenter la productivité de chacun.

Amplification qualitative et quantitative

La coopération consiste donc à amplifier les capacités des agents pris isolément et à augmenter leurs performances. Cette augmentation peut être *quantitative* comme nous venons de le voir, mais elle peut aussi être *qualitative*, c'est-à-dire rendre possibles des actions impossibles à réaliser par des agents isolés.

Les collaborations permettant d'augmenter qualitativement les performances des agents sont essentielles car sans elles l'action n'est tout simplement pas possible. Supposons par exemple que l'on veuille réaliser un magazine. Un agent qui ne sait qu'écrire des articles mais ne sait pas mettre en page ni imprimer ne pourra pas à lui seul obtenir le résultat désiré. Il lui faut l'aide d'un maquettiste et d'un imprimeur. Et cela est vrai aussi pour les agents informatiques: un logiciel qui sait calculer mais ne sait pas imprimer aura besoin des services d'une imprimante et d'un logiciel d'impression.

Les performances qualitatives des agents sont donc fondamentales pour autoriser des performances qualitatives de groupe, mais il ne faut pas négliger les performances quantitatives car elles permettent parfois des résultats qualitatifs impossibles sans cette collaboration. Par exemple, si un robot transporteur veut porter un objet qui pèse 100 kg et qu'il ne dispose que d'une force lui permettant de porter 70 kg, il sera totalement incapable d'accomplir son action: si l'objet est d'un seul tenant,

il est impossible de n'en porter que 70%! De ce fait, l'aide d'autres robots, bien que n'apportant qu'une amplification initialement quantitative, rend possible une action et transforme cette aide quantitative en collaboration qualitative. On dit alors que l'on a à faire à un *effet de masse*, lié à une quantité critique d'agents. Ces phénomènes sont dus à des caractéristiques non linéaires de la collaboration par addition des capacités des différents agents. Cette notion est bien connue en chimie et en physique nucléaire où certaines réactions ne peuvent avoir lieu que si la concentration atteint un certain seuil.

Inversement, il est possible d'obtenir des avantages quantitatifs à partir d'amplification qualitative. C'est alors la présence d'un agent supplémentaire doué de compétences nouvelles, qui entraîne une amélioration des performances de l'ensemble de l'organisation. Par exemple, la présence d'un agent de police à un carrefour permet généralement d'améliorer la fluidité du trafic. C'est la compétence de contrôle que possède cet agent qui permet d'assurer une augmentation des capacités du système. De même, dans le cas de robots transporteurs de minerai, l'introduction d'un robot capable de construire des routes permettra d'améliorer la vitesse de déplacement des robots et ainsi d'accroître les performances du groupe. On dira qu'il s'agit d'un phénomène de *catalyse*, par analogie avec les phénomènes chimiques où la seule présence de certains composants chimiques, les catalyseurs, permet d'améliorer les vitesses de réaction, ces catalyseurs n'intervenant pas directement dans les réactions.

En conséquence, on peut noter que les amplifications, par le biais de la coopération, peuvent être quantitatives ou qualitatives et que les avantages obtenus peuvent être eux aussi qualitatifs ou quantitatifs. On obtient alors le tableau 2.2 qui représente les quatre situations de collaboration possibles en fonction des paramètres d'amplification et d'avantages.

Avantages Amplification	Qualitatif	Quantitatif
Qualitative	Collaboration qualitative	Catalyse
Quantitative	Effet de masse	Collaboration quantitative

Table 2.2: Les quatre formes de collaboration en fonction de l'amplification produite et des avantages obtenus

2.4.5 La résolution de conflit

Du fait de leur autonomie, c'est-à-dire de leur capacité à déterminer leur propre comportement, les agents sont amenés à se trouver dans des situations où leurs intérêts peuvent être contradictoires, ils sont alors en situation objective de conflits. Ces situations proviennent essentiellement d'un problème d'accès à des ressources limitées: deux agents désirent au même moment quelque chose qui ne peut être partagé ou pour lequel toute répartition diminue ce que l'un des agents aurait pu obtenir si l'autre n'avait pas été présent. Par exemple, lorsque deux personnes désirent faire

leurs photocopies sur la même machine au même moment, elles se trouvent en situation conflictuelle. Il en est de même de véhicules qui se croisent à un carrefour, d'avions qui risquent d'entrer en collision, de personnes qui veulent acquérir le même bien, de nations qui veulent occuper le même territoire, de professeurs qui veulent avoir le même cours au même moment dans la même salle de classe ou d'entreprises qui cherchent à obtenir de plus grandes parts de marchés. Dans tous ces cas, les ressources sont limitées et il sera nécessaire d'utiliser des techniques de résolution de conflit pour gérer cette opposition.

Les situations conflictuelles sont à la fois l'effet et la cause d'interactions. Elles naissent d'un manque de ressources et nécessitent des interactions supplémentaires pour sortir de ces conflits, qu'il s'agisse de techniques de négociation, d'arbitrage ou d'utilisation d'un règlement, voire de recours à la compétition et à la force.

Bien qu'il existe des situations pures, qui sont uniquement conflictuelles ou synergiques, la plupart des situations naturelles présentent à la fois des composantes conflictuelles et synergiques. Par exemple, dans une niche écologique, les individus de différentes espèces coopèrent, puisque leur survie dépend des cycles de nutrition qui font intervenir les autres espèces de la niche, mais ils sont aussi adversaires puisqu'il y a compétition entre les espèces voisines pour l'acquisition des ressources de l'écosystème. D'une manière générale, les systèmes complexes sont fondés sur une intégration des situations synergiques et conflictuelles, les premières assurant l'augmentation des performances et les secondes permettant la sélection des agents concernés.

Les *indices conflictuels* traduisent très simplement le nombre d'agents en situation de conflit (on pourra compter le nombre effectif de collisions entre des avions — dans un simulateur de préférence — sans ou avec coordination d'actions) ou le nombre d'individus voulant accéder à la même ressource au même instant (le nombre de processus en attente d'être exécutés sur un processeur par exemple).

2.5 Les méthodes de coopération

Comment coopérer? Si nous avons étudié les conditions de la coopération, nous n'avons encore rien dit des moyens que l'on peut mettre en œuvre pour coopérer. Ces moyens, que nous appellerons *méthodes de coopération* sont au nombre de six: le regroupement et la multiplication, la communication, la spécialisation, la collaboration par partage des tâches et des ressources, la coordination d'actions, la résolution de conflit par arbitrage et négociation.

2.5.1 Le regroupement et la multiplication

La première méthode est évidemment la plus évidente et, pour cela, souvent oubliée. Elle consiste tout simplement pour les agents à se rapprocher physiquement, c'est-à-dire à constituer soit un bloc plus ou moins homogène dans l'espace, soit un réseau de communication permettant à plusieurs agents de se comporter comme s'ils étaient physiquement les uns à côté des autres. Par exemple, le monde animal utilise

le regroupement pour assurer un grand nombre de besoins et en particulier pour améliorer la sécurité de chacun des membres en réalisant une défense de groupe (la masse est plus solide, plus difficile à attaquer), en bénéficiant d'une vigilance accrue (quelques-uns font la garde pendant que les autres peuvent vaquer sans risque à d'autres occupations) et enfin en augmentant la probabilité que, si un prédateur attaque, ce soit l'un des autres membres du groupe qui disparaisse.

Vivre en communauté facilite aussi la prise de nourriture puisqu'il suffit qu'un des membres trouve quelque chose d'intéressant pour que tous les autres en profitent. Par exemple, dès qu'un goéland plonge et ramène du poisson (ou des déchets de nourriture abandonnés par un bateau) les autres goélands se regroupent auprès du premier et bénéficient des découvertes du premier. La chance de trouver un compagnon sexuel pour la reproduction est elle aussi améliorée. La vie en groupe facilite aussi l'acquisition de comportements complexes en permettant aux petits de se confronter aux adultes et d'avoir un grand nombre de modèles de comportement à leur disposition qu'ils peuvent alors imiter.

Enfin, les groupes simplifient les problèmes de navigation comme le montrent les formations d'oiseaux migrateurs, les bancs de poissons ou les troupes d'éléphants. L'effort de navigation est globalement diminué si un seul (ou un petit nombre) d'entre eux seulement décide de l'endroit où aller, les autres suivant le mouvement.

Finalement, on peut considérer qu'un groupe agit comme un organisme distribué. Il y a les spécialistes de l'exploration de nourriture, ceux de la défense ou de la chasse, ceux qui assurent la reproduction et ceux enfin qui guident le groupe pour qu'il trouve des contrées qui seront favorables au groupe. Tous les membres agissent à la fois pour leur bien et pour celui du groupe, car, en s'occupant de la société tout entière, ils s'occupent d'eux et de leur descendance.

D'autre part, la multiplication, c'est-à-dire la simple augmentation quantitative des individus dans un système donné présente des avantages considérables, tant du point de vue de l'augmentation des performances que de celui de sa fiabilité. Par exemple, dans le cas des robots explorateurs, le fait de disposer d'un grand nombre de robots n'est pas neutre quant à la fiabilité du système. Si la performance globale en cas d'absence de pannes est égale à la somme des performances individuelles, c'est-à-dire que l'augmentation différentielle de performances est nulle, il n'en est plus de même si les agents peuvent avoir des pannes. Si l'on ne dispose que d'un seul agent, toute panne peut avoir des conséquences fâcheuses car la tâche ne pourra pas être effectuée. Au contraire, si l'on dispose de plusieurs agents, la redondance permettra d'accroître la fiabilité du système (sans qu'il y ait nécessairement augmentation de la productivité individuelle).

2.5.2 La communication

Le système de communication qui lie un ensemble d'agents agit comme une sorte de système nerveux qui met en contact des individus parfois séparés. La communication en effet agrandit les capacités perceptives des agents en leur permettant de bénéficier des informations et du savoir-faire des autres agents. Les communications sont indispensables à la coopération et il est difficile de concevoir un système

d'agents coopérants s'il n'existe pas un système permettant aux agents d'échanger des informations ou de véhiculer des requêtes. Dans les systèmes cognitifs, les communications s'effectuent par envois de messages, alors que dans les systèmes réactifs, elles résultent de la diffusion d'un signal dans l'environnement. La communication constitue l'un des moyens fondamentaux pour assurer la répartition des tâches et la coordination des actions. Ses différents aspects seront étudiés au chapitre 6.

2.5.3 La spécialisation

La spécialisation est le processus par lequel des agents deviennent de plus en plus adaptés à leurs tâches. Il est souvent difficile de faire des agents qui soient spécialisés dans toutes les tâches. La réalisation performante d'une tâche suppose souvent des caractéristiques structurelles et comportementales qui ne peuvent permettre d'effectuer d'autres tâches avec efficacité. Par exemple, un robot très adapté dans l'excavation et la recherche de minerai en profondeur, dont toute la structure est dédiée au forage de trous, ne sera pas nécessairement très bien adapté au transport de minerai ou à sa détection, qui nécessite des robots capables de passer n'importe où pour diminuer le temps de transport. La spécialisation n'est pas nécessairement le fruit d'un choix a priori. Des agents initialement totipotents peuvent se spécialiser progressivement dans l'accomplissement de leur tâche, ce qui revient à leur associer des rôles temporaires par une sorte d'adaptation individuelle. Cette spécialisation peut être bénéfique à la collectivité en augmentant la capacité du groupe à résoudre plus rapidement un problème semblable. Par exemple, dans le problème du tri collectif (Deneubourg et al. 1991) dans lequel des agents doivent placer en tas des objets "bleus" ou "rouges" à partir d'une répartition aléatoire, leur comportement consiste simplement à se déplacer aléatoirement, et, s'ils rencontrent un tas d'objet, à le prendre et à le redéposer près d'un tas de même couleur. A. Drogoul a montré que, si l'on utilise des agents capables de renforcer leur tendance à s'occuper préférentiellement d'une couleur, alors une société spécialisée est plus efficace qu'une société d'agents totipotents (Drogoul 1993), du fait de la diminution des errements aléatoires résultant d'une meilleure sensibilité à un type particulier d'objet.

2.5.4 La collaboration par partage de tâches et de ressources

La collaboration consiste à travailler à plusieurs sur un projet, une tâche commune. Nous appellerons collaboration l'ensemble des techniques permettant à des agents de (se) répartir des tâches, des informations et des ressources de manière à réaliser une œuvre commune. Résoudre un problème de collaboration consiste donc à répondre à la question "qui fait quoi?" par rapport à un travail donné. Il existe de nombreuses manières de répartir des tâches et des moyens.

Dans les systèmes d'agents cognitifs, l'allocation des tâches passe par des mécanismes d'offre et de demande. On distingue alors les mécanismes de répartition centralisée, dans lesquels un agent coordinateur centralise les offres et les demandes et les répartit ensuite au mieux, des approches distribuées dans lesquelles tout agent peut être à

la fois offrant ou demandeur, sans qu'il existe un organe centralisateur. Deux techniques alors s'opposent: celles qui s'appuient sur des réseaux d'accointances (Gasser et al. 1987), c'est-à-dire sur des représentations mutuelles des capacités de chacun et celles qui reposent sur la notion de marché, par des techniques d'appel d'offre dont la plus connue est le "réseau contractuel" (contract net) (Davis et Smith 1983).

On peut être surpris de penser qu'il est aussi possible de parler d'allocation de tâches dans le cas d'agents réactifs. En effet, on pourrait supposer que leurs capacités limitées ne leur permettraient pas de mettre en œuvre des mécanismes aussi complexes que les algorithmes que nous venons d'analyser. Pourtant, il est possible de faire en sorte, d'une part, que les tâches soient distribuées aux agents et, d'autre part, que les agents eux-mêmes se spécialisent dans l'accomplissement de telle ou telle tâche.

Le chapitre 7 est entièrement consacré à la problématique de la collaboration et développe les différentes techniques d'allocation et de répartition des tâches, des informations et des ressources tant dans les systèmes multi-agents cognitifs que réactifs.

2.5.5 La coordination d'actions

Gérer un ensemble d'agents suppose l'exécution d'un certain nombre de tâches supplémentaires qui ne sont pas directement productives mais qui servent simplement à faire en sorte que les actions productives puissent s'accomplir dans les meilleures conditions. Lorsqu'il s'agit de système monolithique, ces tâches supplémentaires font partie du système organisationnel, mais dans le cas d'un système multi-agent, c'est-à-dire d'un travail nécessairement distribué, on parle de *tâches de coordination*. Celles-ci sont indispensables dès lors que l'on se trouve en présence d'un ensemble d'agents autonomes qui poursuivent leurs propres buts, la réalisation de tâches productives entraînant avec elles tout un cortège de tâches de coordination sans lesquelles les premières ne peuvent être accomplies.

La phase de coordination d'actions est directement impliquée dans la définition de l'ordre des actions à effectuer. Le chapitre 8 est entièrement consacré à la problématique de la coordination d'action dans les systèmes multi-agents cognitifs ou réactifs.

2.5.6 La résolution de conflit par arbitrage et négociation

L'arbitrage et la négociation sont deux des moyens utilisés par les systèmes multi-agents pour résoudre les conflits et empêcher que des désaccords entre individus ne se transforment en luttes ouvertes et que le système dans son entier ne dégrade ses performances. L'arbitrage conduit à la définition de règles de comportement qui agissent comme des contraintes sur l'ensemble des agents, mais dont le résultat global a pour effet de limiter les conflits et de préserver à la fois les individus mais surtout les sociétés d'agents.

Dans les sociétés humaines, le comportement des individus est contraint par un ensemble de lois et réglementations qui régissent les activités sociales et limitent

leur activité pour (idéalement) le bien de tous. Il existe un organe d'arbitrage, la justice, dont la fonction est de faire respecter ces règles collectives et de veiller à leur application. Cet organe agit aussi pour déterminer les responsabilités dans les conflits et décider des punitions que les personnes jugées coupables doivent subir. Dans les systèmes multi-agents, l'ensemble des règles sociales agit comme un ensemble de contraintes comportementales plus ou moins internalisées par l'agent. Evidemment, ces "lois" ne peuvent s'appliquer qu'à des agents cognitifs de très haut niveau, qui disposent d'une sorte de "libre arbitre" et sont capables de peser le pour et le contre des conséquences de leurs actions.

Lorsque des agents cognitifs entrent en conflit d'objectif ou de ressource, on préfère souvent ne pas recourir à un arbitre, mais laisser plutôt les agents résoudre eux-mêmes le conflit par la recherche d'un accord bilatéral au cours d'un processus de négociation. Cette forme de résolution de conflit a été particulièrement étudiée par l'école américaine autour de V. Lesser (Conry et al. 1988), E. Durfee (Durfee et al. 1989) et K.Sycara (Sycara 1989). Malgré l'intérêt évident de ce problème, nous ne le développerons pas dans cet ouvrage.

Mais si dans les systèmes multi-agents cognitifs, les techniques de négociation sont à l'honneur, il n'en est pas de même des systèmes multi-agents réactifs qui ne peuvent entrer dans des phases de résolution de conflit de ce type. Le comportement de l'agent est "câblé", lui interdisant de sortir des limites qui lui ont été assignées par le concepteur et l'empêchant ainsi d'enfreindre ces lois. Tout se passe alors comme si ces lois sociales avaient été directement "compilées" dans le comportement des agents, sous la forme de mécanismes d'inhibition ou d'évitement (cf. chap. 3 et 8).

2.6 Organisations et coopération

2.6.1 Le système des activités de coopération

Les fonctions et les méthodes de coopération sont bien entendu liées entre elles. Certaines méthodes favorisent plus certaines fonctions, d'autres résolvent des problèmes intermédiaires tels que la répartition des tâches ou la coordination des actions. Il est possible de donner une vision intégrée de ces différents éléments sous la forme d'un diagramme systémique représenté figure 2.3. Ce diagramme montre d'une part leur interdépendance et, d'autre part, comment les objectifs de survie, de performances, de fiabilité et de résolution de conflit sont assurés par des méthodes élémentaires qui elles-mêmes entraînent des problèmes qui ne peuvent être assurés que par des corps de techniques plus élaborés. L'augmentation de la probabilité de survie des agents passe en particulier par l'augmentation des performances du groupe, ce qui requiert l'application de méthodes telles que le regroupement, l'augmentation du nombre des agents et la différenciation de leurs rôles. Mais en retour ces méthodes posent d'autres problèmes tels que la répartition des tâches et l'augmentation du nombre des conflits dus à des problèmes d'accès à des ressources et au manque de coordination. Les méthodes d'allocation de tâches, de communication, de coordination d'actions et de résolution de conflits (arbitrage, négociation, hiérarchisation)

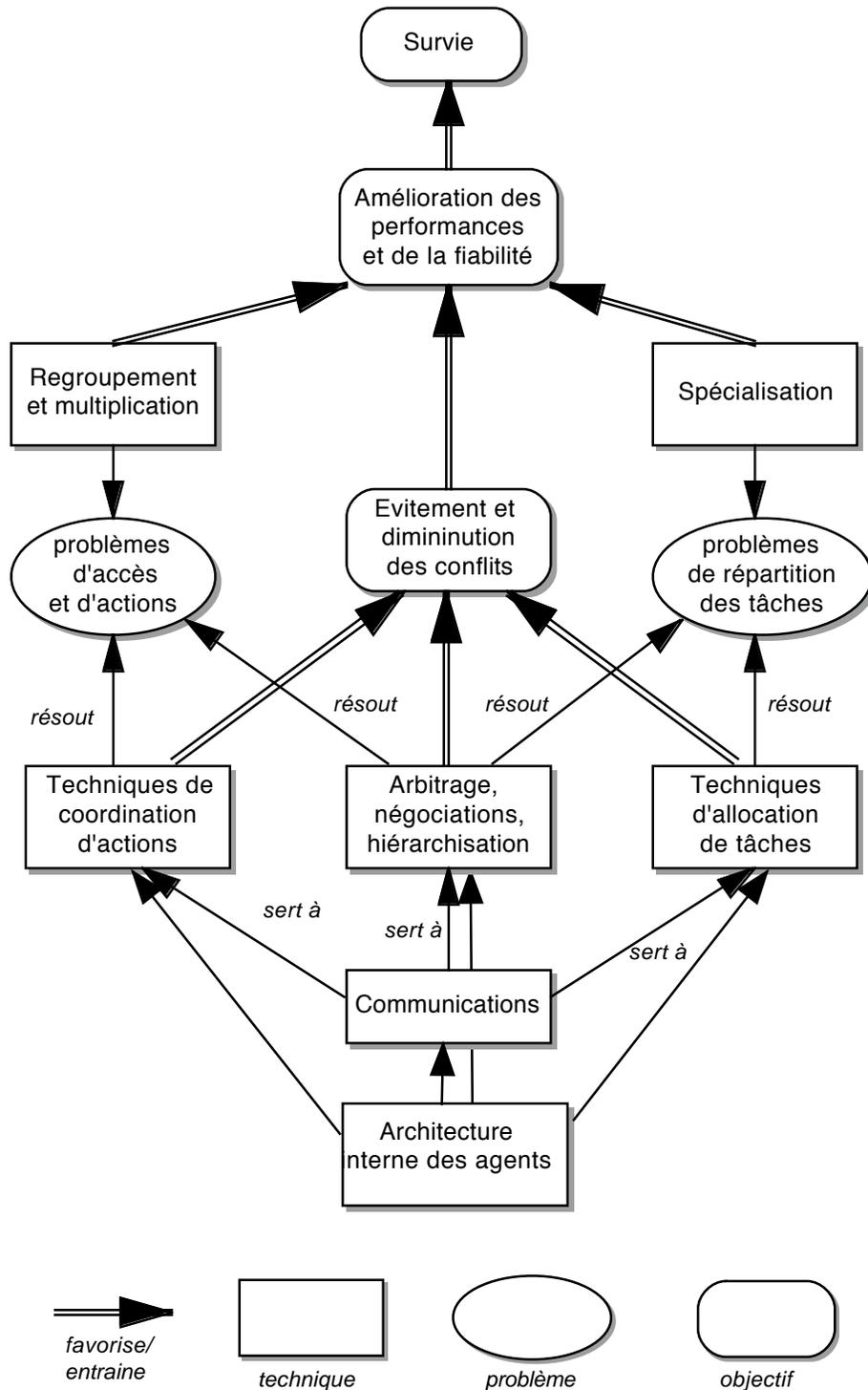


Figure 2.3: Le système des activités de coopération. Les organisations émergent des interactions provenant des techniques d'allocation de tâches (collaboration), d'arbitrage, de négociation et de coordination d'action.

servent à résoudre ces problèmes et participent de ce fait à l'augmentation des performances du groupe. Toutes ces techniques nécessitent d'être structurées au sein d'organisations qui décrivent la manière dont les agents sont positionnés les uns par rapport aux autres et comment ils peuvent effectivement travailler ensemble. Ces organisations peuvent être prédéfinies par le concepteur ou bien émerger des interactions entre agents, comme nous le verrons au chapitre 3. La coopération nécessite donc la mise en œuvre de mécanismes pour améliorer les performances du groupe afin d'accroître la survie de ses membres, mais elle nécessite en contre partie des structures sociales qui présentent des conséquences à la fois positives et négatives pour les agents. Enfin, la réalisation effective de ces méthodes de coopération supposent l'existence de mécanismes de communication ainsi que la définition des structures internes nécessaires aux agents pour qu'ils interagissent.

2.6.2 Avantages

La coopération, en favorisant les performances collectives des agents, offre ainsi un certain nombre d'avantages qui s'expriment soit par des améliorations individuelles, telles que l'accroissement de productivité ou l'augmentation de la probabilité de survie des individus, soit par des améliorations collectives telles que l'accomplissement de tâches impossibles autrement à réaliser. Voilà une liste non exhaustive des avantages de la coopération⁴:

1. *Accomplir des tâches impossibles à réaliser seul.* Il s'agit d'un avantage qualitatif essentiel comme nous l'avons vu. Dans ce type de situation seule la collaboration permet d'accomplir l'action désirée.
2. *Améliorer la productivité de chacun des agents.* Cet avantage quantitatif permet d'obtenir ce que l'on appelle *économie d'échelle* en gestion commerciale: la production augmente alors de manière plus que linéaire en fonction du nombre d'agents ce qui entraîne un accroissement de la productivité. Nous en avons vu un exemple avec l'étude des robots récupérateurs de minerai. Il s'agit d'une caractéristique essentielle de la collaboration en favorisant l'émergence de nouvelles structures grâce aux économies réalisées par les différents agents.
3. *Augmenter le nombre de tâches réalisées dans un délai imparti ou diminuer le temps de réalisation d'une tâche.* La simple multiplication du nombre des agents permet d'augmenter quantitativement le nombre de tâches réalisables et de diminuer ainsi le temps d'exécution nécessaire à l'accomplissement d'un ensemble de tâches données. Mais la collaboration peut s'avérer indispensable s'il existe un temps limité pour effectuer un certain quota de tâches. Par exemple, pour exécuter un contrat portant sur la fourniture d'un produit au bout d'un délai précis, il est parfois nécessaire que plusieurs personnes collaborent à la réalisation du produit simplement pour faire face à l'exécution du contrat.

⁴Durfee, Corkill et Lesser (Durfee et al. 1989) proposent une classification semblable dans le cadre de la résolution distribuée de problèmes, mais ils ne différencient pas les avantages qualitatifs des bénéfices quantitatifs.

Cette amélioration quantitative se transforme du fait du délai associé au contrat en un bénéfice qualitatif, l'inobservation du contrat entraînant toute une série d'inconvénients.

4. *Améliorer l'utilisation des ressources.* En gérant mieux l'utilisation de certaines ressources par collaboration, il est possible de diminuer les coûts de fonctionnement en équilibrant les besoins, ce qui revient alors soit à augmenter la productivité du système (avantage 2) soit à améliorer les performances et rendre possible des actions (avantage 3). Par exemple, si un programme fonctionne sur un multiprocesseur, il peut être intéressant pour les processeurs de collaborer pour que les agents puissent équilibrer leur charge de travail sur les différentes machines par un mécanisme dynamique de régulation des charges (load balancing).

Il existe bien évidemment un lien très étroit entre les performances collectives et les capacités de survie individuelles. L'action des agents contribue à la performance collective qui en retour assure une meilleure survie des individus. Cependant, le gain de cet avantage impose des contraintes aux individus en limitant leur marge de manœuvre.

2.6.3 Contraintes sociales et émergences de structures

De nombreuses structures émergent des interactions. Nous avons vu que ces structures sont nécessaires à la survie du groupe car elles amplifient sa capacité. Par exemple, la file des robots récupérateurs de minerai émerge simplement des interactions tendant à améliorer les performances du groupe. Mais ce sont ces structures qui, par l'effet d'une boucle de rétroaction sociale, définissent des contraintes individuelles. En effet, les mécanismes de coopération, en améliorant les performances du groupe et en augmentant les capacités de survie des individus se traduisent par l'établissement d'un certain nombre de normes et contraintes sociales tendant à limiter l'éventail des comportements possibles d'un individu et à l'intégrer dans une organisation sociale de plus en plus rigide. Par exemple, l'amélioration des performances par spécialisation des agents, comme dans le cas du système MANTA (cf. chap. 7), entraîne un appauvrissement de leurs capacités de totipotence et diminue ainsi leur marge de manœuvre. Ils deviennent alors plus dépendants des autres et donc de moins en moins capables de survivre seuls. La spécialisation les cantonne dans un rôle dont ils peuvent de moins en moins se libérer et dont ils deviennent de plus en plus dépendants. Les mécanismes sociaux mis en place pour assurer la viabilité du groupe restreignent encore les possibilités individuelles des agents en contraignant leurs comportements et en les poussant à se socialiser encore plus, c'est-à-dire à devenir de plus en plus spécialisés et dépendants des autres. Il y a donc un effet de rétroaction positive entre l'accroissement de performance, l'augmentation de la survie individuelle et l'établissement de contraintes sociales tendant à limiter les possibilités individuelles d'action, comme nous l'avons vu au chapitre précédent.

Chapter 3

Organisations multi-agents

Les systèmes multi-agents peuvent procurer à celui qui commence à les analyser une sensation de vertige. Les paramètres qu'il est possible de faire varier sont si nombreux que l'on a l'impression de se trouver devant une infinité de cas particuliers qu'il sera donc impossible d'analyser exhaustivement. En effet, de simples variations dans les capacités de représentation des agents, dans la manière de coordonner leurs actions ou sur le fait, pour eux, de pouvoir s'engager ou non à accomplir des tâches vont produire des systèmes multi-agents très divers et dont les propriétés générales sont bien difficiles à cerner.

Cependant, malgré cette diversité, il est possible de dégager des méthodes d'analyse visant à caractériser l'organisation d'un SMA et d'être ainsi à même d'ordonner ce qui a priori semble relever du chaos.

3.1 Qu'est-ce qu'une organisation?

L'organisation est, avec l'interaction, l'un des concepts de base des systèmes multi-agents. Lorsqu'on parle d'organisation, on suppose qu'il existe un ensemble d'entités formant une certaine unité et dont les différents éléments sont subordonnés entre eux dans un ensemble solidaire et dans une activité convergente. L'organisation nécessite donc un certain ordre entre les entités éventuellement hétérogènes, lequel concourt à la cohérence du tout.

Définition: *Une organisation peut être définie comme un agencement de relations entre composants ou individus qui produit une unité, ou système, dotée de qualités inconnues au niveau des composants ou individus. L'organisation lie de façon interrelationnelle des éléments ou événements ou individus divers qui dès lors deviennent les composants d'un tout. Elle assure solidarité et solidité relative, donc assure au système une certaine possibilité de durée en dépit de perturbations aléatoires (Morin, 1977).*

Dans un système multi-agent, il existe de nombreuses interrelations entre les agents, par le biais de délégation de tâches, de transfert d'informations, d'engagements, de synchronisations d'actions, etc. Ces interrelations ne sont possibles qu'à l'intérieur

d'une organisation, mais réciproquement, les organisations exigent l'existence de ces interrelations. Les organisations constituent donc à la fois le support et la manière dont se passent ces interrelations, c'est-à-dire la façon dont sont réparties les tâches, les informations, les ressources, et la coordination d'actions.

De plus, et c'est ce qui rend ce terme si difficile à cerner, l'*organisation* désigne à la fois le processus d'élaboration d'une structure et le résultat même de ce processus. Cette dualité du terme montre qu'il existe toujours un aspect dynamique dans l'organisation: il n'existe, à proprement parler, pas d'organisations statiques. L'organisation est nécessairement dynamique et toujours en réorganisation de l'ensemble des entités et des liens qui unissent ces entités.

3.1.1 Structures organisationnelles et organisations concrètes

Mais au travers de cette dynamique, il persiste quelque chose d'invariant, que l'on appellera sa *structure organisationnelle* (ou simplement *structure* s'il n'y a pas d'ambiguïté). La structure organisationnelle est ce qui caractérise, sur un plan abstrait, une classe d'organisation. A l'inverse, l'*organisation concrète* (ou simplement organisation) est une instanciation (au sens des langages objets) possible d'une structure organisationnelle, une réalisation comprenant l'ensemble des entités qui participent effectivement à l'organisation ainsi que l'ensemble des liens qui associent ces agents à un moment donné¹. A toute organisation concrète, il est possible d'associer une structure organisationnelle, mais une même structure organisationnelle peut servir de base à la définition d'une multitude d'organisations concrètes. En appliquant aux organisations la définition de P. Delattre (Delattre 1968) pour les structures, on peut dire qu'une structure organisationnelle est définie par la double donnée:

1. d'un ensemble de classes d'agents caractérisées par les rôles affectés aux agents,
2. de l'ensemble de relations abstraites existant entre ces rôles.

Enfin, on classera les structures organisationnelles en fonction de leur capacité à s'adapter et à évoluer en fonction des besoins. Pour obtenir une organisation concrète, il est nécessaire d'y ajouter l'ensemble des agents appartenant aux différentes classes et de décrire les relations concrètes qui existent entre les agents, c'est-à-dire "d'instancier" l'organisation abstraite. La figure 3.1 illustre cette différence. La structure organisationnelle O est décrite par l'ensemble des rôles qui peuvent y être tenus (A, B, C) alors que les structures concrètes comprennent un grand nombre d'entités concrètes (A_1, A_2, B_1, \dots), les A_i étant associées au rôle A . Les liens entre les entités concrètes ($g_1, g_2, h_1, h_2, \dots$) correspondent aux liens abstraits qui existent entre les entités abstraites (g, h, \dots). Par exemple, le lien abstrait g qui lie l'entité abstraite A à B s'exprime concrètement dans l'organisation concrète O_1 sous la forme de deux

¹Ces termes ne sont pas standardisés. Signalons en particulier que Maturana et Varela (Maturana et Varela 1987) appellent organisation ce qui est appelé ici structure organisationnelle et structure ce qui est intitulé organisation. P. Delattre propose une autre définition en appelant structure relationnelle les structures organisationnelles et structure totale les organisations concrètes.

liens g_1 et g_2 qui relient respectivement A_2 à B_1 et A_1 à B_1 , alors que dans O_2 des liens g_1 et g_2 semblables relient respectivement A_1 à B_1 et A_1 à B_2 .

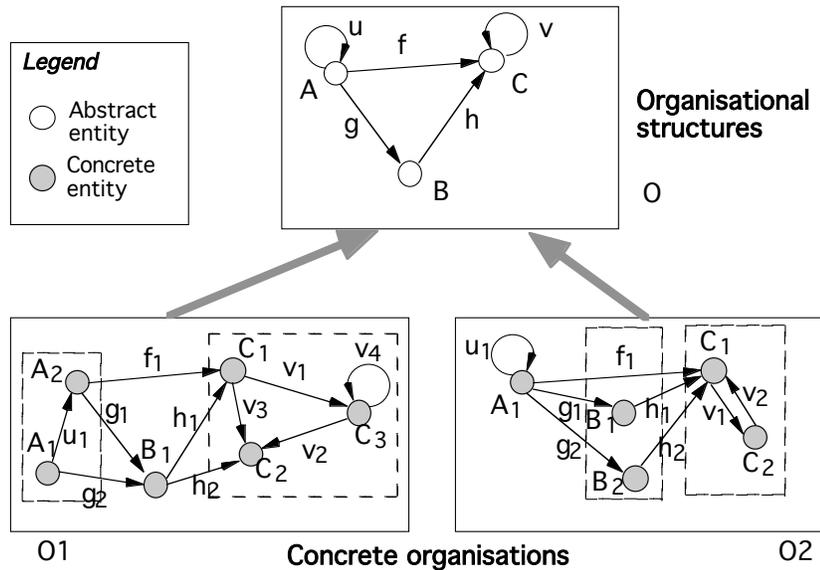


Figure 3.1: Organisations et structures organisationnelles

3.1.2 Les niveaux d'organisation

L'intérêt de la notion d'organisation est de pouvoir intégrer à la fois la notion d'agent et celle de système multi-agent. En effet, du point de vue du concepteur, un agent est non seulement un individu, mais aussi un assemblage de composants. De même, un système multi-agent peut être considéré à la fois comme une composition d'agents et comme une unité, lorsqu'on étudie les interactions de ce système avec un autre dispositif technique.

La notion de niveau d'organisation permet de comprendre l'emboîtement d'un niveau dans un autre. De même qu'en biologie une cellule est considérée comme une organisation de macromolécules et comme un être individuel pour l'organisme multicellulaire dont elle fait partie, de même, on pourra considérer qu'une organisation est une agrégation d'éléments de niveau inférieur, et un composant d'organisations de niveau supérieur (fig. 3.2).

On appellera *composant élémentaire* (ou *module*) les unités de niveau inférieur considérées comme indécomposables (tout du moins par le biais de cette analyse) et *système multi-agent général* le niveau supérieur, c'est-à-dire une organisation ne faisant pas partie (d'un certain point de vue) d'une organisation de plus haut niveau. On distinguera enfin les *organisations (ou entités) individuelles*, qui sont considérées de l'extérieur comme des êtres individuels (c'est-à-dire comme des agents), et les *organisations (ou entités) collectives*, qui sont composées d'un ensemble d'êtres individuels, sans qu'elles soient nécessairement individuéés elles aussi.

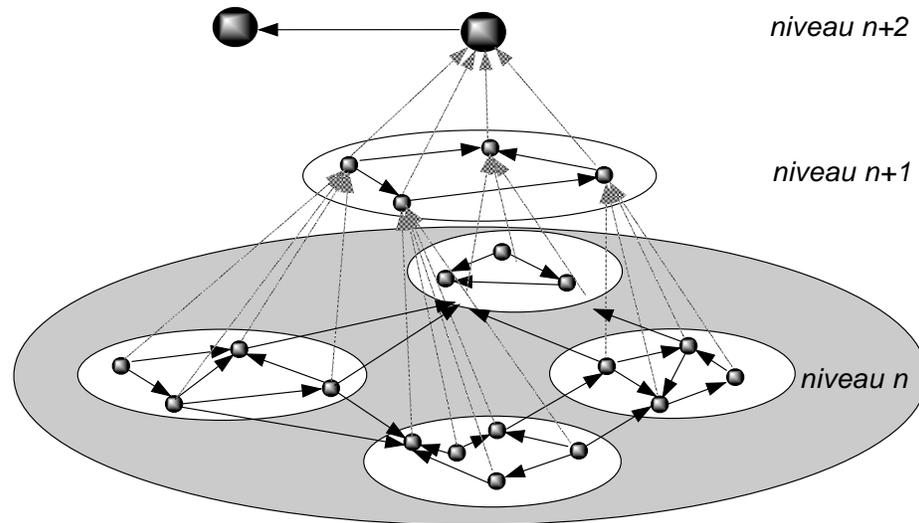


Figure 3.2: Les agents de niveau n sont regroupés en organisations qui au niveau $n + 1$ peuvent être considérées comme des entités individuelles. Inversement, des entités individuelles de niveau $n + 1$ peuvent être vues comme des organisations de niveau n . Le processus peut se répéter sur un nombre quelconque de niveaux.

La différence entre entités individuelles et collectives réside avant tout dans l'existence de ce que les biologistes appellent une *membrane*: le fait qu'une entité individuelle sépare son espace intérieur de l'espace extérieur de telle manière que les interactions entre les composants internes et les entités externes ne puissent s'effectuer que dans des limites très précises dictées par des composants spécifiques chargés des entrées-sorties. Au contraire, les entités collectives ne font pas (ou peu) ce type de différence et elle ne peuvent pas apparaître naturellement comme constituant un individu.

Les agents sont donc avant tout des entités individuelles alors que les systèmes multi-agents sont essentiellement des entités collectives. Néanmoins, un agent composé lui-même d'un ensemble d'agents sera considéré à un certain niveau comme une organisation individuelle et, au niveau inférieur, comme une organisation collective.

3.1.3 Comment étudier une organisation?

Devant l'ensemble des organisations possibles, la démarche d'analyse doit être structurée. C'est pourquoi nous distinguerons trois volets:

1. L'*analyse fonctionnelle*, qui décrit les fonctions d'une organisation multi-agent dans ses différentes dimensions.
2. L'*analyse structurale*, qui distingue les différentes formes d'organisation possibles et identifie quelques paramètres structuraux essentiels.
3. Les *paramètres de concrétisation*, qui traitent du passage d'une structure organisationnelle à une organisation concrète et posent la difficile question de la réalisation effective d'une organisation multi-agent.

3.2 L'analyse fonctionnelle

L'analyse fonctionnelle d'une organisation conduit à l'identification des principales fonctions que les composants de cette organisation doivent remplir. Dans le cadre d'une telle analyse, une organisation peut être vue comme un système de rôles, chaque rôle étant défini par l'ensemble des propriétés que les agents ayant ce rôle manifestent dans cette organisation. Afin d'affiner cette analyse, nous verrons qu'il est possible de développer ces fonctions dans plusieurs dimensions d'analyse qui correspondent aux différents points de vues par lesquels une organisation peut être étudiée.

3.2.1 Les fonctions d'une organisation

Les rôles décrivent la position des agents au sein d'une organisation ainsi que l'ensemble des activités qu'il sont censés exercer afin que l'organisation puisse accomplir ses objectifs. Ils caractérisent ainsi les fonctions que les agents remplissent, indépendamment de leur structure interne et des mécanismes mis en œuvre. Par exemple, un agent qui gère les demandes d'exécution et les répartit aux agents compétents aura un rôle de *médiateur*, un agent qui détermine les actions à entreprendre aura un rôle de *planificateur*, un agent qui choisit parmi un ensemble d'actions possibles sera appelé *décideur*, un agent qui rend un service à un autre, considéré comme un *client*, aura un rôle de *fournisseur* et un agent dont la fonction principale est exécutive sera un *exécutant*. Ces rôles sont définis par l'ensemble des fonctions nécessaires à une organisation pour qu'elle s'adapte à ses finalités (et en particulier aux objectifs et aux besoins de son concepteur) et à son environnement.

Ces rôles ne sont pas nécessairement fixes, mais peuvent évoluer avec la dynamique de l'organisation. Par exemple, un exécutant, par suite d'une évolution de l'organisation, peut se retrouver médiateur, décideur ou client. De même, un fournisseur, pour un ensemble donné d'activités, peut être considéré comme un client pour d'autres tâches. Chaque agent dispose donc d'un ensemble de rôles qui caractérisent les différents services abstraits qu'il peut remplir au sein d'une organisation, ces rôles déterminant des positions dans l'ensemble des relations abstraites qui peuvent s'exprimer entre les agents. Les rôles sont regroupés en sous-systèmes², c'est-à-dire en sous-organisations de l'organisation principale, caractérisés par leurs fonctions qui peuvent être centralisées ou au contraire distribuées parmi les composants d'une organisation. Voici les six fonctions principales d'une organisation et quelques-uns des rôles associés:

1. La *fonction représentationnelle* comprend l'ensemble des fonctions de modélisation de l'environnement et des autres organisations ainsi que la mémorisation des événements qui ont pu affecter l'organisation. Cette fonction est particulièrement développée dans les organisations individuelles cognitives (c'est-à-dire les agents cognitifs) qui disposent d'une représentation interne de leur

²Il faut prendre ici les termes "système" et "sous-système" dans l'acception qu'ils ont lorsqu'on parle de système circulatoire, respiratoire ou végétatif du corps humain.

environnement qu'il leur est possible de manipuler afin de planifier les actions qu'elles devront entreprendre. Dans une organisation collective, la fonction représentationnelle peut être, comme pour les autres fonctions, distribuée parmi tout un ensemble d'agents. Dans les systèmes d'information des entreprises, par exemple, cette fonction est particulièrement développée, puisqu'elle correspond à la conservation et à l'archivage de toutes les informations véhiculées dans l'entreprise. Les systèmes de gestion électronique de documents ne font qu'améliorer cette fonction. Le rôle associé à la fonction représentationnelle est celui d'*archiviste*.

2. La *fonction organisationnelle* se rapporte à tout ce qui a trait à la gestion de l'activité d'une organisation et, en particulier, à la planification, l'allocation et le suivi de tâches, la coordination des actions ainsi que la gestion des engagements. Cette fonction est évidemment essentielle et c'est celle qui dans le domaine des systèmes multi-agents a été plus particulièrement étudiée. De nombreux rôles sont associés à cette fonction, en particulier ceux de *client*, de *fournisseur*, de *médiateur*, de *planificateur* et de *coordinateur*. Cette fonction, ainsi que les rôles associés, sera plus particulièrement traitée dans le chapitre 7.
3. La *fonction conative* (du latin *conari*, se préparer à, entreprendre) se rapporte aux sources, aux limites et à la sélection des activités des organisations. Cette fonction se divise elle-même en trois sous-fonctions: celle des motivations ou *fonction motivationnelle* qui décrit comment les buts ou tendances sont formés, qu'il s'agisse de demandes internes (besoins, désirs, pulsions) ou externes (demandes des autres agents, pressions sociales); celle de l'élaboration des *contraintes* qui décrit les normes et inhibitions qui viennent limiter l'éventail des actions possibles de l'organisation et enfin celle de la prise de décision grâce à la *fonction décisionnelle* qui s'occupe, en liaison avec le système organisationnel, de l'évaluation et du choix des buts à satisfaire et des tâches à entreprendre. Chaque sous-fonction définit un ensemble de rôles associés, dont le plus connu est celui de *décideur* associé à la fonction décisionnelle.
4. La *fonction interactionnelle* sert à faire le lien entre une organisation et ce qui l'entoure. C'est elle qui gère les activités d'interactions avec son environnement ainsi que l'ensemble des communications avec les autres organisations. Elle assure ainsi toutes les fonctions d'entrée-sorties et d'interface. Elle se compose de deux sous-fonctions, la fonction perceptive, qui se charge de l'acquisition d'informations provenant de l'extérieur, et la fonction exécutive qui s'occupe de l'accomplissement des actions choisies par le système conatif et mises en œuvre par le système organisationnel. Les rôles associés à cette fonction sont ceux d'agents d'*interface* (*observateur*, *exécutant* et *communicateur*).
5. La *fonction productive* ou *opératoire* concerne l'ensemble des activités primitives qui doivent être mises en œuvre pour résoudre un problème ou effectuer un travail. Cette fonction dépend de l'application: un SMA dédié à

l'analyse linguistique, qui comprend des agents capables de faire de l'analyse lexicale, syntaxique, sémantique ou pragmatique, n'aura évidemment pas la même fonction productive qu'un ensemble de robots explorateurs, qui doivent explorer, forer et transporter des échantillons de minerai. Il n'y a pas de rôle caractéristique associé à cette fonction, du fait qu'elle dépend du domaine d'application.

6. La *fonction végétative* ou (*conservative*) traite de la conservation de la structure et des agents, de l'acquisition et de l'entretien des ressources et des tâches employées pour maintenir et reproduire l'organisation. Par exemple, pour un animat (cf. chap. 5), manger, se soigner, se reproduire constituent les comportements de base gérés par le système végétatif. Pour une organisation collective, la fonction végétative traite à la fois des apports énergétiques et de l'élimination des déchets, mais aussi de l'intégration de nouveaux agents, par recrutement ou reproduction. Cette fonction, que les militaires appellent l'intendance, s'occupe aussi des réparations qui doivent être entreprises pour que la structure organisationnelle perdure.

Evidemment, toutes ces fonctions sont liées, comme le sont les systèmes nerveux, circulatoire, reproducteur et digestif des êtres vivants, et sont nécessaires au bon fonctionnement d'une organisation, même si certaines sont réduites à leur plus simple expression, comme la fonction représentationnelle, qui s'avère quasi inexistante pour les agents réactifs, ou distribuée dans un ensemble d'agents, comme la fonction décisionnelle dans un système multi-agent réactif. La figure 3.3 illustre les relations principales qui associent ces systèmes entre eux. Prenons un exemple simple. Sup-

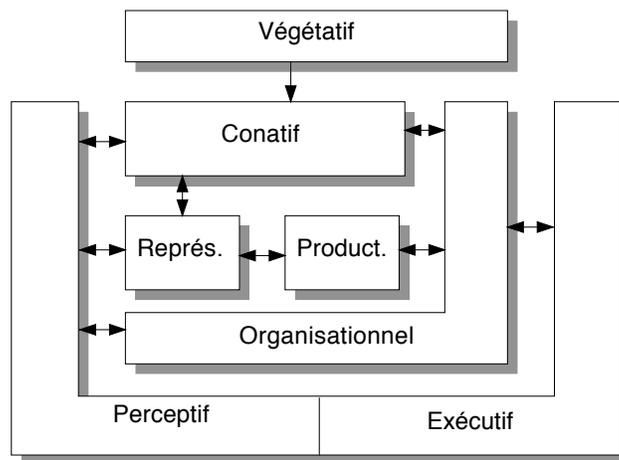


Figure 3.3: Représentation schématique des différentes fonctions d'une organisation. Ce schéma s'applique aussi bien à des agents qu'à des systèmes multi-agents.

posons que l'on réalise un robot, Clotaire, capable d'aller chercher des objets et de les ramener. Peut-être est-il capable de détecter certains stimuli, de reconnaître des outils tels qu'une pince ou un marteau s'ils se trouvent devant lui, ou d'identifier la pièce dans laquelle il se trouve (à partir, par exemple, d'un codage infrarouge ou

d'une caméra et d'un système de reconnaissance de formes). Pour ce faire, il doit disposer d'organes de perception et de modules de reconnaissances des formes. A l'autre bout de la chaîne, on trouve les primitives d'action (aller tout droit, tourner à gauche, tourner à droite, prendre un objet, le déposer, etc.) et les mécanismes de réception et d'envois de messages. Tout ceci constitue sa fonction interactionnelle.

Si son rôle, à l'intérieur d'une organisation de plus haut niveau, est de transporter des objets, sa fonction productive sera composée d'un ensemble d'opérations de transport telles que "aller dans la pièce X ", ou "rapporter l'objet X qui se trouve dans la pièce P ".

Sa fonction organisationnelle sera chargée de la planification des tâches. C'est elle qui assure le suivi des opérations et en réfère au système conatif s'il y a un problème. Par exemple, si l'on demande à Clotaire de ranger tous les outils, les composant chargés de la fonction organisationnelle planifieront les actions à accomplir, en considérant les opérations de transport de sa fonction productive comme des primitives. Il en viendra peut être ainsi à décomposer cette tâche en un ensemble d'actions plus élémentaires telles que: aller chercher la pince qui se trouve dans la remise et l'apporter dans le hangar, puis aller chercher le marteau qui se trouve dans le hangar et l'apporter dans l'atelier. Il faudra ensuite, dans le cadre de la fonction organisationnelle, qu'il assure le suivi des tâches, en vérifiant qu'elles sont bien réalisées.

La fonction représentationnelle gère ses représentations du monde, lesquelles sont directement utilisées par les mécanismes de planification. Par exemple, si Clotaire est effectivement un agent cognitif, sa fonction représentationnelle sera caractérisée par l'ensemble des mécanismes qui tentent de mettre en adéquation ses représentations et la structure de l'environnement, en construisant une carte du lieu où il doit se déplacer, en lui indiquant les passages ouverts ou fermés, en mémorisant l'emplacement des outils, etc. Cette fonction mémorise aussi les différents savoir-faire de l'agent, cette connaissance pouvant être mise à profit lors de l'élaboration des plans. En effet, alors que la fonction productive est concernée par l'exécution des plans "compilés", la fonction représentationnelle possède une description symbolique de ces opérations afin que les composants liés à la fonction organisationnelle puissent planifier les nouvelles opérations.

Enfin, la fonction conative est responsable de l'évaluation des situations et des prises de décisions. Par exemple, il n'est pas sûr que Clotaire accepte de ranger tous les outils s'il se trouve déjà en train d'accomplir d'autres tâches qu'il juge plus importantes. En particulier, si son niveau d'énergie tombe à un trop bas niveau, comme le lui indique les organes de son système végétatif, il laissera tomber les tâches en cours pour tenter de récupérer un niveau d'énergie correct avant de poursuivre son activité productrice.

Cet exemple n'est pas propre à un seul agent. Supposons qu'une équipe d'agents soit chargée de transporter des objets, les différentes fonctions seront réparties parmi les agents. Certains seront plus chargés du transport effectif des objets et constitueront ainsi la fonction productive de cette équipe. La fonction organisationnelle sera réalisée par l'ensemble des agents chargés de la planification des actions, du suivi de leur accomplissement, de la coordination des actions de transports et, d'une

manière générale, de l'administration de l'équipe. La fonction représentationnelle pourra être totalement distribuée parmi les agents de transport ou bien, au contraire, centralisée dans quelques agents qui tiendront à jour une carte des lieux avec la position des objets et des agents. Les agents chargés de la fonction interactionnelle se consacreront aux communications (liaison radio, gestion des télécommunications, etc.) avec les autres équipes et avec la base. Enfin, les agents spécialisés dans la fonction conative détermineront les priorités dans les objectifs et décideront des tâches générales à accomplir.

3.2.2 Les dimensions d'analyse

L'étude de fonction précédentes, et donc des rôles des agents, peut se décliner selon plusieurs dimensions:

1. La *dimension physique* (ϕ) traite de l'implémentation, de l'architecture de l'organisation et de ses ressources personnelles. Cette dimension répond à la question: "Comment ceci est-il réalisé explicitement à l'intérieur de l'organisation?" S'il n'est pas indispensable de répondre à cette question lorsqu'on ne s'intéresse qu'à l'analyse comportementale d'une organisation, un concepteur de SMA ne peut faire l'impasse sur cette question, car l'existence d'une représentation opératoire directement exécutable est à la base de l'informatique et d'une manière générale des sciences synthétiques visant à la constructions d'artefacts.
2. La *dimension sociale* (σ) concerne la place, et donc le rôle, d'une organisation dans une organisation de plus haut niveau. Cette dimension s'occupe aussi de l'aspect social des activités d'une organisation, c'est-à-dire de ses capacités à répondre à des besoins ou fonctions générales nécessaires au bon fonctionnement de l'organisation de plus haut niveau dans laquelle elle se trouve.
3. La *dimension relationnelle* (α) s'occupe des interactions, des échanges qu'une organisation peut avoir avec d'autres organisations de même niveau, de ses communications et de leur coordination. Cette dimension répond à la question: "Avec qui, pourquoi et de quelle manière une organisation interagit-elle avec d'autres organisations?". C'est à ce niveau que sont définis les protocoles d'interaction, c'est-à-dire les enchaînements de comportements mutuels que des organisations produisent lors de leur échanges.
4. La *dimension environnementale* (χ) est liée aux capacités d'une organisation à agir, percevoir et raisonner sur le monde environnant. C'est à ce niveau que sont définies ses capacités d'interactions avec le monde, ses compétences proprement instrumentales. Cette dimension répond donc à la question: "Quelles sont les possibilités d'action d'une organisation, quelles sont ses expertises techniques et de quelle manière est-elle capable d'agir sur le monde?"
5. La *dimension personnelle* ou *dimension du self* (ω) représente tout ce qui a trait à l'organisation en tant que telle et en particulier à ses aspects "méta":

la représentation de soi, l'acquisition des connaissances, la préservation de sa structure et l'activité proprement autopoïétique d'une organisation.

La figure 3.4 illustre ces dimensions selon deux axes. Les dimensions relationnelles et environnementales se situent sur un même plan horizontal car les relations qu'un agent entretient avec les autres agents ou avec l'environnement sont effectuées au même niveau que l'agent lui-même. L'axe vertical correspond aux dimensions sociales et physiques. Ces deux dimensions sont à la fois opposées et complémentaires si l'on considère un agent comme une organisation de niveau supérieur par rapport à ses composants, considérés comme des agents de niveau inférieur.

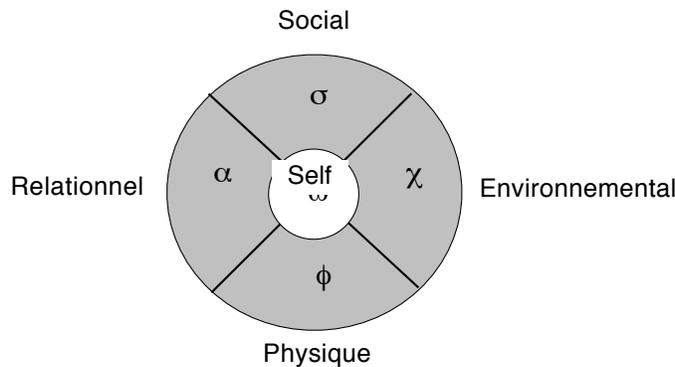


Figure 3.4: Une organisation peut être analysée selon cinq dimensions: sociale, relationnelle, physique, environnementale et personnelle (self).

Dans ce cas, la dimension sociale des agents "composant" recouvre la dimension physique de l'agent "société", comme le montre la figure 3.5. En effet, la fonction sociale qu'un agent "composant" assure au sein d'un agent "société" est un point de vue complémentaire à celui de la dimension physique de l'agent "société", qui voit ses composants comme des organes lui permettant d'assurer son comportement et d'assurer sa viabilité. Enfin, la dimension personnelle est au centre de ces oppositions, le lieu réflexif à partir duquel l'agent peut se construire en tant que sujet agissant au milieu d'autres agents.

3.2.3 Analyse dimensionnelle d'une organisation

L'analyse fonctionnelle d'une organisation le long des dimensions que nous venons de mentionner permet de constituer des grilles d'analyses pour donner un aperçu assez précis de la nature d'une organisation. Il est en effet possible de projeter les six fonctions d'une organisation sur ces cinq dimensions.

Fonction représentationnelle

- ϕ La *dimension représentationnelle/physique* représente tout ce qui se rapporte à la manière dont les connaissances sont représentées ainsi qu'à l'architecture inférentielle utilisée. Il représente la manière dont sont implémentés les mécanismes de raisonnement (moteurs d'inférences, tableaux noirs avec déclenchement

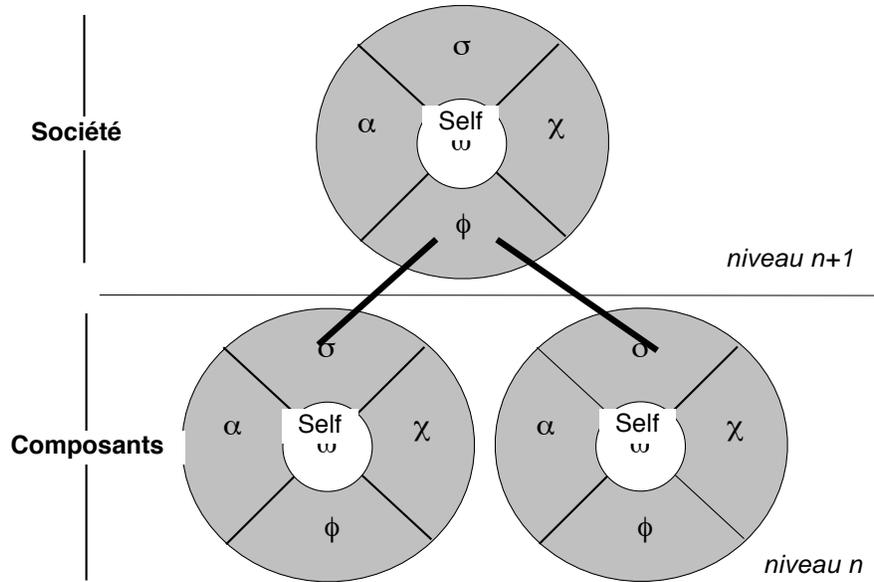


Figure 3.5: La dimension sociale du niveau n est directement liée à la dimension physique du niveau $n + 1$. Les agents de niveau n sont les composants des agents du niveau $n + 1$ et, inversement, les agents de niveau $n + 1$ constituent la société dans laquelle les agents de niveau n évoluent.

sur événements, programmes en LISP ou en C++, etc.). La dimension physique correspond à ce que Newell appelle le *niveau du symbole* (symbol level) par opposition au *niveau conceptuel* (knowledge level) (Newell 1982) et agit donc comme une dimension d'implémentation sur laquelle se projette les autres dimensions.

σ La *dimension représentationnelle/sociale* concerne les représentations qu'une organisation, et notamment un agent cognitif, se fait de la société dans laquelle elle se trouve, c'est-à-dire les modèles lui permettant de comprendre les buts collectifs, les aspirations et les normes sociales qui la contraignent. Peu de systèmes multi-agents implémentent cette dimension.

α La *dimension représentationnelle/relationnelle* a trait aux représentations des autres organisations. Lorsque l'organisation est un agent, on parle alors de représentations des accointances, c'est-à-dire des représentations des agents qu'un agent connaît et avec lesquels il peut communiquer directement. Cette dimension a particulièrement été traitée en IAD depuis les travaux de L. Gasser sur MACE (Gasser et al. 1987).

χ La représentation des objets du monde environnant et des lois de l'univers ambiant s'effectue dans la *dimension représentationnelle/environnementale*. Lorsque l'organisation est un agent, cette dimension peut faire défaut si l'agent est un agent logiciel ou au contraire un agent purement réactif. Si l'organisation est un système multi-agent, la dimension environnementale correspond

à l'ensemble des interactions que le système entretient avec ce qui n'est pas le système, par le biais de capteurs ou d'agents d'interfaces.

- ω La *dimension représentationnelle/personnelle* porte sur les représentations de soi, c'est-à-dire sur les croyances qu'un agent détient sur ses propres capacités, sur son savoir-faire, sur ses plans et ses buts, etc. A ce niveau tout ce qui est normalement inconscient pour une organisation devient explicite, ce qui signifie qu'il lui est ensuite possible de raisonner dessus.

Fonction organisationnelle

- ϕ La *dimension organisationnelle/physique* décrit l'architecture et l'implémentation du mécanisme de contrôle, de planification et de gestion des activités de l'organisation.
- σ La *dimension organisationnelle/sociale* porte sur la planification des tâches sociales, c'est-à-dire sur la définition des rôles permettant à une organisation de niveau inférieur d'accomplir des tâches utiles pour l'organisation de niveau supérieur.
- α La *dimension organisationnelle/relationnelle* est très étudiée en IAD car il s'agit de la planification des interactions et de la coordination des actions et des communications. Une grande partie des travaux portant sur la coopération et la coordination d'action sont situés sur cette dimension. En particulier, les protocoles d'appel d'offre et de coordination d'actions font partie de cette dimension.
- χ La *dimension organisationnelle/environnementale* traite de la planification et du contrôle de l'exécution des actions dans l'environnement, indépendamment des problèmes d'interaction avec les autres agents. Comment disposer des caisses en piles? Comment aller d'un lieu à un autre? sont des questions qui demandent des réponses qui doivent être élaborées le long de cette dimension.
- ω La *dimension organisationnelle/personnelle* correspond aux problèmes de méta-organisation, c'est-à-dire au contrôle et à la gestion des tâches internes. Pour un agent, cette dimension recouvre ce que l'on traite sous le terme de contrôle de niveau méta (Pitrat 1990; Maes et Nardi 1988; Ferber 1989).

Fonction conative

- ϕ La *dimension conative/physique* de la fonction conative correspond à l'implémentation de la production et de la gestion des pulsions, des contraintes et des tendances ou buts.
- σ La *dimension conative/sociale* correspond aux buts collectifs et aux contraintes sociales qu'une organisation de niveau supérieur se donne et qu'elle retransmet ensuite à ses membres. Par exemple, marquer des buts pour une équipe de football est un objectif collectif: chaque joueur va tenter de faire en sorte que

le groupe marque des points même s'il ne cherche pas à mettre lui-même le ballon dans les buts. Mais par ses actions, qui s'inscrivent dans une démarche collective, il va permettre à d'autres de marquer et ainsi d'accomplir les buts collectifs du groupe. Il en va de même des normes qui sont généralement établies sur la base d'une décision collective. Les lois, par exemple, s'appliquent à tous et contraignent ainsi les actions possibles de chacun. Ces contraintes, qui sont un effet secondaire des interactions entre agents, agissent comme un amplificateur social tendant à spécialiser le rôle que les agents détiennent dans une organisation sociale (cf. chap. 7).

- α La *dimension conative/relationnelle* porte sur les demandes et contraintes des autres organisations qui entraînent la production d'objectifs personnels. Les demandes sont généralement des requêtes pour que l'organisation receveuse accomplisse des actions ou fournisse des informations à l'organisation demandeuse. Lorsque les organisations sont des agents, ce type de comportement est décrit par des mécanismes de communication et en particulier par la théorie des actes de langage (cf. chap. 6) lorsqu'il s'agit d'agents cognitifs.
- χ C'est le long de la *dimension conative/environnementale* que sont définies les sources de désirs et de répulsions qu'une organisation rencontre dans le monde environnant. Ceci est plus particulièrement clair pour les animats. Pour un prédateur, la proie est une source de désir, pour un robot explorateur, c'est le minerai. Il est possible aussi d'induire des "désirs" indirects. Pour un robot nettoyeur par exemple, on pourra faire en sorte que les déchets soient des sources de "désir" afin qu'ils motivent son action de nettoyage. A l'inverse, certains objets ou agents peuvent être des sources de répulsion. La proie fuit le prédateur, un avion fuit l'autre appareil avec lequel il risque d'entrer en collision. Cela signifie que les objectifs (tant positifs que négatifs) peuvent se trouver dans l'environnement en liaison avec les besoins et buts internes de la dimension suivante.
- ω C'est au travers de la *dimension conative/personnelle* que sont élaborées les motivations qui viennent des besoins internes de l'organisation à partir généralement des informations provenant de son système végétatif: nécessité de trouver des ressources énergétiques, besoin de protection et de reproduction notamment.

Fonction interactionnelle

- φ La *dimension interactionnelle/physique* décrit l'implémentation des systèmes d'entrées/sorties composés de mécanismes de communications, de perceptions et d'action. Lorsque l'organisation est un agent, c'est à ce niveau que sont décrits ses capteurs et ses actionneurs. C'est ici aussi que sont traités les problèmes de bande passante dans les communications et d'une manière générale tout ce qui concerne les couches inférieures des télécommunications. Ceci ne fait pas partie du domaine de la kénétique, mais renvoie aux recherches menées en robotique et dans les réseaux de télécommunications.

- σ La *dimension interactionnelle/sociale* porte sur les primitives d'interaction entre l'individu et l'organisation à laquelle il appartient. Cette dimension, qui n'est généralement pas implémentée dans les SMA, comprend les mécanismes d'élection, de saisines de tribunaux, etc. et d'une manière générale tous les protocoles qui, dans les sociétés démocratiques, assurent aux individus une certaine capacité d'action sur la société prise comme un tout.
- α La *dimension interactionnelle/relationnelle* porte sur la définition des primitives de communication, c'est-à-dire sur la description des types de messages envoyés et acceptés par les agents.
- χ La *dimension interactionnelle/environnementale* décrit l'ensemble des primitives de perception et d'action disponible pour un agent. C'est par ce sous-système que s'effectue (sur le plan conceptuel et logique, les problèmes matériels étant gérés par la dimension physique) l'analyse et le filtrage des perceptions, ainsi que la définition de ses possibilités d'actions.
- ω La *dimension interactionnelle/personnelle* est en général inexistante dans les SMA actuels. Elle correspondrait à la possibilité (pour l'instant humaine) de se parler à soi-même et d'agir sur son propre corps par l'extérieur. On peut aussi penser à un robot qui serait capable d'agir sur lui-même pour se réparer ou modifier certaines de ses fonctionnalités.

Fonction productive (opératoire)

La fonction productive décrit l'ensemble des tâches qu'une organisation sait accomplir, c'est-à-dire à la fois l'ensemble du savoir-faire qu'elle peut appliquer dans le monde et l'ensemble des objets qu'elle peut produire. La fonction productive représente la "raison d'être" d'une organisation, c'est-à-dire sa fonction par rapport au projet qui a amené sa réalisation, ou son rôle par rapport à l'organisation de niveau supérieur.

C'est le long de cette dimension qu'est particulièrement nette l'articulation entre les différents niveaux d'organisation. Une fonction considérée comme "productive" pour une organisation de niveau n sert souvent à la définition d'une fonction d'une autre nature au niveau $n + 1$. Par exemple, si l'on peut considérer que la fonction de production d'un groupe d'agents chasseurs est de ramener de la nourriture, cette fonction peut être décrite au niveau de la société à laquelle ces agents appartiennent comme une fonction végétative. De même, des agents spécialisés en planification d'action pour le compte d'autrui sont décrits comme ayant une fonction productive de planification, alors qu'au niveau de la société, cette fonction est considérée comme organisationnelle. De ce fait, les fonctions productives de niveau n deviennent généralement des rôles non productifs au niveau n . Seules certaines activités sont considérées comme productives sur plusieurs niveaux. Par exemple, un robot récupérateur de minerai sur une base lunaire ou martienne possède la fonction productive de récupération de minerai qui devient, au niveau $n + 1$, un élément de la fonction productive fondamentale de la société de robots qui consiste justement à

extraire et à produire du minerai. Néanmoins, il existe toujours un niveau où ces fonctions deviennent partie intégrante d'une organisation de niveau $n + 1$ et perdent leur caractère producteur. Par exemple, si l'on se situe au niveau de l'ensemble des êtres humains dans son ensemble, la récupération de minerai peut être intégrée à des fonctions végétatives de très haut niveau (production et consommation de biens permettant à une société de mieux survivre).

- ϕ La *dimension productive/physique* décrit l'implémentation des tâches productives, le mécanisme permettant leur activation et leur régulation si nécessaire. Lorsque ces tâches produisent des objets physiques, cette dimension ne fait pas partie du domaine de la kénétique.
- σ La *dimension productive/sociale* décrit l'ensemble des tâches effectuées pour le compte d'une organisation de niveau supérieur, telles que les tâches de direction, de gestion et d'administration d'une organisation.
- α La *dimension productive/relationnelle* porte sur la définition des tâches relationnelles, qui assurent la coopération et la coordination d'actions entre organisations de même niveau. Attention, le long de la dimension relationnelle, il faut bien distinguer la fonction relationnelle/organisationnelle de la fonction relationnelle/productive. La première porte sur la manière dont les composants d'une organisation sont liés et donc sur les communications internes à une organisation. La seconde porte sur l'ensemble des fonctionnalités supplémentaires attribuées à un composant pour que l'organisation puisse communiquer avec d'autres organisations de même niveau. Par exemple, dans l'armée, un "radio" ou spécialiste en communication s'occupe du traitement des messages sur un plan qui n'est pas accessible aux autres (usage d'un matériel spécialisé, communications codées indéchiffrables pour qui ne connaît pas le code) et il se charge de recevoir et transmettre, de coder et décoder les messages qui sont échangés avec d'autres groupes.
- χ La *dimension productive/environnementale* concerne l'ensemble des tâches permettant à une organisation d'être experte dans des activités qui touchent à son environnement, qu'il s'agisse de tâches de diagnostic, de conception ou de contrôle. Les agents spécialistes d'un système multi-experts sont essentiellement caractérisés par des tâches productives qui se déploient le long de cette dimension.
- ω La *dimension productive/personnelle* concerne les tâches d'auto-modification d'un agent et notamment celles qui permettent l'acquisition personnelle de nouvelles expertises ou de nouvelles capacités.

Fonction végétative (conservative)

La fonction végétative traite de l'ensemble des moyens dont dispose une organisation pour maintenir sa structure organisationnelle, c'est-à-dire pour survivre, en se conservant et en développant ses ressources, en maintenant sa structure et en se

reproduisant. D'une certaine manière, les tâches peuvent toujours être ramenées à la fonction végétative en dernier ressort, puisque les objectifs d'une organisation participent à sa conservation et à sa reproduction³. Certaines sont cependant plus directement impliquées dans la conservation, la reproduction et l'entretien d'une organisation que d'autres.

- ϕ La *dimension végétative/physique* décrit l'implémentation du système de conservation et des tâches assurant la conservation individuelle et collective.
- σ La *dimension végétative/sociale* décrit l'ensemble des tâches permettant la conservation de l'organisation de niveau supérieur et la reproduction des organisations de même niveau (procréation, éducation, etc.).
- α La *dimension végétative/relationnelle* concerne la conservation et l'amélioration des liens existant entre plusieurs organisations de même niveau. Lorsque ces organisations sont des agents, c'est sur cette dimension que s'effectue l'entretien du réseau d'accointances ou la vérification des communications et des protocoles de coopération.
- χ La *dimension végétative/environnementale* détaille les tâches portant sur la conservation des ressources environnementales et, en particulier, si cela est nécessaire, sur la défense du territoire par rapport aux autres organisations de même niveau considérées comme des intruses.
- ω La *dimension végétative/personnelle* pour une organisation implique la conservation de soi, la recherche de nourriture, la réparation et l'auto-entretien.

Le tableau 3.1 résume ces rapports entre systèmes et dimensions.

3.2.4 Grille d'analyse fonctionnelle des organisations

Chaque organisation multi-agent ne dispose pas de toutes les fonctions, ni ne développe ses fonctions dans toutes les directions. Par exemple, les organisations réactives n'ont pas (ou peu) de fonction représentationnelle. De même les organisations purement communicantes (qui n'abordent pas le problème de l'interaction avec un environnement) ne sont pratiquement pas développées dans leur dimension environnementale. Enfin, la plupart des systèmes multi-agents actuels, dont les capacités de socialisation sont en général encore très peu développées, n'assurent

³On pourrait dire que les objectifs du concepteur d'un système multi-agent ne peuvent pas se ramener à un problème de conservation et de reproduction et qu'il s'agit d'objectifs supplémentaires irréductibles à la fonction végétative. Je pense au contraire que tout l'art du concepteur d'un système multi-agent est de faire en sorte que les objectifs fonctionnels (et donc productifs) d'une organisation soient obtenus par un effet de bord de mécanismes de recherche de satisfaction liés à chaque agent, chacun agissant comme si sa survie (ou comme si la survie de l'organisation dont il est membre) en dépendait. Cela entraîne alors une intégration des objectifs externes du concepteur dans la fonction végétative de l'organisation. Il faut toutefois remarquer que cette conception est surtout issue de l'école réactive (Steels 1994; McFarland 1994) mais que l'école cognitive ne retient pas spécialement cette approche.

	physique (φ)	social (σ)	relationnel (α)	environnemental (χ)	self (ω)
représentationnel	structure de représentation, modèles de représentation et d'inférence	représentation des rôles et des fonctions, représentation du groupe et de la société	repr. des autres, (but, croyances, compétences)	représentation du monde, représentation des lois de l'univers	auto-représentation de soi et de ses capacités.
organisationnel	contrôle des tâches, type de planificateurs	planification des actions sociales, répartition des tâches et des rôles entre agents.	contrôle communications, planif des interactions, coordination des actions et allocations des tâches	planification des actions dans l'env., organisation des tâches d'expertises.	méta-planification, contrôle des planifications
conatif	nature et implémentation des objectifs, des pulsions et des contraintes	demandes de la société, buts collectifs et contraintes sociales	demandes des autres, demandes de soi vers les autres. engagements envers les autres	sources de plaisir et de répulsions.	pulsions internes, désirs, contraintes internes
interactionnel	implém. des primitives de communications, des capteurs et des actionneurs.	description des interactions entre l'agent et la société,	description des mécanismes de communication entre agents. Définition des performatifs.	description des mécanismes de perception et d'action	auto-communication, auto-actions
productif	architecture des tâches, régulation des actions.	tâches de gestion et d'administration d'une organisation	tâches relationnelles, de coordination, et de négociation, pour le compte d'autres agents	tâches expertises portant sur l'analyse et conception	auto-modifications, apprentissage.
conservatif/ reproducteur	Implémentation du système de conservation, ressources individuelles, mémoire, CPU	conservation de la société, reproduction de l'espèce	conservation des relations, entretien du réseau d'accointances	conservation des ressources, défense et entretien du territoire.	conservation de soi (nourriture), réparation et entretien.

Table 3.1: Fonctions et dimensions d'une organisation

pratiquement aucune dimension sociale. De ce fait le tableau 3.1 peut être compris comme une véritable grille de lecture permettant de situer les organisations les unes par rapport aux autres, en indiquant les fonctions, et donc les rôles, effectivement développées le long de ces cinq dimensions.

Pour ce faire, on indique dans chaque case du tableau si la fonction correspondante est implémentée explicitement dans la dimension associée. Par exemple, le tableau 3.2 présente la grille d'analyse d'une organisation réactive typique dont les dimensions sociales et relationnelles n'existent pas, tout se situant le long des dimensions environnementale et physique.

	Phys.	Social	Relat.	Env.	Pers.
Représentationnel	—	—	—	—	—
Organisationnel	○	—	—	○	—
Conatif	○	—	—	○	×
Interactionnel	○	—	—	○	—
Productif	○	—	—	○	—
Conservatif	○	—	—	○	—

Présence systématique: ○ optionnelle: × Absence: —

Table 3.2: La grille d'analyse des fonctionnalités d'une organisation réactive typique

Cela signifie que cette organisation ne possède aucune représentation de ce qui l'entoure et qu'elle n'est mue que par la perception d'éléments dans l'environnement qui lui servent de stimuli pour déclencher ses actions, qu'elle ne communique pas directement à l'aide de messages avec d'autres organisations, mais indirectement par le biais d'objets situés dans l'environnement (gradients d'odeurs, par exemple), qu'elle ne planifie aucune tâches mais simplement gère les actions environnementales et que ses capacités productives s'expriment uniquement dans l'environnement. La croix de la case conatif/personnelle indique seulement que certaines de ces organisations peuvent être motivées par des stimulations internes (besoins élémentaires) mais que toutes ne disposent pas de cette capacité. A la lecture de ce tableau, on constate tout de suite que, hormis la colonne de la dimension physique qui doit nécessairement être remplie s'il existe au moins un point sur la ligne correspondante, seule la colonne de la dimension environnementale est remplie. Cela signifie que ce type d'organisation est tout entier tourné vers l'environnement et que tous ses comportements sont relatifs à des actions et à des perceptions dans le monde.

Le tableau 3.3 présente au contraire la grille d'analyse d'une organisation purement communicante (par exemple un agent cognitif purement communicant) qui n'a aucun mécanisme d'action ni de perception avec l'extérieur.

Ici la colonne environnementale est presque vide, puisqu'il n'y a pas de contact avec l'environnement. En revanche, si cette organisation dispose d'une représentation d'elle-même et des autres, son système représentationnel sera plus développé que dans le cas d'organisations réactives. Son système conatif est entièrement tourné

	Phys.	Social	Relat.	Env.	Pers.
Représentationnel	○	○	—	—	○
Organisationnel	○	×	○	—	×
Conatif	○	×	○	—	—
Interactionnel	○	—	○	—	—
Productif	○	×	×	×	—
Conservatif	—	—	—	—	—

Présence systématique: ○ optionnelle: × Absent: —

Table 3.3: Grille d'analyse d'une organisation purement communicante typique

vers le relationnel, ce qui signifie que ses objectifs sont le fruit de demandes provenant d'autres agents (et éventuellement de l'organisation de niveau supérieur), et non de besoins internes et encore moins de signaux environnementaux. Le système productif dépend évidemment de la nature de la spécialité de l'agent. S'il s'agit d'un agent d'observation ou d'exécution, les tâches productives seront définies dans la dimension environnementale. En revanche, s'il s'agit d'un agent médiateur ou coordinateur, le système productif peut être développé le long des dimensions sociales et/ou relationnelles. Enfin, la plupart de ces organisations ne comportent (pour l'instant) aucune tâche de conservation, de maintenance ou de reproduction. On a placé une croix dans les cases organisationnel/personnelle et conatif/personnelle pour indiquer qu'il est courant pour une organisation cognitive de disposer de fonctionnalités lui permettant de réorganiser explicitement les tâches à accomplir et de prendre en compte un ensemble de conditions dépendant des demandes provenant de l'extérieur et des engagements déjà formulés.

	Phys.	Social	Relat.	Env.	Pers.
Représentationnel	○	—	○	○	—
Organisationnel	○	—	○	○	—
Conatif	○	○	○	○	○
Interactionnel	○	—	○	○	○
Productif	—	—	—	—	—
Conservatif	○	○	○	○	○

Présence systématique: ○ Absent: —

Table 3.4: La grille d'un agent semblable à un animal social

Le tableau 3.4 donne la grille d'analyse d'un agent ayant des caractéristiques semblables à celles d'un (véritable) animal social. Il ne possède qu'une capacité de représentation minimale de l'environnement et bien qu'il possède des mécanismes de communication directe, ceux-ci s'avèrent peu développés. Son comportement est

entièrement déterminé par des stimuli environnementaux, des pulsions internes et quelques demandes directes. Il est capable de se reproduire et d'avoir un comportement l'amenant à chercher la conservation de son identité et celle du groupe auquel il appartient. Evidemment il ne produit pas directement quelque chose (bien que sa biomasse puisse servir à ses prédateurs et à ses parasites). Le tableau 3.5 représente la grille d'analyse d'une organisation étonnante parce que complète. Elle possède toutes les qualités et son système productif polyvalent peut se spécialiser en fonction des demandes. Elle offre le meilleur système représentationnel que l'on connaisse et elle est capable d'apprendre et de planifier ses actions. Cette organisation, qui se présente comme parfaite du point de vue de nos critères, représente un être humain ou une société humaine...

	Phys.	Social	Relat.	Env.	Pers.
Représentationnel	○	○	○	○	○
Organisationnel	○	○	○	○	○
Conatif	○	○	○	○	○
Interactionnel	○	○	○	○	○
Productif	×	×	×	×	×
Conservatif	○	○	○	○	○

Présence systématique: ○ optionnelle: ×

Table 3.5: La grille d'analyse d'une organisation humaine

3.3 L'analyse structurale

Une organisation multi-agent apparaît comme un système d'une grande complexité. L'analyse structurale tente de donner un ordre à l'ensemble des interactions possibles entre agents en dégagant les relations abstraites qui les relient, et la manière dont elles évoluent au cours du temps. Mais avant d'entrer dans le détail dans cette analyse, il est nécessaire de bien comprendre les différentes manières d'aborder la conception d'une organisation multi-agent.

3.3.1 Agents et tâches

La décomposition d'un problème en termes multi-agents, aborde la difficile mais essentielle question de savoir ce que représente véritablement un agent en relation avec le problème à résoudre. Il s'agit plus exactement de définir le rapport entre la fonction productive individuelle de l'agent et la fonction productive de l'organisation à laquelle il appartient. En effet, la définition d'une organisation multi-agent doit répondre aux questions suivantes:

1. *Que doit-on considérer comme un agent pour le domaine considéré?* Il s'agit d'identifier ce qu'est un agent pour un problème donné, l'analyse s'effectuant à partir d'une approche fonctionnelle (centrée sur la fonction) ou objet (centrée sur l'individu et le produit).
2. *Quelle est la relation entre les agents et les tâches?* Quelles sont les tâches à accomplir, comment les répartir entre les agents et quelle est la redondance des agents par rapport aux tâches?

Un agent se caractérise bien évidemment par son architecture et son comportement. Mais pour le concepteur d'un système multi-agent, l'agent est surtout caractérisé par ce qu'il peut faire, c'est-à-dire par sa fonction productive, et donc par son apport dans une organisation multi-agent. Il s'agit alors d'identifier ce que l'on doit considérer comme un agent par rapport à un problème considéré, à partir d'un point de vue à la fois conceptuel et méthodologique qui ordonne toutes les questions futures. Ce problème est à rapprocher de la phase d'analyse que l'on rencontre dans les méthodes de génie logiciel et, en particulier, dans les méthodes objets où l'on doit *réifier* les entités que l'on représentera dans le système informatique sous la forme d'objets (Ferber 1990).

Approche objet ou fonctionnelle

Dans le cadre du développement de programmes informatiques, on distingue les méthodes de *conception fonctionnelle*, dans lesquelles l'unité de décomposition est la fonction caractérisée par ses entrées-sorties et sa fonction de transfert, et les méthodes de *conception objet* qui mettent l'accent sur l'entité dont on parle, sur les liens qu'elle entretient avec les autres entités et sur les opérations dont elle est capable. Il semble naturel de concevoir les systèmes multi-agents comme une extension naturelle de la conception par objets. Certains chercheurs considèrent en effet que l'agent est un objet doué d'autonomie capable de raisonner sur son environnement et de choisir entre plusieurs alternatives. Mais, de nombreux systèmes qui se réclament de l'approche multi-agent tendent néanmoins à associer à chaque agent un rôle précis caractérisé par sa fonction dans l'organisation globale.

Cela signifie qu'il n'existe pas une seule et unique manière de considérer ce qu'un agent représente. On peut aborder l'identification des agents suivant plusieurs approches, chacune conduisant à la réalisation de systèmes multi-agents différents. En particulier, les travaux de Lesser et Durfee (Durfee et al. 1989) et ceux de Steeb et Camarata (Steeb et al. 1988), ont montré qu'il était possible d'analyser un même problème selon plusieurs angles différents, en privilégiant certaines caractéristiques du problème par rapport à d'autres.

L'exemple du système de contrôle aérien, traité initialement par Steeb et Camarata, illustre cette diversité d'approche (fig. 3.6). Quels doivent être les agents responsables de l'évitement et du contrôle des différentes opérations? Les avions, les secteurs d'espace ou les différentes phases de pilotages telles que le décollage, l'atterrissage ou le suivi de plan de vol?

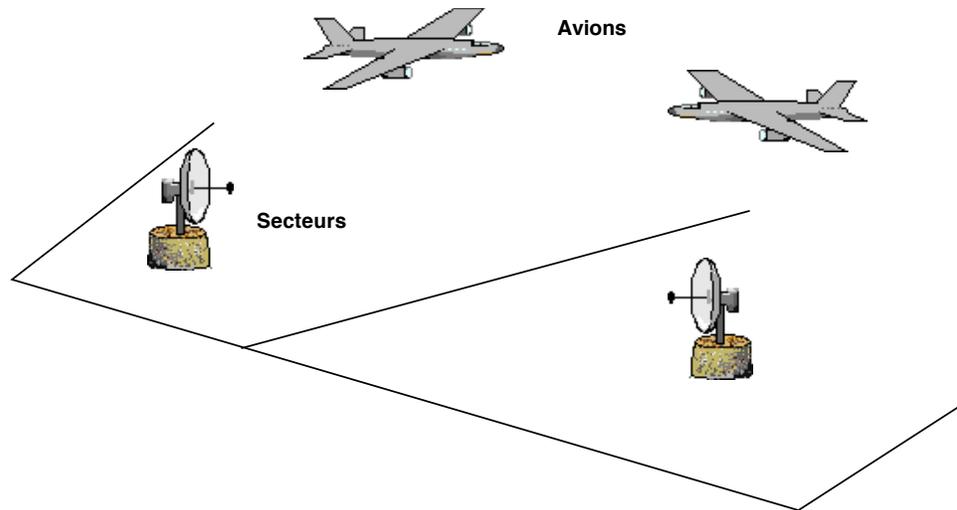


Figure 3.6: Dans un système de contrôle aérien, on peut considérer comme agents les avions, les secteurs ou les phases de vols, selon que l'on privilégie une approche objet, spatiale ou fonctionnelle.

Dans le premier cas, on dira que l'approche est *centrée sur l'objet*: c'est l'avion qui est responsable de ses propres mouvements et qui décide des actions qu'il doit entreprendre à partir d'une interaction avec les autres appareils. Chaque avion dispose alors de toutes les fonctionnalités nécessaires à son activité et, en particulier, des compétences pour gérer les diverses phases de pilotage.

Dans le second, l'approche est *centrée sur l'espace*. Chaque partie de l'espace est une portion d'environnement qui contrôle les mouvements de tous les objets qui se trouvent dans son secteur, les avions n'étant alors que des objets passifs qui ne font qu'obéir aux ordres de la tour de contrôle. Les opérations de gestion des différentes phases de pilotage ne sont plus associées aux avions, mais aux agents représentant les différentes portions d'espace.

Enfin la troisième catégorie est caractéristique d'une *approche fonctionnelle*: les phases de pilotage représentent les experts qui gèrent de manière centralisée les mouvements de tous les avions. Dès qu'un avion se trouve dans une phase spécifique (suivi de plan de vol, attente, atterrissage, roulage, procédures d'urgence...) il est pris en charge par un spécialiste de la question.

Seule la première et la troisième catégorie peuvent être appliquées dans presque tous les cas de figure, alors que la seconde représente des parcelles d'environnement comme un agent. Bien que cela soit possible pour les agents situés⁴, nous ne la traiterons pas dans ce chapitre, car elle est de fait assez peu employée. Souvent, les approches fonctionnelles et objets sont antinomiques l'une de l'autre. Prenons un autre exemple caractéristique des systèmes multi-agents, tels que celui du diagnostic de défaillance dans un réseau de distribution. Faut-il considérer les composants de ce réseau, c'est-à-dire les nœuds de distribution ou de transformation, comme des agents (approche objet), ou au contraire préférer une approche fonctionnelle qui

⁴Cette approche sera reprise lors de la modélisation des environnements, au chapitre 4.

considérera que les agents sont des spécialistes en diagnostic, chacun utilisant une méthode particulière pour résoudre le problème dans sa globalité (modèle structuro-fonctionnel, modèle de défaillance (AMDEC), analyse par cas (case base reasoning), etc.)? La figure 3.7 illustre ces deux possibilités.

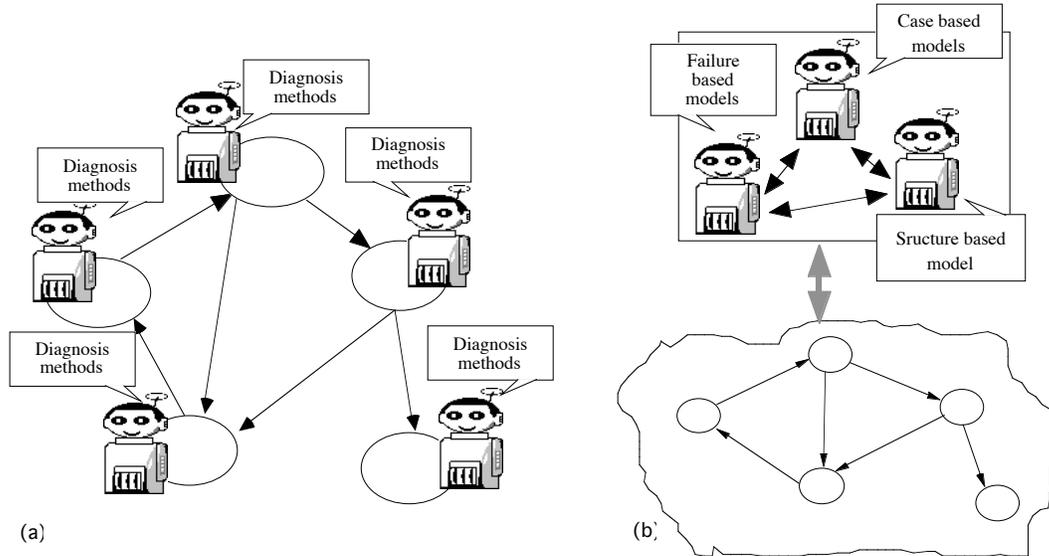


Figure 3.7: Représentation schématique d'organisations fondées sur l'approche objet (A) ou fonctionnelle (B), appliquées au diagnostic de réseau

L'approche fonctionnelle est souvent considérée comme plus simple lorsque les agents ne sont pas physiquement situés. Les agents représentent alors des spécialistes caractérisés par leur fonction et leur rôle dans l'organisation générale. Cette décomposition s'apparente aux méthodes d'analyse fonctionnelle de l'informatique traditionnelle et à l'analyse systémique classique qui identifie les différentes activités et fonctions des systèmes et non pas les entités sur lesquelles s'appliquent ces fonctions. En revanche, l'approche objet dans le cadre des systèmes multi-agents se situe dans le prolongement des analyses et conceptions objets caractéristiques de la tendance actuelle du génie logiciel.

Les perspectives objet et fonctionnelle peuvent néanmoins être combinées ensemble, en considérant que l'approche fonctionnelle définit une sous-division de l'approche objet. Par exemple, on pourra suivre une analyse objet, dans laquelle la première décomposition consiste à prendre les avions comme agents, et entreprendre ensuite une décomposition fonctionnelle de chaque agent-avion en considérant le système de contrôle de chaque appareil comme un système multi-agent et ainsi analyser chacun des organes de contrôle de l'avion comme un agent à part entière. Ceci résulte simplement de l'emboîtement des niveaux d'organisation, comme nous l'avons vu plus haut, et du fait qu'un niveau analysé de manière objet peut être lui-même décomposé en entités de niveau inférieur qui sont alors identifiées comme des fonctions.

La perspective envisagée pour concevoir un système multi-agent, n'est cependant pas neutre. Ce choix engage l'ensemble de la réalisation du système. Les approches centrées sur les objets sont généralement plus "radicales" en ce sens qu'elles supposent une parfaite autonomie des agents impliqués. Elles reposent presque entièrement sur la définition des modèles d'interaction et sur leurs capacités à pouvoir résoudre le problème à partir d'une démarche totalement distribuée. En revanche, les modèles fonctionnels s'avèrent plus "classiques" car ils ne remettent généralement pas en cause les méthodes d'analyse des compétences.

Décomposition verticale et horizontale

Une démarche fonctionnelle peut s'exprimer de deux manières différentes, selon que l'on privilégie une conception *horizontale* ou *verticale* des fonctions. Une division fonctionnelle verticale associe à chaque fonction une activité complexe centrée sur une application particulière. Dans l'exemple du trafic aérien, cela revient à lier, comme nous l'avons fait plus haut, une fonction à une phase du plan de vol.

Au contraire, une décomposition horizontale met l'accent sur ce qui est partagé par les fonctions verticales. C'est cette démarche que nous avons choisi lors de notre analyse fonctionnelle. Les six fonctions (représentationnelle, conative, relationnelle, organisationnelle, interactionnelle et végétative) ont été découpées horizontalement et caractérisent donc les fonctions horizontales d'une organisation artificielle.

Cette dualité horizontale/verticale a été tout particulièrement étudiée dans le cadre de la structuration des agents (Fodor 1983). Par exemple, pour un robot mobile et manipulateur, une décomposition verticale reviendra à associer une fonction pour chacune de ses actions. Il y aura par exemple la fonction de recherche et de transport d'objet, la fonction de placement d'un objet sur un autre objet, la fonction de forage, la fonction d'évitement d'obstacle, etc. Au contraire, dans le cadre d'une approche horizontale on mettra en évidence la fonction de mémorisation et d'inférence, celle de perception, celle de coordination des actions concrètes, etc., sans que ces fonctions soient définies pour des tâches précises.

Il est évidemment bien rare de voir un système artificiel totalement organisé selon un seul de ces deux axes: les décompositions hybrides sont plus courantes que les architectures purement verticales ou horizontales. Par exemple, il est possible de concevoir un système décomposé en premier lieu de manière horizontale en distinguant les activités périphériques (interprétation et production de messages, perception, mécanismes moteurs), des activités cognitives expertes (interprétation de situations, savoir-faire), des mécanismes intentionnels et de contrôle. Dans un second temps on redécompose verticalement chacun de ces modules en distinguant les différentes formes de perception et de motricité ou les différentes expertises (expertises pour se mouvoir dans un univers, pour analyser des scènes visuelles, pour effectuer des tâches spécifiques) du domaine de compétence.

Néanmoins, les décompositions verticales sont plus liées à la résolution d'un problème particulier, alors que les décompositions horizontales sont plus génériques dans leur approche. C'est pourquoi, les agents réactifs ou ceux qui sont conçus avec des techniques informatiques traditionnelles sont plus souvent décomposés de

manière verticale, alors que les agents plus “IA” ont plutôt tendance à être élaborés de manière horizontale.

3.3.2 Relations abstraites

Les relations organisationnelles décrivent au niveau abstrait les formes d'inter-relations qui peuvent avoir lieu entre les classes fonctionnelles d'agents, c'est-à-dire entre les rôles. On distingue deux types de relations organisationnelles: les *relations statiques*, qui forment le squelette d'une organisation, caractérisent la structure organisationnelle préexistante à tout fonctionnement et sont définies en dehors de toute exécution, et les *relations dynamiques*, qui constituent la chair des organisations et décrivent les relations qui sont modifiées lors des interactions. Le tableau 3.6 décrit les principales relations que l'on rencontre dans les organisations multi-agents.

Relations	Statique	Dynamique
Accointance	—	—
Communicationnelle	—	—
Subordination	maître/esclave	demande de service
Opérative	dépendance entre tâches	engagement de faire
Informationnelle	dépendance entre savoirs	engagement de validité
Conflictuelle	—	—
Compétitive	—	—

Table 3.6: Ensemble des relations organisationnelles. Un tiret indique qu'il n'y a pas de nom particulier pour distinguer les formes statiques ou dynamiques d'une relation.

On peut distinguer sept types principaux de relations:

1. Une *relation d'accointance* entre deux agents A et B indique que A connaît B , qu'il en a une représentation et qu'il peut adresser ses communications directement à cet agent. Il s'agit de la relation minimale entre deux agents cognitifs. Elle sert généralement de support à toutes les autres relations (fig. 3.8.a).
2. L'existence d'une *relation communicationnelle*, ou plus exactement d'un *canal de communication*, entre deux agents A et B indique que A peut envoyer des messages à B . Les relations d'accointances sont les supports privilégiés de canaux de communication (fig. 3.8.b).
3. Les *relations de subordination* décrivent un transfert d'exécution entre un agent demandeur A et un agent exécutant B . Lorsque cette relation est statique, cela

signifie que B ne peut pas refuser la demande de A et donc que A et B sont dans une relation maître/esclave (fig. 3.8.d). Lorsqu'elle est dynamique, cela indique que A ne fait que demander des services à B et que B peut refuser les services de A (fig. 3.8.e).

4. Les *relations opératives* représentent des dépendances liées aux tâches. Un agent A qui doit accomplir une tâche T_x peut avoir besoin, pour qu'elle soit réalisée, des résultats d'une tâche T_y d'un agent B . On dira dans ce cas, que A dépend de B pour ses actions. La forme dynamique de ces relations recouvrent alors ce que l'on appelle les *dépendances d'engagements*. Un agent, en s'engageant à accomplir quelque chose, permet à d'autres agents de tabler sur cette réalisation et donc de planifier leurs propres actions, tout en dépendant de l'accomplissement de cet engagement (fig. 3.8.f).
5. Les *relations informationnelles* portent sur les dépendances de validité entre les différents savoirs d'un agent. Lorsque le savoir d'un agent A dépend de celui de B , cela signifie que A n'a pas expérimenté par lui-même, mais qu'il s'en remet à B en ce qui concerne certaines croyances. On traitera au chapitre 5 de ces dépendances.
6. Les *relations conflictuelles* indiquent que plusieurs agents sont en conflit pour accéder à certaines ressources. L'existence de telles relations indique qu'il est nécessaire de coordonner les accès aux ressources, éventuellement à partir d'une négociation entre agents.
7. Les *relations compétitives* correspondent à une situation de compétition entre agents et signalent que leurs buts sont incompatibles.

3.3.3 Les modes de couplage

Si les organisations peuvent être caractérisées par la qualité des liens qui unissent les agents, elles sont aussi décrites par leurs capacités à se modifier dans le temps et à s'adapter à l'évolution des besoins, que ces besoins proviennent de sources endogènes (modification du nombre d'agents, modification des rôles des agents) ou exogènes (modification des caractéristiques environnementales). Pour s'adapter, les organisations doivent pouvoir se modifier. Il est possible de caractériser trois formes d'organisations en fonction du couplage des composants.

1. Dans une organisation à *couplages fixes*, chaque composant possède un (ou éventuellement plusieurs) rôle dont il ne bouge plus. Les relations entre agents sont elles aussi figées, les réorganisations étant impossibles tant au niveau abstrait de la structure organisationnelle qu'au niveau de l'organisation concrète. Ces organisations, qui ne peuvent que très difficilement s'adapter à leur environnement, sont plus caractéristiques du génie logiciel classique que de l'approche kénétique. Nous ne les utiliserons que pour décrire le fonctionnement d'agents — c'est-à-dire comme des modèles d'architecture d'agents (cf. section 3.6) — mais non comme des modèles de systèmes multi-agents.

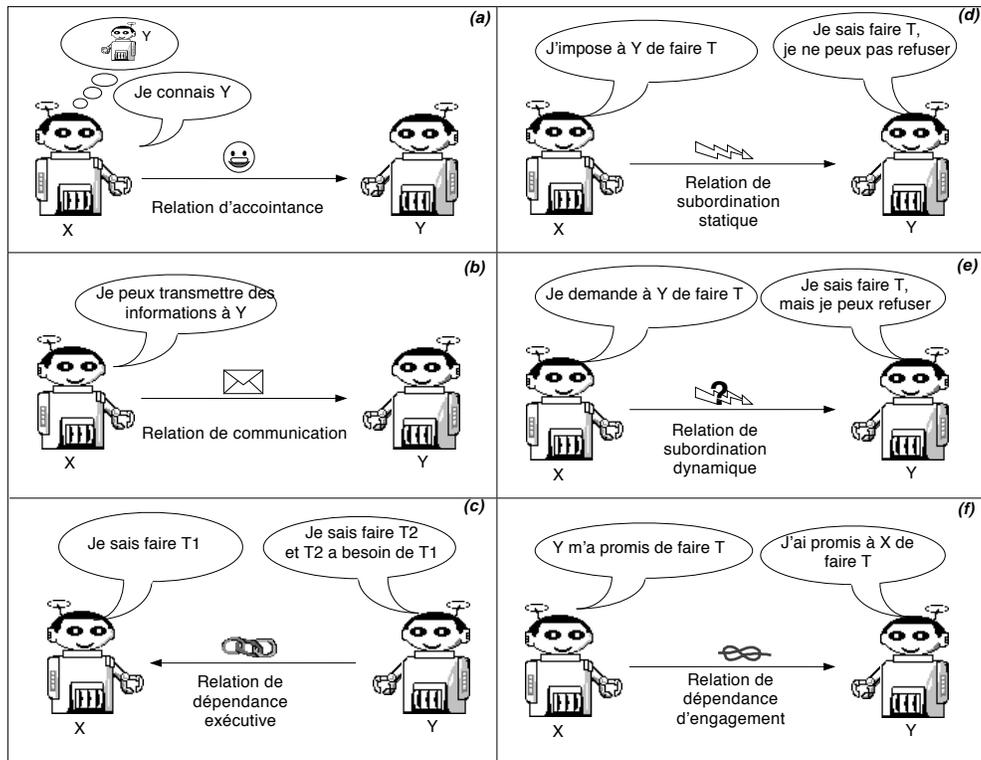


Figure 3.8: Visualisation icônique des principales relations entre agents (les relations informationnelles, conflictuelles et compétitives ne sont pas représentées).

2. Les *couplages variables* correspondent à des structure organisationnelles fixes dont les concrétisations sont variables. Les relations entre agents peuvent évoluer mais dans le cadre de mécanismes bien précisés. Ce type de couplage est très employé dans les systèmes multi-agents. Nous en verrons des exemples constamment.
3. Les *couplages évolutifs* caractérisent des structures organisationnelles variables, l'organisation concrète pouvant être ou non variable. Les relations abstraites entre agents peuvent alors évoluer en fonction des performances de l'organisation.

3.3.4 Structures de subordination et prises de décision

Les relations de subordinations caractérisent la structure de contrôle, c'est-à-dire la manière dont les agents peuvent faire effectuer certaines tâches à d'autres agents. Certaines structures s'avèrent particulièrement classiques. Nous distinguerons essentiellement entre les *structures hiérarchiques*, qui supposent que les relations de subordination forment une arborescence, et les *structures égalitaires*, dans lesquelles un agent peut demander à n'importe quel autre agent d'accomplir une tâche, ce dernier pouvant éventuellement refuser.

Associées à un couplage fixe, les *organisations hiérarchisées* caractérisent des

systèmes très centralisés, très “militaires” pourrait-on dire, dans lesquels un chef commande à un sous-chef, qui commande à un sous-sous-chef et ainsi de suite. On retrouve cette organisation dans la majorité des programmes informatiques, la relation de subordination s’exprimant alors comme un appel à un sous-programme. Au contraire, avec un couplage variable, cette structure de subordination fait généralement intervenir une compétition entre les composants de bas niveau, les composants de haut niveau se chargeant d’arbitrer cette compétition et de sélectionner les vainqueurs. On parle alors parfois de *structures compétitives* pour décrire les organisations à structure hiérarchique et à couplage variable.

Les *structures égalitaires* sont plus caractéristiques des organisations dans lesquelles tous les agents interviennent uniformément dans la décision finale. Lorsque le couplage est fixe, les composants agissent de manière distribuée dans la prise de décision. On retrouve ce type d’organisation dans des architectures de type connexioniste (cf. section 3.6). Avec un couplage variable, cette structure engendre des organisations fondées sur le modèle des marchés économiques, où ceux qui font des demandes et ceux qui font des propositions s’arrangent finalement pour y trouver leur compte.

3.3.5 Types de constitution des structures organisationnelles

Les structures organisationnelles peuvent être constituées de deux manières:

- a. Elles peuvent être définies a priori par le concepteur et l’on parle alors d’*organisations prédéfinies*, ce qui signifie que les relations abstraites, qu’elles soient statiques ou dynamiques sont déterminées à l’avance. Dans le cas d’organisations évolutives, c’est l’ensemble des relations abstraites possibles et l’ensemble des transformations qui sont connues à l’avance. On pourra se référer à (Le Strugeon 1995) pour une première analyse des organisations évolutives prédéfinies.
- b. Elles peuvent aussi être définies a posteriori et l’on parle alors d’*organisations émergentes*. Celles-ci, qui ne comprennent le plus souvent que des agents réactifs, sont caractérisées par l’absence de structure organisationnelle prédéfinie, celle-ci résultant totalement des interactions entre agents. Dans ce cas, les positions et les relations ne sont pas déterminées à l’avance mais apparaissent comme le produit des comportements de chacun des agents. Plus exactement, la répartition des fonctions et des tâches suit un processus d’auto-organisation qui permet à une organisation d’évoluer et de s’adapter facilement aux modifications de l’environnement et aux besoins d’un groupe d’agents.

Bien qu’il existe aussi une certaine forme d’émergence dans les organisations prédéfinies variables ou évolutives, cette “émergence” est toujours contrôlée par des schémas de communication bien établis et les organisations qui en résultent sont toujours décrites par des structures organisationnelles très précises, ce qui n’est pas le cas des organisations émergentes.

3.4 Les paramètres de concrétisation

La question qui se pose à un concepteur de système multi-agent est la suivante: sachant la tâche générale à résoudre, comment répartir l'ensemble des compétences aux différents agents de telle façon que le travail soit effectivement accompli? Préfère-t-on des agents très spécialisés qui ne possèdent qu'une compétence, ou bien favorise-t-on des approches avec des agents *totipotents* dans lesquelles chaque agent possède toutes les compétences voulues, seul le nombre faisant la différence? Si pour concevoir une voiture de course, on fait appel à des spécialistes disposant de compétences différentes (spécialistes des moteurs, des châssis, des pneumatiques, etc.), chacun apportant des compétences propres qui l'empêchent de remplacer un autre spécialiste dans son travail, on peut au contraire imaginer des agents peu spécialisés pour transporter des objets de différentes formes. Chaque agent peut alors prendre la place d'un autre agent si celui-ci fait défaut, ce qui augmente la fiabilité du système en cas de panne.

Afin de mieux expliquer ce que nous entendons par spécialisation et redondance, il est nécessaire de préciser quelques points. Afin de simplifier la présentation, on supposera que chaque tâche élémentaire peut être entièrement effectuée par un agent seul et qu'il n'y a donc pas d'augmentation qualitative à faire travailler plusieurs agents entre eux. Par exemple, le nettoyage d'une surface peut être réalisée par un seul agent, le fait de faire intervenir un groupe d'agents ne présentant qu'une amélioration quantitative. On éliminera donc de cette discussion toutes les tâches élémentaires qui réclament explicitement plusieurs agents travaillant ensemble pour être accomplies⁵.

Dans ce cas, si l'on appelle C_a l'ensemble des compétences dont dispose un agent a , C^P l'ensemble des compétences nécessaires à la réalisation d'un travail P et C_A l'union des compétences d'un ensemble d'agents A , alors on peut dire que le travail P peut être accompli si $C^P \subseteq C_A$, c'est-à-dire si toutes les compétences nécessaires à l'accomplissement d'un travail sont au moins présente dans l'ensemble A . On dira qu'un agent a est *totipotent* s'il dispose de toutes les compétences nécessaires pour effectuer ce travail par lui-même, c'est-à-dire si $C^P \subseteq C_a$. On notera par C_a^P l'ensemble des compétences utiles à un agent a pour effectuer un travail P : $C_a^P = C_a \cap C^P$.

A partir de ces définitions, il est possible de caractériser la manière dont un travail est réparti dans un système multi-agent par deux paramètres: le degré de spécialisation et celui de redondance.

1. Le *degré de spécialisation* (H_a^P) d'un agent a pour un problème P indique le taux de compétence qu'un agent possède par rapport au nombre de compétences nécessaires pour résoudre un problème. On le définit ainsi:

$$H_a^P = \frac{\text{card}(C^P) - \text{card}(C_a^P)}{\text{card}(C^P) - 1}$$

⁵Cette limitation n'est pas trop pénalisante. Si une tâche s'avère trop complexe pour un seul agent, on la décomposera en éléments plus simples de manière à se retrouver dans le cas favorable. Si des actions nécessitent absolument l'intervention simultanée de deux agents, alors on considérera qu'il s'agit d'un problème de coordination d'action et nous le traiterons au chapitre 8.

Si $H_a^P = 0$, l'agent est donc *totipotent* pour le problème P puisqu'il possède toutes les compétences requises. Inversement, lorsqu'il ne possède qu'une seule compétence, le taux est égal à 1, et plus ce taux est proche de 1, plus l'agent est *spécialisé* et plus sa contribution à l'ensemble du problème sera faible. On peut généraliser ce degré de spécialisation à un système multi-agent en considérant que le degré de spécialisation général est égal à la moyenne des spécialisations de chacun des agents:

$$H^P = \frac{1}{\text{card}(A)} \sum_{a \in A} H_a^P$$

2. Le *degré de redondance* caractérise le nombre d'agents possédant les mêmes compétences. Si l'on appelle P_c le nombre d'agents disposant de la compétence c ($P_c = \text{Card}(\{a \mid c \in C_a\})$), alors le degré de redondance pour une compétence peut lui aussi être compris entre 0 et 1 en utilisant la formule:

$$R^c = \frac{P_c - 1}{\text{card}(C) - 1}$$

Si R^c vaut 0 cela signifie qu'il n'existe qu'un seul agent capable d'effectuer une tâche réclamant une compétence c . Le degré de redondance vaut 1 lorsque tous les agents possèdent cette compétence, la redondance étant maximale. On peut ensuite calculer le degré de redondance moyen de la manière suivante:

$$R = \frac{1}{\text{card}(C)} \sum_{c \in C} R^c$$

Par exemple, dans le cas d'un système fondé sur la coopération entre spécialistes, les agents ne peuvent se remplacer, la spécialisation est élevée (proche de 1) et la redondance faible (égale à zéro s'il n'existe qu'un seul agent par spécialité). Au contraire, des agents plus polyvalents verront leur spécialisation baisser et leur redondance s'élever. On peut résumer les différents types d'organisation des compétences par le diagramme de la figure 3.9. Il existe quatre types extrêmes d'organisation des

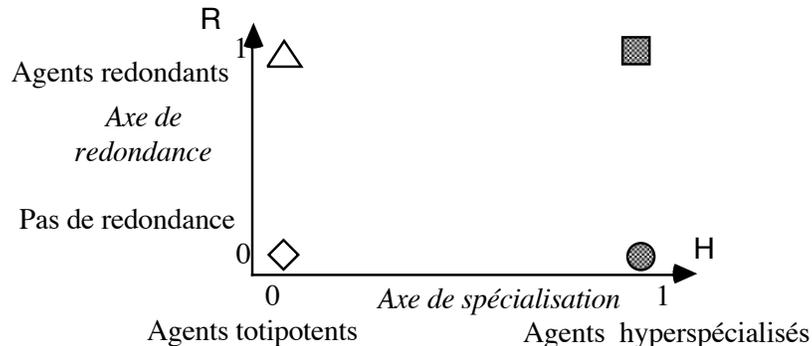


Figure 3.9: Les différents types d'organisation des compétences

compétences en fonction des degrés de spécialisation et de redondance.

- a. *Organisation hyperspécialisée non redondante*. Spécialisation 1, redondance 0: un agent ne sait faire qu'une tâche et chaque tâche n'est effectuée que par un seul agent. Ce type d'organisation est caractéristique des approches purement fonctionnelles dans lesquelles chaque fonction est représentée sous la forme d'un agent.
- b. *Organisation spécialisée redondante*. Spécialisation 1, redondance 1: chaque agent ne sait faire qu'une tâche et chaque tâche est effectuée par tous les agents. Il s'agit d'un cas limite dans lequel tous les agents ont exactement une seule et unique compétence. On utilise parfois ce type d'organisation localement pour augmenter la fiabilité d'un système (par exemple, un système de calcul comprenant plusieurs processeurs en parallèle qui calculent la même fonction et votent pour décider du meilleur résultat est fondé sur ce type d'organisation).
- c. *Organisation généraliste redondante*. Spécialisation 0, redondance 1: un agent sait faire plusieurs tâches et chaque tâche est effectuée par un grand nombre d'agents. Lorsque la redondance est égale au nombre de tâches, alors tous les agents sont totipotents. Le problème réside alors dans la répartition des tâches au sein d'agents pouvant a priori tout faire. Nous en verrons un exemple dans le cas de répartition réactive avec le système MANTA (cf. chap. 7).
- d. *Organisation généraliste non redondante*. Spécialisation 0, redondance 0: un agent sait faire toutes les tâches, mais chaque tâche n'est effectuée que par un seul agent. C'est aussi un cas limite où le système multi-agent se résume à un seul et unique agent capable de tout faire par lui-même.

Le cas général se situe vers le milieu de la figure, c'est-à-dire pour une spécialisation et redondance quelconque: les agents savent accomplir plusieurs tâches, mais pas toutes, et chaque tâche peut être réalisée par plusieurs agents. Le problème du concepteur et de l'organisateur d'un système multi-agent est de parvenir à une organisation performante et qui soit capable de faire montre d'adaptabilité lorsque les conditions environnementales viennent à être modifiées. Pour l'aider dans cette tâche, on peut définir des paramètres quantitatifs qui prennent en compte les qualités de la solution obtenue ou l'efficacité du système. Dans l'exemple des robots explorateurs on peut vouloir connaître le temps mis par un groupe d'agents pour récupérer une certaine quantité de minerai; dans celui de l'évitement de collision entre avions, on préférera connaître le nombre de conflits (le nombre de fois où des avions se croisent à une distance inférieure à une valeur limite) ou la déviation par rapport aux trajectoires nominales établies par leur plan de vol.

On peut ainsi établir une fonction de performance à partir de nombreux paramètres tels que le nombre ou la densité des agents, la bande passante et les temps de communication, les caractéristiques individuelles, etc. Si l'on se restreint aux deux seuls paramètres R et H , c'est-à-dire à la redondance et à la spécialisation des agents, on peut définir une fonction de performance organisationnelle que l'on notera $\mu_s(R, H)$ qui indique la performance du système S en fonction seulement des paramètres R et H . Le problème du concepteur consiste donc à déterminer une organisation

permettant d'optimiser la valeur de cette fonction de performance, comme le montre la figure 3.10. Si le système peut évoluer, par modification de la compétence des agents, alors il lui est possible de se déplacer dans l'espace $R \times H$ par ses propres moyens et donc de trouver une meilleure valeur de cette fonction de performance. Par exemple, le passage d'un ensemble d'agents totipotents à une plus grande spécialisation permet généralement d'améliorer la performance du système en lui évitant des tâches coordinatrices inutiles. Inversement, l'augmentation de la totipotence améliore l'adaptabilité du système. Enfin, la redondance augmente les performances qualitatives du système, mais également les besoins en coordination.

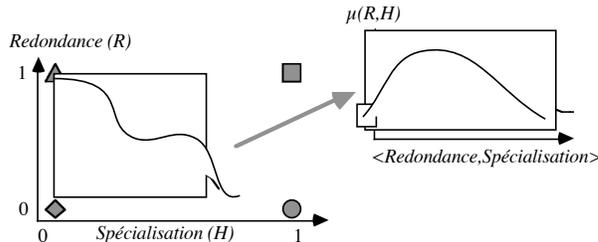


Figure 3.10: Le problème du concepteur d'un système multi-agent consiste à optimiser la valeur de la fonction de performance organisationnelle en fonction de la redondance et du degré de spécialisation des agents.

3.5 Analyse d'une organisation concrète

Comment, à partir de cet ensemble de paramètres, est-il possible de définir une organisation concrète? Il suffit pour cela d'associer aux caractéristiques issues des analyses fonctionnelle et structurale, les facteurs de spécialisation et de redondance correspondant à l'instanciation d'une structure organisationnelle. Ainsi, une organisation peut être décrite par la donnée de:

1. L'approche choisie (objet, fonctionnelle verticale, fonctionnelle horizontale) et la fonction productive de l'organisation, c'est-à-dire l'ensemble des moyens techniques, des opérations, des plans, des savoirs et des savoir-faire qui doivent être mis en œuvre pour satisfaire l'objectif global du système et la raison d'être du système. Cette fonction est liée au contexte de l'organisation, c'est-à-dire à l'ensemble des contraintes et des conditions caractérisant la manière dont le système multi-agent devra opérer.
2. La grille d'analyse de l'organisation qui décrit la manière dont les six fonctions essentielles sont projetées le long des cinq dimensions d'analyse.
3. L'ensemble des rôles associés aux fonctions ainsi que les relations abstraites qui relie ces différents rôles.

4. La manière dont les relations de subordination structurent le rapport d'autorité entre les agents.
5. La valeur des paramètres structuraux que sont le couplage et la constitution.
6. La valeur des paramètres de concrétisation, c'est-à-dire la spécialisation et la redondance.

La figure 3.11 résume l'ensemble de ces caractéristiques.

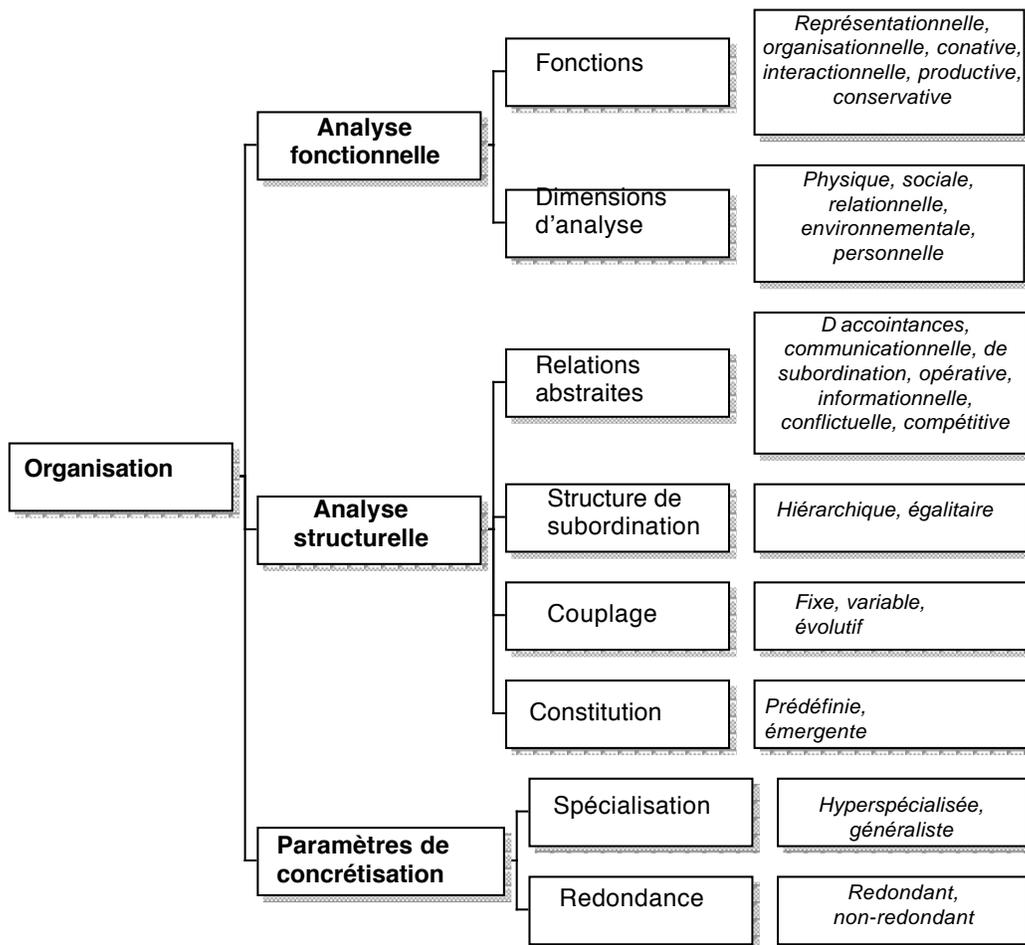


Figure 3.11: Les différents éléments fonctionnels, structuraux et de concrétisation d'une organisation artificielle

3.5.1 L'exemple des robots explorateurs

Pour montrer comment ces différentes caractéristiques peuvent être expliquées dans un exemple, nous reprendrons celui des robots explorateurs (cf. chap. 1). Il ne s'agira pas de traiter intégralement ce problème puisque l'ensemble des organisations possibles, même pour un exemple aussi simple, est déjà considérable. En revanche, nous verrons un ensemble de structures caractéristiques afin de comprendre comment les différents paramètres que nous avons examinés peuvent se combiner entre eux.

Fonction productive et contexte

La première question consiste à savoir ce que doit faire le système: Quel est son objectif général et quelles sont ses conditions de fonctionnement? On suppose qu'un ensemble de robots doit explorer la planète Mars pour récupérer du minerai. Pour cela, les robots doivent trouver un lieu où se trouve du minerai, forer pour ramener le minerai à la surface et transporter le minerai jusqu'à la base. On suppose que chacune de ces tâches peut être accomplie indépendamment les unes des autres et qu'elle ne nécessite qu'un seul robot pour être accomplie. La fonction productive P_{explo} de l'organisation se ramène donc aux trois tâches:

$$P_{\text{explo}} = \{\text{trouver-minerai}, \text{ forer}, \text{ transporter-minerai}\}$$

L'approche choisie est fonctionnelle verticale: chaque robot dispose d'un certain nombre de compétences lui permettant d'accomplir au moins l'une des trois tâches. On suppose que le système doit travailler de manière autonome, qu'il n'existe aucune carte de Mars et qu'on ne sait initialement pas où se trouvent les zones intéressantes de minerai. De plus, on suppose que des robots peuvent devenir inopérants pour des raisons diverses (écrasement par une météorite, chute dans une crevasse, panne diverse, etc.).

Analyse fonctionnelle

Dès l'analyse fonctionnelle, il faut exprimer ses options et choisir entre agents cognitifs et réactifs. Ce sont les fonctions représentationnelle et communicationnelle qui font la différence ici. Des agents cognitifs tenteront de construire une carte précise du terrain et de se positionner par rapport à cette carte. Par exemple, un agent cognitif ayant détecté du minerai indiquera la position de l'endroit où il a trouvé du minerai aux autres sous la forme de coordonnées cartésiennes ou polaires. En revanche, un agent réactif se contentera d'envoyer un signal dont l'intensité varie de manière inverse à la distance, afin de guider les agents dans sa direction.

Autre intérêt de l'analyse fonctionnelle: déterminer les contraintes imposées à la fonction végétative. Dans quelle mesure les agents sont-ils autonomes sur le plan énergétique et quels sont les moyens mis à leur disposition pour éventuellement se recharger. Faut-il prévoir des agents ravitailleurs et donc ajouter de nouvelles tâches? Nous ne le supposerons pas dans notre analyse structurale, mais c'est un point qu'il faudrait résoudre dans un cas réel.

Analyse structurale

C'est lors de l'analyse structurale que l'on choisit la forme d'organisation désirée. Afin de montrer quelques-unes des variantes possibles, nous traiterons les structures organisationnelles suivantes:

- Organisations par équipe fixe (structure fixe-hiérarchique-prédéfinie).
- Organisations à structure variable-égalitaire-émergente.

- Organisations par appel d'offre (structure variable-égalitaire-prédéfinie).
- Organisations à structure évolutive.

3.5.2 Organisations à structure fixe-hiérarchique-prédéfinie

Dans ce type d'organisation, que l'on appelle aussi *organisation par équipes fixes*, les rôles ainsi que les relations peuvent être définis de manière statique, à l'avance. La figure 3.12 montre une organisation possible comprenant deux équipes indépendantes dont l'organisation est fixe et chaque agent spécialisé. Chaque fonction productive est réalisée par un agent, mais plusieurs agents peuvent effectuer la même fonction. Chaque robot détecteur est associé à un ensemble bien défini de foreurs, chaque foreur est associé à un ensemble de transporteurs. Les détecteurs commandent les foreurs et les foreurs commandent les transporteurs. Les foreurs dépendent des détecteurs, puisque l'action de forage ne peut se faire que si l'on a détecté du minerai et, de même, les transporteurs dépendent des foreurs.

La structure de subordination est hiérarchique, les détecteurs se situant au niveau supérieur et les transporteurs au niveau inférieur. Les communications vont et viennent le long de cette hiérarchie définie par les relations d'accointance, mais les relations de subordination suivent toutes le même trajet. Les détecteurs disent aux foreurs ce qu'ils doivent faire et les foreurs donnent des ordres aux transporteurs. Il n'y a donc nécessairement qu'un seul détecteur par équipe et les transporteurs sont associés une bonne fois pour toutes à un foreur.

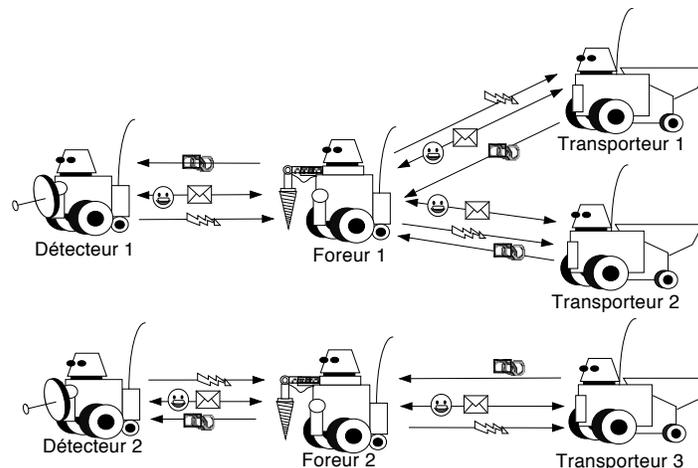


Figure 3.12: Une organisation fixe dont toutes les relations sont statiques. De plus, ici, la structure générale est hiérarchique.

Paramètres de concrétisation

Ce type d'organisation est hyperspécialisé puisqu'un agent n'accomplit qu'une seule action ($H = 1$), mais la redondance n'est pas précisée. Si pour la détection la redondance dans une équipe est de 0 (mais peut être plus élevée en fonction du

nombre d'équipe), elle est totalement fonction de la composition des équipes et donc de la concrétisation de la structure organisationnelle. On peut imaginer des équipes minimales ne comprenant qu'un seul foreur et qu'un seul transporteur (et dans ce cas la redondance de l'équipe est égale à 0), ou bien avoir des équipes plus importantes si l'on pense que les gisements de minerai sont riches et qu'il nécessitent un travail important. La capacité d'adaptation d'une organisation fixe est par définition minimale: qu'un agent essentiel vienne à disparaître et toute une équipe est perdue. D'autre part, elle est particulièrement peu performante, puisque si le détecteur d'une équipe tarde à trouver du minerai, ses foreurs et ses transporteurs resteront inutilisés.

Il existe de nombreuses variantes de ce type d'organisation. On peut par exemple imaginer que des agents coordonnateurs indépendants dirigent les opérations directement. Ces agents coordinateurs peuvent être des robots qui accompagnent les équipes sur le terrain ou des agents logiciels qui restent à la base, reçoivent des informations concernant la situation et donnent des ordres en retour. Les remarques restent valides pour ces variantes.

En fait, les organisations fixes ne sont pas bien adaptées au problème des robots explorateurs. On les rencontre surtout lorsque le nombre d'agents n'est pas très important et, tout particulièrement, dans les systèmes de coopération entre spécialistes, où chaque agent "sait" à qui il doit demander des informations et envoyer des résultats. On se trouve donc en face de systèmes très rigides, plus "modulaires" en fait que véritablement multi-agents. Nous ne rencontrerons ces organisations par la suite que pour la constitution d'agents, les systèmes multi-agents étant définis à l'aide d'organisations plus adaptatives.

3.5.3 Organisations à structure variable-égalitaire-émergente

Les organisations à structure organisationnelle variable et à constitution émergente sont caractéristiques des systèmes peuplés d'agents réactifs qui s'auto-organisent. Si les robots explorateurs sont totipotents, cela signifie que chaque agent peut détecter, forer et récupérer du minerai tout seul. Le problème de ce type d'organisation est de faire en sorte que les agents réussissent à coopérer afin que si un agent trouve du minerai, les autres puissent bénéficier de cette découverte. Nous verrons au chapitre 8 comment traiter ce problème.

La redondance du système est égale à 1, et l'on se trouve donc en face d'un système généraliste très redondant. Afin d'améliorer les performances de manière dynamique, les robots peuvent se spécialiser au fur et à mesure qu'ils travaillent, de façon que ceux qui ont déjà transporté du minerai deviennent de plus en plus transporteurs et que ceux qui forent deviennent de plus en plus foreurs. Nous verrons au chapitre 7 un exemple semblable dans le cas d'une simulation d'un nid de fourmis.

3.5.4 Organisations à structure variable-égalitaire-prédéfinie

Cette structure est caractéristique des organisations par appel d'offre qui sont caractérisées par une très grande variabilité des liens d'accointance. Elles fonctionnent à

partir d'une métaphore des marchés économiques: un agent qui a besoin d'accomplir une tâche envoie à d'autres agents un appel d'offre et les agents qui se sentent concernés peuvent répondre en envoyant des propositions au demandeur qui décide en dernier retour avec lequel il désire passer un contrat. C'est ce type d'organisation qui est le plus utilisé dans les SMA cognitifs.

Paramètres de concrétisation

Il est possible de faire varier le paramètre de spécialisation et ainsi d'obtenir des configurations très différentes.

Si le degré de spécialisation est 0, c'est-à-dire si les agents sont totipotents, chaque agent peut demander explicitement de l'aide aux autres si le besoin s'en fait sentir. Par exemple, si un agent détecteur trouve du minerai, il peut donner la position du filon aux autres, de telle manière que le minerai soit récupéré le plus vite possible. Nous verrons au chapitre 7 les problèmes qu'une telle organisation pose et la façon de les résoudre.

Si au contraire les agents sont plus spécialisés et, en particulier, si le degré de spécialisation est voisin de 1, les agents ne peuvent accomplir qu'une seule action. Dans ce cas, si l'on suppose, comme pour les organisations à structure fixe, que les détecteurs jouent un rôle de chef, ces derniers peuvent demander aux robots foreurs de venir récupérer du minerai. Seuls les foreurs libres et proches répondront à cette demande. Ils agiront de même avec les robots transporteurs (fig. 3.13). Cette organisation permet ainsi de traiter avec beaucoup d'efficacité le fait que l'on ne connaisse pas initialement la topologie du terrain et l'emplacement des gisements de minerai. De plus, la disparition d'un robot, et même d'un robot détecteur, ne pénalise pas trop le déroulement des opérations, puisqu'aucun robot ne devient totalement indispensable à l'affaire. Le degré de redondance est alors garant des capacités du système à fonctionner avec des pannes.

3.5.5 Organisations à structure évolutive

Les organisations évolutives ont été encore peu traitées dans le cadre des systèmes multi-agents. Il existe essentiellement deux manières de les aborder.

La première solution consiste à donner à l'organisation des mécanismes prédéfinis afin qu'elle puisse modifier son fonctionnement et sa structure lorsque certains paramètres organisationnels atteignent un niveau critique. C'est cette approche qu'a choisie E. Le Strugeon (Le Strugeon 1995) en proposant qu'une organisation évolutive soit définie par un ensemble de modèles d'organisations fixes ou variables et que le système décide de l'application de l'une d'entre elles en fonction de la valeur de certains paramètres internes.

La seconde a été étudiée essentiellement dans le cadre de la vie artificielle. En incorporant au sein de l'organisation (ou de ses agents) des algorithmes génétiques, on laisse la société évoluer et adopter l'organisation qui semble la plus adaptée en fonction des conditions environnementales. C. Delaye notamment, avec son système ALIA (Delaye 1993), a étudié la manière dont l'introduction d'algorithmes

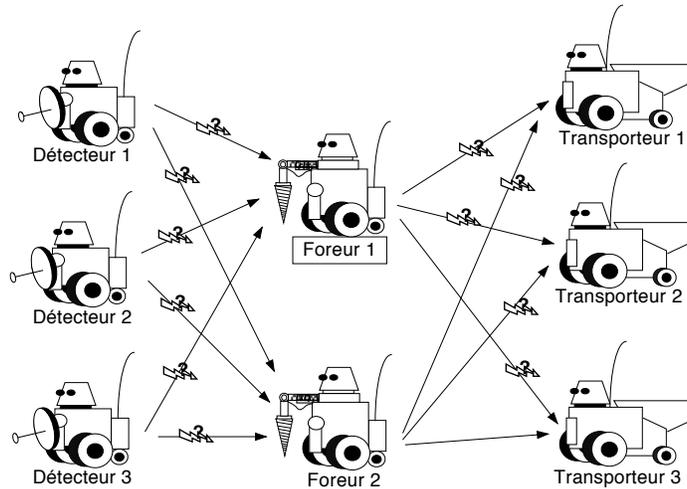


Figure 3.13: Une organisation à structure variable de constitution prédéfinie. Les relations de subordination sont variables et l'allocation des tâches peut s'effectuer en fonction des demandes. Pour plus de clarté, seules les relations de subordination ont été affichées.

génétiques pouvaient affecter la structure et le fonctionnement d'une organisation de robots explorateurs réactifs et totipotents.

3.5.6 Autres travaux sur les organisations

Il existe relativement peu de travaux qui aient abordé le problème de l'organisation avec une démarche systématique comme la nôtre. On peut citer cependant les travaux de Fox (Fox 1981) qui classe les organisations depuis le groupe, composé d'agents indépendants et non coopérants, jusqu'au marché général dans lequel les agents suivent des buts individuels, en fonction de leurs structures de décision. Mintzberg (Mintzberg 1982) a proposé une classification qui s'applique en fait aux organisations humaines, mais qui a eu une certaine influence en IAD (Bouron 1992; Le Strugeon 1995). Il distingue cinq structures organisationnelles caractéristiques: la structure simple (centralisation), la bureaucratie mécaniste (hiérarchie à décentralisation horizontale limitée), la bureaucratie professionnelle (décentralisation verticale et horizontale), la structure divisionnée (décentralisation verticale limitée) et l'adhocratie (fonctionnement par ajustements mutuels entre des groupes de projets).

Gasser (Gasser 1992) propose une classification en quatre catégories: centralisée (qui correspond au mode d'adaptation fixe, avec des liens de subordination de type maître/esclave), de marché (ce qui correspond aux organisations variables prédéfinies avec appel d'offre), de communauté pluraliste (correspond à nos organisations variables émergentes) et de communauté à règle de comportement (organisations variables prédéfinies avec interactions selon des protocoles explicites). Il soutient aussi que ce que j'appelle la kénétique devrait intégrer les résultats de la théorie des organisations afin de constituer un domaine de recherche sur les organisations synthétiques (computational organization research) (Gasser 1995) qui serve à la fois à mieux à

comprendre les organisations humaines, mais aussi à apporter des concepts et des théories fondamentales à la conception d'organisations artificielles.

Enfin, les travaux de Pattison, Corkill et Lesser (Pattison et al. 1987) utilisent une démarche très semblable à celle proposée ici pour l'analyse structurale. Les organisations sont constituées de composants caractérisés par différents paramètres (responsabilités, ressources, connaissances) liés entre eux par des relations. Ils distinguent ainsi les relations de communications, d'autorité (correspondant à nos relations de subordination) et de proximité (nos relations d'accointances). Ils donnent ensuite un langage permettant de définir aisément des organisations.

3.6 Organisations individuelles

Les agents constituent un cas particulier, mais essentiel, d'organisations. Celles-ci sont importantes non seulement parce qu'elles sont le point de départ de la constitution d'organisations collectives, mais aussi parce qu'elles apparaissent comme le résultat d'une organisation de composants de niveau inférieur. De ce fait, l'analyse structurale d'un agent ressemble à celle d'une organisation quelconque, à ceci près que les composants d'un agent ne sont généralement pas des agents eux-mêmes, mais des modules (horizontaux ou verticaux) plus ou moins spécialisés. On retrouve cependant toutes les caractéristiques et tous les paramètres que nous avons examinés précédemment pour les organisations quelconques.

Le terme généralement employé pour décrire l'organisation interne d'un agent est celui d'*architecture*, par analogie avec la structure des ordinateurs. Un agent peut en effet être comparé à un ordinateur, avec ses organes de perception et d'action et son système de délibération qui ressemble aux unités d'entrées-sorties, de mémorisation et de calcul d'un ordinateur. Mais il s'agit à la fois d'un ordinateur moins généralisé (puisque l'on peut simuler le fonctionnement d'un agent et même d'un système multi-agent sur un ordinateur), et plus évolué puisqu'il est capable de suivre ses propres objectifs de manière autonome.

La ressemblance qui existe entre ordinateur et agent ne doit donc pas être poussée trop loin, même si elle s'avère féconde pour celui qui veut comprendre l'organisation d'un agent et être à même de construire un agent à partir d'une architecture donnée, voire d'inventer de nouvelles architectures. Si les architectures d'ordinateurs sont relativement homogènes, du fait de l'hégémonie de l'architecture dite de von Neuman sur la quasi-totalité des ordinateurs réalisés aujourd'hui, il n'en est pas de même pour les architectures multi-agents qui présentent un éventail très riche de solutions pour concevoir l'organisation interne d'un agent afin qu'il exhibe le comportement désiré.

Un agent cognitif dispose naturellement d'une architecture plus élaborée que celle d'un agent réactif. Mais la division cognitif/réactif n'épuise absolument pas le débat sur les architectures. D'abord parce qu'il en existe une vaste gamme aussi bien pour les agents cognitifs et réactifs, mais surtout parce que l'on peut très bien réaliser des agents réactifs à partir d'architectures initialement destinées à des agents cognitifs et vice versa. Mis à part quelques architectures, telles que celles fondées

sur des réseaux de neurones, pour lesquelles on ne peut toujours pas construire des agents présentant un comportement très élaboré, la plupart peuvent s'appliquer à des agents aussi bien cognitifs que réactifs. Néanmoins, il faut tempérer cette affirmation en reconnaissant que certaines architectures ont été développées dans un certain cadre et qu'elles tendent donc à être tout spécialement dédiées à certains types d'agents. Par exemple, nous verrons que les systèmes à base de tableau noir sont plutôt réservés aux agents cognitifs, alors que les architectures de subsomption sont utilisées pour des agents réactifs, même si rien n'empêche théoriquement de les employer l'un pour l'autre.

3.6.1 Tableau des principales architectures

On retrouve dans une architecture d'agent les paramètres généraux valables pour l'analyse structurale des organisations artificielles: le type d'approche, les structures de subordination, le couplage et la constitution.

Type d'architecture	Approche	Type de composant	Structure de subordination	Couplage	Constitution
<i>Modulaire horizontale</i>	fonct. horiz.	Module	hiérarchique	fixe (mais progr.)	prédéfinie
<i>Tableau noir</i>	fonct.	Tâche	hiérarchique (méta)	variable	prédéfinie
<i>Subsomption</i>	fonct. vert.	Tâche primitive	hiérarchique	fixes	prédéfinie
<i>Tâches compétitives</i>	fonct. vert.	Tâche + actions primitives	hiérarchique (compétition)	variable	prédéfinie
<i>Règles de production</i>	fonct.	Règle	hiérarchique (méta)	variable	prédéfinie
<i>Classifieurs</i>	fonct. vert.	Règle	hiérarchique	évolutif	prédéfinie
<i>Connexioniste</i>	fonct. vert.	Neurone formel	égalitaire	fixe (à poids)	prédéfinie
<i>Système dynamique</i>	fonct. vert.	Fonction stimuli-commande	égalitaire	fixe (mais progr.)	émergente
<i>Multi-agents</i>	objet/fonct.	Agent	égalitaire	variable	émergente

Table 3.7: Principales architectures d'agents

Parmi l'ensemble des architectures possibles obtenues en faisant varier ces paramètres, seul un petit nombre a donné lieu à suffisamment d'implémentations pour pouvoir être catégorisé. Il s'agit des *architectures à base de modules horizontaux* (ou

plus simplement *architectures modulaires*), *de tableau noir*, *de subsomption*, *de tâches compétitives*, *de règles de production*, *de classifieurs*, *de système dynamique et de multi-agents* et les *architectures connexionistes*. Elles sont regroupées avec leurs paramètres associés dans le tableau 3.7. On pourra noter que tous les types d'architecture (sauf éventuellement le type "multi-agent") emploient une approche fonctionnelle, qu'elle soit horizontale ou verticale.

3.6.2 Architecture modulaire horizontale

Ce type d'architecture est certainement l'un des plus répandus. Qu'il s'agisse de travaux théoriques ou d'applications pratiques, la plupart des architectures proposées pour la définition d'agents cognitifs sont fondées sur la notion d'ensemble de modules horizontaux liés par des connexions préétablies.

Les architectures modulaires horizontales, que nous appellerons modulaires tout simplement par souci de simplification, sont conçues comme un assemblage de modules, chacun réalisant une fonction horizontale particulière. Les modules les plus courants sont:

- Les fonctions perceptives et motrices s'il y a lieu.
- L'émission et l'interprétation des communications.
- La base de croyances comprenant la modélisation de l'environnement et des autres agents.
- La gestion des engagements.
- Les expertises du domaine de compétence.
- La gestion des buts et la prise de décision.
- La planification des actions, etc.

Dans ce type d'architecture, toutes les liaisons sont fixes, c'est-à-dire que le mode de circulation des informations est prédéfini par le concepteur. La figure 3.14 montre un exemple caractéristique d'une telle architecture, caractérisé par un flux d'information montant puis descendant.

Dans la phase ascendante, les signaux provenant de l'extérieur par l'intermédiaire de capteurs ou de boîtes aux lettres comprenant des messages sont filtrés de manière à obtenir une information de nature de plus en plus abstraite, jusqu'à ce qu'elle puisse s'intégrer aux modélisations de l'agent. La fonction la plus élevée est effectuée par le module de prise de décision qui, à partir des informations qu'il reçoit et des objectifs qui lui sont propres, décide d'agir. On passe ensuite à la phase descendante correspondant à la mise en application des décisions. Le module de planification ordonne les actions à effectuer pour satisfaire l'objectif choisi, celles-ci sont ensuite transmises au module d'exécution.

Un grand nombre d'architectures proposées pour les agents cognitifs, et en particulier celles que nous proposerons au chapitre 5, s'appuient essentiellement sur ce modèle.

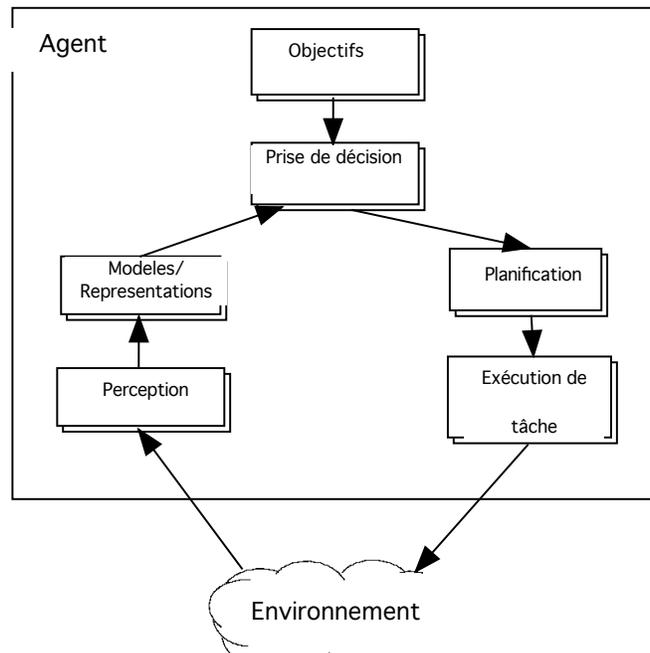


Figure 3.14: Représentation caractéristique d'un agent à architecture modulaire horizontale

3.6.3 Les architectures à base de tableaux noirs

L'architecture de tableau noir est l'une des plus utilisée dans les systèmes multi-agents cognitifs symboliques et elle a donné lieu à une abondante littérature. Originellement développée dans le cadre de l'intelligence artificielle traditionnelle (c'est-à-dire, du point de vue de l'IAD, pour réaliser des systèmes mono-agents), pour la reconnaissance de la parole avec le système Hearsay II (Erman et al. 1980), l'architecture de tableau noir s'est rapidement imposée en IAD comme une architecture suffisamment souple et puissante pour pouvoir implémenter les mécanismes de raisonnement et de calculs intervenant à l'intérieur des agents, notamment avec le système DVMT (Lesser et Corkill 1983).

Le modèle de tableau noir est fondé sur un découpage en modules indépendants qui ne communiquent directement aucune informations, mais qui interagissent indirectement en partageant des informations. Ces modules, appelés *sources de connaissance* ou *KS* (pour *Knowledge Sources*), travaillent sur un espace qui comprend tous les éléments nécessaires à la résolution d'un problème. L'architecture d'un système à base de tableau noir comprend trois sous-systèmes (fig. 3.15):

- *Les sources de connaissance (KS).*
- *La base partagée* (le “tableau” proprement dit) qui comprend les états partiels d'un problème en cours de résolution, les hypothèses et les résultats intermédiaires et d'une manière générale toutes les informations que s'échangent les KS. Ces bases sont décomposées en hiérarchies (conceptuelles, partitives, causales etc.) qui structurent la modélisation du domaine d'application comme

l'espace des hypothèses/solutions.

- *Un dispositif de contrôle* qui gère les conflits d'accès entre les KS, ces derniers intervenant de manière "opportuniste" c'est-à-dire sans être déclenchés effectivement par un système centralisé de contrôle. C'est cette partie qui a connu le plus de modifications au cours de l'évolution de ces architectures.

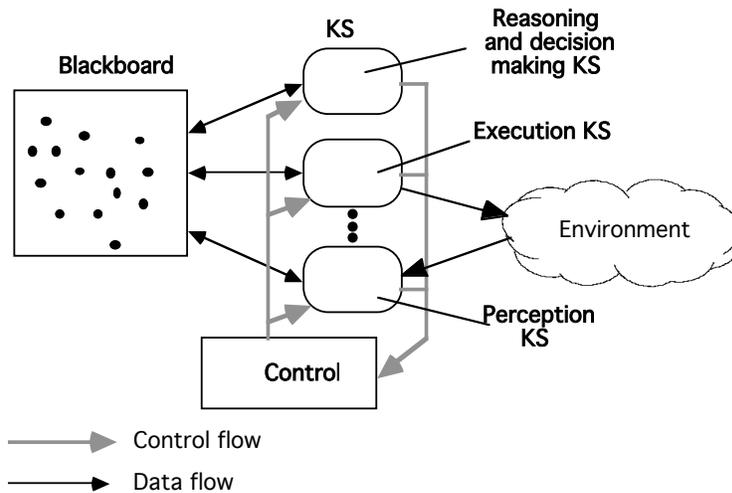


Figure 3.15: Une architecture de système à base de tableau noir

Le contrôle dans les tableaux noirs

Si le modèle de tableau noir est très général, puisqu'il ne dit rien sur la manière dont sont organisés ces sous-systèmes, de nombreuses réalisations ont vu le jour, chacune présentant une vision différente du contrôle, la partie en fait la plus ardue d'un tableau noir. Le problème du contrôle dans un tableau noir revient à essayer de savoir ce qu'il convient de faire ensuite, c'est-à-dire en fait de déterminer quelle source de connaissance doit être déclenchée. Au début, avec les premières implémentations de HEARSAY II, le contrôle était "câblé" au sein d'une procédure, puis très rapidement il fut donné sous la forme d'un ensemble de règles. Mais ce n'est qu'avec le système BB1 de B. Hayes-Roth que l'organisation du contrôle dans les tableaux noirs acquit ses lettres de noblesses, en considérant le contrôle comme un problème suffisamment important pour qu'il dispose de son propre tableau (Hayes-Roth 1985). L'architecture de BB1 comprend donc deux tableaux: le premier est destiné au traitement du problème du domaine et le second à la gestion du contrôle. De ce fait, il est possible de traiter l'activation des sources de connaissance comme s'il s'agissait d'un problème indépendant du domaine d'application.

Pour pallier certains des problèmes d'efficacité des approches à base de tableau de contrôle à la BB1 et considérant que la plupart des systèmes de contrôle étaient en fait hiérarchiques, J.-P. Haton, H. Laasri et B. Maitre du Crin à Nancy ont cherché une architecture présentant un bon compromis entre l'expressivité et l'efficacité. Ils

voulaient pouvoir décrire le contrôle sous forme de KS sans perdre en efficacité par rapport à un contrôle procédural pur. De cette réflexion est né le système ATOME (Lâasri et Maître 1989; Lâasri et al. 1987) qui a été depuis utilisé dans un grand nombre d'applications. Citons notamment un système d'aide à la décision pour la gestion des réfections de la voirie de Nancy (Ferraris et Haton 1992) et un outil de gestion temps réel des contre-mesures pour un pilote d'avion de chasse (Lalanda et al. 1992).

On trouvera une bonne explication de la notion de tableau noir comme système indépendant, de ses mécanismes et de ses différentes versions dans (Engelmore et Morgan 1988) et (Jagannathan et al. 1989).

Tableaux noirs et systèmes multi-agents

Si, dans un premier temps, les systèmes à base de tableaux noirs furent considérés comme des systèmes d'IAD, chaque KS pouvant être perçu comme un agent qui interagit avec les autres KS, il n'en est plus de même aujourd'hui. Du fait de leur mécanisme de contrôle très centralisé et de leur manque de mémoire locale et donc de localité des informations, ces systèmes sont maintenant envisagés comme des architectures pratiques pour la réalisation de systèmes "intelligents" et, en particulier, pour implémenter la structure interne d'agents cognitifs symboliques. Nombre de systèmes multi-agents ont été implémentés de cette manière aux Etats-Unis (Lesser et Corkill 1983; Hayes-Roth et al. 1988) et en Europe (Chevrier 1993; Iffenecker 1992; Cambier 1994).

L'architecture de tableau noir présente de nombreux avantages dont, en tout premier lieu, une remarquable souplesse pour décrire des modules et articuler leur fonctionnement. Son intérêt réside dans ce qu'elle est à la fois opportuniste et centralisée, les liaisons entre les modules (entre les KS donc) étant mutables. Elle est opportuniste au sens où les KS se déclenchent lorsque des configurations du tableau les intéressent, suite à des modifications provoquées par d'autres KS. Elle est centralisée par l'intermédiaire d'un module de contrôle qui ordonne l'ordre dans lequel les KS seront effectivement activés, en essayant de déterminer quelle est la meilleure action à effectuer compte tenu de l'état du système.

Son principal inconvénient provient de sa relative inefficacité, due à la très grande expressivité de son contrôle. De ce fait, ce type d'architecture s'avère particulièrement utile lors de la phase de prototypage de la réalisation de systèmes ou lorsque les temps de réponses ne sont pas trop contraints. Néanmoins, des versions de BB1 ont montré que même dans des cas où il était nécessaire d'avoir des temps de réponses en temps réel, la gestion fine du contrôle pouvait accélérer de manière drastique son comportement en prenant les bonnes décisions et en choisissant les tâches importantes et urgentes au bon moment (Hayes-Roth et Collinot 1993). Le système GUARDIAN, qui surveille un patient dans une unité de soin intensif en est un bon exemple (Hayes-Roth et al. 1992).

En tout état de cause, les avantages de ce type d'architecture sont indéniables: du fait de sa très grande plasticité, il est possible d'implémenter n'importe quelle structure d'agent en termes d'éléments de tableaux et de KS. En particulier, toutes

les autres architectures peuvent être modélisées en termes de tableaux noirs au prix parfois d'une certaine lenteur dans l'exécution. Le tableau noir se présente donc comme une sorte de "méta-architecture", c'est-à-dire une architecture pour implémenter des architectures.

3.6.4 L'architecture de subsomption

L'*architecture de subsomption* a été proposée pour la première fois par R. Brooks pour la constitution d'agents tropiques-réactifs (Brooks et Connell 1986). A l'inverse de l'architecture modulaire hiérarchique qui divise un agent en modules horizontaux, l'architecture de subsomption⁶ décompose un agent en modules verticaux, chacun d'eux n'étant responsable que d'un type de comportement très limité.

Les interactions entre les modules sont fixes et s'effectuent par l'intermédiaire d'un rapport de dominance défini lors de la conception. Les modules effectuent leurs tâches en parallèles, mais si deux modules sont en conflit (c'est-à-dire que deux modules produisent des résultats contradictoires), alors seules les informations fournies par le module dominant seront considérées. Si le module inférieur produit un résultat sans que le module dominant fonctionne, son effet sera pris en compte, mais si ce dernier produit lui aussi des informations, ceux-ci seront prioritaires devant ceux du module inférieur, comme le montre la figure 3.16.

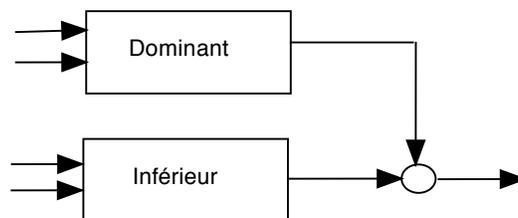


Figure 3.16: Dans une architecture de subsomption, les modules supérieurs sont dominants et inhibent, si nécessaire, la sortie des modules inférieurs.

Il est ainsi possible de construire des systèmes relativement complexes avec une telle architecture. Cependant, c'est encore le concepteur qui définit l'ensemble des modules et surtout les relations de domination qui s'exercent entre eux. La figure 3.17 présente un exemple caractéristique d'architecture de subsomption pour un robot explorateur.

Cette technique a surtout été utilisée pour décrire des agents réactifs, mais il serait possible de l'utiliser pour des agents cognitifs, en considérant que les modules supérieurs sont les plus réflexes et les modules inférieurs sont les plus cognitifs, en induisant une priorité dans les modules.

⁶Subsomption, de subsumer, mettre sous. Ne pas confondre avec son usage dans le domaine de la représentation des connaissances, où l'on parle de relation de subsomption entre concepts.

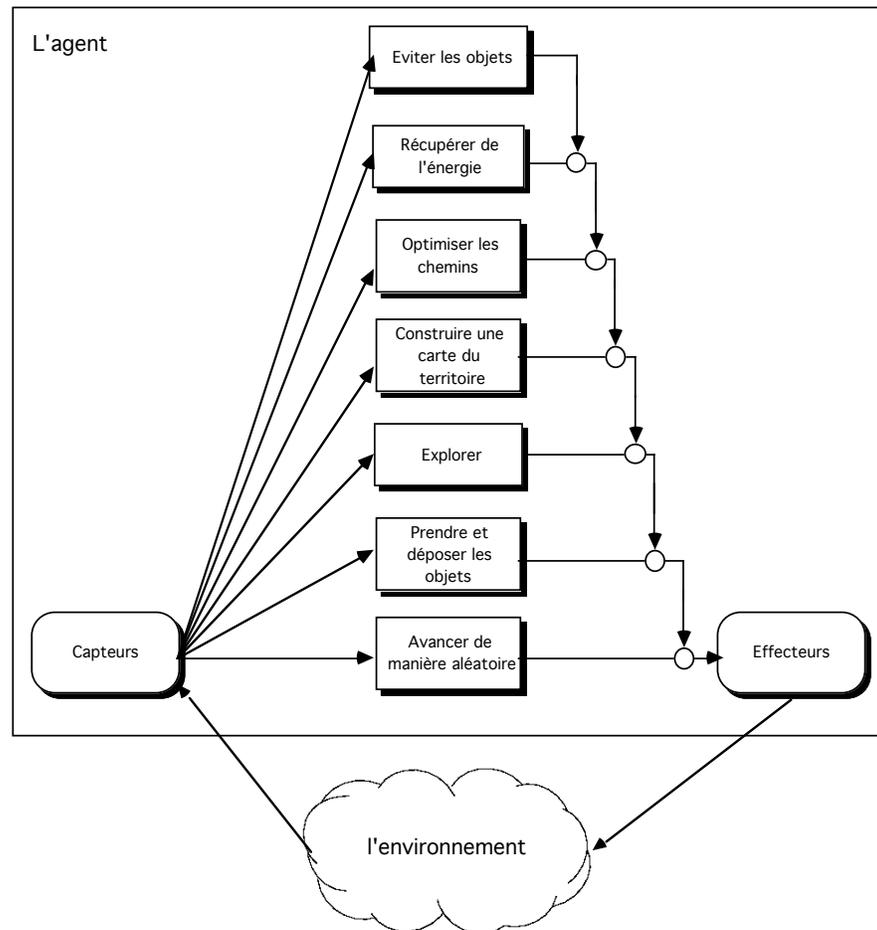


Figure 3.17: Modèle de robot explorateur à base d'architecture de subsumption

3.6.5 Les tâches compétitives

Alors que dans les deux architectures modulaires précédentes, les liens sont fixes, d'autres architectures ont été proposées pour essayer d'introduire une certaine mutabilité dans les liens entre modules afin d'obtenir plus de souplesse dans le choix de la sélection du module. Ces modules représentent des fonctions verticales, ce que certains appellent des "comportements" ou "tâches".

Dans la structure des tâches compétitives, un agent est composé d'un ensemble de tâches parmi lesquelles seule une peut être active à la fois. Ces tâches sont ainsi en compétition pour être élues par un mécanisme de décision qui prend en compte différents paramètres: le poids de la tâche à un moment donné, le contexte d'application, les informations provenant de l'extérieur, etc. Ce type d'architecture est décrit tout particulièrement dans (Drogoul et Ferber 1992). Une tâche correspond à une action macroscopique qui présente une certaine unité. Par exemple, `allerChercherObjet`, `récupérerEnergie`, ou `explorer` peuvent être considérées comme des tâches, alors que `bougerBras` ou `seDéplacer` sont des actions trop élémentaires pour constituer des tâches à elles seules. Ce sont au contraire des *primitives d'action*, c'est-à-dire les mots du langage avec lequel il est possible de

définir des tâches. La figure 3.18 montre le processus d'élection de la tâche courante. Le module de sélection reçoit les valeurs des paramètres des différentes tâches pour chacune d'entre elles et sélectionne celle dont la fonction d'évaluation appliquée aux paramètres est la plus élevée.

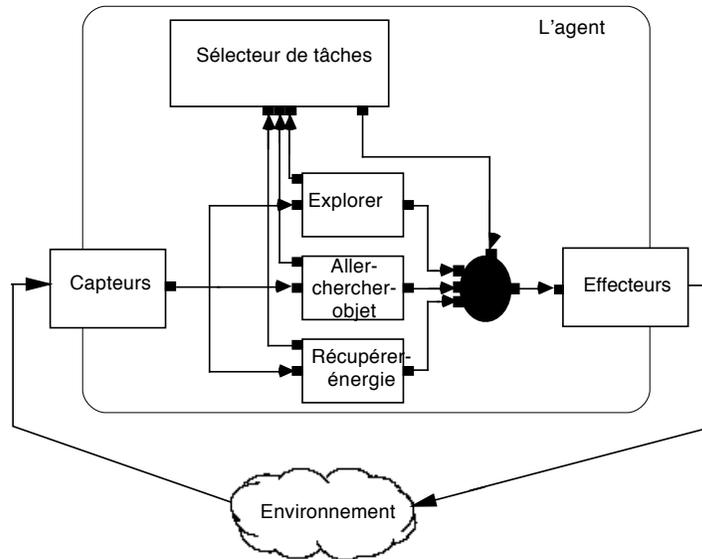


Figure 3.18: Un agent à base de tâches compétitives. Le sélecteur de tâche choisit la tâche qui présente la valeur (issue d'une fonction d'évaluation) la plus élevée.

La tâche élue, une fois sélectionnée, est activée et la tâche courante précédente désactivée. Mais une tâche désactivée n'est pas totalement "éteinte". Elle présente une activité de veille qui consiste à recevoir les informations provenant des capteurs et à calculer les valeurs des paramètres de sélection qui en dépendent afin de demeurer dans la compétition. C'est ce type d'architecture qui est utilisé dans le système MANTA sur la simulation des fourmis (Drogoul et Ferber 1994) (cf. chap. 7).

D'autres architectures fondées plus ou moins sur le même principe ont été proposées. A partir d'un point de vue éthologique (McFarland 1990; McFarland 1994), elles consistent toutes à proposer un mécanisme de sélection d'actions pour agents réactifs qui permette à ces derniers d'organiser leur comportement en fonction des stimuli environnementaux qu'ils reçoivent et des pulsions internes qui les motivent (besoin d'énergie, reproduction,...). On retrouvera cette problématique dans les architectures hybrides ci-dessous et dans la problématique du passage à l'acte réactif (cf. chap. 5).

3.6.6 Les systèmes de production

Les systèmes de production font certainement partie des architectures les plus connues de l'intelligence artificielle. Un système de production est défini par la combinaison d'une base de faits (BF), d'une base de règles de production (BR) et d'un interprète, le moteur d'inférence (MI). Bien qu'il y ait une grande variété

de syntaxes pour la définition de règles de production, les règles sont généralement données sous la forme suivante:

si <liste-conditions> alors <liste-actions>

où <liste-conditions> est associée à des éléments de la base de faits et <liste-actions> comprend des actions élémentaires telles qu'ajouter ou supprimer des éléments de la base de faits.

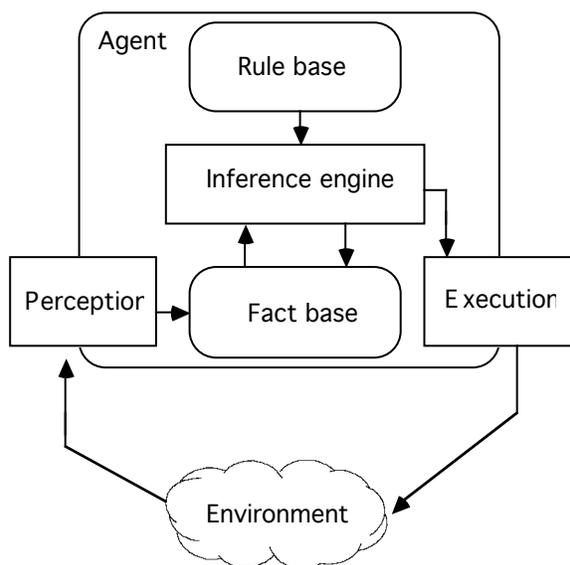


Figure 3.19: Un agent à base de systèmes de production

Certaines actions peuvent aussi directement activer les commandes d'exécution de l'agent. Lorsqu'une règle peut valider chacune des conditions de sa liste, elle exécute les actions correspondantes. Si plusieurs règles peuvent être activées, on dit qu'elles sont en conflit, et le système de contrôle du moteur d'inférence déclenche la règle qu'il considère comme la plus prioritaire (en fonction de paramètres internes à la règle, ou tout simplement en prenant la première).

Dans le cadre d'un système multi-agent, chaque agent est représenté sous la forme d'un système de production, muni des fonctions d'interprétation et d'exécution (fig. 3.19). La fonction de perception se borne à placer les informations perçues ou les messages à l'intérieur de la base de faits pendant le fonctionnement du moteur d'inférence, pour que la base de règles puisse le prendre en compte directement.

Bien qu'ils aient été longtemps considérés comme l'outil idéal pour l'écriture de grandes bases de connaissances, les systèmes de production présentent néanmoins deux inconvénients majeurs:

- La partie action des règles comportant généralement des mécanismes de suppression d'action, on dit que le moteur est *non monotone*. Cependant, dans un moteur non monotone, le résultat obtenu n'est pas indépendant de l'ordre d'application des règles. De ce fait, il est nécessaire de connaître l'ordre d'application des règles pour être sûr que le résultat obtenu soit conforme

au désir de réalisation, ce qui est contraire au principe de “mise en vrac” des règles. Pour tenir compte de ce problème, on regroupe généralement les règles à l’aide de paquets de règles. Mais de toute façon, il est indispensable de connaître les algorithmes du moteur d’inférence, c’est-à-dire les choix et l’ordre d’application des règles, pour savoir utiliser efficacement un système de production.

- Les règles, à la différence des procédures et des fonctions des langages de programmation, ne sont pas combinables. En effet, alors qu’une fonction LISP est définie comme une combinaison de fonctions, les règles implémentent directement la relation entre les conditions et les actions sans qu’il soit possible de définir des abstractions sous la forme de “règles de règles”. La solution classique à ce dilemme consiste comme précédemment à construire des paquets de règles réunissant un ensemble de règles liées fonctionnellement entre elles. Mais cette notion de paquet introduit un mécanisme procédural qui s’éloigne des vertus initiales.

De ce fait, la portée de l’application des systèmes de production est nécessairement réduite et les programmes réalisés de cette manière doivent être soigneusement testés pour vérifier que les systèmes sont cohérents. Cependant, ils offrent l’avantage, lorsque le nombre de règles est petit ou lorsque le découpage de la base de règles en paquets de règles permet de diminuer le nombre de règles à envisager simultanément, de fournir une forme d’écriture simple à la définition de mécanismes de raisonnement complexes. Dans le cadre de systèmes multi-agents, les bases de connaissance associées à un agent sont relativement simples ou en tout cas décomposables en paquets de règles qui représentent alors les modules d’une des architectures modulaires vues précédemment.

3.6.7 Les systèmes à base de classifieurs

Les systèmes à base de “classifieurs”⁷ sont des cas particuliers de systèmes à règles de production originellement introduits par Holland (Holland 1968; Holland 1978; Holland et al. 1986) pour concevoir des systèmes autonomes évolutifs.

Un système à base de classifieurs se présente comme une variation d’un système à règles de production dans lequel les faits (appelés messages) sont des chaînes de caractères (et les caractères sont généralement pris dans un alphabet binaire 0, 1) de taille fixe et les règles (appelées *classifieurs*) sont des couples de chaînes de caractères dont les éléments sont pris dans l’alphabet précédent auquel on a ajouté un caractère “joker” (‡) qui peut s’associer à n’importe quel élément. Les règles comportent un poids correspondant à leur probabilité de déclenchement. La figure 3.20 montre un tel système. Les informations provenant de l’environnement sont codées par le système de perception sous la forme d’un ensemble de faits. Les règles s’appliquent sur ces faits et produisent d’autres faits qui peuvent à leur tour

⁷Il s’agit d’une francisation du néologisme américain “classifier” inventé par J.H. Holland. Je reprends le terme utilisé par E. Sutton dans son rapport sur les algorithmes génétiques (Sutton 1994).

déclencher d'autres classifieurs ou bien produire des actions dans l'environnement via le système d'exécution. Les systèmes à base de classifieurs ne différeraient pas des systèmes de production si deux mécanismes ne venaient profondément modifier leur perspective, à savoir un système d'attribution de crédit et un système de reproduction de règles par algorithmes génétiques.

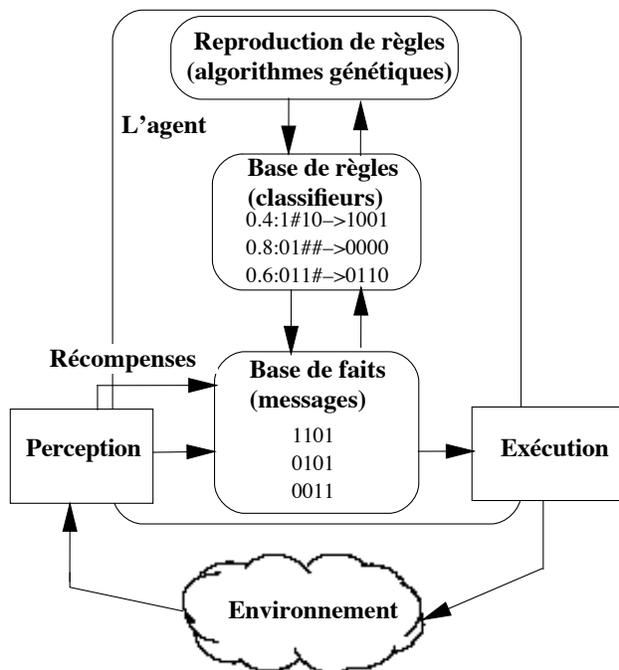


Figure 3.20: Un agent conçu autour d'un système à classifieurs

Le système d'attribution de crédit récompense les règles qui ont donné lieu à une "bonne" action, c'est-à-dire une action considérée comme ayant permis d'aboutir à un but tel qu'obtenir de la nourriture par exemple. Dans ce cas, les poids de ces règles sont augmentés, alors que dans le cas contraire, les règles n'ayant apporté aucun bénéfice pour l'agent voient leur poids décroître. On déclenche prioritairement les règles ayant le poids le plus fort. De plus, le système de reproduction des classifieurs par algorithmes génétiques est utilisé pour produire de nouvelles règles à partir des règles ayant le poids le plus fort par des mécanismes de mutations et de croisements (cross over) (Goldberg 1989). Ces systèmes de classifieurs sont essentiellement utilisés pour créer des comportements adaptatifs. Par exemple, dans (Wilson 1991) des animats situés dans un environnement apprennent à trouver leur nourriture le plus rapidement possible avec un tel système.

3.6.8 Les architectures connexionnistes

Les architectures connexionnistes, qui sont fondées sur la métaphore du cerveau, sont formées d'un réseau d'éléments tous identiques que l'on appelle souvent *neurones formels*, à cause de leur ressemblance (assez vague en fait) avec les neurones du système nerveux des animaux. Chaque neurone détermine sa valeur de sortie en

fonction des valeurs d'entrée qu'il reçoit d'autres neurones. La fonction de transfert du neurone est donnée par l'équation:

$$y_j = f\left(\sum_i w_{ij}x_i\right)$$

où les x_i sont les valeurs de sortie des unités i , les w_{ij} sont les *poids* des connexions reliant les neurones i aux neurones j , et f est la fonction d'activité. Les fonctions d'activité les plus classiques sont les fonctions à seuil ou les sigmoïdes, qui indiquent qu'un neurone est actif (ou d'autant plus actif) à partir du moment où la somme des valeurs qu'il reçoit sur ses entrées dépasse une certaine valeur. Il existe un grand nombre de modèles de réseaux de neurones. Les plus classiques sont les réseaux à couches. Ils supposent que les neurones sont regroupés en ensembles, appelés couches, et que les entrées des neurones de la couche n sont reliées aux sorties de la couche $n - 1$. Dans un réseau à n couches, la première couche est directement reliée aux capteurs et la couche n aux effecteurs. Les couches intermédiaires, appelées couches internes ou cachées, servent à mémoriser des états internes. La figure 3.21 montre un réseau à trois couches.

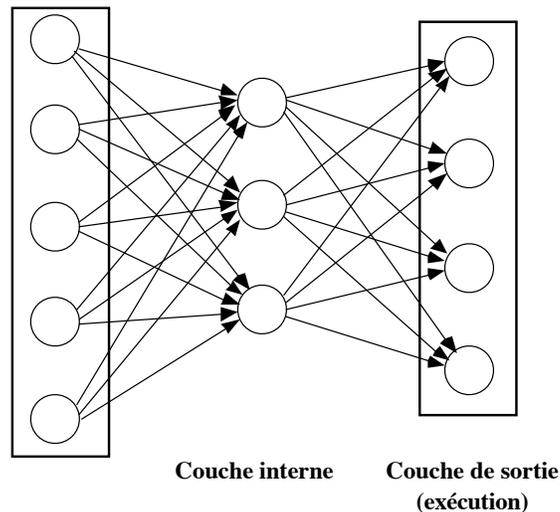


Figure 3.21: Un réseau de neurones à trois couches comprenant une couche interne (ou cachée)

En tant qu'architecture d'agents, on peut définir les poids des connexions entre neurones de trois manières différentes:

1. La première technique consiste tout simplement à définir ces poids à la main ou à les faire évoluer avec des techniques extrêmement simples. Lorsque l'architecture est très simple, on peut parfois s'en contenter. Braitenberg, notamment, décrit des petits robots autonomes situés qui naviguent avec un très petit nombre de neurones dont l'architecture est donnée à la main (Braitenberg 1984). Par exemple, la figure 3.22 montre certainement l'une des architectures les plus simples que l'on puisse réaliser avec des neurones. Avec seulement deux neurones croisés et reliant des capteurs à des effecteurs, on peut construire un petit véhicule qui se dirige vers

une source lumineuse, chaque neurone activant un moteur proportionnellement à la puissance des informations reçues sur le capteur correspondant. Plusieurs auteurs s'en sont inspirés, notamment (Beer et Chiel 1992) et (Cherian et Troxell 1993), pour développer des agents réactifs capables de s'adapter à leur environnement.

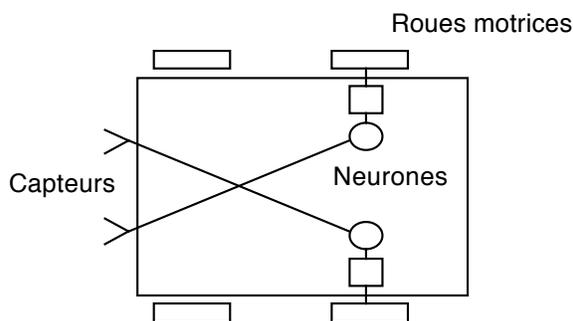


Figure 3.22: L'un des petits véhicules de Braitenberg. Il est capable de se diriger vers une source lumineuse bien que son contrôle se résume à deux neurones.

2. Dans la deuxième technique, le réseau apprend lui-même ces poids à l'aide d'un mécanisme de rétropropagation de l'erreur (backprop). On présente un ensemble de stimuli en entrée et une réponse attendue en sortie. L'écart entre la réponse attendue et celle donnée par le réseau est rétropropagée sur les poids des connexions entre les différentes couches. Si ces techniques sont bien adaptées à des situations d'apprentissage relativement statiques comme la reconnaissance de visages ou d'écriture, elles ne conviennent pas toujours à des agents autonomes mobiles qui doivent apprendre un comportement dynamique. On trouvera néanmoins dans (Lin 1992) une utilisation de la technique de rétropropagation à la définition de comportement d'agents réactifs adaptatifs.

3. La troisième technique, qui prend d'ailleurs de l'essor, consiste à faire évoluer les poids à l'aide d'algorithmes génétiques, ce qui permet d'utiliser directement les réponses environnementales comme mécanismes de renforcement, comme dans les systèmes de classifieurs (Todd et Miller 1991). Mais la tendance est aussi de faire directement évoluer la structure du réseau neuronal par l'utilisation d'algorithmes génétiques, en développant par apprentissage les neurones cachés nécessaires à l'accomplissement de la tâche (Cliff et al. 1993) (Werner et Dyer 1992) (Werner 1994).

3.6.9 Les architectures à base de système dynamique

Dire qu'il existe des architectures dynamiques est un peu présomptueux pour l'instant puisqu'il n'existe qu'une seule version d'architecture de ce type, celle qu'a développé L. Steels pour la réalisation de robots autonomes coopératifs (Steels 1994). Le principe consiste précisément à faire table rase de la notion d'architecture et à simplement implémenter le comportement d'un agent en décrivant directement les équations qui relient les valeurs (standardisées pour des raisons d'efficacité) des capteurs à celles des commandes appliquées aux effecteurs.

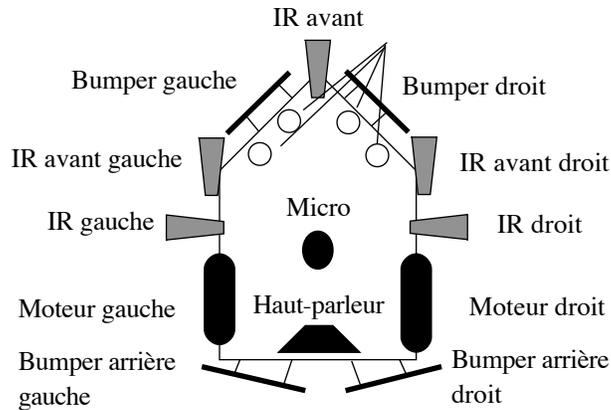


Figure 3.23: Synoptique général d'un robot "dynamique" de L. Steels

Par exemple, L. Steels utilise un robot comportant un ensemble de capteurs et d'effecteurs lui permettant de sentir les chocs (bumper) ou de percevoir des obstacles par infrarouge (IR), d'entendre et d'émettre des bruits, de percevoir des lumières et surtout de bouger en jouant directement sur la vitesse des moteurs droits et gauches du robot (fig. 3.23). Pour exprimer un comportement, il suffit de donner la valeur au temps $t + 1$ du paramètre de commande en fonction de sa valeur au temps t et de la valeur de tous les autres capteurs au temps t . Par exemple, pour dire au robot d'avancer, il suffit de donner les équations du mouvement pour les vitesses des moteurs de droite (VD) et de gauche (VG).

$$VD(t + 1) = VD(t) - \frac{VD - VD_{Default}}{10}$$

et de même pour la commande de la vitesse du moteur de gauche. Ainsi, que le moteur aille trop vite ou trop lentement, il va peu à peu s'adapter à sa vitesse nominale indiquée par le paramètre $VD_{Default}$ (qui vaut 150). Lorsque le capteur de contact (un "bumper" dans la terminologie des roboticiens) de gauche touche quelque chose, il faut que les moteurs s'arrêtent. Avec une architecture comme celle de subsumption, on ferait en sorte que le comportement `éviterObstacle` inhibe le comportement `allerToutDroit`. Ici au contraire, on modifie directement les valeurs de commande des moteurs en ajoutant ou retranchant des valeurs numériques de telle manière que les moteurs aient tendance à aller en sens contraire. Par exemple, le fait d'aller à droite si on touche à gauche s'exprime à l'aide du comportement suivant:

```
Comportement allerADroiteSiToucheAGauche
  Si le bumper de gauche est touché
  alors
    VG(t+1) = VG(t) - 300 * valeur(bumper de gauche)
    VD(t+1) = VD(t) - 400
```

Il suffit ensuite d'additionner directement toutes les valeurs des moteurs pour obtenir le comportement désiré. Dans ce cas, si le robot touche le bumper de gauche

il ira en marche arrière (puisqu’il y a des valeurs de retrait très supérieures aux valeurs nominales pour aller en avant) avec un petit mouvement de rotation lui amenant le nez vers la droite. Aux temps suivants, si le robot ne touche plus l’obstacle, il va retrouver progressivement sa vitesse nominale en marche avant à l’aide des équations de mouvement avant. De ce fait, toutes les commandes partielles (`allerToutDroit`, `éviterObstacle`, etc.) sont toujours sommées, et le comportement global résulte directement de la somme de tous les comportements partiels. L. Steels a appliqué cette architecture à la réalisation d’un couple de (véritables) robots coopérants qui déambulent de manière naturelle pour aller recharger leurs batteries et accomplir une fonction “utile” pour le concepteur.

Evidemment, cette approche est très liée au matériel utilisé puisqu’il n’existe pas de représentations intermédiaires à partir desquelles il serait possible d’abstraire un comportement indépendant. Néanmoins, et bien que ce type d’architecture en soit encore à ses premiers débuts, on peut gager qu’il recueillera rapidement de nombreux adeptes, du fait de sa simplicité et de sa capacité à pouvoir effectuer de manière naturelle plusieurs actions en même temps (cf. chap. 5).

3.6.10 Architectures multi-agents et acteurs

On appelle architecture multi-agent d’un agent l’application de la notion de système multi-agent à la définition de l’architecture des agents eux-mêmes, un agent étant alors considéré comme un système multi-agent à part entière. Le premier à avoir considéré le psychisme d’un être (humain ou artificiel) comme le résultat d’interactions entre petits agents individuels est M. Minsky⁸ dans son célèbre livre *La Société de l’esprit* (Minsky 1988). Il y décrit, de manière assez décousue, un ensemble de mécanismes résultant de conflits et de coopérations entre petites entités calculatoires qu’il appelle agents. Chacun est responsable d’une activité, d’un souvenir, d’une propriété reconnue d’un objet, sans qu’il y ait de véritable système centralisateur coordonnant l’ensemble. De ce fait, Minsky considère que le fonctionnement de l’appareil psychique n’est pas le résultat d’un ensemble d’inférences portant sur des symboles, mais plutôt le fruit de confrontations auto-organisatrices entre processus autonomes. Bien que très peu de systèmes aient été conçus à partir des travaux de Minsky (et on peut d’ailleurs se demander si elles sont réellement implémentables dans leur totalité), ses idées ont certainement influencé tout un courant de pensée, et en particulier les travaux de P. Maes dont on parle à la section suivante, qui considère que le comportement d’un agent résulte d’un ensemble d’activités internes que l’on peut associer au fonctionnement d’un système multi-agent.

En particulier tous les travaux portant sur les langages d’acteurs et sur leur utilisation dans des systèmes multi-agents peuvent dans une certaine mesure se rapporter à ce courant. Initialement, la notion d’acteur est apparue avec les travaux de Hewitt dès 1973 (Hewitt et al. 1973). Ils ont véritablement été rendus publics vers 1976 avec l’article initiateur *Viewing Control Structures as Patterns of Passing*

⁸En fait, G. Bateson (Bateson 1979) avait déjà avancé cette idée en disant qu’un esprit est un ensemble de parties ou de composantes en interaction, mais il n’en avait pas dégagé les conséquences opératoires nécessaires à une réalisation informatique.

Messages (Hewitt 1977), qui développait une certaine conception des programmes conçus comme des sociétés de spécialistes résolvant un problème en communiquant par envois de messages. A cet effet, Hewitt montrait que son modèle d'exécution permettait de considérer les structures de contrôle des langages traditionnels comme des schémas (patterns) de communications entre entités autonomes appelées *acteurs*. Depuis, les langages d'acteurs, en particulier sous l'influence de Agha (Agha 1986) de Tokoro (Ishikawa et Tokoro 1986) et de Yonezawa (Yonezawa et Tokoro 1987), ont surtout été étudiés comme des modèles d'exécution pour la programmation par objets concurrents. Mais quelques travaux ont voulu rester dans les idées initiales que prônaient Hewitt et qu'il confirma avec ses notions de "sémantique des systèmes ouverts" (Hewitt 1991; Hewitt 1985). P. Carle (Carle 1992), S. Giroux (Giroux et Senteni 1992) et J. Ferber (Ferber 1987), tout en estimant que les langages d'acteurs sont effectivement de très bons outils pour l'implémentation de calculs parallèles, considèrent néanmoins qu'ils présentent des caractéristiques tellement originales qu'ils modifient par leur présence la notion même d'architecture multi-agent en envisageant les agents et les systèmes multi-agents comme des extensions naturelles de la notion d'acteur.

Mais qu'est-ce qu'un acteur? Un acteur est une entité informatique qui se compose de deux parties: une structure qui comprend l'adresse de tous les acteurs qu'il connaît⁹ et à qui il peut envoyer des messages et une partie active, le *script*, qui décrit son comportement lors de la réception d'un message. Le comportement de chaque acteur, qui s'exécute indépendamment et en parallèle avec les autres, se résume à un ensemble d'actions extrêmement réduit: envoyer des messages, créer des acteurs et modifier son état (ou déterminer un nouveau comportement pour le message suivant). C'est tout! Et c'est suffisant pour pouvoir exprimer n'importe quel calcul parallèle comme une combinaison de ces actions primitives.

La communication entre acteurs s'effectue par envois de messages asynchrones, c'est-à-dire que l'émetteur n'a pas besoin que le récepteur soit prêt pour recevoir un message. Mais l'originalité la plus importante des langages d'acteurs est certainement l'usage de la continuation locale du calcul. Lorsqu'un acteur A a besoin d'une réponse à un message qu'il envoie à un acteur B, il n'attend pas la réponse: il passe en argument un nouvel acteur C, appelé "customer" chez Hewitt, qui se charge de traiter la réponse et la suite du calcul. Cet acteur C agit donc comme la continuation locale du calcul initié par l'envoi de message de A vers B. L'usage des continuations locales donnent une très grande fluidité aux calculs exprimés en langages d'acteurs puisque les acteurs n'ont aucunement besoin d'attendre les réponses. De nombreux langages d'acteurs ont été proposés. Les plus célèbres et actuellement opérationnels sont ABCL (Yonezawa 1990), MERING IV (Ferber et Carle 1991) et ACTALK (Briot 1989), ce dernier étant conçu comme une extension de SMALLTALK (Goldberg et Robson 1980).

⁹En termes de langages d'acteurs, on appelle ces acteurs les *accointances* d'un acteur. Il ne faut cependant pas confondre cette notion d'accointance, qui se résume à une simple adresse, avec la gestion des accointances telle qu'on la rencontre dans les systèmes multi-agents, laquelle suppose qu'il existe une représentation explicite des caractéristiques des autres agents (cf. le chapitre 6 sur la représentation des connaissances d'un agent).

En étendant la notion d'acteur vers celle d'agent grâce à l'utilisation de la réflexivité organisationnelle, S. Giroux (Giroux 1993) propose de considérer un agent comme un écosystème, c'est-à-dire comme un ensemble d'agents en interactions, et un écosystème comme un agent. Son système REACTALK, fondé sur une extension d'ACTALK, fournit aux agents la possibilité d'adapter leur comportement en fonction du type de message qu'ils reçoivent et ainsi de pouvoir dialoguer aussi en mode synchrone si cela s'avère nécessaire. Néanmoins, ces agents ne disposent pas véritablement de comportements autonomes (ils n'ont pas de systèmes motivationnels) ni de représentations de leur environnement, l'idée d'écosystème étant plutôt prise comme une métaphore. Néanmoins ces travaux montrent qu'il existe un lien de continuité entre la notion d'acteur et celle d'agents. J'avais d'ailleurs, dans (Ferber 1989), proposé un système, PALADIN, écrit comme une extension du langage MERLING IV dans lequel les acteurs pouvaient poursuivre un but et, si nécessaire, raisonner sur leurs propres compétences et expliquer leur comportement à d'autres entités du système. Les acteurs avaient véritablement pris le statut d'agent, la réflexivité étant ici encore le lien nécessaire au passage d'un niveau "acteur" dans lequel les entités n'effectuent que du calcul, à un niveau "agent" où les entités disposent d'une plus grande latitude comportementale, voire de capacités de raisonnement ou de motivations (Ferber et Carle 1991).

Enfin, il faut signaler, d'une part, que L. Gasser dans son système MACE (Gasser et al. 1987) a été fortement inspiré par les idées de Hewitt concernant les acteurs et, d'autre part, que le langage ACTALK est à la base d'un grand nombre de plateformes multi-agents, la première en date ayant été MAGES III (Bouron et al. 1990). Le rapport entre acteurs et agents est ainsi des plus féconds et nul doute que l'avenir verra d'autres travaux tendant à montrer les liens très étroits qui unissent ces deux concepts.

Chapter 4

Action et comportement

4.1 La modélisation

4.1.1 Des modèles...

Un modèle, en science, est une image stylisée et abstraite d'une portion de réalité. L'activité scientifique consiste principalement à faire des modèles des phénomènes et des objets qu'elle étudie. Ces modèles peuvent être très formels ou simplement fournir une représentation simple et pratique. Par exemple, la physique utilise des modèles très mathématiques pour prédire le mouvement des corps ou déterminer le rayonnement d'une source électromagnétique. En revanche la biologie utilise des modèles de transfert d'information ou des images simples comme celle de la clé et de la serrure pour comprendre le fonctionnement de la cellule, la reproduction et l'évolution d'un organisme.

L'intérêt d'un modèle est d'abord d'être plus explicite, plus simple et plus facile à manipuler que la réalité qu'il est censé représenter. Les modèles éliminent ainsi un grand nombre de détails considérés comme inutiles par le modélisateur afin de mieux se consacrer aux données que celui-ci juge pertinentes relativement au problème qu'il désire résoudre. Par exemple, il est difficile de prédire comment un corps sous la pression de forces différentes va se déplacer et se déformer. Mais la mécanique utilise des modèles simples de forces qui, une fois conjugués, permettent de prévoir précisément le mouvement de ce corps et les déformations que ce dernier subira. C'est parce que l'on a pu traduire le corps dans un modèle dynamique et parce que l'on sait comment utiliser ce type de modèle pour prévoir l'avenir que la construction et l'utilisation de ces modèles s'avèrent particulièrement utile.

Les modèles sont ainsi des images homomorphes de la réalité, c'est-à-dire qu'il existe un homomorphisme entre l'objet d'étude et le modèle qui permet d'appliquer les résultats du modèle à l'objet lui-même, comme le montre la figure 4.1.

Le phénomène observé est traduit sous la forme d'une abstraction, laquelle peut être manipulée (par transformations algébriques, par calculs numériques ou par simples inférences logiques) pour obtenir des résultats qui peuvent alors servir à mieux comprendre ou à prédire des situations futures. C'est par des opérations de cette nature que l'on a pu calculer les orbites des planètes autour du Soleil ou mieux

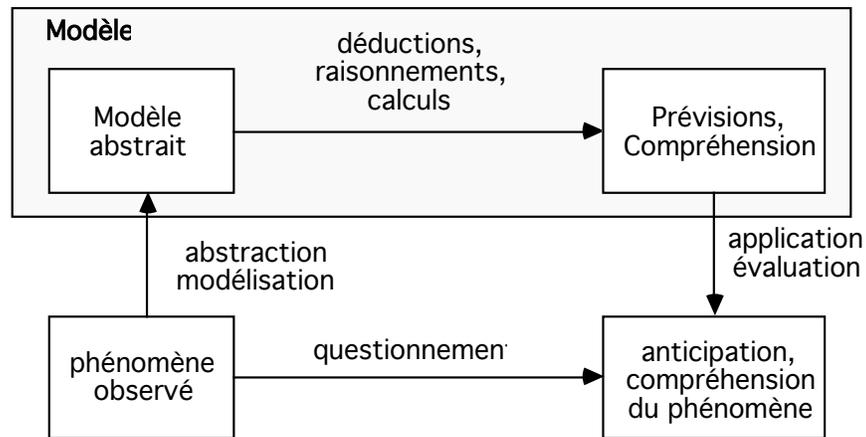


Figure 4.1: La prévision et la compréhension de phénomènes passe par l'élaboration de modèles.

comprendre le comportement des oies sauvages.

4.1.2 ... et de leur utilité pour les SMA

On peut se demander pourquoi les systèmes multi-agents auraient besoin de modèles? Ne se suffisent-ils pas à eux-mêmes? Le code des programmes permettant aux agents de s'exécuter n'est-il pas suffisant à la compréhension de leur fonctionnement?

Malheureusement tout n'est pas si simple. Pour un système multi-agents, comme d'ailleurs pour n'importe quel système informatique un tant soit peu complexe, analyser le comportement d'un programme à partir de son implémentation présente les inconvénients suivants:

1. Il est particulièrement difficile d'entrer dans le cœur d'un programme et d'analyser son comportement simplement en regardant son implémentation.
2. Le code d'un programme introduit un grand nombre de détails inutiles à la compréhension de son fonctionnement.
3. Des implémentations différentes, en termes de langages et de choix de structures de données, présentent des manifestations semblables, alors que des modifications légères dans un programme peuvent avoir des conséquences importantes sur le comportement de ce dernier
4. L'exécution des agents s'effectuant en parallèle (que ce parallélisme soit simulé sur une machine monoprocesseur ou effectif sur une machine multiprocesseur) il est encore plus difficile de comprendre leur fonctionnement à partir de leur code, du fait du non-déterminisme inhérent au parallélisme.
5. Enfin, et surtout, les systèmes multi-agents sont des logiciels complexes, difficiles à appréhender et à concevoir. Il est donc nécessaire de réduire leur complexité en dégageant leur structure et en analysant séparément leurs différents composants sans pour autant perdre de vue l'organisation générale.

4.1.3 Que modéliser?

Quels sont les caractères essentiels d'un SMA? Quels sont les paramètres qui les différencient les uns des autres? Que faut-il décrire précisément? Qu'est-ce qui se passe effectivement et comment cela se passe-t-il? En d'autres termes, les questions fondamentales portent sur l'intelligibilité et la conception des mécanismes mis en œuvre et sur les conséquences visibles et attendues de leur fonctionnement, c'est-à-dire entre le point de vue de l'ingénieur attaché à concevoir les rouages d'un système et celui de l'observateur qui analyse de l'extérieur le comportement qu'il manifeste. C'est à ces questions que j'essayerai de répondre dans le reste du livre. On peut distinguer quatre notions principales pour lesquelles la modélisation s'avère particulièrement pertinente:

1. La modélisation des actions des agents et des conséquences de celles-ci dans l'environnement. Ce dernier peut être complexe et présenter aussi une évolution autonome, distincte des conséquences des actions des agents. Cette modélisation met en jeu des langages de type L3 (modélisation opératoire) et L5 (formalisation) (cf. chap. 1).
2. Le fonctionnement d'un agent, aussi bien dans le comportement observable que dans la mécanique interne de l'agent, en utilisant des langages de type L3 (modélisation opératoire) et éventuellement L4 (représentation des connaissances).
3. L'interaction des agents entre eux et en particulier les différents modes de communication, préludes aux formes organisées d'interactions que sont la coopération, la coordination d'action et la résolution de conflits, avec des langages de type L2 (communications) et L3 (modélisation opératoire).
4. L'évolution du système multi-agent, ce dernier étant considéré dans son ensemble, à partir d'expérimentations et de formalisations (langages de type L5).

Il existe un grand nombre de modèles applicables à la compréhension et à la conception de systèmes multi-agents. On peut distinguer deux grandes familles: les modèles algébriques, qui tendent à décrire un agent en termes mathématiques (langages de type L5), et les modèles opératoires (langages de type L3) qui utilisent des structures a priori plus informatiques, même si certains, tels les automates à états finis ou les réseaux de Petri, sont eux-mêmes formalisables en termes algébriques. Les premiers sont essentiels car ils déterminent tous les développements ultérieurs en définissant les fondements à partir desquels ceux-ci seront rendus possibles, qu'il s'agisse de modèles opératoires, d'architectures, des états mentaux des agents cognitifs ou de la définition des interactions entre agents.

4.1.4 Agents et actions: des notions faussement élémentaires

La question ne cesse d'être répétée: Qu'est-ce qu'un agent? Qu'est-ce qu'un agent pour son concepteur et pour celui qui l'observe? Comment décrire son comportement,

son architecture, ses caractéristiques internes et ce qui fait qu'il se comporte comme il le fait? Il y a de nombreuses réponses et nous ne chercherons dans ce chapitre qu'à en donner une première esquisse: un agent sera simplement considéré comme une entité qui, en permanence, perçoit, délibère et exécute. Il s'agit donc là d'une version très appauvrie de la notion d'agent mais elle sera suffisante pour donner une modélisation formelle des agents. Nous aurons l'occasion, au cours des chapitres suivants, de l'approfondir.

A côté de l'agent, une autre question soulève encore plus d'interrogations, celle de l'action. Comment décrire le fait que le monde évolue, qu'il se transforme sous la poussée de nos gestes et sous la pression de cette myriade de changements que les astres, la matière, les plantes, les animaux et les êtres humains provoquent à chaque moment et dont ils subissent les conséquences? Or l'action, si elle se trouve à la base de notre existence, sert aussi de fondement aux systèmes multi-agents. En effet, par rapport aux conceptions classiques de l'intelligence artificielle qui postulent que la manifestation de l'intelligence repose sur un raisonnement logique, les systèmes multi-agents prennent comme point de départ non point le raisonnement mais le comportement des agents, les actions qu'ils accomplissent dans le monde et les interactions qui en découlent. Toute analyse des systèmes multi-agents passe donc nécessairement par une étude en profondeur de la notion d'action et par la définition d'une théorie de l'action qui permette de traiter les interactions et leurs conséquences sur la société.

Mais qu'est-ce qu'une action? Bien qu'il s'agisse d'une notion qui paraisse évidente pour chacun de nous, une définition précise de ce concept pose de nombreux problèmes et dépend des conceptions générales, quasi philosophiques, que nous pouvons avoir sur l'agir et ses conséquences. Comme le signale E. Morin (Morin 1977): *Aussi loin que nous puissions concevoir les profondeurs de la physis, nous trouvons agitation et interactions particulières. Immobilité, fixité et repos sont des apparences locales et provisoires, à l'échelle de nos durées et perceptions.*

Tout bouge, le monde est instable et sujet aux changements qui sont le fruit des actions et donc des interactions, réactions et rétroactions des agents les uns avec les autres. Les atomes, les molécules chimiques, les cellules, les organismes multicellulaires, les sociétés animales et humaines et les écosystèmes sont les fruits d'un grouillement d'êtres en activité permanente, les résultats de mouvements incessants et d'agitations turbulentes qui s'organisent à partir des actions élémentaires d'entités autonomes. Mais comment décrire ce flux d'animation, comment représenter de manière rigoureuse l'activité d'un agent et les interactions entre agents?

L'action est d'abord modification. Initialement, on a tendance à considérer le monde comme immobile, les actions des agents venant modifier cet état stable pour qu'il prenne ensuite un autre état stable. L'action est alors ce qui modifie l'état global du monde. Même s'il s'agit d'une des théories les plus en vigueur actuellement au sein de l'intelligence artificielle, nous verrons qu'elle est mal adaptée à la prise en compte des interactions dans lesquelles plusieurs agents peuvent accomplir des actions en même temps et donc se trouver en situation de conflit. C'est pourquoi d'autres théories tentent de s'affranchir du problème de l'action en l'éliminant, soit en ayant recours à des modèles mécaniques du mouvement, soit en ne considérant que

l'enchaînement des événements sans prendre en compte l'état de l'environnement.

Un autre problème concerne la réaction du milieu aux actions des agents et le fait qu'une même action puisse avoir des résultats différents. Car l'action est aussi un geste, une tentative d'influer sur le monde pour le modifier selon nos désirs. Et la conséquence de ce geste ne suit pas toujours l'intention qui le cause. Le but, source de l'intention, porte sur un résultat attendu; le geste est un moyen de l'accomplir. Mais entre le geste et le résultat peuvent s'insérer des impondérables qui font que le résultat n'est pas ce que nous escomptions. Car l'univers est soumis à des lois, auxquelles nous ne pouvons pas nous soustraire. Nous proposons, mais l'univers dispose. Il combine l'ensemble de nos gestes, de nos tentatives et produit une réponse qui est la conséquence de cette combinaison en fonction des lois de l'univers dans lequel ces gestes ont été accomplis.

Mais sans geste, le monde ne reste pas dans un état stable comme semble le croire l'impression première. Il évolue, se transforme sans que l'on ait besoin de supposer un quelconque geste extérieur ou, comme le prétendait Aristote, un premier moteur qui viendrait le mettre en mouvement. De ce fait, il faut distinguer le geste et les conséquences du geste, l'acte et la réponse de l'univers à nos actes. C'est pourquoi nous proposerons une théorie personnelle qui modélise l'action comme un système de réponses à la production d'influences produites par les agents. L'action n'est plus alors envisagée comme le résultat de ce que font les agents, mais comme le résultat des réactions de l'univers aux influences des agents.

4.1.5 Modéliser l'action

Il n'existe pas qu'une seule voie pour modéliser les actions et leurs conséquences. Plusieurs formalismes, plus ou moins mathématiques sont disponibles pour rendre compte de l'action, de l'activité des agents et de l'évolution d'un SMA. Voici quelques-uns des formalismes examinés par la suite:

1. L'action comme *transformation d'un état global*. Le modèle fonctionnel par transformation d'états est certainement le plus classique, et nous l'examinerons en premier. Toute la théorie de la planification en IA a été développée dans ce cadre et la plupart des théories sur la planification multi-agent s'y réfèrent. Il est donc indispensable de bien le connaître. De plus, il sert de base à presque tous les autres modèles de l'action qui à la fois s'y rapportent et s'en distinguent.
2. L'action comme *réponse à des influences*. Mais le modèle par transformation d'états souffre de nombreuses limitations. Pour les résoudre, je proposerai un modèle fondé sur la notion d'*influence* qui se présente comme une extension du modèle précédent.
3. L'action comme *processus informatique*. L'action peut être aussi vue comme un ensemble d'événements produits et consommés par des processus informatiques. On utilise alors des outils théoriques tels que les automates à états finis ou les réseaux de Petri pour formaliser les comportements des agents et celui de l'environnement.

4. L'action comme *modification locale*. Dans les modèles locaux, l'action est considérée comme une modification locale qui se propage le long d'un réseau d'automates. S'ils posent de nombreux problèmes quant à la description des agents, ces modèles, en particulier avec les automates cellulaires, permettent de représenter facilement des environnements.
5. L'action comme *déplacement physique*. Dans les modèles géométriques et physique, les actions sont considérées comme des déplacements dans un espace géométrique. Ce modèle est évidemment très pratique lorsqu'il s'agit de décrire les actions d'agents mobiles.
6. L'action comme *commande*. Dans le modèle cybernétique, l'action est une commande. On ne s'intéresse plus aux transformations apportées par l'agent dans l'environnement mais seulement à ce qu'il perçoit de cet environnement et aux commandes qu'il envoie à ses organes effecteurs de manière à satisfaire un but ou une pulsion interne. Ce modèle est surtout utilisé pour décrire le comportement d'agents réactifs.

Enfin, je présenterai un modèle personnel qui tente de synthétiser les modèles précédents de manière à simplifier la description et l'intelligibilité des systèmes multi-agents. Il décrit les agents, leur comportement et l'environnement comme un ensemble de composants interconnectés. Ce modèle, qui utilise le formalisme BRIC, sera utilisé tout au long de cet ouvrage pour décrire les comportements des agents et leurs interactions.

4.2 L'action comme transformation d'un état global

4.2.1 Une représentation fonctionnelle de l'action

Les conceptions classiques de l'action en intelligence artificielle reposent sur une approche par *état et transformation d'états*. On suppose qu'il est possible de caractériser l'ensemble (généralement discret) Σ des états possibles du monde. Une action op est alors définie comme une transition d'état, c'est-à-dire comme un opérateur dont l'exécution produit un nouvel état. Ainsi, à partir d'une situation σ_1 de Σ , l'exécution de l'action op produit un nouvel état σ_2 de Σ . Par exemple, si à l'état σ_1 , l'agent A se trouve au lieu L_1 et qu'on applique l'action $op = aller(A, L_1, L_2)$, la situation σ_2 qui en résulte verra l'agent déplacé au lieu L_2 (à condition bien sûr que l'opérateur soit applicable, c'est-à-dire qu'il n'y ait pas d'obstacle à l'accomplissement de l'action):

$$\sigma_2 = Exec(aller(A, L_1, L_2), \sigma_1)$$

où $Exec$ est l'opérateur d'exécution des opérateurs d'actions, c'est-à-dire une fonction de type $Op \times \Sigma \rightarrow \Sigma^1$.

¹Un exemple analogue est donné par la fonction `Apply` en LISP.

On peut faire un parallèle entre cette approche et celle de la pellicule d'un film. Chaque image représente un état du monde et les actions décrivent le passage d'une image à une autre. C'est alors le déroulement du film qui donne l'illusion d'une continuité du mouvement.

4.2.2 Les opérateurs de type STRIPS

Mais comment décrire les situations, c'est-à-dire les états du monde et les transformations? En 1971, pour répondre à des problèmes de planification, Fikes et Nilsson (Fikes et Nilsson 1971) proposèrent une représentation, connue encore sous le terme de "représentation à la STRIPS" en l'honneur du premier planificateur à l'avoir utilisée. Dans cette représentation, qui est encore la plus employée dans les problèmes de planification en IA et en IAD, un état du monde est décrit comme un ensemble de formules atomiques dont la conjonction est supposée affirmer la validité de cet état. Supposons par exemple que l'on veuille représenter un monde dans lequel un robot, Clotaire, est capable de se déplacer de pièce en pièce pour aller chercher des outils ou d'autres objets du même genre, comme le montre la figure 4.2. On suppose que Clotaire est capable de se diriger dans l'une des quatre directions (nord, sud, est, ouest) et de prendre ou de déposer des objets. On suppose aussi qu'une pièce est un lieu atomique, un point dans un espace, et que les positions de Clotaire et des outils sont relatives aux pièces représentées par leur numéro. Un état du monde

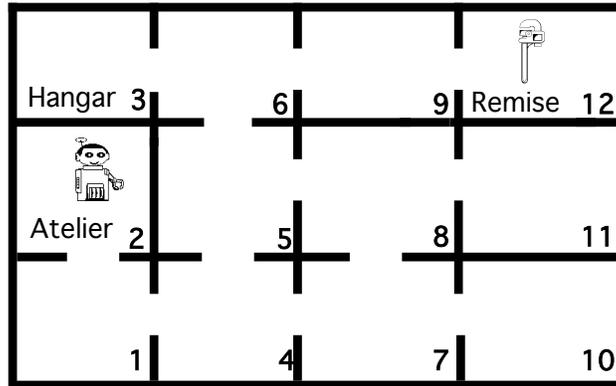


Figure 4.2: Clotaire est un agent qui peut transporter des objets d'un lieu à un autre.

est décrit par l'ensemble des formules atomiques qui sont vraies dans cet état². Par

²Plus précisément, il faudrait définir un état non pas comme un ensemble de formules mais comme une structure algébrique pour laquelle ces formules sont satisfaites. Il suffit pour cela de définir chaque état comme une structure algébrique $\sigma = \langle D, R, F, C \rangle$ où D est un ensemble d'éléments appelés domaine de la structure et qui représente les ensembles des entités dont on parle, R et F sont respectivement l'ensemble des relations et des fonctions définies sur cette structure, et C est un ensemble d'éléments distingués de D , les constantes. On dit alors qu'une formule de la forme $p(t_1, \dots, t_n)$ est satisfaite dans l'état σ , ce que l'on note $\sigma \models p(t_1, \dots, t_n)$ si et seulement si $\langle I(t_1), \dots, I(t_n) \rangle \in R$. Il en est de même pour les constantes où $\sigma \models c_i$ si et seulement si $I(c_i) \in C$ et où I est une fonction d'interprétation des formules. Pour gérer un ensemble Σ d'états, il suffit de définir Σ comme un ensemble de structures algébriques $\sigma_i = \langle D_i, R_i, F_i, C_i \rangle$ telles que

exemple, pour dire que Clotaire se trouve dans la pièce numéro 2 à l'état **s1** il suffit de préciser que la formule:

`positionRobot(Clotaire,2)`

appartient à **s1**, et il en est de même pour la définition de la position des outils ainsi que de toutes les relations géographiques que l'on peut exprimer entre ces pièces lorsqu'elles communiquent:

`s1 = { positionRobot(Clotaire,2), positionOutil(Pince,12),
nord(1,2), sud(2,1), est(1,4),... }`

Dans cette représentation, les opérateurs caractérisant les actions sont décrits sous la forme de triplets:

`op = <pré, suppr, ajouts>`

où `pré`, `suppr` et `ajouts` sont des ensembles de formules atomiques. `pré` décrit la précondition de l'opérateur, c'est-à-dire la condition qui doit être vérifiée pour que l'on puisse appliquer l'opérateur. `suppr` et `ajouts` sont deux ensembles de propositions qui décrivent respectivement les faits qu'il est nécessaire de retirer ou d'ajouter à l'état courant du monde pour passer dans l'état suivant. `suppr` et `ajouts` sont donc ce qui caractérise la transformation pour passer d'un état au suivant et cette transformation se résume à la suppression et à l'ajout de nouvelles propositions à l'ensemble des propositions décrivant l'état précédent. Par exemple, un opérateur permettant à Clotaire d'aller vers le sud serait formulé ainsi:

```
Opérateur allerVersSud(x)
  pré:    positionRobot(x,l1), sud(l1,l2)
  suppr:  lieu(x,l1)
  ajouts: lieu(x,l2)
fin
```

Et l'application de cet opérateur à la situation initiale **s1** produit un nouvel état **s2** dans lequel Clotaire s'est déplacé dans la pièce 2 (ou plus exactement dans lequel la formule `positionRobot(Clotaire,2)` ne s'y trouve pas, celle-ci étant remplacée par la formule `positionLieu(Clotaire,1)`):

`s2 = Exec(allerVersSud(Clotaire),s1)`

et donc

$\forall \sigma_i, \sigma_j \in \Sigma, D_i = D_j$ et $C_i = C_j$ (il n'y a que les fonctions et les relations qui peuvent évoluer entre deux états, les domaines et les constantes restent inchangées). On interprète alors les opérations *op* comme des descriptions de fonctions de Σ dans Σ , qui associent à un état σ de Σ un état $op(\sigma)$ de Σ . Cette construction des états est utilisée par E. Pednault (Pednault 1986) pour son langage ADL et on pourra aussi consulter (Jacopin 1993) pour une présentation de quelques sémantiques associées à des notions d'action voisines de STRIPS. Par la suite, nous en resterons à la définition simplifiée dans laquelle on appelle état un ensemble de formules. Pour plus de précisions sur la description logique des états, on pourra lire le livre (Genesereth et Nilsson 1987), qui présente simplement le problème de la définition d'actions par un appareil logique.

```
s2 = { positionRobot(Clotaire,1),
       positionOutil(pince,12),
       nord(1,2), sud(2,1), est(1,4),... }
```

Les opérateurs ne décrivent pas entièrement les états initiaux et finaux d'une transition, mais seulement ce qui se transforme, et on suppose que tout ce qui ne bouge pas reste inchangé d'un état à l'autre.

4.2.3 Planifier avec des opérateurs de type STRIPS

Malgré sa pauvreté, cette modélisation de l'action est bien adaptée aux problèmes de planification tels que l'IA se les ait posés depuis ses débuts. En effet un problème de planification classique repose sur la donnée d'une situation initiale σ_{init} et d'une situation finale σ_{fin} à laquelle on désire aboutir, toutes deux appartenant à Σ , et d'un ensemble d'opérateurs Op définis comme ci-dessus par des fonctions de Σ dans Σ . Résoudre un problème dans un tel contexte, c'est trouver la suite d'opérateurs $X_1, \dots, X_i, \dots, X_n$ où les X_i sont des variables qui prennent leurs valeurs dans Op , telles que:

$$\sigma_{fin} = Exec(X_n \circ \dots \circ X_i \circ \dots \circ X_1, \sigma_{init})$$

où le signe 'o' représente la composition fonctionnelle. Dans ce cadre, un problème de planification peut être décrit formellement sous la forme du quadruplet:

$$PP = \langle \Sigma, \sigma_{init}, \sigma_{fin}, Op \rangle$$

La solution d'un problème de planification est donnée par la production d'un ensemble de plans P , chaque plan $p \in P$ étant représenté sous la forme d'une suite X_1, \dots, X_n de variables prenant leurs valeurs dans Op telles que:

1. l'action associée à X_1 est applicable à l'état initial σ_{init} , ou autrement dit, σ_{init} se trouve dans le domaine de l'opérateur représenté par X_1 ;
2. $\forall i \ 1 < i < n$, l'opérateur représenté par X_i est applicable à tous les états résultants de l'application de X_{i-1} , c'est-à-dire que l'état résultant de l'application de X_{i-1} appartient au domaine de l'action représentée par X_i ;
3. l'application de X_n produit un état final dans lequel se trouve σ_{fin} , ce qui signifie que σ_{fin} se trouve dans l'espace d'arrivée (ou codomaine) de l'action représentée par X_n .

La figure 4.3 représente la forme générale d'un problème de planification. A partir de la spécification des situations initiales, des buts et de l'ensemble des actions disponibles, le planificateur trouve un ensemble de plans $\{p_i\}$ parmi lesquels le plan p à exécuter est choisi. Ce plan est alors transmis à l'exécuteur qui se charge de l'accomplir. Dans de nombreux systèmes, le planificateur se contente de fournir un seul et unique plan. Dans ce cas, le module sélectionneur peut être supprimé.

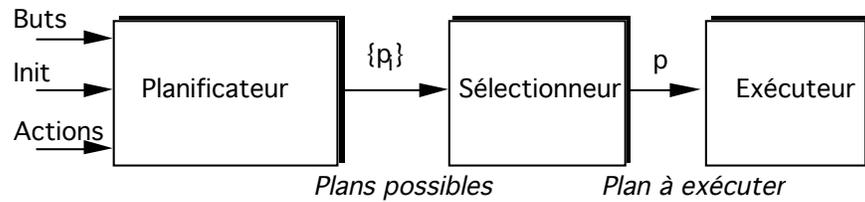


Figure 4.3: Planifier, c'est produire un ensemble de plans parmi lesquels un seul sera exécuté.

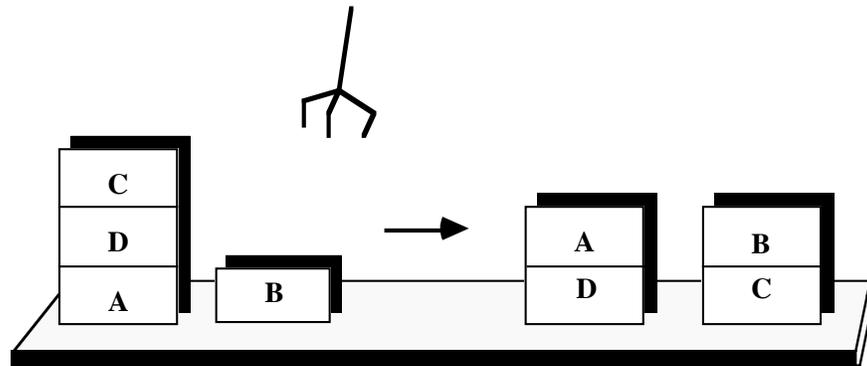


Figure 4.4: Un exemple de manipulation de cubes

L'exemple le plus connu dans le domaine de la planification est celui du monde des cubes. Le problème consiste à empiler des cubes les uns sur les autres de manière à obtenir un empilement précis à partir d'un empilement initial en ne bougeant qu'un seul cube à la fois. La figure 4.4 donne un exemple de situation initiale et finale.

Ces situations sont définies simplement sous la forme d'un ensemble de formules atomiques:

$$\sigma_{init} = \{\text{sur}(C,D), \text{sur}(D,A), \text{sur}(A,\text{Table}), \text{sur}(B,\text{Table})\} \quad \sigma_{fin} = \{\text{sur}(A,D), \text{sur}(D,\text{Table}), \text{sur}(B,C), \text{sur}(C,\text{Table})\}$$

Les opérateurs correspondant aux deux actions classiques d'empilement d'un cube sur l'autre et de dépôt du cube sur la table peuvent s'écrire ainsi:

```
Opérateur poser(x,y)
  pré:   libre(x), libre(y), sur(x,z)
  suppr: libre(y), sur(x,z)
  ajouts: libre(z), sur(x,y)
fin
```

```
Opérateur poserTable(x)
  pré:   libre(x), sur(x,y)
  suppr: sur(x,y)
  ajouts: libre(y), sur(x,Table)
```

fin

L'opérateur `poser(x,y)` indique que pour poser un cube `x` sur un cube `y`, il faut que `x` soit libre, que `y` soit libre également et que `x` soit initialement sur `z`. Après application de l'opérateur `poser(x,y)`, l'état du monde est tel que `x` n'est plus sur `z` mais sur `y`, et tel que `y` n'est plus libre mais `z` l'est devenu. Mais comment fonctionne un planificateur de type STRIPS? Comme il existe de nombreuses variantes et que le propos n'est pas ici de développer l'ensemble des techniques de planification, nous nous contenterons de donner un aperçu suffisant pour comprendre la problématique de la planification multi-agent. Afin d'illustrer cette technique nous reprendrons l'exemple classique de l'empilement des cubes. Supposons que l'on veuille passer d'un empilement initial à un autre empilement, comme le montre la figure 4.4. La technique générale de planification consiste à partir de l'état final et à rechercher les opérateurs qui permettent d'obtenir cet état, en appliquant une technique récursive. Pour un état final donné, on prend tous les opérateurs nécessaires pour obtenir cet état, puis on essaye de savoir s'ils sont applicables et de déterminer les conséquences de leur application. Par exemple, pour obtenir la situation finale

{sur(A,D), sur(D,Table), sur(B,C), sur(C,Table)}

il suffit d'appliquer les opérateurs `poser(A,D)`, `poser(B,C)`, `poser(C,Table)` et `poser(D,Table)` à la situation précédente:

```
Opérateur: poser(A,D)
  pré:    libre(A),libre(D), sur(A,z)
  suppr:  libre(D), sur(A,z)
  ajouts: sur(A,D),libre(z)
fin
```

```
Opérateur: poser(B,C)
  pré:    libre(B),libre(C), sur(B,z)
  suppr:  libre(C), sur(B,z)
  ajouts: sur(B,C),libre(z)
fin
```

```
Opérateur: poserTable(C)
  pré:    libre(C), sur(C,y)
  suppr:  sur(C,y)
  ajouts: libre(y),sur(C,Table)
fin
```

```
Opérateur: poserTable(D)
  pré:    libre(D), sur(D,y)
  suppr:  sur(D,y)
  ajouts: libre(y),sur(D,Table)
fin
```

Mais ces opérateurs sont-ils effectivement applicables? Sachant que la situation initiale est la suivante

$$\{\text{sur}(C,D), \text{sur}(A,Table), \text{sur}(B,Table), \text{sur}(D,A), \\ \text{libre}(C), \text{libre}(B)\}$$

on peut appliquer soit l'opérateur $\text{poser}(B,C)$ soit l'opérateur $\text{poserTable}(C)$. En appliquant le premier on obtient l'état

$$\{\text{sur}(B,C), \text{sur}(C,D), \text{sur}(A,Table), \text{libre}(Table), \\ \text{sur}(D,A), \text{libre}(B)\}$$

qui empêche l'application de l'opérateur $\text{poser}(C,Table)$ puisque C n'est plus libre et qu'il n'est plus possible d'appliquer l'opérateur $\text{poserTable}(C)$. Donc il est préférable d'appliquer le deuxième opérateur et d'obtenir l'état:

$$\{\text{sur}(C,Table), \text{sur}(A,Table), \text{sur}(B,Table), \text{sur}(D,A), \\ \text{libre}(C), \text{libre}(B), \text{libre}(D)\}$$

On peut donc appliquer maintenant soit l'opérateur $\text{poserTable}(D)$, soit l'opérateur $\text{poser}(B,C)$. En effet ces deux opérateurs sont totalement indépendants puisque l'application de l'un n'empêche pas l'application de l'autre et que le résultat final est indépendant de leur ordre d'application. L'application de ces deux opérateurs dans un ordre quelconque produit l'état:

$$\{\text{sur}(B,C) \text{ sur}(C,Table), \text{sur}(D,Table), \text{sur}(A,Table), \\ \text{libre}(B), \text{libre}(A), \text{libre}(D), \text{libre}(Table)\}$$

Il ne reste alors plus qu'à appliquer l'opérateur $\text{poser}(A,D)$ pour achever la construction du plan (fig. 4.5).

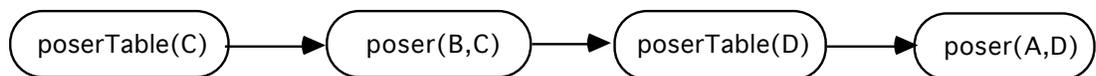


Figure 4.5: Un plan linéaire pour la résolution du problème des cubes de la figure 4.4

4.2.4 Quelques catégories de plans

Il est possible de classer les plans en plusieurs catégories en fonction de la structure des plans et de la manière dont les buts interagissent.

On parle de *planification linéaire* si les buts peuvent être satisfaits indépendamment les uns des autres, c'est-à-dire s'ils sont tels que l'application des opérateurs n'induit aucun conflit dans la satisfaction de ces buts. Par exemple, le problème des cubes de la figure 4.6 peut être résolu par un planificateur linéaire car tous les buts de la situation finale

$$\{\text{sur}(B,A), \text{sur}(D,C), \text{sur}(A,Table), \text{sur}(C,Table)\}$$

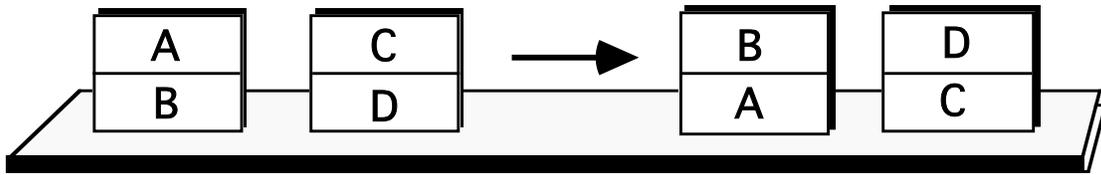


Figure 4.6: Un problème pouvant être résolu à l'aide d'un planificateur linéaire

peuvent être résolus séparément.

En effet pour que B soit sur A, il faut déposer A sur la table et placer B sur A. D'autre part pour que D soit sur C, il faut poser C sur la table et placer D sur C. Ces deux lignes de plan sont totalement compatibles, ce qui permet de traiter chacun des problèmes indépendamment, les actions servant à résoudre un but n'interagissant pas avec celles qui satisfont l'autre but. Evidemment, la plupart des situations courantes ne sont pas linéaires, car les buts ne sont pas indépendants. Dans ce cas, on parle de planification non linéaire. Par exemple, le problème de la figure 4.7, apparemment plus simple puisqu'il ne fait intervenir que 3 cubes, ne peut pas être résolu par un planificateur linéaire. En effet, la situation finale est donnée par

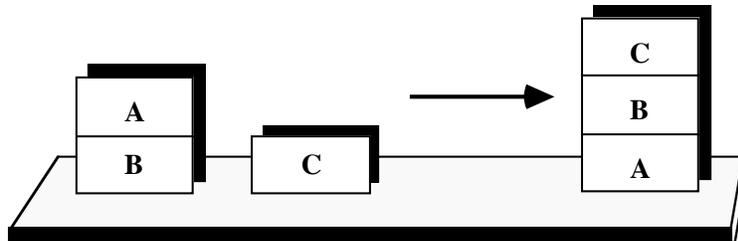


Figure 4.7: Un problème de planification non linéaire

$$\{\text{sur}(C,D), \text{sur}(B,A), \text{sur}(A, \text{Table})\}$$

Dès que A est sur la table, il existe deux actions possibles qui correspondent au but final: soit placer C sur B, soit placer B sur A. Si l'on choisit la première possibilité, on se trouve dans une impasse, car il n'est plus possible de placer B sur A ensuite. De ce fait, l'ordre d'application des opérateurs influe sur le résultat final, et la construction du plan doit tenir compte de cet ordre, d'où le terme de non-linéarité qui est associé à ce type de plan et surtout aux planificateurs capables de traiter cette situation.

Il existe une autre sorte de distinction qui présente un intérêt certain dans le cas de systèmes multi-agents et porte sur la différence entre *plans totaux* et *plans partiels*. Les premiers sont constitués d'une suite ordonnée des opérateurs à appliquer. Le plan de l'exemple de la figure 4.4 est un plan total car il est composé d'une séquence d'opérateurs disposés dans un *ordre total*. Mais, comme nous l'avons vu, il existe plusieurs plans équivalents qui permettent d'obtenir le même résultat et qui ne diffèrent que par l'ordre d'exécution. Dans ce cas, il n'est pas nécessaire d'ordonner totalement les opérateurs, et il est préférable de les conserver sous la forme d'un *ordre partiel*, modélisé sous la forme d'un graphe acyclique comme le montre la

figure 4.8. On obtient alors des branches parallèles d'exécution dont nous verrons que les systèmes multi-agents peuvent aisément tirer profit.

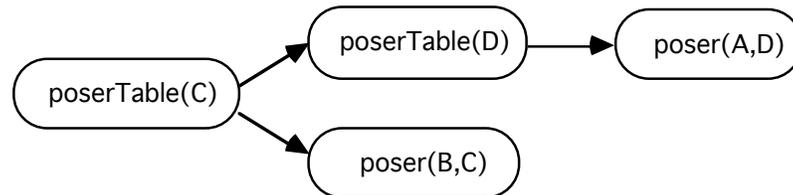


Figure 4.8: Un plan partiel correspondant à l'exemple de la figure 4.6

Au moment de l'exécution du plan, on décidera de la manière de parcourir ce graphe, de manière à *linéariser* le plan. Un autre intérêt des plans partiels est d'être considérés comme plus simples à produire même (et surtout) s'il s'agit de plans non linéaires. Le principe consiste à retarder au maximum les choix portant sur l'ordre dans lequel des actions doivent être appliquées. Dans ce cas, planifier s'effectue en deux étapes. Dans un premier temps on construit un plan partiel sans tenir compte des conflits éventuels, puis, dans un second temps, on affine ce plan de manière à déterminer les conflits éventuels et à linéariser le plan. Nous reverrons au chapitre 8 la problématique de la planification, mais cette fois-ci appliquée à plusieurs agents. Les problèmes qui se posent sont nettement plus complexes et encore peu de modèles généraux permettent de les résoudre.

4.2.5 Limites des planificateurs de type STRIPS

La planification par opérateurs de type STRIPS suppose que les opérateurs décrivent le passage d'un état à un autre en explicitant uniquement ce qui est modifié lors de ce passage. On dit alors que ce type de planification suit les hypothèses de STRIPS (Waldinger 1977):

1. On ne peut appliquer qu'un seul opérateur à un moment donné, lequel précise la manière dont on passe d'un état du monde à un autre. On ne peut donc pas effectuer plusieurs actions en parallèle.
2. Toutes les conséquences des modifications apportées par les actions sont explicitement précisées dans la partie ajout et suppression des opérateurs.
3. Un opérateur représente une action instantanée. Comme nous l'avons précisé plus haut, l'action n'est pas représentée dans son déroulement (elle ne prend aucun temps) mais uniquement par le résultat des transformations qu'elle apporte.

Evidemment, ces hypothèses contraignent sérieusement les possibilités de représentation des actions. La deuxième hypothèse notamment est particulièrement restrictive. Elle précise que toutes les conséquences des actions doivent être directement codées dans les opérateurs. Parfois cela pose de sérieux problèmes. Par exemple,

supposons que l'on veuille représenter un monde "cher aux écoliers" de robinets et de baignoires qui se remplissent et qui fuient. L'opération d'ouvrir un robinet peut s'écrire ainsi:

```
Opérateur ouvrirRobinet
  pré:   robinet(r), fermé(r)
  suppr: fermé(r)
  ajouts: coule(r,Eau), ouvert(r)
fin
```

Supposons maintenant que nous ayons un monde constitué d'un robinet R1 initialement fermé et d'un verre V1 vide qui se trouve juste au-dessous de R1. L'état initial du monde est alors:

$$S1 = \{\text{verre}(V1), \text{vide}(V1), \text{robinet}(R1), \text{fermé}(R1), \text{sous}(V1,R1)\}$$

L'application de l'opérateur ouvrirRobinet, produit le nouvel état S1:

$$S2 = \{\text{verre}(V1), \text{vide}(V1), \text{robinet}(R1), \text{ouvert}(R1), \\ \text{coule}(R1,Eau), \text{sous}(V1,R1)\}$$

dans lequel on peut constater que le verre reste vide bien que l'eau coule effectivement! En effet, pour déduire qu'un verre se remplit s'il se trouve sous un robinet duquel de l'eau coule, il faudrait un opérateur supplémentaire de la forme:

```
Opérateur remplirVerreSousRobinet
  pré :   robinet(r), verre(v), sous(v,r),
          coule(r,Eau), vide(v)
  suppr: vide(v)
  ajouts: rempli(v,Eau)
fin
```

et une information supplémentaire qui nous indique que l'application de cet opérateur est entraînée par l'application de l'opérateur précédent, afin de disposer d'une représentation qui sache gérer les conséquences des actions.

Une telle extension est en dehors du cadre des planificateurs à la STRIPS et peu de planificateurs généraux savent s'accommoder de la causalité, bien que de nombreuses recherches aient été accomplies dans le problème de la représentation des liens causaux (Shoham 1988) qui nécessitent une description plus complexe de l'action.

4.2.6 Limites des représentations classiques de l'action

Au-delà des problèmes de planification, cette représentation d'action par transformation d'états montrent de nombreuses faiblesses dès que l'on s'écarte de l'univers initial pour lequel elle a été définie. Elles se situent à deux niveaux:

1. Au niveau du langage de description de l'action lorsqu'on se borne à des opérateurs de type STRIPS.

2. Au niveau de la conception générale de l'action considérée comme une transformation d'un état dans un autre.

La première limite a été souvent reconnue comme terriblement contraignante car elle ne permet de décrire que des actions très élémentaires. Pour juger de la limite de cette notation, il faut seulement remarquer qu'il n'est pas possible de définir dans ce langage une action telle que: *poser sur la table tous les cubes qui sont eux-mêmes sur un cube*. De ce fait d'autres langages ont été proposés pour pallier cet inconvénient, en améliorant les capacités de description des opérateurs tout en restant capable de trouver des plans d'action (pour avoir une synthèse de ces langages de description, de leurs possibilités, et des problèmes qu'ils soulèvent, voir (Jacopin 1993)).

La seconde limite est plus fine car beaucoup moins étudiée. Elle repose sur trois postulats qui posent problème dans le cas de systèmes multi-agents.

1. **Postulat de "staticité"**. le modèle STRIPS considère que le monde est a priori statique et que les seules évolutions possibles sont le fruit des actions des agents. Pour modéliser un comportement spécifique du monde (par exemple des changements atmosphériques, l'évolution du niveau d'un fleuve), il faut donc expressément introduire des agents extérieurs, tels que l'atmosphère ou le fleuve. Mais ce qui est plus grave, c'est l'incapacité de ce système de prendre en compte la première loi de Newton: *Tout corps persévère en son état de repos ou de mouvement rectiligne uniforme, sauf si des forces "imprimées" le contraignent d'en changer*. En effet, pour un mobile animé d'un mouvement rectiligne uniforme, le passage d'une position à une autre ne nécessite pas l'intervention d'un agent car ses différences positions successives sont les simples résultantes du mouvement. Or les opérateurs de type STRIPS ne décrivent pas les mouvement des objets du monde, mais seulement les transformations obtenues par application d'actions. De ce fait, les représentations par transformation d'états sont pré-newtonniennes dans leur expression et toute description de mouvement devra utiliser une autre technique.

2. **Postulat de séquentialité**. Un autre inconvénient provient de la difficulté à traduire les actions simultanées. Par exemple, supposons que deux agents A et B aient à pousser un cube comme le montre la figure 4.9.

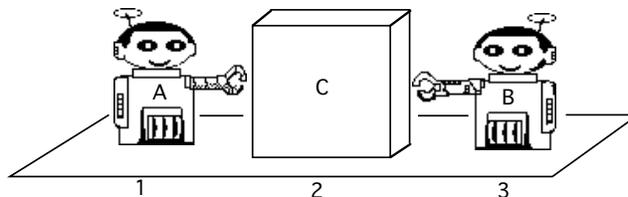


Figure 4.9: Deux agents tentent de pousser un cube en sens contraire.

Si l'on applique ce modèle de l'action et que l'on dispose de l'opérateur *déplacer*(x, y, d) où d est défini comme un déplacement dans la direction opposée où se trouve l'agent x , quel sera le résultat de l'application de ces opérations aux deux agents A et B et au cube C? Les modèles fondés sur l'approche STRIPS supposent que l'on appliquera soit *déplacer*(A,C,+1), soit *déplacer*(B,C,-1) ou l'inverse. Dans le premier cas,

le cube ira d'abord en 3 puis reviendra en 2, et dans le second il ira d'abord en 1 avant de s'en retourner en 2, écrasant peut-être au passage l'un des agents, mais il ne restera jamais immobile sous la poussée contraire des deux agents.

3. Postulat d'universalité. Les actions ne sont pas décrites dans leur déroulement mais par leurs résultats. On ne distingue pas les actions elles-mêmes, c'est-à-dire ce que font les agents, des conséquences de ces actions, ce qui nécessite de décrire dans les actions toutes les catastrophes éventuelles qui peuvent survenir. En étant dépendantes de l'environnement et des lois de l'univers, les descriptions d'action doivent prendre en compte tous les cas de figure possibles, même ceux qui résultent d'une erreur ou de situations a priori inattendues. Ainsi, l'action **Avancer** d'un robot doit contenir le fait qu'il lui est possible de tomber dans un trou, d'être bloqué contre un mur ou d'être ensablé. Par exemple, si une personne peut, en se déplaçant, écraser une fourmi, il faudra spécifier explicitement tout ce qui peut éventuellement être écrasé par le déplacement d'une personne.

Le postulat de séquentialité associé à celui d'universalité pose un autre problème lorsque deux agents agissent simultanément: Comment décrire les conséquences d'une collision éventuelle entre deux corps mobiles (fig. 4.10)? Si l'opérateur **Avancer**(a, dx, dy) représente le déplacement d'un corps a d'un quantième d'espace représenté par son vecteur vitesse $\langle dx, dy \rangle$, comment décrire la rencontre entre deux mobiles et ses répercussions sur la direction et la vitesse des mobiles?

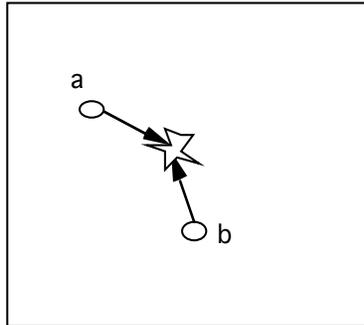


Figure 4.10: Deux mobiles a et b se rencontrent et provoquent une collision.

Il existe évidemment quelques solutions, l'informaticien restant rarement démuné: en ajoutant des arguments supplémentaires pour synchroniser les actions (Georgeff 1983) ou réifier le temps, on peut résoudre certains de ces problèmes au prix d'une écriture plus complexe, et donc moins naturelle, des actions. Mais cela n'est pas toujours suffisant et il faut parfois améliorer la puissance d'expression du langage. C'est ce que fait E. Pednault avec ADL (Pednault 1986) dont il étend la puissance de description de manière à pouvoir appliquer des quantificateurs universels et existentiels sur les variables, ainsi qu'à représenter explicitement des modifications des valeurs fonctionnelles (champ **update** en ADL).

En passant aussi le temps comme paramètre, il peut ainsi caractériser des actions qui se passent au même temps t , même si, à l'exécution, les actions sont appliquées séquentiellement. De plus, il offre la possibilité de décrire des mécanismes ressemblant à des affectations de variables globales dans un langage de programmation, ce

qui lui donne la capacité de représenter presque tout ce que l'on peut coder dans un langage informatique. Pour résoudre le problème de collision, on peut coder dans le corps des actions le fait que s'il existe un autre objet mobile au même endroit, alors il y aura un choc (et éventuellement une explosion), c'est-à-dire en décrivant explicitement dans les opérateurs de mouvement les tests permettant de vérifier s'il y a choc ou non. Mais il faut pour cela décrire explicitement dans les opérateurs tout ce qui concerne les chocs et leurs conséquences, ce qui devient rapidement très lourd, et nous ramène au problème posé par le postulat d'universalité. En outre, il faut faire très attention aux problèmes de synchronisation d'actions, sous peine de "rater" des collisions, une difficulté bien connue des programmeurs de jeux vidéo.

Mais dans tous ces cas, il nous semble qu'il s'agit plus de "trucs" de représentation qui consistent à coder dans les objets de l'environnement la synchronisation éventuelle des actions, que d'une véritable prise en compte de la simultanéité des actions. Dans le cadre des systèmes multi-agents, le problème est patent: alors que nous supposons en permanence que les actions des différents agents sont effectuées en parallèles, nous ne disposons pas de formalisme adéquat pour décrire facilement des actions simultanées et donc représenter des actions collectives. C'est à la définition d'un tel formalisme que la section suivante est consacrée.

4.3 L'action comme réponse à des influences

4.3.1 Présentation générale

Dans cette section, nous proposons un modèle de l'action fondée sur des principes d'influences et de réactions aux influences. Ce modèle reprend certains des éléments des approches fondées sur la transformation d'états en l'étendant de manière à prendre en compte les conséquences des actions simultanées des agents et donc de modéliser les phénomènes dus à des interactions entre agents. De plus, ce modèle permet de décrire aussi bien les actions considérées comme des déplacements dans un espace physique (et donc d'intégrer les formalismes de la mécanique classique) que celles qui relèvent d'autres espaces (tels que l'action d'écrire un programme par exemple). De plus, comme nous le verrons, il ne s'agit que d'une extension du modèle de l'action par transformation d'états, et nous verrons que l'on retrouve ce modèle classique comme un cas particulier, lorsque les modifications du monde ne proviennent que d'un agent unique. Il s'agit donc d'une théorie générale de l'action permettant d'en représenter ses différents aspects tout en bénéficiant des travaux effectués dans chacun de ces domaines tels que celui du mouvement physique, des propagations locales par automates cellulaires ou même de la planification classique, qui n'en sont que des cas particuliers.

Le problème des théories précédentes viennent de ce qu'elles confondent dans l'action ce qui est produit par les agents de ce qui se produit effectivement. Elles mélangent le geste et le résultat du geste. Il suffit donc de séparer ces deux notions pour que la plupart des problèmes rencontrés précédemment disparaissent. Un agent, qu'il s'agisse d'un robot qui pousse une porte ou d'une personne qui change de place, exécute effectivement un geste. Mais le résultat de ce geste dépend des autres gestes

réalisés par les autres agents, comme des capacités de réaction de l'environnement. Les gestes sont ce que nous appelons des *influences*, c'est-à-dire des moyens de modifier le cours des choses s'il n'y avait pas eu ces influences, et les résultats de ces gestes sont obtenus en appliquant les *lois de l'univers*. Ces dernières sont donc des réponses de l'environnement qui agit en composant l'ensemble des influences qu'il reçoit des agents, puis, en fonction de ses propres particularités, il en déduit un nouvel état global du monde.

L'avantage de ce modèle qui est proposé ici est de pouvoir servir de cadre global pour intégrer les travaux développés dans différents domaines: par exemple, dans le domaine de la mécanique classique, l'état du monde est donné par la position et la vitesse d'un ensemble de corps. Les influences sont alors les forces qui s'exercent sur les corps à un instant t et l'application des lois de l'univers consistent simplement à faire le bilan des forces en présence et à en déduire, en fonction des positions et vitesses, la position et la vitesse des corps à l'instant $t + \Delta t$. Dans un autre domaine, celui du contrôle, les influences sont décrites comme des modifications des variables de commandes du système à contrôler, et les lois de l'univers correspondent à la réponse du système à ces valeurs d'entrées. Enfin, comme nous l'avons dit, si l'on se résume à un système mono-agent, et que les seules actions à prendre en compte dans le monde sont celles accomplies par l'agent, alors ce modèle de l'action devient identique à celui qui fonctionne par transformation d'états, les seules influences à considérer provenant d'un seul agent.

4.3.2 Les états

Le modèle influence/réaction étant une extension du modèle STRIPS, nous reprendrons la définition des états de ce dernier. Un *état du monde* σ de Σ est décrit à l'aide d'un ensemble de formules atomiques de la forme $p(a_1, \dots, a_n)$ où p est un prédicat d'arité n et les a_1, \dots, a_n sont des constantes ou des termes fonctionnels ne comportant pas de variables. Dans ce cadre, la situation σ_1 présentée à la figure 4.9, peut être décrite ainsi:

$$\sigma_1 = \{ \text{cube}(\text{C}), \text{agent}(\text{A}), \text{agent}(\text{B}), \text{lieu}(\text{C}, 2), \\ \text{lieu}(\text{A}, 1), \text{lieu}(\text{B}, 3) \}$$

Mais alors que les théories classiques ne comprennent qu'une seule structure, l'état du monde, nous introduisons une autre structure appelé *influence* qui représente l'ensemble des gestes, ou tentatives d'actions, accomplis par les agents à partir de l'état courant du monde. Ces structures sont décrites elles aussi à l'aide d'un ensemble de formules atomiques définies à l'aide d'un langage équivalent à celui des états du monde, et on appelle Γ l'ensemble de ces structures d'influence qui peuvent être décrites à l'aide de ces formules. Les actions résultent de la combinaison des gestes en influences et de la réaction de l'environnement à ces influences.

4.3.3 Actions et réactions

Pour des raisons de simplicité, un opérateur prendra l'allure et la syntaxe STRIPS. Il sera représenté sous la forme d'un triplet $\langle nom, pré, post \rangle$ où nom est une expression de la forme $f(x_1, \dots, x_k)$ et où les x_i sont les seules variables autorisées à apparaître dans les formes $pré$ et $post$. $pré$ et $post$ sont des ensembles de formules de la forme $p(a_1, \dots, a_n)$ où p est un prédicat d'arité n et les a_i sont soit des constantes soit des variables. L'application de l'opérateur produit un ensemble d'influences. $pré$ décrit les conditions qui doivent être vérifiées pour que l'opérateur s'applique et $post$ l'ensemble des influences qui seront produites directement par cet opérateur lors de son application. L'application d'un opérateur peut être décrit par une fonction $Exec$ qui prend un opérateur, un état courant et retourne une influence:

$$Exec : Op \times \Sigma \rightarrow \Gamma$$

$$Exec(op, \sigma) = \gamma$$

qui est tel que

$$Exec(\langle nom, pré, post \rangle, \sigma) = Post \text{ si } pré(\sigma) \text{ est vérifié}$$

$$\text{et } \{\} \text{ sinon}$$

Par exemple, les opérateurs de déplacement de la figure 4.9 peuvent être décrits ainsi:

$$\text{déplacer} = \langle \text{nom: } \text{déplacer}(x, y, d),$$

$$\text{pré: } \text{agent}(x), \text{ cube}(y),$$

$$\text{post: } \text{poussée}(y, d) \rangle$$

et la tentative de déplacement du cube C par l'agent A à partir d'une situation σ_1 s'exprime alors ainsi:

$$Exec(\text{déplacer}(A, C, +1), \sigma_1)$$

et produit l'influence γ_1 :

$$\gamma_1 = \{\text{Poussée}(C, +1)\}$$

Les actions peuvent être composées pour définir une influence résultant de la simultanéité des actions exécutées à un état du monde donné. L'opérateur de combinaison d'actions simultanées, que l'on écrit \parallel fait simplement l'union des influences des actions prises séparément. La fonction $Exec$ peut alors être étendue pour devenir un morphisme de l'espace des actions muni de l'opérateur de simultanéité vers l'ensemble des influences Γ :

$$Exec : (Op, \parallel) \times \Sigma \rightarrow \Gamma$$

$$Exec(a \parallel b, \sigma) = Exec(a, \sigma) \cup Exec(b, \sigma)$$

Les lois de l'univers sont décrites à l'aide d'une fonction $React$ qui précise comment le monde se transforme d'un état à l'autre, et en particulier comment il réagit aux influences.

$$React : \Sigma \times \Gamma \rightarrow \Sigma$$

Cette fonction fait donc l'objet d'une description particulière pour chaque type d'environnement. Par exemple, dans un environnement dans lequel les cubes réagissent aux poussées en se déplaçant d'une distance proportionnelle aux poussées, on pourrait décrire cette fonction de la manière suivante:

```

React =
  < état : Lieu(c,x)
    influences: Poussee(c,d)
    faire: supprimer{Lieu(c,x)}
          ajouter{Lieu(c,x+dep)}
          avec dep = +/{d | Poussée(c,d)}
  >

```

où '/' représente un opérateur de réduction permettant d'appliquer un opérateur binaire à un ensemble de valeurs (ex: $+/\{1\ 2\ 3\ 4\} = 10$). Cette loi se lit ainsi: le nouvel état σ est égal à l'ancien état σ dans lequel on a supprimé toutes les indications de lieu, et dans lequel on ajoute toutes les descriptions de localisation après avoir sommé les poussées s'exerçant sur les objets du monde. Ainsi, le résultat de l'exécution simultanée de n actions est défini ainsi:

$$\sigma' = \text{React}(\sigma, \text{Exec}(op_1 || \dots || op_n, \sigma))$$

La résolution du problème de la figure 4.9 peut donc s'exprimer ainsi: les deux agents A et B exécutent simultanément leurs actions et produisent un nouvel état d'influence γ_1 :

$$\begin{aligned} \gamma_1 &= \text{Exec}(\text{déplacer}(A, C, +1) || \text{déplacer}(B, C, -1), \sigma_1) \\ &= \{\text{Poussée}(C, +1), \text{Poussée}(C, -1)\} \end{aligned}$$

L'application des lois de l'univers à l'état σ_1 initial produit le nouvel état d'influence σ_2 suivant:

$$\begin{aligned} \sigma_2 &= \text{React}(\sigma_1, \gamma_1) \\ &= \{\text{cube}(C), \text{agent}(A), \text{agent}(B), \text{lieu}(C, 2), \\ &\quad \text{lieu}(A, 1), \text{lieu}(B, 3)\} \end{aligned}$$

Le traitement de l'exemple des collisions entre mobiles, présenté à la figure 4.10, s'écrit de la même manière: on décrit l'opérateur de mouvement sans se soucier des collisions, laissant leur traitement aux lois de l'univers. L'opérateur de mouvement s'écrit donc ainsi:

```

avancer =
  < nom:  avancer(a,dx,dy)
    pré:  mobile(a,x,y)
    post: trans(a,dx,dy)
  >

```

et la loi de l'univers est définie par la fonction `React` suivante, qui indique que les deux mobiles produisent une explosion s'ils se trouvent à la même position et qu'ils changent simplement de position autrement.

```

React =
< état : lieu(a1,x1,y1), lieu(a2,x2,y2)
  influences: trans(a1,dx1,dy1), trans(a2,dx2,dy2)
  faire:
    Supprimer{lieu(a1,x1,y1), lieu(a2,x2,y2)}
    si x1=x2 et y1=y2 alors
      Ajouter{explosion(a1,x1,y1), explosion(a2,x2,y2)}
    sinon
      Ajouter{lieu(a1,x1+dx1,y1+dy1), lieu(a2,x2+dx2,y2+dy2)}
>

```

et qui se lit ainsi: s'il existe deux mobiles `a1` et `a2`, supprimer leur indication de position de l'état courant; si leurs positions sont identiques, alors dire qu'il y a explosion, sinon ajouter les influences de déplacement à leurs positions initiales. Par exemple, si à l'état initial deux mobiles `a` et `b` se trouvent aux points (50,50) et (70,70), l'état σ_1 peut être décrit ainsi:

$$\sigma_1 = \{\text{lieu}(a,50,50), \text{lieu}(b,70,70)\}$$

L'application des opérateurs `avancer(a,10,-20)` et `avancer(b,5,10)` conduisent à la production des influences:

$$\{\text{trans}(a,10,-20), \text{trans}(b,5,10)\}$$

et, par application des lois de l'univers, on obtient le nouvel état:

$$\sigma_2 = \{\text{lieu}(a,60,-50), \text{lieu}(b,75,80)\}$$

Si par contre les déplacements consistent à déplacer les mobiles à l'aide des actions `Avancer(a,20,0)` et `Avancer(b,0,-20)`, les influences seront évidemment de la forme:

$$\{\text{trans}(a,20,0), \text{trans}(b,0,20)\}$$

et, par application des lois de l'univers, on obtient:

$$\sigma_2 = \{\text{explosion}(a), \text{explosion}(b)\}$$

du fait que les mobiles se trouvent au même endroit.

4.3.4 Intérêt du modèle influences/réactions pour les SMA

L'intérêt majeur du modèle des influences/réactions est de bien distinguer ce qui appartient en propre aux agents des phénomènes qui se passent dans l'environnement. En différenciant les influences produites par l'agent des réactions du milieu, il est plus facile de concevoir des systèmes multi-agents hétérogènes, dans lequel les agents présentent des structures et des organisations internes différentes. Il suffit simplement de décrire l'ensemble des influences que les agents peuvent effectivement produire et d'utiliser l'environnement comme un système intégrateur dont les lois de l'univers sont les mêmes pour tous.

4.4 L'action comme processus informatique

L'informatique classique ne prend jamais en compte l'état global du monde. Elle conçoit l'univers (informatique) comme un ensemble d'activités qui s'exécutent en parallèle et que l'on nomme processus. Il est presque impossible de donner une définition précise de ce qu'est un processus, car il s'agit d'un concept primitif comme celui d'objet ou de machine. Disons que pour l'informaticien, un processus est un programme en cours d'exécution, que pour l'automaticien, un processus est un ensemble de tâches qui se déroulent séquentiellement ou en parallèle, et que pour un théoricien tel que Hoare, un processus représente le comportement d'un objet (Hoare 1985). L'intérêt de ce concept est de pouvoir décrire un comportement en ne s'intéressant qu'aux états initiaux et finaux d'actions considérées comme atomiques. Ces actions en s'exécutant donnent lieu à la production d'événements observables, la succession de ces événements permettant de définir un processus du point de vue de l'observateur.

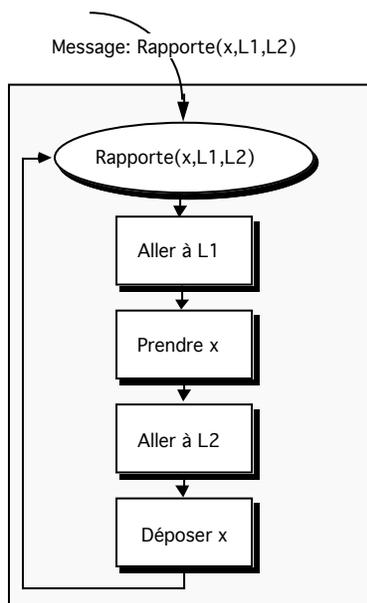


Figure 4.11: Le comportement de Clotaire représenté sous la forme d'un processus

Bien que la notion de processus soit un peu floue, elle correspond néanmoins à quelque chose de fondamental dès qu'on généralise son application à des systèmes multi-agents. En considérant que le monde est composé d'un ensemble de processus, on ne s'intéresse plus comme dans la section précédente à l'état global du monde, mais à l'ensemble des entités qui le composent, à leurs comportement et, surtout, aux interactions qui s'établissent entre ces processus. Il est possible de représenter un processus sous la forme d'une sorte d'organigramme, dans lequel les carrés correspondent aux actions atomiques et les flèches à l'ordre de succession.

Reprenons l'exemple de Clotaire qui doit rapporter des objets en se dirigeant dans une suite de pièces et en supposant qu'il n'ait plus de problèmes pour se repérer.

Lorsque Clotaire reçoit un message de la forme³

Clotaire << Rapporte(x,L1,L2)

qui indique qu'il doit aller chercher l'objet x qui se trouve dans le lieu $L1$ et l'apporter au lieu $L2$, il exécute le processus de la figure 4.11.

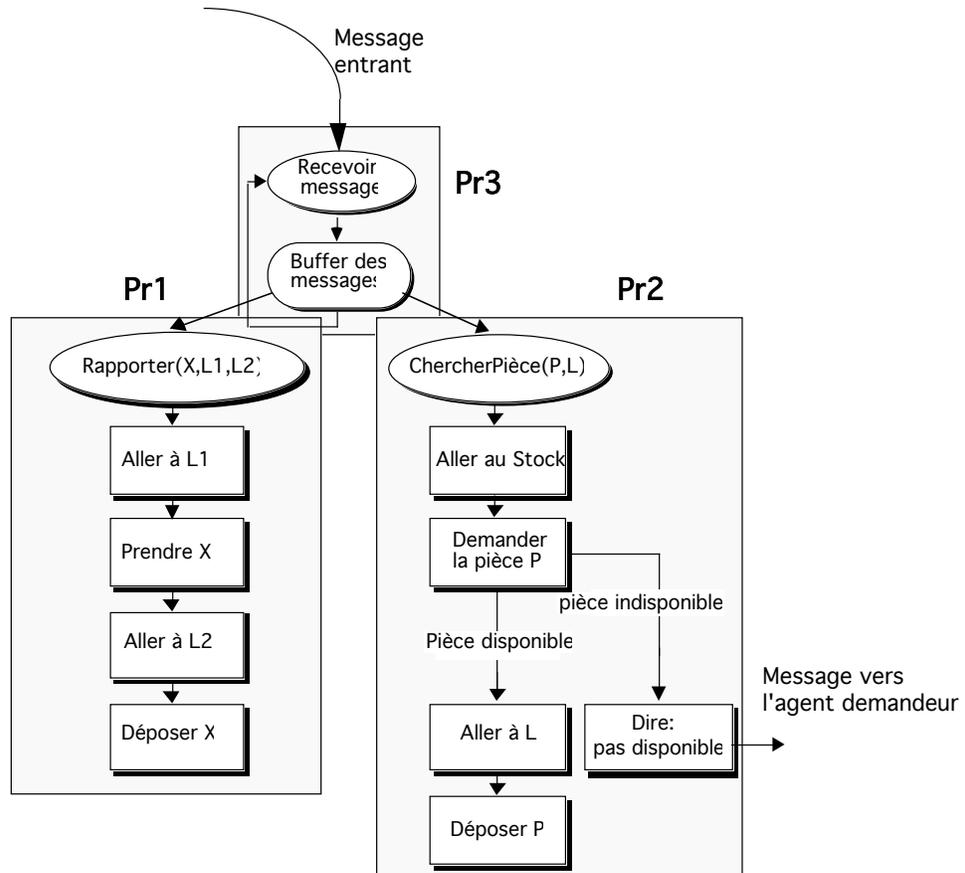


Figure 4.12: Le comportement de Clotaire est ici décrit par la combinaison de trois processus: deux processus Pr_1 et Pr_2 , qui s'excluent mutuellement, et un processus de gestion des messages, Pr_3 .

Mais l'intérêt des processus vient de la possibilité de pouvoir les coupler entre eux en leur donnant la possibilité de s'exécuter en parallèle. Par exemple, supposons que Clotaire puisse accomplir d'autres tâches, telles qu'aller s'approvisionner au stock de pièces détachées. Supposons de plus que Clotaire puisse mettre en attente les messages qu'il reçoit et les exécuter au fur et à mesure de ses possibilités. Le processus résulte alors du couplage de trois processus élémentaires: celui qui sert à rapporter des objets (Pr_1), celui qui consiste à aller chercher des pièces au stock (Pr_2), et celui de la gestion des messages en attente (Pr_3) (fig. 4.12). Pr_3 se déroule en parallèle par rapport à Pr_1 et Pr_2 qui s'excluent mutuellement. **Buffer** est un

³La syntaxe et la sémantique des messages sont décrites ci-dessous, section 4.11.2, et au chapitre 6.

composant qui sert à mémoriser des messages et à synchroniser la réception et l'interprétation des messages qui s'effectuent de manière concurrente⁴. Il est possible aussi de coupler le comportement de plusieurs agents de la même manière, en considérant que les messages produits par les uns vont directement au processus de réception des autres. Il existe de nombreuses manières de représenter des processus. Les plus courantes sont les automates à états finis, les automates à registres (ou réseaux de transitions augmentés) et les réseaux de Petri.

4.4.1 Représentation de processus par automates à états finis

L'intérêt de la notion d'automate à états finis est, d'une part, de pouvoir décrire très facilement le comportement d'un agent capable de mémoriser l'état dans lequel il se trouve et, d'autre part, d'être soutenu à la fois par un support théorique important et d'avoir été utilisé dans de nombreux domaines de l'informatique, tels que les architectures d'ordinateurs, l'automatisme, l'algorithmique, les langages formels, les techniques de compilation, les réseaux, etc.

Etats et transitions

Un automate à états finis est décrit par l'ensemble de ses états et par ce que l'on appelle sa fonction de transition qui spécifie l'état suivant en fonction de l'état actuel et des informations qu'il reçoit en entrée. On représente un automate à états finis sous la forme d'un graphe orienté dont les nœuds sont les états de l'automate et les arcs ses transitions. Sur les arcs, on indique à la fois les événements qui le font passer d'un état à un autre et les actions qui doivent être entreprises lors de cette transition. Par exemple, supposons que l'on veuille représenter le comportement d'un distributeur de bonbons, tel que celui de la figure 4.13.

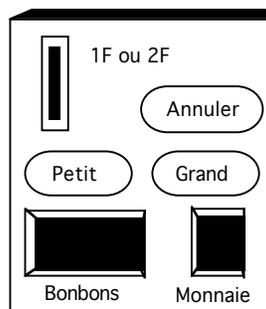


Figure 4.13: Un distributeur de bonbons. Il accepte des pièces de 1 F ou 2 F, et distribue des petits (prix 1 F) ou des grands bonbons (prix 2 F).

⁴J'utiliserai indifféremment les termes *parallèle* et *concurrent* pour parler de processus qui peuvent s'exécuter simultanément. Avec les Anglo-saxons, on différencie parfois ces deux termes en attribuant *parallèle* aux exécutions qui s'effectuent physiquement en même temps sur plusieurs processeurs distincts, et *concurrent* à l'aspect logique de cette simultanéité, que les processus s'exécutent en même temps ou non, mais je ne ferai pas ici cette distinction.

Il accepte des pièces de 1 F ou de 2 F et il distribue des petits ou des grands bonbons. Lorsque il reçoit trop d'argent, il rend la monnaie et, lorsqu'on appuie sur annuler, il rend la somme déjà introduite.

Ce distributeur peut être représenté comme une boîte noire comportant des entrées et des sorties (fig. 4.14). Les entrées sont les événements externes (mettre une pièce de 1 F ou 2 F, demander un petit ou un grand bonbon, annuler tout) et les sorties sont des commandes qui déclenchent les opérations de la machine (donner un bonbon, rendre la monnaie). Les entrées comme les sorties sont des variables booléennes. On suppose qu'une seule des entrées est valide à un instant donné, mais que plusieurs sorties peuvent être vraies simultanément.

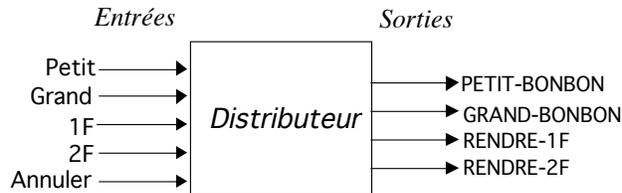


Figure 4.14: Représentation en termes d'entrées-sorties du distributeur

L'automate décrivant cette machine est présenté figure 4.15. Il comprend trois états, marqués s_0 , s_1 et s_2 . L'état s_0 est l'état initial, c'est-à-dire celui dans lequel se trouve la machine lorsqu'on la met en marche. Les états s_1 et s_2 représentent le fait que la machine a reçu respectivement 1 F et 2 F. Chaque transition est décrite par un couple $\langle \text{événement}, \text{action} \rangle$ qui décrit l'événement initiateur du changement d'état ainsi que l'action qui est accomplie par la machine. Formellement un automate à

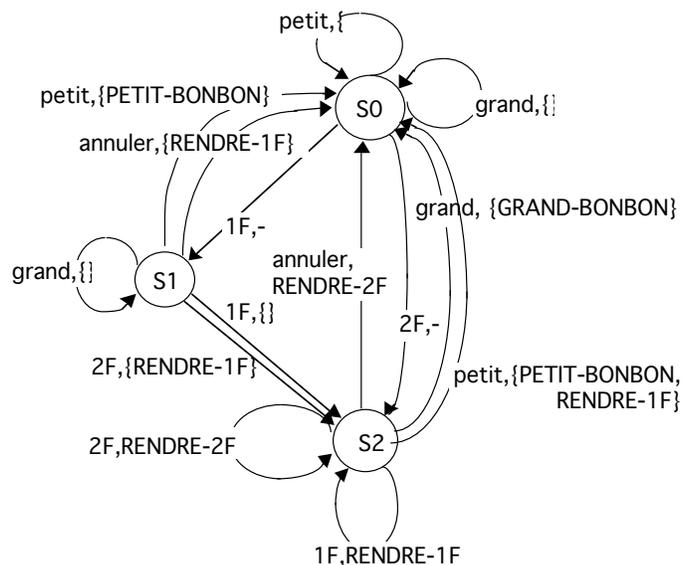


Figure 4.15: La représentation graphique de l'automate de distribution de bonbons

états finis comportant k entrées et p sorties est défini comme un sextuplet:

$$A = \langle E, S, C, \psi, \phi, s_0 \rangle$$

où E est un ensemble fini d'événements (décrits par des mots comprenant k caractères), S est un ensemble fini d'états, C est l'ensemble des commandes égal à B^p (où B est le domaine $0,1$), s_0 est un élément particulier de S que l'on appelle état initial, ψ est la *fonction de transitions*, c'est-à-dire une fonction de la forme:

$$\psi : S \times E \rightarrow S$$

et ϕ est la fonction d'activité, c'est-à-dire la fonction qui lance une commande et applique une opération lors d'une transition:

$$\phi : S \times E \rightarrow C$$

Par exemple, l'automate défini ci-dessus pour représenter le comportement d'un distributeur de machine peut être donné ainsi:

$$\begin{aligned} E &= \text{Petit, Grand, 1F, 2F, Annuler} \\ S &= s_0, s_1, s_2 \\ C &= \langle \text{PETIT-BONBON, GRAND-BONBON,} \\ &\quad \text{RENDRE-1F, RENDRE-2F} \rangle \end{aligned}$$

Les deux fonctions ψ et ϕ peuvent être représentées par le tableau 4.1. En ligne, on trouve les états, en colonne les événements et dans les cases, les couples formés de l'état suivant et de l'ensemble des actions à accomplir.

Ψ, Φ	Petit	grand	1F	2F	annuler
s₀	$s_0, \{\}$	$s_0, \{\}$	$s_1, \{\}$	$s_2, \{\}$	$s_0, -$
s₁	$s_0, \{\text{PETIT-BONBON}\}$	$s_1, \{\}$	$s_2, \{\}$	$s_2, \{\text{RENDRE-1F}\}$	$s_0, \{\text{RENDRE-1F}\}$
s₂	$s_0, \{\text{PETIT-BONBON, RENDRE-1F}\}$	$s_0, \{\text{GRAND-BONBON}\}$	$s_2, \{\text{RENDRE-1F}\}$	$s_2, \{\text{RENDRE-2F}\}$	$s_0, \{\text{RENDRE-2F}\}$

Table 4.1: La table des transitions d'un automate à états finis

Avantages et inconvénients des automates à états finis

Les automates à états finis sont très simples à manipuler et ils permettent de décrire assez facilement des comportements de systèmes logiques simples. Les architectures d'ordinateurs en font notamment un grand usage. Il est en effet possible de les simplifier ou de vérifier des propriétés⁵. Cependant ils s'avèrent très rapidement limités pour représenter des comportements plus complexes, et encore plus pour décrire l'évolution d'un système multi-agent:

⁵Pour en savoir plus, on pourra consulter n'importe quel livre parlant d'architectures des ordinateurs tels que le (Tanenbaum 1991) qui traite de la définition des automates à partir de fonctions booléennes, et de leur simplification par des méthodes telles que celle des tables de Karnaugh.

1. Leur capacité de calcul est limitée, du fait que leur nombre d'états est fini. Ils ne peuvent donc se souvenir d'une suite d'événements arbitrairement longue, ni vérifier, par exemple, qu'une suite de parenthèses quelconque est bien formée. Cette limitation constitue rarement une gêne, bien qu'elle empêche des robots de mémoriser le chemin qu'ils ont pris pour pouvoir revenir sur leurs pas. Si l'on veut dépasser cette limite, il faut étendre le formalisme des automates à états finis.
2. Les automates ne peuvent décrire que des processus séquentiels. De ce fait, tous les calculs, toutes les actions parallèles leur sont pratiquement interdits. Par exemple, il est impossible de représenter un agent qui agisse et qui communique en même temps avec un autre agent, sans passer par des mécanismes de fusion d'automate extrêmement lourds et rapidement impraticables (la fusion de deux automates de N états chacun produit un automate de N^2 états). Il faut passer alors à des formalismes plus puissants tels que celui des réseaux de Petri décrits plus loin (cf. section 4.4.3).
3. Enfin, toutes les mémorisations, telles que des changements de valeurs pour des registres, passent nécessairement par la transformation d'un état en un autre. Par exemple, si le prix des distributeurs de bonbons était de 2,40 F et que la machine accepte des pièces de 10 c, 20 c, 50 c, 1 F et 2 F, il faudrait augmenter considérablement le nombre d'états pour prendre en compte toutes les situations intermédiaires entre 10 c et 2,40 F, par pas de 10 c et en tenant compte des additions intermédiaires. Par exemple: si l'on est dans l'état 60 c et que l'on mette 50 c, on passe alors dans l'état 1,10 F. On pourra s'amuser à dessiner le graphe complet de cet automate qui n'est pourtant encore pas très compliqué! Pour dépasser cet inconvénient il faut pouvoir ici aussi étendre les capacités du formalisme, et nous allons maintenant voir que les automates à registres permettent de pallier cet inconvénient.

4.4.2 Automates à registres

Toutes ces limitations, et en particulier la dernière, ont amené à étendre la notion d'automates à états finis en introduisant des registres pouvant contenir des valeurs quelconques telles que des nombres, des chaînes de caractères, des symboles, des vecteurs, etc. On parle alors d'*automates à registres* ou de *réseaux de transitions augmentés* (ou ATN pour "Augmented Transition Network"). Ce formalisme permet d'obtenir un pouvoir de description plus étendu, au prix d'une perte des propriétés (et en particulier des propriétés de synthèse et de simplification des automates) associées à celui des automates à états finis.

Les automates à registres sont décrits, comme les automates à états finis, par un ensemble d'états et une fonction de transition sur ces états. La différence majeure vient du fait que l'on ajoute aux états un ensemble de registres qui peuvent être manipulés lors de déclenchement des transitions, ce qui permet d'appréhender des comportements plus élaborés qu'avec les automates à états finis. Les automates à

registres sont définis comme des n-uplets de la forme:

$$AR = \langle E, S, C, \psi, \phi, s_0, R, D, T \rangle$$

où les E, S, C, ϕ, s_0 , sont identiques à ceux des automates à états finis, où D est une famille de domaines de valeurs, où R est un ensemble de registres r_i à valeurs dans l'un des domaines D_i de D , où C est un ensemble de fonctions booléennes qui testent les valeurs des registres. La fonction ψ est alors légèrement modifiée, puisqu'elle possède le type suivant:

$$\psi = S \times E \times C \rightarrow S \times [R \rightarrow D]$$

c'est-à-dire qu'elle s'occupe non seulement de faire passer l'automate d'un état à l'autre, mais aussi de modifier la valeur des registres, le type fonctionnel $R \rightarrow D$ exprimant la définition d'un nouveau jeu de valeurs des registres.

La figure 4.16 donne la représentation graphique d'un automate à registres décrivant le fonctionnement d'un distributeur de bonbons acceptant des pièces de valeurs quelconques, sachant que le petit bonbon vaut 1 F et le grand bonbon 2,40 F. On a ajouté un registre V qui comptabilise la somme déjà introduite dans la machine. Les transitions sont décrites par des quadruplets

$\langle \text{événement, condition, modification des registres, commandes} \rangle$

qui indiquent l'événement initiateur du changement d'état, la condition portant sur des valeurs des registres, les modifications apportées à ces registres si la transition s'effectue et les commandes accomplies par la machine. Les événements portent maintenant des paramètres dont les valeurs peuvent être placées dans des registres.

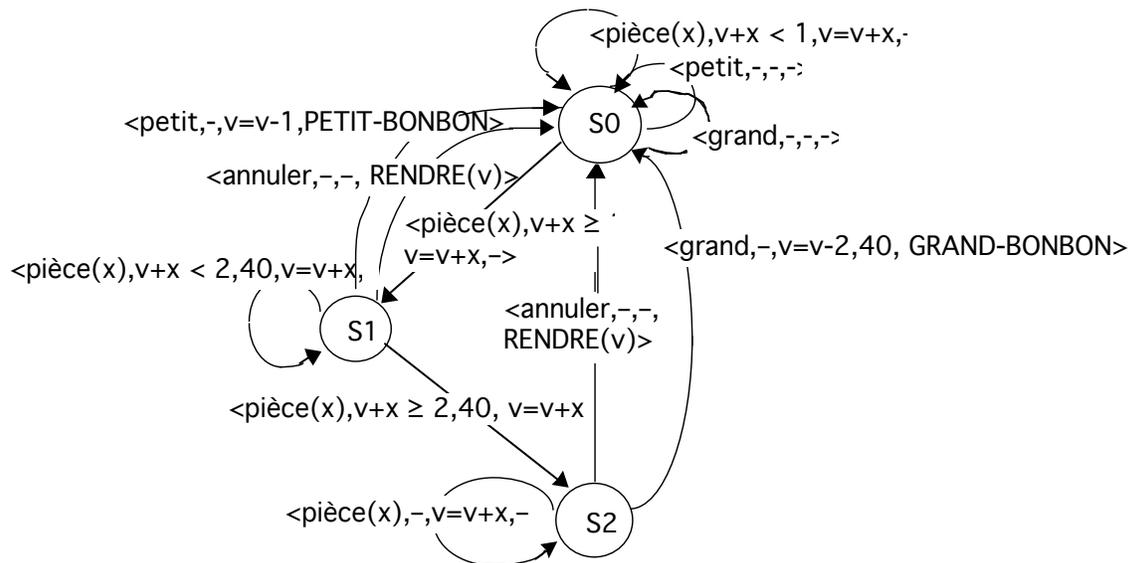


Figure 4.16: Un distributeur de bonbons modélisé par un automate à registre

L'intérêt des automates à registres est bien évidemment d'être beaucoup plus simple que leurs homologues à états finis dès que le nombre d'états devient trop

important. Grâce à leurs registres et aux conditions supplémentaires associées aux transitions, ils “factorisent” des informations qui doivent être réparties sur plusieurs états dans les automates à états finis.

4.4.3 Représentation de processus par réseaux de Petri

Les automates à états finis comme les automates à registres ne peuvent représenter que des processus séquentiels et sont donc bien limités pour prendre en compte les aspects de parallélisme caractéristiques des systèmes multi-agents. De ce fait, lorsque les agents deviennent plus complexes, ou tout simplement lorsque l'on cherche à préciser ce qui se passe lors d'interactions entre agents, il faut disposer de formalismes plus élaborés pour décrire des comportements et des organisations mettant en œuvre plusieurs processus.

Principes généraux

L'un des modèles les plus formels mais aussi les plus développés pour représenter des systèmes multi-processus est celui des réseaux de Petri. Ces derniers peuvent être considérés comme des outils de modélisation de processus essentiellement asynchrones fondés sur une représentation à la fois graphique et mathématique. L'aspect graphique permet de visualiser simplement les processus et de faciliter la conception et la communication de modèles de comportement, tandis que la formulation mathématique assure à ce formalisme des fondements rigoureux (cf. par exemple (Murata 1989)). Ils sont utilisés dans de nombreux domaines où il est nécessaire de représenter des mécanismes qui s'exécutent en parallèle: commandes de procédés industriels, protocoles de communications dans des réseaux, sémantique de programmes et de langages, prototypage de systèmes parallèles, etc. Enfin, il existe non pas un modèle de réseau de Petri, mais toute une famille permettant de modéliser des mécanismes aussi complexes que cela s'avère nécessaire.

Un réseau de Petri est défini comme un graphe orienté comprenant deux sortes de nœuds: des *places* et des *transitions*. Ce graphe est constitué de telle sorte que les arcs du graphe ne peuvent relier que des places aux transitions ou des transitions aux places, suivant en cela les règles de formation des réseaux de transitions augmentés définis précédemment. On représente graphiquement les places par des cercles et les transitions par des barres (fig. 4.17).

Les réseaux de Petri, à l'encontre des automates, permettent la représentation des aspects dynamiques des processus en déplaçant des *marques* de places en places. Dans les réseaux de Petri ordinaires, ceux qui sont à la base de toute la famille des réseaux de Petri, ces marques ne sont que des *jetons*, c'est-à-dire des éléments simples et indiscernables, alors que dans d'autres types de réseaux, tels que les réseaux colorés que nous verrons plus loin, ces marques sont des structures nettement plus complexes qui s'apparentent à des “records” à la Pascal. Les marques évoluent de place en place en suivant ce que l'on appelle des *règles de franchissement*.

On dit qu'une transition T peut être franchie, ou encore qu'une transition est *validée*, si toutes ses places d'entrées P pour lesquelles il existe un arc orienté de

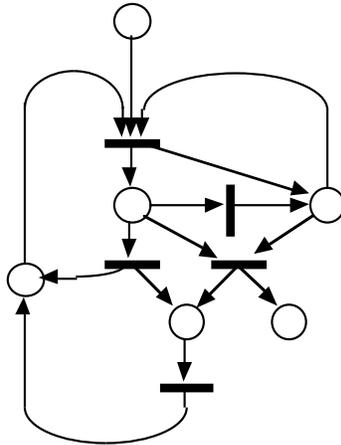


Figure 4.17: Un exemple de réseau de Petri comprenant 6 places et 5 transitions

P vers T possèdent une marque. Parmi toutes les places validées on en prend une au hasard qui sera effectivement déclenchée, et on dira que les marques franchissent cette transition. Franchir une transition revient à enlever une marque sur chaque place d'entrée et à en ajouter une sur chaque place de sortie (les places P pour lesquelles il existe un arc de T vers P). La figure 4.18 montre un exemple de franchissement de transition.

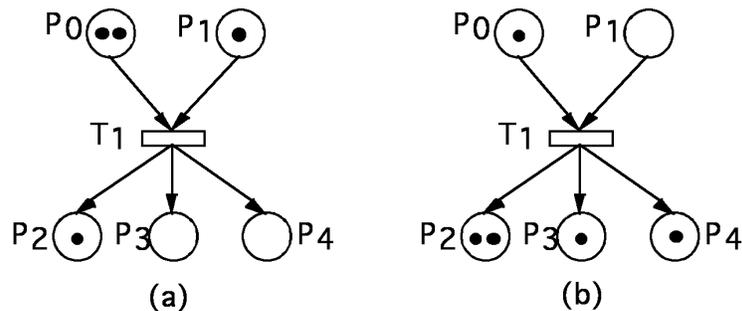


Figure 4.18: Exemple de franchissement de transition, avant (a) et après (b)

Lorsqu'il y a plusieurs transitions possibles en sortie d'une place, la règle de tirage au sort permet de déterminer laquelle des transitions doit être franchie, comme le montre la figure 4.19.

On appelle *marquage* une distribution de jetons sur les places, et marquage initial la distribution initiale de jetons dans le réseau.

Réseaux de Petri colorés

Si on souhaite distinguer les jetons entre eux, il est possible d'utiliser des *réseaux de Petri colorés*. Dans ce cas, on peut associer une valuation dans un domaine fini, que l'on appelle "couleur" du jeton. Il est aussi possible, et c'est ce que nous ferons par la suite, d'utiliser des n-uplets de variables colorées (sous la forme $\langle a, a, b \rangle$ par exemple) en lieu et place des jetons ordinaires (Jensen 1992). L'intérêt des

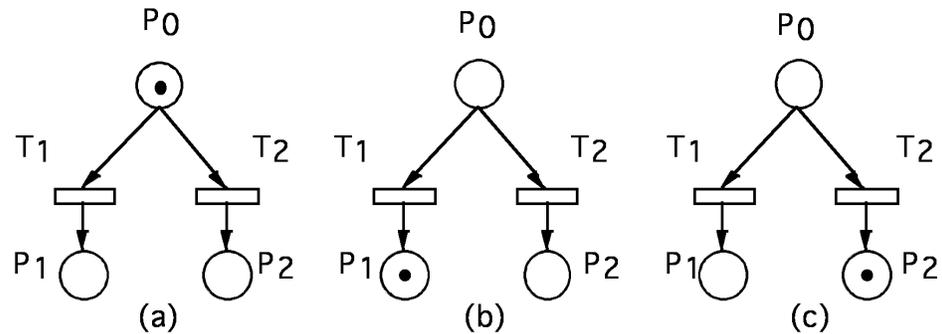


Figure 4.19: Les transitions T_1 et T_2 étant validées en (a), le tirage au sort des transitions à franchir produit le marquage (b) ou (c).

réseaux colorés est de pouvoir décrire des dynamiques relativement complexes avec un petit nombre de places et de transitions. Dans ce modèle de réseau de Petri, les transitions peuvent être considérées comme des sortes de règles qui ne peuvent être déclenchées que si les conditions associées à leurs arcs entrants sont remplies. Elles produisent des marques correspondant aux informations associées à leurs arcs sortants. L'exemple de la figure 4.20 montre le mécanisme de franchissement. La transition T_1 ne peut être franchie que s'il existe une marque de la forme $\langle x, x, n \rangle$ sur la place P_0 et une marque de la forme $\langle x, y \rangle$ sur la place P_1 . Dans l'exemple, cela est possible puisqu'il existe une marque $\langle a, a, 2 \rangle$ sur P_0 et une marque $\langle a, c \rangle$ sur P_1 à la situation (a). Ces marques sont alors supprimées des places d'entrées, et des marques correspondant aux indications des arcs sortants sont produites sur les places de sorties. Dans l'exemple, les marques $\langle a, c \rangle$, $\langle 3 \rangle$ et $\langle c, c \rangle$ sont respectivement ajoutées à la liste des marques des places P_2 , P_3 et P_4 (b).

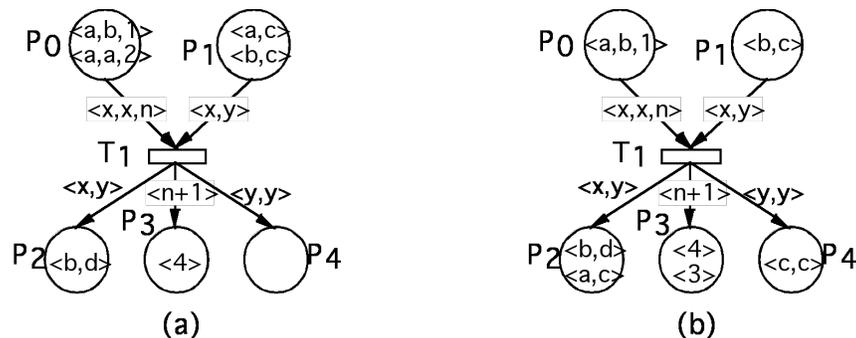


Figure 4.20: Franchissement d'une transition dans un réseau coloré

Il existe bien d'autres modèles de réseaux de Petri. En particulier les réseaux de Petri à objets qui supposent que les marques sont des objets qui peuvent eux-mêmes contenir des références à d'autres objets et dont les champs peuvent être modifiés (Sibertin-Blanc 1985), les réseaux de Petri à prédicats, les réseaux synchrones et temporisés, voire même les réseaux continus (David et Alla 1989). Pour l'instant, notre choix s'est porté sur les réseaux de Petri colorés, qui présentent l'avantage

de fournir déjà une bonne puissance de modélisation tout en étant encore capables d'offrir des possibilités de vérification formelle sur leurs propriétés.

Modélisation de processus par réseaux de Petri

Si le modèle des réseaux de Petri est assez simple à comprendre, il pose deux problèmes: il s'avère difficile de modéliser un comportement à cause de la grande souplesse et du grand pouvoir d'expression du modèle et il devient malaisé de voir ce qui se passe *de visu* dès que le processus à modéliser s'avère un peu complexe. Le premier de ces problèmes requiert un peu d'expertise. Il s'agit d'une autre forme de programmation: modéliser un ensemble de processus sous la forme d'un réseau de Petri relève simplement d'un art au même sens que l'écriture d'un programme ou de l'établissement d'une spécification. Le second peut être résolu en définissant un formalisme de représentation de plus haut niveau mais dont la structure peut être transformée en un réseau de Petri. On dispose alors d'un langage plus intelligible, tout en bénéficiant des propriétés formelles associées à ces réseaux.

L'autre intérêt des réseaux de Petri est de pouvoir intégrer le formalisme des automates à états finis comme une sous-classe des réseaux de Petri. On représente alors les états de l'automate sous la forme de place dans le réseau, et sa fonction de transition comme un ensemble de transitions, chacune ne possédant qu'une place en entrée et une place en sortie. On les appelle alors machines à états.

Les réseaux de Petri ont un pouvoir d'expression bien supérieur à celui des automates à états finis car ils peuvent facilement modéliser un ensemble de processus et décrire des conflits éventuels d'accès aux ressources.

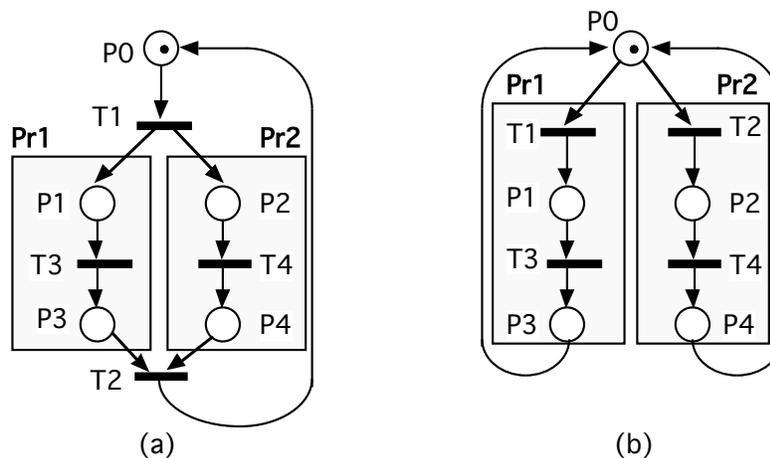


Figure 4.21: Pr1 et Pr2 s'exécutent simultanément en (a) et alternativement en (b).

Par exemple, la figure 4.21 montre deux exemples d'ensembles de processus modélisables par réseaux de Petri. Dans le premier cas (a) les processus Pr_1 et Pr_2 ⁶ s'exécutent simultanément. En effet, un jeton placé en P_0 franchit la transition T_1 et occupe les places P_1 et P_2 simultanément. Ensuite, les transitions T_3 et

⁶Dans ces exemples et les suivants, nous avons fait figurer les processus en grisé pour faciliter la compréhension des figures. Mais la notion de processus ne fait pas partie des réseaux de Petri

T_4 s'exécutent indépendamment l'une de l'autre, mais la transition T_2 sert de synchronisation. Elle ne pourra être franchie que si les places P_3 et P_4 comportent toutes les deux au moins un jeton. En revanche, dans le second cas (b) les processus Pr_1 et Pr_2 s'exécutent indépendamment l'un de l'autre, et s'il n'existe qu'un seul jeton en P_0 , ils s'exclurent mutuellement. En effet, le jeton en P_0 passera soit par Pr_1 ou par Pr_2 , et il faudra attendre que le jeton ait franchi toutes les transitions associées au processus pour que le jeton revienne en P_0 et que les deux processus soient de nouveau exécutables. Mais si P_0 contient plusieurs jetons, les deux processus Pr_1 et Pr_2 pourront éventuellement s'exécuter en parallèle.

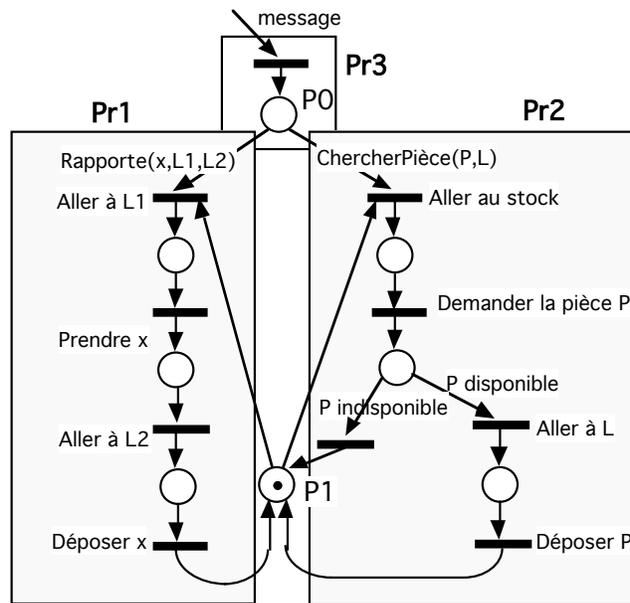


Figure 4.22: La représentation du comportement de Clotaire exprimé sous la forme d'un réseau de Petri

Il est ainsi possible de décrire précisément le comportement de Clotaire tel qu'on l'avait présenté à la figure 4.12, où l'on avait supposé qu'il pouvait rapporter des objets et aller chercher des pièces en stock tout en recevant des messages. Il s'agit d'une modification du schéma de la figure 4.21.b. Les deux processus travaillent en exclusion mutuelle, laquelle est gérée par la place P_1 , qui contient initialement un jeton. Lorsqu'un message arrive, il est rangé dans la place P_0 , et si P_1 contient un jeton, l'un des deux processus Pr_1 ou Pr_2 est choisi en fonction de la nature du message. S'il s'agit d'un message **Rapporte**, alors le processus Pr_1 est choisi et activé et le jeton de la place P_1 est consommé ainsi que le message. Les marques passent alors de place en place jusqu'à atteindre la place P_1 . Pendant ce temps, puisque P_1 ne contient aucun jeton, aucun autre message ne peut être pris en compte par l'un des deux processus. Mais dès qu'il récupère un jeton, c'est-à-dire lorsque Pr_1 ou Pr_2 sont terminés, et si P_0 contient des messages, l'un de ces deux processus pourra de nouveau être activé, en fonction de la forme du message.

stricto sensu, qui ne font intervenir que des places, des transitions (et leurs arcs associées) et des

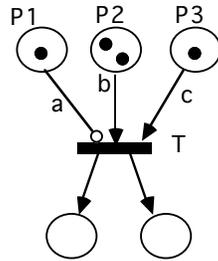


Figure 4.23: Un réseau de Petri dont l'arc a est inhibiteur

Les réseaux de Petri que nous venons de voir ne permettent pas de représenter facilement des mécanismes de priorité entre actions. Il faut en effet pouvoir indiquer que quelque chose *empêche* une transition, ce qui n'est pas possible dans le modèle normal. On peut alors ajouter des arcs inhibiteurs qui relient une place à une transition et qui fonctionnent de manière inversée par rapport aux arcs normaux, en empêchant le franchissement d'une transition si la place "inhibitrice" contient une marque. Dans la figure 4.23, l'arc (a) est inhibiteur, ce qui bloque la transition tant que la place P_1 possède une marque. Les arcs (b) et (c) sont des arcs normaux.

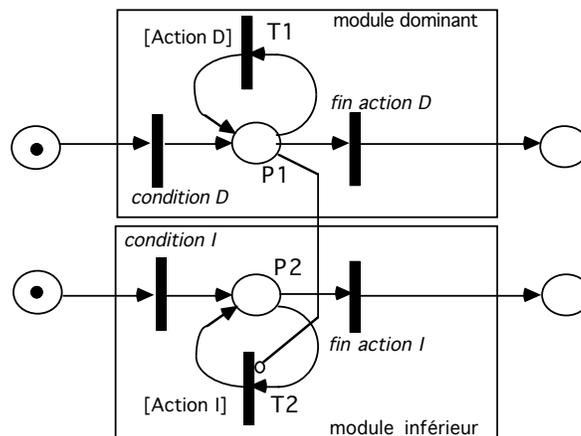


Figure 4.24: Un modèle d'architecture de subsomption utilisant un arc inhibiteur

Par exemple, un modèle d'architecture de subsomption (cf. chap. 3), qui repose sur la définition d'une hiérarchie entre modules, peut être décrit à l'aide de réseaux de Petri à arcs inhibiteurs (fig. 4.24). On constate que si le module dominant est activé et donc qu'une marque se trouve sur la place P_1 , alors la transition T_2 est inhibée, ce qui empêche le module inférieur de fonctionner.

Très vite, les modélisation par réseaux de Petri deviennent fort complexes et il est alors nécessaire de décomposer le réseau en modules distincts. C'est ce que nous ferons avec la définition du formalisme BRIC à la section 4.11.

4.4.4 Autres modèles événementiels

Il existe de nombreux autres modèles de mécanismes concurrents qui peuvent être utilisés pour décrire des systèmes multi-agents. En particulier les modèles CCS (Milner 1989) et CSP (Hoare 1985) qui servent à décrire de manière formelle le fonctionnement de systèmes parallèles et distribués en interactions, pourraient faire l'objet d'un emploi plus important dans le cadre des systèmes multi-agents. Néanmoins, il reste un travail important à entreprendre, du fait que ces outils formels ont été développés dans le cadre des processus distribués et de la sémantique des langages parallèles, et qu'ils ne sont pas toujours très bien adaptés à la problématique des systèmes multi-agents et notamment celle des systèmes multi-agents situés, pour lesquels les agents accomplissent des actions dans un environnement.

4.5 L'action comme déplacement physique

Pourtant bien antérieur dans son développement, le concept d'action tel qu'il est utilisé en physique a peu influencé le monde de l'intelligence artificielle, peut-être trop obnubilé par les formalismes logiques. Pour la physique, l'action est avant tout ce qui entraîne le mouvement, ce qui change l'état dynamique d'un corps, c'est-à-dire une accélération. La première loi de Newton le décrit bien: *Un corps persévère dans son état de repos ou de mouvement rectiligne uniforme, sauf si des forces le contraignent d'en changer* (Newton 1985). Pourquoi ne pas s'inspirer de ces principes pour exprimer des comportements d'agents situés pour lesquels l'environnement est conçu comme un espace métrique? En effet, cette approche peut s'avérer particulièrement féconde dans le cas d'agents réactifs qui doivent planifier leur trajectoire, éviter d'entrer en collision, et si besoin, être capables de se rencontrer, et d'évoluer ensemble. Il suffit de penser au vol d'une escadrille d'avions ou à un flux de voitures sur une route principale pour avoir une idée de cette application mécanique de l'action.

En effet, lorsque des agents se déplacent, qu'il s'agisse de robots mobiles ou de véhicules qui tentent de s'éviter, leur comportement se manifeste naturellement comme un mouvement dans un espace euclidien. Pourtant, bien qu'il puisse paraître naturel de considérer les systèmes à base de robots mobiles comme des solides matériels se déplaçant dans l'espace au cours du temps, pendant longtemps les recherches en planification de mouvement de robots mobiles se sont cantonnées dans une analyse symbolique et logique de l'analyse des déplacements. Ce n'est que récemment que des points de vue réactifs utilisant des techniques issues de champs de potentiels ont vu le jour, et elles sont encore considérées comme presque "révolutionnaires", alors qu'elles peuvent en fait apparaître bien simples car elles ne mettent en œuvre que du calcul vectoriel.

Il est en effet naturel pour des systèmes réactifs de se tourner vers des représentations "analogiques" de la réalité, les déplacements étant traités comme des additions de vecteurs dans un espace euclidien (fig. 4.25).

Si au temps t_0 l'agent se trouve au point M , il se trouvera au point M' au temps $t_0 + \Delta t$ ayant accompli alors le déplacement D , ce qui permet d'écrire vectoriellement

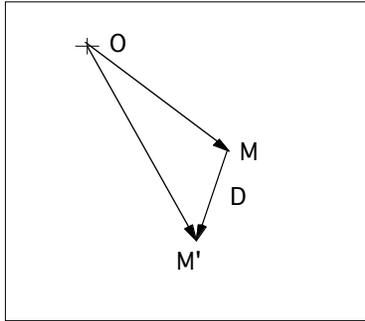


Figure 4.25: Le déplacement géométrique est une forme d'action qui s'exprime comme une transformation géométrique dans un espace.

que:

$$O\vec{M}' = O\vec{M} + \vec{D}$$

Il est possible de pousser encore plus loin l'analogie avec les particules physiques en considérant la notion de champ. Qu'il s'agisse de champs gravitationnel, électriques ou magnétiques, les champs servent en physique à expliquer l'origine des forces et donc le mouvement des particules. On distingue généralement deux types de champ: les champs scalaires et les champs de vecteurs. Ces champs associent à tous points de l'espace une valeur. Lorsqu'il s'agit d'un champ scalaire, la valeur est un scalaire, c'est-à-dire un nombre qui représente l'intensité du champ au point considéré. Dans le cas d'un champ vectoriel, la valeur est un vecteur.

Ces champs peuvent se combiner et se superposer s'il s'agit de champs de même nature. Généralement ces combinaisons s'expriment sous forme additive en additionnant les grandeurs des champs en chacun des points de l'espace. Si l'on appelle $\mathbf{U}(p)$ et $\mathbf{V}(p)$ deux champs définis par leurs valeurs en tout point quelconque p de l'espace, alors la somme des champs est un champ $\mathbf{W}(p)$ dont la valeur est simplement la somme des valeurs en chaque point:

$$\mathbf{W}(p) = \mathbf{U}(p) + \mathbf{V}(p)$$

Il est aussi possible de combiner des champs de même nature en ne retenant que la valeur la plus importante s'il s'agit de champs scalaires. Dans ce cas, l'opération *max* est définie ainsi:

$$\max(\mathbf{U}(p), \mathbf{V}(p)) = \mathbf{U}(p) \text{ si } \mathbf{U}(p) \geq \mathbf{V}(p), \mathbf{V}(p) \text{ sinon}$$

La visualisation de la valeur d'un champ scalaire en fonction de sa position dans un espace à 2 dimensions donne un effet de relief: les pics sont les lieux de plus grandes intensités, et les vallées les zones de moins grande influence du champ. Les pentes de ces montagnes donnent ce que l'on appelle le *gradient* d'un champ, c'est-à-dire sa variation en fonction du lieu et détermine la direction de plus grande pente associée à chaque point. Le gradient est une fonction vectorielle qui associe ainsi un champ

de vecteurs au champ scalaire initial. Si le champ scalaire définit un potentiel, alors le gradient définit une force correspondant au potentiel:

$$\vec{F}(p) = -\vec{\nabla}U(p) \quad (4.1)$$

où ∇ représente l'opérateur gradient. Ces champs vont déterminer des tendances attractives ou répulsives (que l'on exprime souvent sous la forme de forces) qui font se mouvoir les particules dans une direction correspondant au point de la plus grande attraction d'un champ. Le gradient joue alors un rôle essentiel, car plus le gradient est important, c'est-à-dire plus on passe rapidement d'une vallée à un mont, plus la force attractive ou répulsive est forte.

4.5.1 Déplacements dans un champ de potentiel

Cette notion de champ est surtout utilisée dans les systèmes réactifs pour déterminer le comportement des agents. Il s'agit d'une technique relativement récente qui prend sa source dans une critique radicale des techniques classiques de planification pour déterminer un chemin ou une route à suivre (Latombe 1991; Barraquand et Latombe 1989). Le principe général est de considérer que les objets de l'environnement émettent des signaux dont l'intensité est proportionnelle à la distance (ou plus généralement au carré de la distance) au but, ces signaux sont diffusés dans tout l'espace et définissent un potentiel $U(p)$, dont il est possible de tirer un champ de vecteur force $F(p)$ à partir de l'équation (4.1). On considère alors que les buts définissent des champs de potentiel attractifs, alors que les obstacles correspondent à des champs de potentiel répulsifs. De ce fait, le champ global U est défini comme la somme vectorielle des champs d'attraction et de répulsion:

$$U(p) = U_{attr}(p) + U_{repul}(p)$$

Ces champs sont généralement définis en fonction de la distance au but ou aux obstacles. Les champs attractifs peuvent présenter une intensité proportionnelle à la distance avec le but ou, par analogie avec les champs "naturels", proportionnelle au carré de cette distance:

$$U_{attr}(p) = k * dist(p, p_{but})^2$$

Les champs de potentiel des obstacles sont souvent un peu plus compliqués, car ils doivent servir seulement à empêcher le mobile d'entrer en collision avec l'obstacle sans qu'il influe trop sur sa trajectoire à une longue distance. Il suffit pour cela de définir une barrière d'influence, c'est-à-dire une distance $dist_{influence}$ à partir de laquelle l'influence des obstacles ne se fera plus sentir.

$$\begin{aligned} U_{repul}(p) &= k' \frac{1}{dist(p, p_{obs})^2} \text{ si } dist(p, p_{obs}) \leq dist_{influence} \\ &= 0 \text{ sinon} \end{aligned}$$

Dans ce cas, l'intensité du champ croît très fortement lorsqu'on se rapproche des obstacles (elle est même "infinie" à l'endroit de l'obstacle). Le déplacement d'un

robot revient alors à “descendre” le champ de potentiel en suivant la ligne de plus grande pente, le but représentant la vallée et les obstacles les montagnes qu’il faut fuir. Cette technique est donc très simple à mettre en œuvre et surtout très facile à comprendre du fait de son caractère éminemment visuel. En effet, si les déplacements des robots s’effectuent dans un plan, il est possible de représenter ces champs à l’aide de graphiques en trois dimensions, la hauteur représentant l’intensité du champ. La figure 4.26 montre un tel champ de potentiel pour un but et deux obstacles.

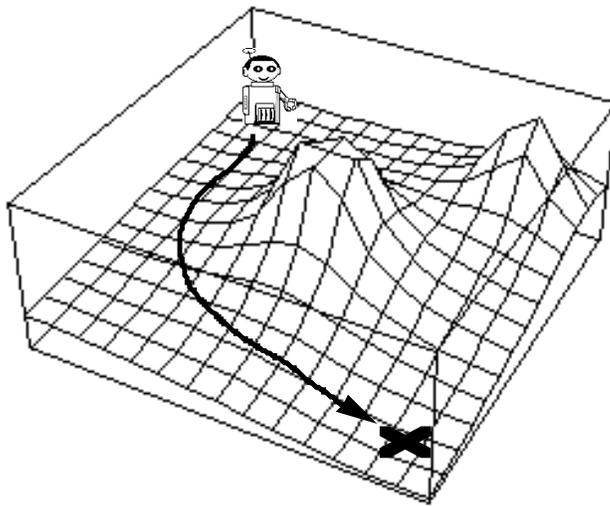


Figure 4.26: Dans un système à champs de potentiels, l’agent est attiré par le but et repoussé par les obstacles.

On peut voir que l’agent essaye de trouver une trajectoire présentant le meilleur compromis entre l’attraction du but d’une part et la répulsion des obstacles d’autre part. Si l’on désire que la trajectoire passe plus près des obstacles, il suffit de donner une valeur plus pentue aux champs répulsifs et inversement de les “arrondir” si l’on désire s’en éloigner.

Si les champs de potentiel ont le mérite de la simplicité, ils présentent malheureusement plusieurs inconvénients dont le principal est l’existence de minima locaux. En effet, face à certains obstacles concaves, un robot peut être bloqué dans un puits de potentiel qui l’empêche de revenir en arrière et donc de faire le tour de l’obstacle. Pour résoudre ce problème, il suffit de construire les champs attractifs différemment en considérant non pas la distance absolue entre le but et les points de l’espace, mais la distance minimale des chemins qui vont du but à ces points de l’espace en contournant les obstacles. Cette distance peut être calculée très simplement à partir d’un système de “vague” (Barraquand et Latombe 1989). On parcourt tous les points de l’espace en parallèle à partir du (ou des) but(s), en incrémentant à chaque fois la valeur de l’intensité du champ et en contournant les obstacles. Ce calcul s’effectue donc comme la propagation du front d’une vague, ayant pour origine le but, qui emplit tout l’espace libre en contournant les obstacles. Pour construire un tel champ, il faut au préalable “digitaliser” l’espace en le décomposant en

un ensemble de cellules de tailles égales ayant généralement la forme de carrés ou mieux d'hexagones (s'il s'agit d'un espace de dimension 2, mais on peut généraliser facilement à la dimension 3). L'algorithme est alors distribué de telle manière que chaque cellule agisse de manière autonome. Lorsqu'une cellule reçoit une nouvelle valeur qui est inférieure à sa propre valeur de potentiel, elle la stocke et propage à toutes les cellules voisines cette valeur incrémentée d'une unité:

```
Type Cellule
Champs
    potentiel: Nombre
    voisins: Liste de Cellule

Procédure valeur(x, v)
    si x.potentiel n'est pas défini
    ou si v < x.potentiel
    alors
        x.potentiel := v
        pour tout y dans x.voisins
            valeur(y,v+1)
    sinon ne rien faire
Fin
```

Il est possible de faire tourner cet algorithme de manière totalement parallèle et, si l'on dispose des calculateurs adéquats, d'obtenir rapidement la valeur du potentiel en chaque point de l'espace⁷. Le lecteur désirant en savoir plus sur les techniques de déplacement d'agents mobiles et en particulier sur les techniques fondées sur la notion de champ de potentiel pourra se référer à l'excellent livre de Jean-Claude Latombe (Latombe 1991).

Nous reverrons souvent cette technique appliquée aux SMA réactifs, comme celui de la simulation d'un monde de fourmis (cf. chap. 7), ou d'un agent situé qui doit se déplacer dans un monde comprenant des obstacles (cf. section 4.9.3). Ces techniques peuvent ainsi s'appliquer aussi bien aux mouvements dans un plan qu'à la diffusion de marqueurs "chimiques" qui agissent comme des champs d'attraction pour le comportement des agents.

4.5.2 Intérêt de cette conception de l'action

Cette approche physique de l'action s'avère très efficace dans deux cas:

1. Lorsque les actions se ramènent naturellement à des déplacements dans un espace métrique. En particulier, tous les systèmes multi-agents qui conçoivent

⁷Cette version fonctionne très bien mais ne permet pas de savoir si les calculs sont effectivement réalisés en parallèle, lorsque le programme termine. Pour avoir une telle information, il faut employer des algorithmes un peu plus compliqués qui estampillent les sources des signaux et retournent les valeurs locales de terminaison, comme nous le verrons au chapitre 7 pour les algorithmes d'allocation de tâches. On pourra se référer à (Raynal 1987) pour avoir de plus amples informations sur les problèmes de terminaisons dans les algorithmes de propagation de vagues.

les agents comme des sortes de petits robots qui se déplacent dans un environnement peuvent en tirer facilement profit.

2. Lorsque les actions doivent être décomposées dans le temps et qu'il est possible d'utiliser des notions de champs pour décrire les tendances des agents à effectuer quelque chose.

Les champs de vecteurs offrent une gestion assez efficace de la coordination d'actions entre agents multiples lorsque les actions se résument à des déplacements dans un espace physique. Leur utilisation, pour résoudre des problèmes de coordination de déplacements de plusieurs agents mobiles tels que celui de la régulation de trafic aérien, s'avère particulièrement féconde comme nous le verrons au chapitre 8. Néanmoins, les champs de vecteurs présentent quelques inconvénients.

Le premier est de ne pas s'intégrer naturellement à une conception logique de l'intention et de l'action. Comment relier en effet la description d'un état du monde sous forme de formules logiques et les actions sous formes de déplacements dans un espace muni d'un champ de potentiels et comment planifier ces actions dans de telles situations?

Le deuxième, plus fondamental, est de se montrer inefficace dans les situations qui ne savent pas représenter les situations dans un espace métrique. Par exemple, il est clair que l'on ne sait pas résoudre des problèmes administratifs de cette manière. En particulier, tous les systèmes multi-agents fondés sur la notion de coopération d'expertises (cf. chap. 1) ne savent absolument pas tirer parti de tels modèles mécanistes.

Le troisième enfin provient de ce qu'il est difficile de rendre compte de l'évolution globale d'un système lorsque les agents sont individuellement capables de décider de leurs propres actions. En effet, un agent, à la différence d'une particule simple dont l'évolution est déterminée par les caractéristiques des champs dans lesquels elle évolue, peut disposer d'une véritable autonomie de décision qui rend très difficile toute prévision quant au comportement global du système. Sauf à réduire la capacité de ces agents en les rendant tous semblables (et très simplistes souvent), il n'existe pas de véritable théorie qui sache prévoir le comportement d'un système dans le cadre d'une telle autonomie. On dit alors que le comportement global du système *émerge* de l'ensemble des interactions locales.

4.6 L'action comme modification locale

A l'encontre des conceptions qui considèrent que chaque action transforme de manière globale le monde, notre intuition commune appréhende l'action comme une modification locale: mes actes ne transforment que les entités qui se trouvent proches de moi. Si je fais tomber un verre par terre, je ne suppose pas que le cours des planètes s'en trouvera modifié. Toute action ne produit qu'une perturbation locale, une altération qui se trouve en contact ou de toute manière à une distance finie de la cause de cette action. Même si elle présente des conséquences importantes, telles qu'une explosion nucléaire par exemple, l'action peut être vue comme une cause

locale dont les effets sont propagés de proche en proche dans une grande portion d'espace.

A la différence de l'approche par transformation d'états qui décrit le monde comme un ensemble de formules valides, on suppose ici que le monde est constitué d'un réseau dont les nœuds représentent des entités "réelles" (des portions d'espace, des objets physiques, des êtres, des animaux, des personnes, mais aussi des idées, des situations, des événements, des tâches) et les arcs correspondent aux liens qui existent entre ces entités: liens de contacts, de relations sociales (les accointances), de dépendances, etc. Chaque nœud agit de manière indépendante et modifie son propre état en fonction de ses perceptions. On retrouve ici l'idée même de système multi-agent comme fondement de la notion d'action. Bien qu'il existe peu de modèles permettant de traiter de ce type d'action dans toute sa généralité, il existe des outils théoriques et pratiques qui, en réduisant les objectifs, permettent de prendre en compte cette idée de l'action.

4.6.1 Les automates cellulaires

Les automates cellulaires sont des systèmes dynamiques dont le comportement est totalement déterminé en termes de relations locales. Un système à automates cellulaires est composé d'un ensemble d'automates à états finis répartis sur les nœuds d'un réseau périodique, généralement une grille à deux dimensions à maille carrée (fig. 4.27). Chaque automate, appelé *cellule*, est lié aux automates qui l'entourent et ses entrées sont liées aux états de ses voisins. De ce fait, l'état d'une cellule à l'instant $t + 1$ dépend non seulement de son état précédent mais aussi des états de tous ses voisins à l'instant t . De plus, on suppose que tous les automates sont identiques, qu'ils obéissent à la même fonction de transition, et que leurs transitions sont synchrones, c'est-à-dire qu'ils changent d'états tous en même temps.

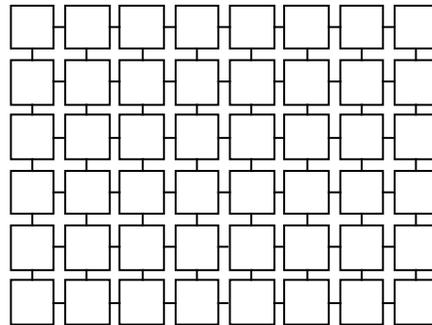


Figure 4.27: Un automate cellulaire plan de connexité 4

L'un des exemples les plus connus des automates cellulaires est celui du "jeu de la vie" dû à John Conway dont le fonctionnement tente d'imiter celui des organisations vivantes. Ce jeu fut très célèbre dans les années 60, en particulier grâce à la publicité que leur donna Martin Gardner, journaliste scientifique de *Scientific American*, qui les décrit en détail dans sa rubrique "récréations mathématiques".

Une cellule du jeu de la vie ne peut avoir que deux états: 1 ou 0, l'état 1 représentant une cellule vivante et l'état 0 une cellule morte. La fonction de transition des automates est la suivante:

- *Naissance*: si l'automate est mort, il devient vivant (il passe à l'état 1) si exactement trois de ses voisins sont vivants. Dans tous les autres cas, il reste mort.
- *Mort*: si l'automate est vivant, il reste vivant si 2 ou 3 de ses voisins sont vivants et il meurt dans les autres cas.

La figure 4.28 montre une évolution possible du jeu de la vie sur quelques instants. L'intérêt de ce jeu est d'avoir montré qu'il était possible d'avoir des configurations dynamiques relativement stables. En particulier, l'une d'entre elle, le "planeur" (glider), se reproduit à l'identique à l'instant $t + 4$ mais en s'étant déplacée d'un cran. Ce jeu est fort intéressant car il montre qu'à partir de règles locales très simples, il est possible d'obtenir des configurations complexes et stables.

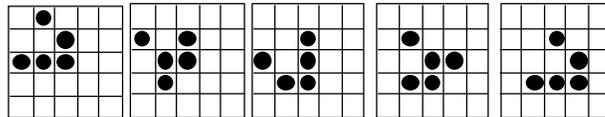


Figure 4.28: Le déplacement d'un "planeur" (glider), une figure classique du jeu de la vie qui se reproduit à l'identique au bout de 4 cycles en se déplaçant d'une case en diagonale.

Les automates cellulaires sont très utilisés de nos jours pour réaliser des simulations en physique et plus particulièrement dans le cadre de la dynamique des fluides. Des extensions des automates cellulaires pour incorporer non plus des états discrets mais des états continus ont été utilisées dans le cadre de la dynamique des populations. Ils sont de plus en plus utilisés pour étudier l'émergence de phénomènes collectifs tels que les turbulences, le chaos, etc. En particulier, le domaine de la vie artificielle fait un usage intensif de ces techniques (Langton 1988). Dans tous les domaines où il est possible de caractériser des règles locales fondées sur la proximité et la géographie, les automates cellulaires peuvent être utilisés avec profit.

4.6.2 Représentation d'un automate cellulaire

La cellule d'un automate est représentée par une fonction de transition ϕ qui associe à son état et à l'état de ses cellules voisines son état suivant. Chaque cellule est donc définie par une fonction de transition unique ϕ définie ainsi:

$$\phi(s_0, s_1, \dots, s_n) = s'_0$$

où s_0 est l'état de la cellule au temps $t - 1$ et les s_i sont les états des cellules voisines au même instant, et où s'_0 est l'état de la cellule au temps t . Il est possible d'en

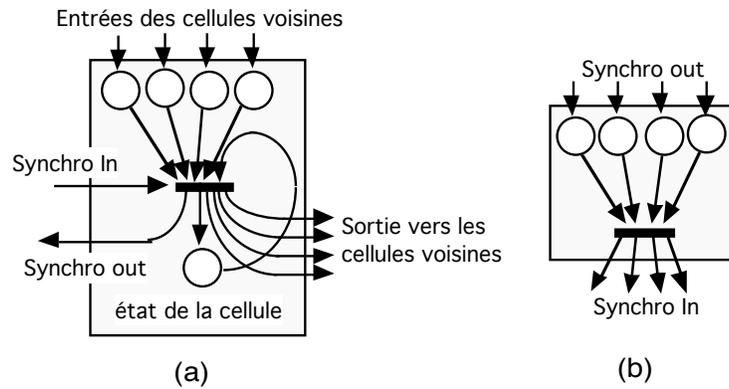


Figure 4.29: Représentation d’une cellule d’automate cellulaire (a) et de son système de synchronisation (b) en termes de réseaux de Petri

donner une image sous la forme d’un réseau de Petri, comme le montre la figure 4.29.a.

Les entrées sont fournies par les états au temps $t - 1$ des cellules voisines qui sont bufferisées pour chaque cellule (les places d’entrées du réseau servent de buffer). La fonction de transition de la cellule est décrite comme une simple transition qui modifie l’état courant de la cellule et propage ce nouvel état à toutes les cellules voisines. Pour être bien sûr que toutes les cellules fonctionnent en même temps, il faut les synchroniser. C’est à cette fonction que sont dédiées les bornes **Synchro In** et **Synchro Out**, qui respectivement proviennent et vont vers un composant de synchronisation présenté à la figure 4.29.b. Les entrées **Synchro Out** du synchronisateur sont directement liées aux sorties du même nom des cellules, et les sorties **Synchro In** du composant vont directement aux entrées de même nom de chacune des cellules. Lorsque les cellules ont fini leur transition, elles envoient un signal sur leur sortie **Synchro Out**. Ce n’est que lorsque toutes les cellules ont envoyé ce signal que la transition interne du synchronisateur peut se déclencher et envoyer un signal sur les bornes **Synchro In** des cellules, leur permettant d’effectuer leur transition.

4.6.3 Automates cellulaires et systèmes multi-agents

La question que l’on se pose souvent est de comprendre la relation qui existe entre systèmes multi-agents et automates cellulaires. Les systèmes multi-agents sont-ils des sortes d’automates cellulaires? Ou l’inverse? Malgré leur ressemblance, les automates cellulaires ne sont pas des systèmes multi-agents. Si l’environnement des SMA peut facilement être représenté sous la forme d’une grille de cellules, il n’en est pas de même des agents, qui sont des entités autonomes et peuvent se déplacer tout en conservant leur identité et surtout modifier les liens qu’ils établissent avec d’autres agents. De ce fait, les automates cellulaires peuvent être considérés soit comme des systèmes multi-agents “dégénérés” dans lesquels les agents seraient devenus fixes, soit, au contraire, comme de bons modèles d’environnement, dans lesquels

il est possible de définir avec précision des règles de propagation de signaux ou plus généralement les lois de l'univers.

Par exemple, la diffusion de potentiel par vague, comme nous l'avons vu à la section précédente, s'exprime assez facilement sous la forme d'un automate cellulaire. Le comportement d'une cellule se borne alors à prendre la valeur maximale de l'intensité d'un signal sur ces valeurs voisines et à décrémenter sa propre valeur d'une unité:

$$\text{valeur cellule courante} = \max(\text{valeurs des cellules voisines}) - 1$$

Lorsque le réseau se stabilise, les cellules ont comme valeur l'intensité du signal en ce point.

4.7 L'action comme commande

La théorie du contrôle et la cybernétique (Couffignal 1963) se sont intéressées à l'action d'un point de vue particulier en considérant le problème de la commande de procédés industriels et la réalisation de machines régulées. Le problème de l'action consiste alors à faire varier un certain nombre de paramètres d'entrées, appelés variables de *commandes*, d'un système physique pour obtenir des valeurs particulières des variables de sorties. Ainsi dans un problème de régulation, les variables de sorties doivent atteindre et garder des valeurs de consigne fixées.

Prenons l'exemple du pilotage automatique d'un bateau. Pour aller dans un endroit déterminé, le pilote d'un bateau manipule la barre de telle manière que le bateau se dirige dans la direction voulue. Cette action est obtenue en empêchant le bon déroulement des flux liquides dus à la vitesse du bateau, ce qui exerce une force sur la barre et modifie le déplacement du navire.

Mais le milieu liquide est lui-même agité de mouvements et modifie le cap du bateau, ce qui oblige le pilote à compenser la direction par le biais d'un mécanisme de rétroaction, comme le montre la figure 4.30.

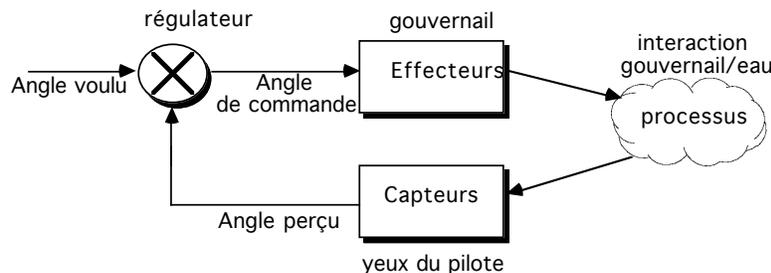


Figure 4.30: La cybernétique et en particulier la théorie de la commande, considère l'action comme une commande que l'on applique à un système pour réguler et contrôler son fonctionnement.

Ici l'action du pilote consiste donc à contrôler le système physique à partir des informations provenant de ses capteurs (ses organes de vues) et à corriger le

comportement du bateau en agissant sur les effecteurs (le gouvernail) pour aller dans la direction voulue.

L'action n'est donc plus seulement une transformation d'états du monde, ni un simple mouvement, mais une activité complexe entièrement dirigée vers un but, qui tient compte des réactions du milieu et des corrections à apporter aux actions précédentes. Cette conception de l'action comme commande est à la base de tous les systèmes de régulations et, d'une manière générale, de tout ce qui touche au contrôle de machines. La cybernétique, puis la théorie générale des systèmes, s'est d'ailleurs faite le chantre de cette conception de l'action avec les travaux de N. Wiener (Wiener 1948), de L. von Bertalanffy (Bertalanffy 1968), ou de M. D. Mesarovic et Y. Takahara (Mesarovic et Takahara 1975).

Cette conception cybernétique de l'action est évidemment très importante pour les SMA: les actions des agents doivent tenir compte des réactions du milieu et savoir y apporter les corrections nécessaires. Toute action s'effectue dans un milieu et suppose un ensemble de mécanismes de régulation de son exécution qui puisse corriger les défauts éventuels de son application. Nous verrons en particulier que les agents eux-mêmes peuvent être considérés comme des êtres cybernétiques. Mais, malgré son intérêt, la cybernétique et la vision de l'action qu'elle propose ne peuvent rendre compte de la richesse des interactions qui s'expriment entre agents. Si en dernier ressort, agir pour un agent réside dans la capacité à faire varier les paramètres de commande de ses effecteurs, cette conception ne permet pas de prendre en compte les interactions entre agents. Il lui manque notamment la possibilité de traiter la simultanéité des actions et en particulier celle d'être à même de décrire des coordination d'actions, de pouvoir répartir des tâches entre plusieurs agents, toutes choses pour lesquelles une simple technique de régulation est loin d'être suffisante.

4.8 Agents tropiques et hystérétiques

Après avoir étudié différents modèles d'action, nous pouvons passer à la description des agents et des systèmes multi-agents. L'approche empruntée consiste à partir du modèle influences/réactions de l'action, en introduisant les agents comme des entités capables de produire des influences et de distinguer parmi les états du monde ceux qui sont utiles pour leurs actions. Ces modèles d'agents sont fortement inspirés du travail de Genesereth et Nilsson (Genesereth et Nilsson 1987) qui, les premiers, ont donné une représentation algébrique de la structure et des comportements des agents. Cependant, comme ils sont restés dans une conception de l'action comme transformation d'états, ils n'ont pu donner que des modèles de systèmes mono-agents, dans lesquels un seul agent est responsable des modifications qu'il apporte à l'environnement. Au contraire, en partant de l'action comme production d'influences il sera possible d'introduire les actions simultanées et donc de présenter des modèles effectivement multi-agents.

Les agents sont des entités capables de prendre en compte ce qui les entoure, ce qui se traduit pour les agents purement situés par la capacité de *percevoir* l'environnement et d'agir en *exécutant des actions* qui tendent à modifier l'état

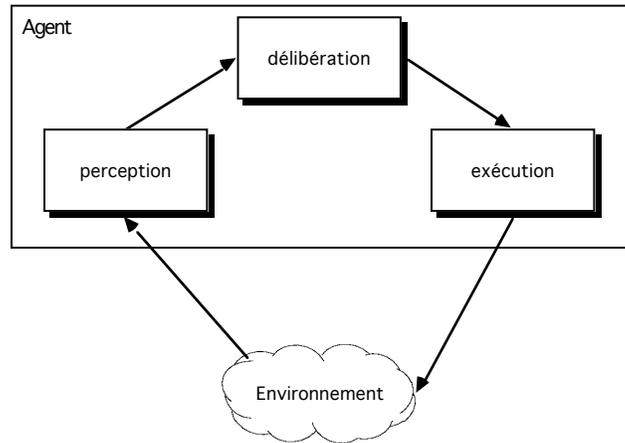


Figure 4.31: La structure perception-délibération-exécution d'un agent

du système multi-agent (soit en agissant directement sur l'environnement, soit en émettant des messages à l'adresse d'autres agents), après avoir *délibéré* de ce qu'il convenait de faire. On dit ainsi qu'un agent se compose des trois volets: perception, délibération, action, comme le montre la figure 4.31.

La perception est la qualité pour un agent de pouvoir classer et distinguer les états du monde à la fois en fonction des traits saillants de l'environnement, mais aussi en fonction des actions qu'il entreprend. De ce fait, on considérera la perception comme une fonction qui associe à l'ensemble des états du monde Σ un ensemble de valeurs que l'on appelle *percepts* ou *stimuli*⁸. Si l'on appelle P_a l'ensemble des percepts associés à un agent a , alors la fonction de perception d'un agent peut être définie ainsi:

$$Percept_a : \Sigma \rightarrow P_a$$

qui associe un percept à chaque état du monde. Les capacités d'exécution ont été traitées précédemment: elles consistent à produire des influences qui seront combinées et traitées par l'environnement à partir des lois qui le caractérise. De ce fait, l'exécution consiste à appliquer un opérateur par la fonction *Exec* définie précédemment:

$$Exec : Op \times \Sigma \rightarrow \Gamma$$

où Op et Γ décrivent respectivement l'ensemble des opérations réalisables par les agents et l'ensemble des influences produites par les agents.

Ce qui se situe entre l'entrée et la sortie, c'est-à-dire entre la perception et l'exécution, s'appelle *délibération* et constitue la partie la plus essentielle d'un agent. C'est elle en effet qui rend compte de son comportement effectif, et constitue donc la partie la plus étudiée et la plus complexe. Alors que les capacités de perception et d'exécution peuvent souvent être considérées comme de simple fonctions, il n'en

⁸On préférera *percepts* dans le cadre des agents cognitifs et *stimuli* dans celui des agents réactifs. Néanmoins, j'utiliserai indifféremment l'un ou l'autre pour désigner ce qui est perçu par un agent.

est plus de même de la délibération, car, sauf à considérer des agents extrêmement simples, c'est là que s'élaborent les objectifs, les prises de décisions, les facultés de mémorisation et, en ce qui concerne les agents cognitifs, les représentations du monde et les concepts utilisés pour décider de l'action à entreprendre.

On distinguera, dans un premier temps, deux grandes catégories d'agents: les *agents tropiques* et les *agents hystériques*⁹. Les premiers caractérisent des agents réactifs élémentaires qui sont toujours dans l'*ici et le maintenant*, sans pouvoir mémoriser quoi que ce soit, alors que les seconds qualifient des agents aux comportements aussi complexes que l'on veut et qui utilisent leur expérience passée pour anticiper sur l'avenir.

4.9 Agents tropiques

4.9.1 Approche formelle

Les agents tropiques sont des agents qui agissent de manière totalement réflexe aux états du monde. On suppose donc que les agents n'ont aucun but, ni même aucun état interne. Dans ce cas, l'exécution d'un agent est directement liée à ses perceptions, ce qui signifie que la faculté de délibération peut être représentée comme une simple fonction *Reflexe* qui associe une opération à un percept:

$$Reflexe_a : P_a \rightarrow Op$$

Un agent tropique a dans un système d'action $\langle \Sigma, \Gamma, Op, Exec, React \rangle$, où Σ , Γ , Op , $Exec$ et $React$ sont définis comme nous l'avons vu plus haut, peut donc être décrit comme un triplet:

$$a = \langle P_a, Percept_a, Reflexe_a \rangle$$

c'est-à-dire comme un ensemble de percepts, une fonction de perception et une fonction de décision qui se borne à réagir de manière réflexe aux perceptions qu'elle reçoit.

De ce fait, si l'on appelle $Comport_a$ la fonction qui représente le comportement de l'agent a , alors cette fonction combine simplement le réflexe et la perception de cet agent pour produire une nouvelle opération.

$$\begin{aligned} Comport_a : \Sigma &\rightarrow Op \\ Comport_a(\sigma) &= Reflexe_a(Percept_a(\sigma)) \end{aligned}$$

4.9.2 Un système multi-agent tropique

La dynamique d'un système multi-agent tropique (et donc situé) est donné par l'évolution de l'état de l'environnement $\sigma(t)$ en fonction du temps. Pour un agent a , l'équation de cette évolution est:

$$\sigma(t+1) = React(Exec(Comport_a(\sigma(t)), \sigma(t)), \sigma(t))$$

⁹Cette distinction vient de (Genesereth et Nilsson 1987).

L'état de l'environnement est donc le résultat de l'application des lois de l'univers aux influences produites par l'application de l'opération choisie par le comportement de l'agent. Lorsque plusieurs agents a_1, \dots, a_n agissent simultanément, il faut combiner les influences produites par chacun des agents:

$$Exec(Comport_1(\sigma(t)) || \dots || Comport_n(\sigma(t))) = \bigcup_{i=1}^n Exec(Comporti(\sigma(t)))$$

d'où on déduit la dynamique de l'état de l'environnement:

$$\sigma(t+1) = React\left(\bigcup_{i=1}^n Exec(Comporti(\sigma(t)))\right)$$

L'évolution d'un système multi-agent tropique consiste donc en une suite d'états, chacun dépendant à la fois des lois de l'univers et du comportement de chacun des agents. La figure 4.32 donne une représentation graphique d'un système multi-agent comprenant trois agents tropiques.

Nous avons donné une version discrétisée de cette évolution dans laquelle tout ce qui se passe entre l'instant t et l'instant $t+1$ est considéré comme "instantané". Il serait possible d'en donner une version continue, mais il faudrait pour cela que les opérateurs *React* et *Exec* aient une expression continue.

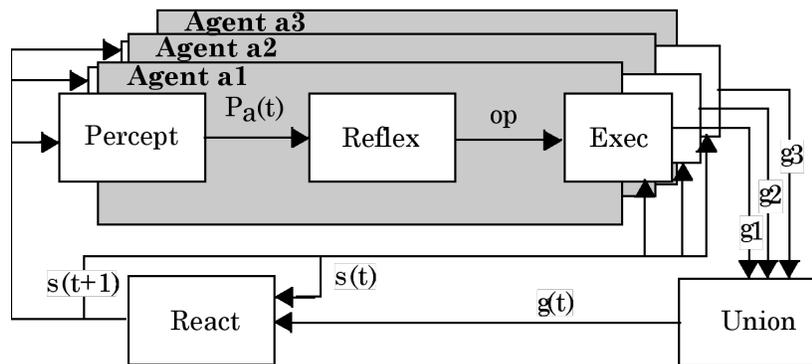


Figure 4.32: Représentation graphique d'un SMA comprenant trois agents tropiques

4.9.3 Agents tropiques et actions situées

Bien que l'on puisse a priori penser que les agents tropiques, du fait de leur simplicité, ne soient pas très utiles, tout un domaine de recherches issu initialement de la planification a pu montrer qu'il était possible de définir des agents tropiques exhibant un comportement relativement complexe en utilisant l'environnement comme repère et mémoire. On dit alors que les actions d'un agent tropique sont situées, c'est-à-dire qu'elles sont relatives à sa position et à l'état du monde qu'il perçoit. Pour cette notion, une expression a été forgée: celle d'*action située*. Introduites initialement en 1987 par Suchman (Suchman 1987), reprises par P. Agre et D. Chapman (Agre et Chapman 1987), soutenues par P. Maes (Maes 1990), et adaptées à l'univers

multi-agent par Connah et Wavish (Wavish et Connah 1990), les actions situées constituent le modèle le plus élémentaire des actions réactives. Le principe en est très simple et repose sur un schéma de type stimulus/réponse.

La perception joue un rôle essentiel dans ce modèle. C'est en effet par le biais de la perception d'indices caractéristiques situés dans l'environnement que l'action associée peut être déclenchée. Cette notion d'*indice* sert de fondement à la compréhension du déclenchement des actions situées. Un indice est un trait saillant de l'environnement utile à l'action qu'est en train de mener l'agent. Par exemple, si pour me rendre chez un ami je sais que je dois tourner à droite juste après la pharmacie qui se trouve en face du tabac, cette pharmacie devient un indice pour le déclenchement de mon action **tourner-à-droite**. Un indice est toujours relatif à l'agent et à ses actions. Il n'y a pas d'indice en soi. Par exemple, cette pharmacie ne devient un indice pour mon action de tourner à droite que parce que je suis en train de me rendre chez mon ami et elle perd son caractère d'indice une fois que j'aurai tourné à droite. De même un herbivore ne constitue pas nécessairement un indice pour un prédateur. Il ne prend son caractère d'indice que lorsque le prédateur a faim, c'est-à-dire lorsqu'il s'est engagé dans une action de chasse et qu'il considère cet herbivore comme une proie. De ce fait, les indices sont des déclencheurs d'actions relatifs à un agent. Les percepts sont alors définis comme des combinaisons d'indices significatifs pour l'accomplissement des actions.

La fonction de perception peut donc être définie comme une combinaison de fonctions de reconnaissances d'indices, où chaque fonction partielle de reconnaissance d'indice, $PerceptIndice_i$, indique si l'indice est présent ou non dans l'environnement par rapport à l'agent:

$$PerceptIndice_i : \Sigma \times Ag \rightarrow Bool$$

Bool représente l'ensemble des valeurs $\{Vrai, Faux\}$. Dans le cas de Clotaire, par exemple, la fonction qui reconnaît s'il y a ou non une pince à l'endroit où se trouve Clotaire retournera *Vrai* si la pince se trouve effectivement dans la pièce où se trouve Clotaire et *Faux* sinon. Il est possible de composer ces fonctions de reconnaissance partielle à l'aide d'opérateurs logiques tels que \wedge (et), \vee (ou) ou \neg (non). La perception est alors définie comme une combinaison de fonctions partielles de tests d'indices. Par exemple, le percept

être-dans-la-remise-et-repérer-une-pince

est obtenu si les deux fonctions de perceptions d'indice **être-dans-la-remise** et **repérer-une-pince** retournent toutes les deux la valeur *Vrai*. La classification des situations se fait donc en fonction de la position de l'agent, ce qui suppose que les positions des agents soient codées dans les états du monde.

Le comportement de délibération d'un agent peut donc être donné sous la forme d'un ensemble d'actions situées, c'est-à-dire comme un ensemble de fonctions partielles qui associent une action particulière à un percept. Pour les commodités de sa description, on représentera chaque action située comme une règle:

si <percept> alors <action>

où `<percept>` est donné sous la forme d'une combinaison logique de tests d'indices. Ces idées ont été originellement développées par Agre et Chapman dans un programme informatique, PENG1, fondé sur un jeu vidéo, PENG0, dans lequel un pingouin tente de capturer un maximum de diamants en se déplaçant dans un labyrinthe constitué de cubes de glaces et peuplé d'abeilles qui pourchassent le pingouin et tentent de le tuer en le piquant ou en lui lançant des cubes de glaces. Ces derniers peuvent en effet être poussés par les abeilles et par le pingouin. Ils se déplacent alors en ligne droite jusqu'à toucher un autre cube ou le bord de l'écran et tuent les animaux, abeilles ou pingouin, qui se trouvent sur leur passage (fig. 4.33). L'intérêt de ce jeu, comme d'ailleurs de la plupart des jeux vidéo d'action, provient de l'impossibilité de planifier à l'avance l'ensemble des actions du pingouin. Il faut réagir rapidement aux situations qui peuvent se présenter sur un mode "réflexe".

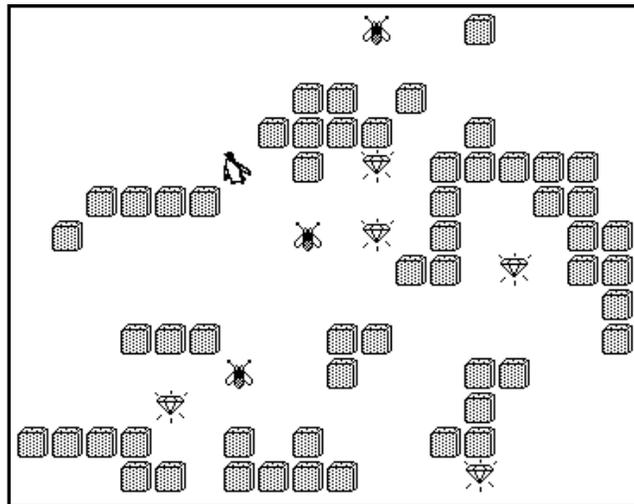


Figure 4.33: Le jeu PENG1. Le pingouin doit capturer le plus grand nombre de diamants en évitant les abeilles (extrait de (Drogoul et al. 1991)).

Dans la version d'Agre et Chapman, les actions du pingouin sont codées sous la forme d'actions situées. Par exemple, l'action consistant à essayer d'écraser une abeille peut s'exprimer sous la forme d'une règle¹⁰ de la forme:

```
Règle tuer-abeille-1
  si il y a un cube-de-glace-à-côté-de-moi
  et il y a une abeille-de-l'autre-côté-
    du-cube-de-glace-qui-est-à-côté-de-moi
  alors
    pousser le cube-de-glace-à-côté-de-moi
```

Les deux premières prémisses de la règle indiquent les prémisses déclenchantes de l'action et la conclusion précise l'action qu'il y a lieu d'activer.

¹⁰Bien qu'il n'y ait pas à proprement parler de règles dans leur système, il est plus facile de l'interpréter de cette manière.

Quoique très séduisante de par sa simplicité, l'écriture de comportements d'agents situés pose de nombreux problèmes. Le premier porte sur la cohérence. Par exemple, si le pingouin se trouve près d'un diamant et qu'il repère une abeille située de l'autre côté d'un cube de glace, doit-il prendre le diamant ou essayer d'abord de tuer l'abeille? La solution la plus simple consiste à donner un ordre de préférence entre les actions de manière à éviter les conflits éventuels. Cet ordre se traduit souvent en attribuant un coefficient de priorité aux actions ou en utilisant une architecture de subsomption (cf. chap. 3).

Un autre problème réside dans le fait que les actions situées ne considèrent ni la notion de but interne, ni celle de mémoire: un agent agit uniquement en fonction de ce qu'il perçoit ici et maintenant et il ne peut donc pas prendre en compte l'état dans lequel il se trouve. Par exemple, supposons que l'on veuille écrire le comportement de Clotaire à qui l'on demande de porter une pince de la remise à l'atelier (fig. 4.2). Une solution est évidemment de décrire son comportement à l'aide de l'ensemble des règles R1 suivant:

```

Règle chercherPince                                (R1)
  si je ne suis pas à la remise
  et je ne porte rien
  alors aller à la remise

Règle prendrePince
  si je suis à la remise
  et je ne porte rien,
  alors prendre la pince

Règle allerAtelier
  si je ne suis pas à l'atelier
  et je porte quelque chose
  alors aller à l'atelier

Règle déposerObjet
  si je suis à l'atelier
  et je porte un objet
  alors déposer ce que je porte

```

Maintenant, supposons que l'on veuille que Clotaire ramasse ce qu'il laisse tomber. Avec les règles précédentes, dès que Clotaire n'a plus rien, il retourne chercher une pince à la remise, et ne tient absolument pas compte du fait qu'il en portait une quelques instants auparavant. En revanche s'il "savait" qu'il portait une pince, il ne retournerait pas directement à la remise, mais chercherait autour de lui pour retrouver la pince. La solution consisterait à coder un état interne sous la forme d'une variable booléenne, *en-train-deramener*, et à réécrire l'ensemble des règles de la manière suivante:

```

Règle chercherPince                                (R2)

```

```

si je ne suis pas à la remise
et je ne porte rien
et je ne suis pas en-train-de-ramener
alors aller à la remise

```

Règle prendrePince

```

si je suis à la remise
et je ne porte rien,
alors prendre la pince
et me placer dans l'état en-train-de-ramener

```

Règle allerAtelier

```

si je ne suis pas à l'atelier
et je porte quelque chose
alors aller à l'atelier

```

Règle déposerObjet

```

si je suis à l'atelier
et je porte un objet
alors déposer ce que je porte
et me placer dans l'état pas en-train-de-ramener

```

Règle rechercherPince

```

si je ne porte rien
et je suis en-train-de-ramener
et je ne perçois pas la pince par terre
alors chercher autour de moi la pince

```

Règle reprendrePince

```

si je ne porte rien
et je suis en-train-de-ramener
et je perçois la pince par terre
alors prendre la pince

```

Malheureusement, les agents tropiques n'ont pas de mémoire, ce qui les prive de tout état interne! Doit-on pour autant faire le deuil des agents tropiques? Non, il suffit de faire bien attention à la manière dont on code les actions situées et de faire en sorte que l'on donne à l'agent les capacités de reprendre la pince s'il ne se trouve pas à l'atelier. Il suffit de reprendre l'ensemble de règles R1 et de lui ajouter la règle suivante pour résoudre le problème:

Règle reprendrePince

(R3)

```

si je ne porte rien
et je perçois une pince
et je ne suis pas à l'atelier
alors prendre la pince

```

Ainsi, dès qu'il perçoit une pince en dehors de l'atelier, qu'il l'ait porté juste avant ou qu'au contraire il l'ait rencontré par hasard, Clotaire reprendra la pince et donc son chemin vers l'atelier, sans nécessiter un quelconque état interne.

4.9.4 Souplesse des actions situées

L'intérêt des systèmes d'actions situées vient de leur souplesse pour accomplir des tâches et leur faculté de prendre en compte des modifications non prévues du monde. Par exemple, supposons que l'on demande d'abord à Clotaire d'apporter une pince de la remise à l'atelier comme précédemment, et que l'on veuille qu'il aille *ensuite* chercher un marteau dans le hangar. Une des possibilités consiste, en s'écartant du principe des actions situées, à donner à l'agent une variable booléenne interne, *pince-déposée*, qui, si elle prend la valeur 1 lui indique que la pince a été déposée et qu'il peut aller chercher la pince. Les règles pour aller chercher un objet sont alors les suivantes (ensemble de règles R4):

```
Règle chercherPince                                     (R4)
  si je ne suis pas à la remise
  et je ne porte rien
  et je ne suis pas en-train-de-ramener
  et non pince-déposée
  alors aller à la remise
```

```
Règle chercherMarteau
  si je ne suis pas dans le hangar
  et je ne porte rien
  et je suis pas-en-train-de-ramener
  et pince-déposée
  alors aller au hangar
```

Clotaire peut savoir ainsi s'il doit aller chercher une pince ou un marteau en tenant compte de son état interne. Cet état constitue une *représentation* du monde (cf. chap. 5): la variable *pince-déposée* représente le fait que la pince se trouve effectivement dans l'atelier. Supposons maintenant que pour une raison ou une autre, il y ait eu une erreur et que la pince n'ait pas été déposée. Clotaire se mettra en route pour aller chercher le marteau alors que la pince n'a pas encore été apportée à l'endroit voulu, ce que l'on peut considérer comme une erreur ou un "bug" dans le comportement de Clotaire. Mais cette erreur n'est pas de son fait: il se contente d'appliquer son programme à la lettre. L'erreur vient d'une incohérence entre sa représentation (la valeur de *pince-déposée* est à *Vrai*) et l'état du monde (la pince n'a pas été déposée). Or c'est justement l'intérêt des agents tropiques que de ne pas avoir de représentation du monde, et donc de pouvoir éliminer les incohérences (et celles de leur gestion) dues à une inadéquation entre la réalité et sa représentation. Pour éviter ce problème, il est possible de s'en tenir simplement à une perception en considérant l'environnement comme une mémoire. Il suffit de faire en sorte que Clotaire aille d'abord voir dans l'atelier si une pince a été déposée ou non, et de

déclencher son comportement de recherche de pince ou de marteau en fonction de cette perception. Ce que l'on obtient en codant les règles de recherche de la manière suivante (ensemble de règles R5):

```

Règle chercherObjet                                (R5)
  si je ne porte rien
  et je suis pas-en-train-de-ramener
  et je ne suis pas à l'atelier
  alors aller à l'atelier

Règle chercherPince
  si je ne porte rien et
  et je ne suis pas en-train-de-ramener
  et je suis à l'atelier
  et je ne perçois pas de pince
  alors aller à remise

Règle chercherMarteau
  si je ne porte rien et je suis pas-en-train-de-porter
  et je suis à l'atelier
  et je perçois une pince // il est temps d'aller chercher
  alors aller à hangar // le marteau

```

C'est ici l'environnement qui fait office de mémoire, en indiquant si la pince a été apportée dans l'atelier ou non, la perception de la pince déposée jouant pour Clotaire le rôle d'une variable booléenne. Le monde "connaît" son propre état, qu'il suffit de percevoir. Une représentation est donc inutile puisqu'il suffit pour Clotaire de percevoir cet état pour agir de manière cohérente, sans qu'il soit nécessaire de vérifier l'adéquation entre une représentation et le monde environnant. Cet avantage se paye par une augmentation des déplacements de Clotaire qui doit initialement aller vérifier dans l'atelier si la pince s'y trouve. Ces déplacements supplémentaires permettent néanmoins à Clotaire de s'assurer que la pince n'est pas dans l'atelier et qu'il doit effectivement aller la chercher dans la remise, ce qui peut s'avérer avantageux lorsque le monde évolue indépendamment des actions de Clotaire (par exemple, parce qu'il y a des ouvriers qui utilisent les outils transportés par Clotaire) et que les ordres que reçoit Clotaire s'avèrent incohérents. L'environnement est ainsi une mémoire formidable que les agents peuvent utiliser à leur avantage, en déposant éventuellement des marques qui leur serviront d'indices pour accomplir leurs actions ultérieures. Comme le Petit Poucet, l'environnement peut enregistrer des états et donner la possibilité à des agents ayant très peu de capacités cognitives d'accomplir des tâches intéressantes. On verra au chapitre (cf. chap. 8) que ces capacités de mémorisation de l'environnement peuvent être utilisées avec profit pour coordonner des actions collectives et amplifier ainsi les capacités naturelles des agents.

4.9.5 Les buts sont dans l'environnement

Puisqu'il ne savent agir qu'en fonction de l'état de l'environnement, on imagine les agents réactifs soit comme des sortes de marionnettes mécaniques totalement dépourvues d'intérêt, soit comme des êtres totalement lunatiques qui se laissent balloter au gré des impulsions que leur procure l'environnement. Il est difficile de comprendre que les agents tropiques situés utilisent l'environnement pour accomplir des tâches utiles et parfois relativement complexes. Comment en effet, un agent tropique serait-il capable de poursuivre un but? Les actions de déplacements de Clotaire, comme celle d'aller à la remise, étaient précédemment considérées comme atomiques: on supposait qu'elles étaient très simples et pouvaient être codées directement. Mais si Clotaire ne peut plus aller d'une pièce à une autre en une seule action, comment peut-il parvenir à la pièce indiquée? Il existe de nombreuses solutions dont certaines, qui ne nécessitent aucune mémorisation, peuvent être appliquées aux agents tropiques.

Codage en dur

La première solution consiste à coder dans le comportement de Clotaire la suite des pièces qu'il doit traverser. Par exemple pour aller de l'atelier à la remise il doit parcourir les pièces $\{1, 4, 5, 6, 9, 12\}$ et pour aller de l'atelier au hangar il doit parcourir les pièces $\{1, 4, 5, 6, 3\}$. Seulement cette technique est bien trop câblée: il faudrait définir tous les trajets possibles permettant d'aller d'une pièce à l'autre, ce qui croît exponentiellement avec le nombre de pièces. De plus, toute modification de l'environnement nécessiterait une complète reprogrammation de l'agent.

Exploration et mémorisation

La seconde utilise la technique d'exploration de labyrinthe par mémorisation des pièces traversées. Il suffit à Clotaire d'explorer l'ensemble des pièces en empilant tous les choix qu'il a effectué à un carrefour et en mémorisant toutes les pièces qu'il a déjà explorées, pour éviter de tourner en rond. Cette méthode, qui repose sur une mémorisation et donc s'écarte des actions situées, est très coûteuse, parce qu'il faudra peut être explorer toutes les pièces avant que Clotaire ne trouve la remise. Elle peut néanmoins être intéressante en permettant à des agents d'avoir des représentations de leur environnement afin qu'ils puissent optimiser leurs déplacements.

Exploration et marquage

La troisième est encore une technique d'exploration de labyrinthe, mais qui à l'inverse de la précédente utilise l'environnement pour marquer les pièces par lesquelles Clotaire est passé. Il lui suffit de déposer une marque chaque fois qu'il arrive dans une nouvelle pièce, et de systématiquement mettre une croix dans toutes les pièces où il passe. S'il tombe dans un cul-de-sac, il retourne en arrière en suivant les marques, et il enlève ces marques jusqu'à revenir à un lieu qui donne dans une pièce qu'il n'a

pas encore exploré. De cette manière, il utilise les marques pour revenir en arrière et les croix pour éviter de tourner en rond. Cette technique, qui est fondée sur les actions situées, est pourtant semblable à la précédente: les marques jouent le rôle de la pile des choix et les croix représentent la mémorisation des pièces déjà vues. Elle présente donc les mêmes inconvénients que la précédente en supposant des parcours qui croissent exponentiellement avec le nombre de pièces. En outre, plus le nombre d'issues par pièce est important, plus le nombre de choix à envisager sera conséquent et donc plus il sera difficile d'aller d'un point à un autre. Enfin, pour un agent tropique, tout nouveau déplacement requerra une nouvelle exploration sans qu'il puisse mémoriser les chemins déjà parcourus.

Hill climbing

La quatrième méthode repose sur la technique de la boussole, aussi appelée "hill climbing" par analogie avec la technique des montagnards qui grimpent en suivant les lignes de plus grande pente. Si l'on connaît la position géographique de la pièce où l'on désire se rendre, il suffit de prendre la direction qui mène le plus directement à cette pièce en "vol d'oiseau". C'est une technique évidemment très simple et qui fonctionne bien lorsque l'environnement n'est pas trop accidenté. Dans ce cas, en effet, il n'est plus nécessaire d'explorer toutes les pièces, et il suffit d'aller dans la direction indiquée par la boussole. Evidemment, comme tout le monde le sait, cette technique ne permet pas de toujours se diriger en ville à cause des sens interdits et des culs-de-sac. Il en est de même pour Clotaire, qui peut se trouver facilement bloqué dans des voies sans issues. Dans cet exemple, s'il utilise cette technique et s'engage malencontreusement dans la pièce 11, il se trouvera irrémédiablement bloqué sans pouvoir se rendre dans la remise.

Diffusion par vague

La cinquième méthode, qui pallie les inconvénients de la précédente, est une variante de celle qui a été développée pour contourner des obstacles à l'aide de champs de potentiels (cf. section 4.5). Elle est très simple et s'avère très utile pour le déplacement d'agents situés. Son principe ressemble à une diffusion de liquide dans un récipient, la source du liquide correspondant au but que l'on cherche à atteindre. Le liquide, à la différence des boussoles, épouse les obstacles et indique ensuite le chemin qu'il faut prendre pour éviter les barrières et les voies sans issues. On marque l'environnement en associant une intensité à une portion d'espace. La source prend la valeur 0 et chaque fois que l'on s'éloigne on affecte à la nouvelle portion d'espace une valeur incrémentée d'une unité par rapport à la précédente. Si elle s'avérait déjà marquée, on ne laisse que la valeur la plus faible. La figure 4.34 montre une diffusion par vague permettant à Clotaire de se rendre à la remise. Il suffit à Clotaire de suivre la ligne de plus grande pente, c'est-à-dire de prendre le chemin dont la valeur descend. Il ira ainsi de la pièce 2 (valeur 6) à la pièce 1 (valeur 5), puis il ira en 4 (valeur 4), en 5 (valeur 3), en 6 (valeur 2) en 9 (valeur 1) et enfin en 12 (valeur 0). Il prend ainsi le chemin le plus direct et évite tous les culs-de-sac.

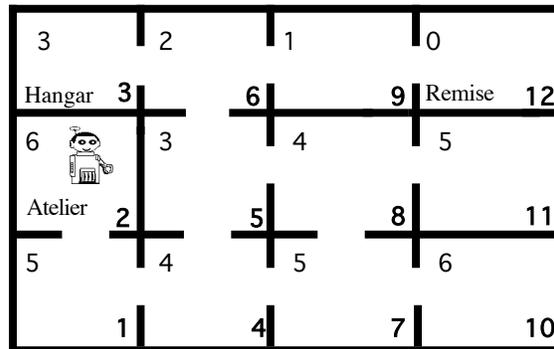


Figure 4.34: Diffusion par vague par rapport à la remise

Cette technique de marquage présente l'avantage de ne pas dépendre de l'agent et de sa position, mais seulement de celle des buts. Par exemple, si Clotaire veut aller de la pièce 10 à la remise, il suivra le même marquage. En revanche, s'il veut aller au hangar, il faudra propager un autre marquage à partir du hangar. Les déplacements de Clotaire entre l'atelier, le hangar et la remise peuvent être entièrement déterminés par 3 marquages correspondant respectivement à chacune de ces pièces. Cette méthode est donc particulièrement intéressante pour déplacer un agent tropique dans un espace quelconque, et a été utilisée dans de nombreuses applications (Barraquand et Latombe 1989; Drogoul et al. 1991). Son inconvénient majeur réside dans le marquage, dont la complexité est proportionnelle au nombre de portions d'espace envisagé. Lorsque les buts sont statiques cela ne pose pas de problème, mais lorsqu'il s'agit de simuler des situations plus dynamiques, la propagation des marques devient certainement l'activité la plus coûteuse du fonctionnement d'un système multi-agent de simulation. Enfin, à cause des phénomènes de propagation très spécifiques des signaux dans la nature (rebond des signaux, bruit, variations d'intensités, etc.), de la limitation des capteurs, il n'est pas toujours possible d'utiliser cette technique dans le cas de robots situés dans le monde réel.

Nous avons présenté précédemment le jeu PENGUIN et son traitement par le biais de routines déclenchées par des indices de l'environnement. En fait, il s'est avéré à la fois plus simple, plus performant et plus immédiat d'implémenter PENGUIN à l'aide d'une technique de diffusion par vague (Drogoul et al. 1991). Le pingouin est attiré par les diamants et repoussé par les abeilles. Bien que les comportements des agents du monde de PENGUIN soient élémentaires, le pingouin est capable de jouer tout à fait correctement, comparé à un joueur humain. On pourrait néanmoins penser qu'un modèle fondé sur des comportements locaux, sans aucune planification, a ses limites (fig. 4.35.1).

Dans cet univers particulier, le pingouin ne doit pas pousser les cubes droit devant lui, sous peine de créer des murs infranchissables (fig. 4.35.2) (il ne peut pousser qu'un cube à la fois). Avec un comportement simpliste, il ne peut atteindre le diamant. La solution est de le doter d'un comportement d'agression envers les cubes un peu plus sophistiqué : il ne doit pas pousser de cube dans la direction qu'il emprunte pour rejoindre un diamant, mais de manière perpendiculaire au gradient

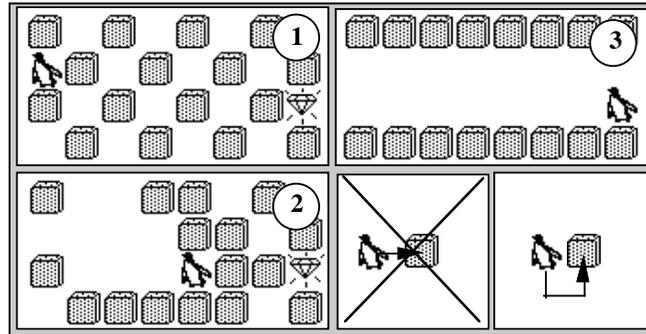


Figure 4.35: Le pingouin peut se tirer de situations très complexes (1) à condition qu’il évite de pousser les cubes droit devant lui (2). Pour cela, il lui suffit de pousser les cubes perpendiculairement à la direction de son but (3).

attractif¹¹. Ceci complique un peu la définition de son comportement mais permet de résoudre le problème sans planification (fig. 4.35.3).

Cartographie

La sixième et dernière méthode consiste à donner à Clotaire le plan des lieux afin qu’il détermine lui-même, sans se déplacer, le chemin à entreprendre. Il suffit alors pour Clotaire de planifier son trajet en utilisant une technique de planification à la STRIPS ou l’une des méthodes précédentes. La carte est alors employée comme une sorte de monde virtuel dans lequel Clotaire peut anticiper l’ensemble des actions à réaliser avant de les accomplir effectivement. Evidemment, cela suppose que Clotaire dispose de capacités cognitives importantes pour effectuer cette activité de planification, ce qui est impossible dans le cas d’agents tropiques.

Quelle technique pour les agents tropiques?

De fait, seules les techniques utilisant le codage en dur, l’exploration et le marquage, le “hill climbing” et la diffusion par vagues peuvent s’appliquer à des agents tropiques. Elles montrent qu’il est possible de donner un but à un agent à condition que ce but puisse être défini dans l’environnement. En effet, donner un but à un agent, c’est souvent définir dans l’environnement les objets ou les lieux auxquels il doit s’attacher. Ces méthodes utilisent généralement des techniques de marquage de l’environnement, ce qui permet aux agents de s’y retrouver et d’accomplir les buts qui leur ont été assignés.

Il faut mettre au crédit des travaux portant sur les agents réactifs d’avoir mis en évidence l’importance de l’environnement pour diriger l’action avec des capacités “psychiques” très réduites. Certaines facultés considérées comme essentiellement cognitives telles que la mémoire ou le comportement apparemment intentionnel peuvent aisément être expliquées à partir d’indices situés dans l’environnement,

¹¹Cf. le chapitre 8 pour une utilisation semblable de forces perpendiculaires aux champs de potentiels dans le cadre de l’évitement de collision.

et indiquent qu'il est possible d'accomplir des tâches complexes avec très peu de compétences cognitives si l'environnement est lui-même suffisamment complexes pour qu'un agent simple puisse y trouver les nombreux indices dont il a besoin pour guider son action¹².

4.10 Agents hystérétiques

Malgré leur intérêt, les agents tropiques sont tout de même relativement limités. Si l'on veut que les agents puissent acquérir personnellement une expérience ou qu'ils soient capables de résoudre des problèmes sans avoir à marquer l'environnement, il faut leur donner la capacité de conserver des informations, et donc d'avoir de la mémoire. On les appelle *agents hystérétiques*.

4.10.1 Approche formelle

Pour un agent, mémoriser une information, ou si l'on préfère, agir en fonction non seulement des perceptions mais aussi des expériences passées suppose que les agents hystérétiques peuvent être caractérisés par un état interne s_a appartenant à S_a l'ensemble des états interne d'un agent a . Mémoriser des informations ou une expérience, c'est alors passer d'un état interne à un autre. La capacité de délibération ne peut plus être décrite comme une simple fonction *Reflexe*, mais comme deux fonctions, celle de mémorisation et celle de décision. La fonction de mémorisation, *Mem*, prend un percept, un état interne et retourne un nouvel état interne, alors que la fonction de décision, *Decision*, prend un percept, un état interne et décide de l'opération à exécuter. Ces deux fonctions peuvent être décrites ainsi:

$$\begin{aligned} Mem_a &: P_a \times S_a \rightarrow S_a \\ Decision_a &: P_a \times S_a \rightarrow Op_a \end{aligned}$$

La perception peut alors être définie, à l'instar des agents tropiques, comme une fonction qui produit des percepts à partir des seuls états de l'environnement. Il est aussi possible de considérer que la perception dépend aussi des états internes de l'agent, par exemple pour caractériser le phénomène de la focalisation de la perception. Dans ce cas, la fonction *Percept* prend deux arguments, un état de l'environnement et un état interne:

$$Percept_a : \Sigma \times S_a \rightarrow P_a$$

Un agent hystérétique a , dans un système d'action $\langle \Sigma, \Gamma, Op, Exec, React \rangle$, peut donc être défini comme un n-uplet:

$$a = \langle P_a, S_a, Percept_a, Mem_a, Decision_a \rangle$$

¹²Cette notion est très courante en éthologie (McFarland 1990).

Le comportement est alors plus complexe que dans le cas d'un agent tropique, puisqu'il s'agit d'une fonction qui associe à un couple <état de l'environnement, état interne> un autre couple <ensemble d'influences, nouvel état interne>:

$$\begin{aligned} \text{Comport}_a &: \Sigma \times S_a \rightarrow \Gamma \times S_a \\ \text{Comport}_a(\sigma, s) &= \langle \text{Decision}_a(p, s), \text{Mem}_a(p, s) \rangle \\ &\text{avec } p = \text{Percept}_a(\sigma) \end{aligned}$$

4.10.2 Un système multi-agent hystérétique

Le modèle des multi-agents hystérétiques situés en découle simplement. Un système multi-agent composé d'agents hystérétiques est défini par le n-uplet:

$$\langle \text{Ag}, \Sigma, \Gamma, \text{Op}, \text{Exec}, \text{React}, \text{Cycle} \rangle$$

où Σ , Γ , Op , Exec et React sont définis comme précédemment, et où Ag est un ensemble d'agents hystérétiques, chaque agent a_i étant défini ainsi:

$$a_i = \langle P_i, S_i, \text{Percept}_i, \text{Decision}_i, \text{Mem}_i \rangle$$

La dynamique d'un système multi-agent comprenant des agents hystérétiques est définie par $n + 1$ équations: une première équation décrit l'état de l'environnement en fonction du temps et des comportements de chaque agent et les n autres correspondent aux modifications dans les états internes des agents.

$$\begin{aligned} \sigma(t+1) &= \text{React}\left(\bigcup_{i=1}^n \text{Exec}(\text{Decision}_i(p_i(t), s_i(t)))\right) \\ s_1(t+1) &= \text{Mem}_1(p_1(t), s_1(t)) \\ &\dots \\ s_n(t+1) &= \text{Mem}_n(p_n(t), p_n(t)) \\ &\text{avec } p_i(t) = \text{Percept}_i(\sigma(t)) \end{aligned}$$

Le problème consiste donc à déterminer l'ensemble des états internes d'un agent et à décrire les deux fonctions de décision et de mémorisation de telle manière que l'agent se comportent comme l'ait voulu le concepteur et que les phénomènes collectifs désirés se mettent bien en place. La figure 4.36 montre la représentation graphique d'un SMA comprenant trois agents hystérétiques.

4.10.3 Modélisation d'agents hystérétiques par automates

Pour définir les deux fonctions de mémorisation et de décision, le plus simple consiste certainement à faire appel à la notion d'automate à états finis que nous avons présenté section 4.4.1. On identifie alors l'ensemble des états internes d'un agent à celui des états de l'automate, et les fonctions de mémorisation et de décision respectivement aux fonctions ψ et ϕ de ce dernier. Les événements conduisant au changement d'état sont donnés par les percepts issus de la fonction de perception.

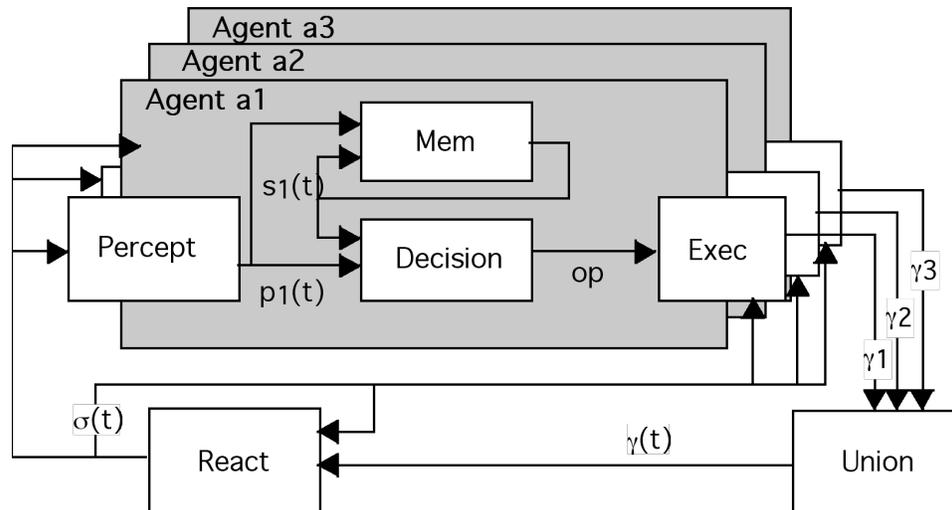


Figure 4.36: Représentation graphique de la formalisation d'un SMA comprenant trois agents hystériques (à mémoire)

Prenons l'exemple d'un robot explorateur qui doit chercher du minerai et le rapporter à sa base. Le robot explore d'abord son environnement dès qu'il a reçu l'autorisation de partir. S'il aperçoit du minerai, il se dirige vers le lieu où se trouve le minerai. Puis il remplit sa benne jusqu'à ce qu'elle soit pleine ou qu'il n'y ait plus de minerai. Lorsqu'il a terminé ce remplissage il se dirige alors vers sa base. La figure 4.37 représente graphiquement l'automate associé à cet agent. On a représenté les percepts sous la forme d'un vecteur de booléens comprenant les indices suivants:

`<autorisation-partir, minerai-en-vue, sur-lieu-minerai,
à-la-base, benne-pleine, benne-vide>`

Par exemple, si l'agent aperçoit du minerai et qu'il a encore la benne vide, son percept sera:

`<0,1,0,0,0,1>`

Sur la figure, les percepts sont représentés par les indices ayant une valeur non nulle. L'ensemble des états internes S_a de l'agent a est égal à:

$S_a = \{\text{Prêt à partir, en exploration, pistage, chargement, retour, déchargement}\}$

et les opérations qu'il peut réaliser sont:

$Op_a = \{\text{explorer, pister, charger, retourner, décharger}\}$

Les automates à états finis, malgré leur intérêt, sont relativement limités car ils ne savent pas mémoriser des valeurs autres que booléennes. Les capacités de mémorisation de tels agents sont donc contraintes par le nombre d'états discrets que l'on sait décrire a priori, toute l'information mémorisée se trouvant dans les états

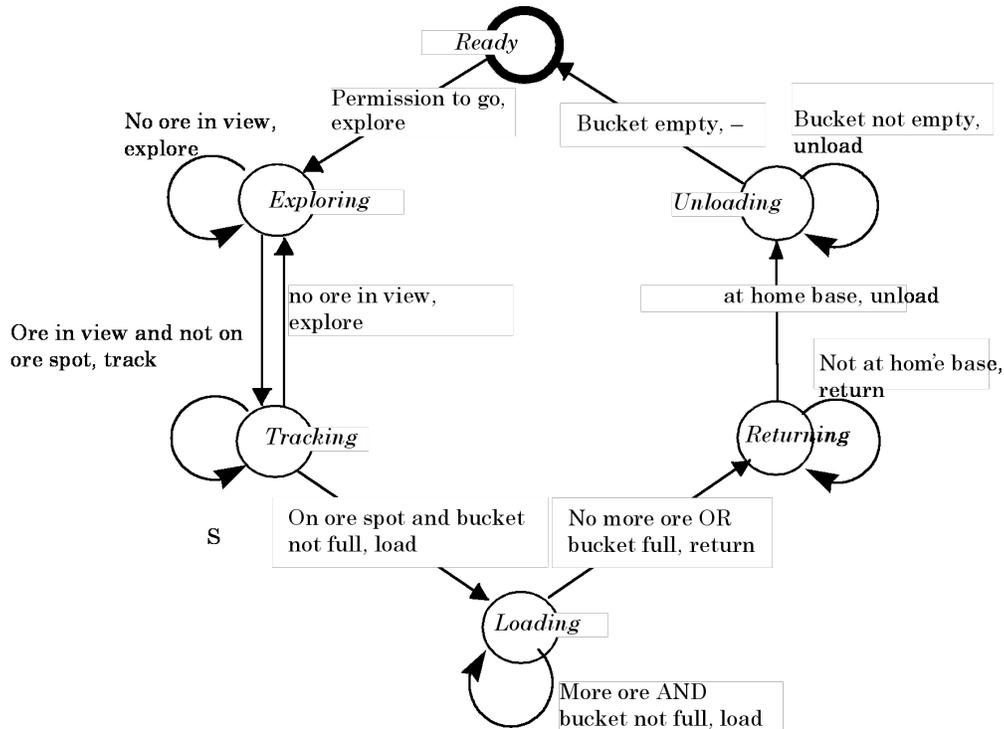


Figure 4.37: Le comportement d'un agent explorateur à mémoire représenté sous la forme d'un automate à état fini

de l'automate. Cette limite entraîne des conséquences non négligeables: l'agent ne peut facilement compter les objets qu'il voit, il ne peut mémoriser les chemins qu'il a pris, il ne peut pas savoir quel est son niveau d'énergie interne et donc réagir de manière à récupérer de l'énergie d'une manière ou d'une autre. Même la position de l'agent doit être donnée comme une information de perception, puisque l'agent est pratiquement incapable de mémoriser ce type d'information. Pour résoudre ces problèmes, il suffit d'introduire des registres et donc de modéliser un agent à l'aide d'automates à registres (cf. section 4.4.2). Néanmoins, ces modélisations ne permettent pas de décrire des agents qui font plusieurs choses en même temps. Par exemple, à l'aide d'un système à base d'automates à états finis, il est pratiquement impossible de représenter un robot explorateur qui communique avec d'autres agents, ou qui planifie ses actions pendant qu'il exécute les plans déjà établis. Pour cela il faut utiliser un formalisme plus puissant, tel que BRIC qui intègre la notion de réseaux de Petri.

4.11 Modélisation de SMA en BRIC

4.11.1 Décrire des SMA à l'aide de composants

Le formalisme BRIC (Basic Representation of Interactive Components), que j'ai développé, a pour fonction de donner une représentation opératoire du fonction-

nement d'un agent et d'un système multi-agent à partir d'une approche componentielle. La métaphore de BRIC est celle des circuits électroniques que l'on peut composer relativement facilement les uns avec les autres. Comme des composants électroniques, les composants BRIC s'accrochent les uns aux autres à l'aide de bornes et de "fils" le long desquels transitent des messages. Le formalisme BRIC est constitué de deux langages de type L3 (cf. chap. 1).

1. Un langage graphique de haut niveau décrit la structure modulaire des composants BRIC. Les agents et l'environnement, mais aussi les composants internes d'un agent sont décrits sous la forme de "briques" connectées les unes aux autres, chaque "brique" étant définie par ses interfaces et ses composants. C'est ce langage qui, en s'inspirant de l'électronique, facilite la construction de composants complexes par assemblage de composants plus élémentaires.
2. Le comportement des composants de base est décrit à l'aide d'un formalisme de plus bas niveau qui utilise les réseaux de Petri colorés, lesquels sont bien adaptés à l'expression du parallélisme.

Une présentation plus complète de ce formalisme est donnée en annexe. Nous n'en donnerons ici qu'un aperçu rapide. Chaque composant BRIC peut être considéré comme un module, c'est-à-dire une entité logicielle définie par l'ensemble des opérations qu'elle peut accomplir, et qui sont figurées sous la forme de bornes d'entrées/sorties (fig. 4.38). Les composants interagissent par le biais de liens de communication établis entre leurs bornes, une borne de sortie étant liée à une (ou plusieurs) bornes d'entrées d'un autre composant. Les composants sont des modules opaques: il n'est pas possible d'aller directement modifier la structure interne d'un composant. De ce fait, les composants sont tous indépendants les uns des autres, et le comportement d'un composant ne peut pas être altéré par celui d'un autre. En termes de génie logiciel, on dit que leur structure interne est "encapsulée" par les messages qu'on peut leur envoyer. Le fonctionnement des composants s'effectue de manière événementielle. Ce sont les messages véhiculés le long des lignes de communication qui déclenchent leur activité. De ce fait, chaque composant BRIC agit en parallèle avec les autres agents, le parallélisme étant vu comme une activité normale du fonctionnement d'un agent, qu'il s'agisse d'un agent réactif ou cognitif.

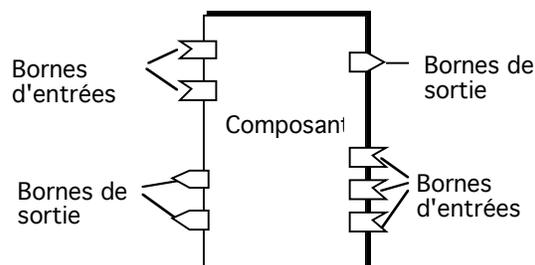


Figure 4.38: Structure d'un composant et de ses bornes

Le comportement de ces composants peut être défini soit par l'assemblage d'autres composants dont le comportement est déjà spécifié et l'on parle alors

de *composant composite*, soit directement par l'intermédiaire d'un réseau de Petri associé. Dans ce dernier cas, on dira qu'il s'agit de *composants élémentaires*. Enfin, il est aussi possible d'introduire des *composants primitifs*, dont le comportement n'est pas décrit par un réseau de Petri, mais dont l'intérêt est de prendre en charge des fonctionnalités externes à l'agent ou au système. Par exemple, les activités de base d'un agent mobile (par exemple, aller tout droit, tourner à gauche, suivre un gradient...) peuvent être décrites sous la forme de composants primitifs. Les composants BRIC peuvent être utilisés pour modéliser des agents et des systèmes multi-agents. L'intérêt de l'approche est d'être récursive: un agent est un composant BRIC, défini lui-même en termes de composants plus élémentaires. Un SMA est alors décrit comme un ensemble de composants interconnectés. Le reste de cet ouvrage fait un emploi constant de ce formalisme qui constitue le langage principal de description opératoire des SMA. Nous allons aborder le cadre général de modélisation de systèmes multi-agents en BRIC, laissant aux chapitres suivants le soin d'en développer leurs particularités. Nous modéliserons des SMA mettant en œuvre trois types d'agents: des agents purement communicants, puis des agents purement situés avant de finir par des agents communicants et situés.

4.11.2 Modélisation de SMA purement communicants

Modélisation d'agents purement communicants

Les agents purement communicants peuvent être décrits de deux manières différentes (fig. 4.39). Dans la première (a) ils sont représentés sous la forme d'un composant ne comportant qu'une seule borne d'entrée, le récepteur des messages, et qu'une seule borne de sortie, le système d'envoi des messages. Le module de réception se borne à conserver les messages dans sa file d'attente et à les distribuer aux modules internes, représentant les fonctions de mémorisation et de décision. Inversement le module d'envoi des messages se charge de la collecte de toutes les demandes et informations provenant des modules internes et les envoie aux agents concernés. Dans la seconde (b), les modules de réception et d'envoi de messages ont disparu. Bien que cette deuxième forme soit moins précise, puisque les modules de communication ont disparu, nous l'utiliserons souvent à cause de sa simplicité.

Modélisation de l'unité d'acheminement des messages

Les communications s'effectuent par l'intermédiaire d'une unité d'acheminement des messages qui joue le rôle d'une poste en se chargeant de la collecte et de la distribution des messages entre agents. Les messages externes envoyés par un agent à un autre agent sont donnés sous la forme:

$$A \ll M(x_1, \dots, x_k)$$

où A est le destinataire, M est l'en-tête décrivant le type du message (cf. chap. 6) et les x_i sont les arguments du message.

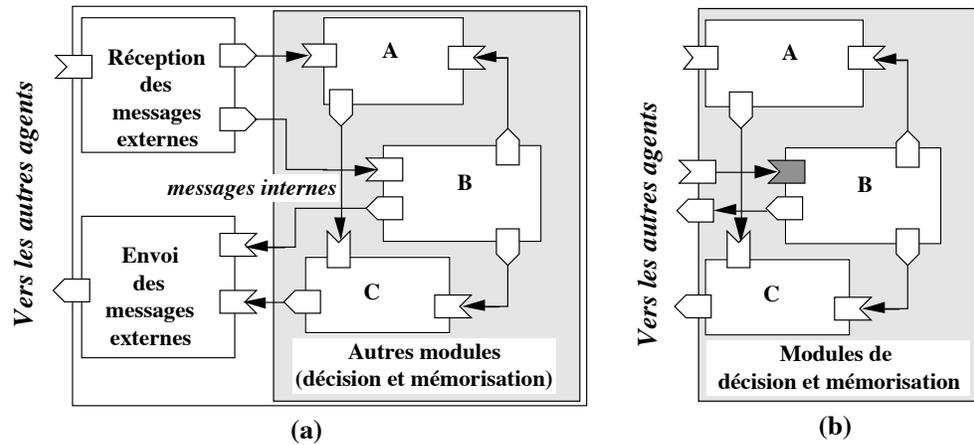


Figure 4.39: Représentation en BRIC d'agents purement communicants

Le module d'acheminement des messages est très simple, puisqu'il se limite à récupérer les messages de tous les agents et à les envoyer aux agents concernés à partir de l'adresse du destinataire, comme le montre la figure 4.40.

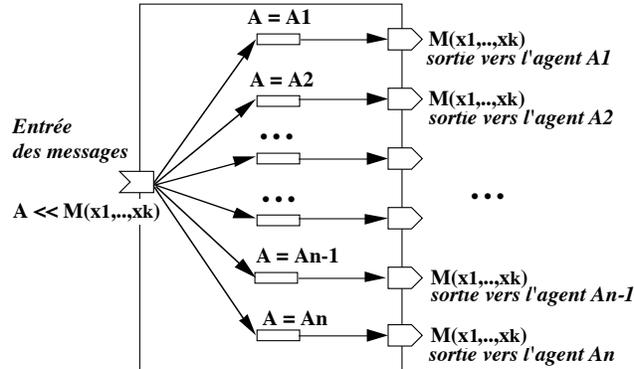


Figure 4.40: Le module d'acheminement des messages

Modélisation des systèmes multi-agents communicants

Il est alors possible d'en déduire la structure d'un système multi-agent communicant. Chaque agent (en utilisant la représentation (a)) voit sa borne de sortie connectée à la borne d'entrée des messages du modules d'acheminement, et sa borne d'entrée reliée à la borne de sortie correspondante de ce module, comme le montre la figure 4.41.

4.11.3 Modélisation des environnements

Avant de passer à la modélisation d'agents situés, il faut d'abord passer par un problème clé, l'environnement. Celui-ci constitue en effet une partie essentielle des systèmes multi-agents situés. Malheureusement, très peu de travaux ont été

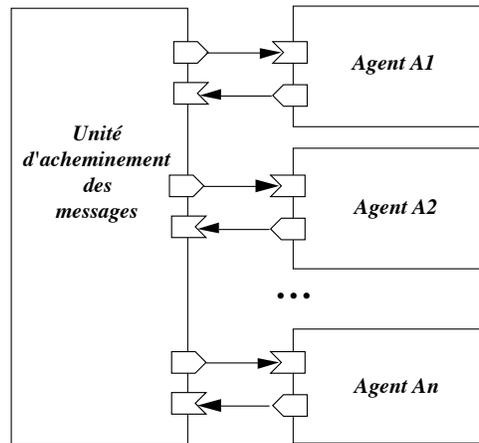


Figure 4.41: L'architecture d'un système multi-agent communicant

consacrés à leur modélisation, et les exposés portant sur l'environnement sont en général perdus dans les explications des systèmes les ayant implémentés, voire totalement noyées dans le code de leur implémentation. C'est pourquoi il nous a paru important de bien dégager les différents modèles d'environnement possibles et d'en présenter une modélisation précise en nous plaçant dans une conception de l'action par influences.

Il est possible de représenter un environnement de deux manières: soit en le considérant comme un bloc monolithique, et l'on parlera alors d'*environnement centralisé*, soit en le modélisant comme un ensemble de cellules réunies en réseau pour former une sorte d'automate cellulaire et constituer ainsi un *environnement distribué*.

Dans un environnement centralisé, tous les agents ont accès à la même structure. Ils produisent des influences, qui peuvent s'exprimer comme des demandes d'actions à l'environnement, et celui-ci répond en renvoyant les éléments consommés ou perçus comme le montre la figure 4.42.

Les environnements centralisés sont de loin les plus utilisés dans les systèmes où la propagation de signaux n'est pas trop complexe, c'est-à-dire des systèmes dans lesquels le nombre de stimuli possibles est réduit. Les problèmes de poursuites (proies/prédateurs) comme ceux relatifs à la représentation de robots mobiles (avions, véhicules terrestres, bateaux) évoluant dans un environnement ne permettant qu'une détection optique ou électromagnétique sont généralement modélisés à l'aide d'environnements centralisés. Par exemple, les systèmes MACE (Gasser et al. 1987), MICE (Durfee et Montgomery 1989), CRAASH (Zeghal et Ferber 1993), PANDORA II (Maruichi 1989), MAGES (Bouron 1991; Bouron et al. 1990) utilisent de telles techniques. Au contraire, un environnement distribué est constitué d'un assemblage de cellules disposées en réseau, à la manière des automates cellulaires. Chaque cellule se comporte comme un environnement centralisé en miniature, c'est-à-dire qu'elle gère les influences des agents qui sont localisés sur cette cellule, et envoie les marques consommées aux agents qui le demandent (fig. 4.43). Cette technique est plutôt utilisée dans les SMA qui font une décomposition de l'environnement en

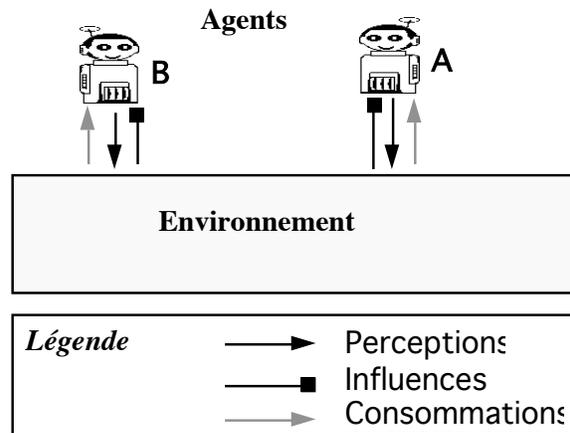


Figure 4.42: Représentation schématique d'un environnement centralisé. Les agents peuvent percevoir ce qui se passe autour d'eux dans l'environnement; ils peuvent consommer des éléments de l'environnement et influencer son déroulement en accomplissant des actions.

petits espaces connectés dans les jeux de type “war game”.

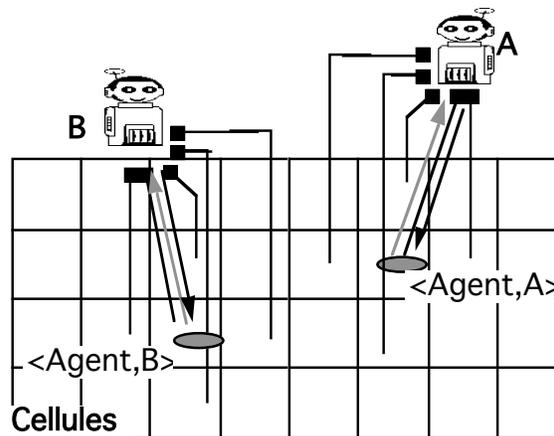


Figure 4.43: Représentation d'un environnement distribué

Les différences essentielles qui séparent les environnements distribués de leurs homologues centralisés sont les suivantes:

1. L'état de la cellule dépend des autres cellules qui l'environnent et donc aussi des influences produites par les agents dans les cellules voisines.
2. La perception des agents s'étend généralement au-delà d'une cellule, ce qui fait qu'il n'est pas possible d'envoyer à l'agent uniquement les informations de l'état de la cellule sur laquelle il se trouve.
3. Les agents se déplacent de cellule en cellule, ce qui fait qu'il faut gérer les liens que les agents entretiennent avec une cellule particulière.

4. Des signaux peuvent se propager de cellule en cellule. Cette propagation prend “un certain temps” et il faut alors synchroniser le déplacement des signaux avec les mouvements des agents.

On ne traitera ici que de la modélisation d’environnements centralisés.

Modélisation d’un environnement centralisé

Puisque les actions peuvent s’accomplir en parallèle, il faut disposer d’un formalisme capable de supporter ce parallélisme. BRIC est un de ceux-là, et je l’utiliserai pour présenter les différents modèles d’environnements. Néanmoins, ce qui est dit ici est indépendant du formalisme utilisé et peut être généralisé à d’autres représentations. Il existe essentiellement deux manières de modéliser un environnement:

- Soit on modélise l’environnement de manière abstraite, indépendamment des influences qu’il peut subir.
- Soit on tend au contraire à particulariser ses caractéristiques en le spécialisant pour un ensemble d’actions bien défini.

Pour modéliser l’environnement, il faut d’abord savoir comment représenter ses différents états $\sigma(t)$ et, en particulier, comment représenter la manière dont sont symbolisés les objets, les agents et leur positionnement dans l’environnement. Pour cela, j’utiliserai une technique classique semblable à celle qui avait été présentée avec les représentations d’action à la STRIPS: les objets et les agents sont donnés sous la forme de constantes et les relations qui existent entre ces objets et ces agents sont représentées par un ensemble de formules prédicatives que l’on appellera *faits*. Les variables seront notées par des minuscules et le nom des constantes sera précédé d’une minuscule. On supposera qu’il existe pour chaque agent un fait de la forme **agent(a,p)** où **a** représente l’identificateur de l’agent (qui doit être unique dans le système) et **p** la position de l’agent (définie par exemple sous la forme de coordonnées cartésiennes dans le plan ou dans l’espace). Par exemple, l’environnement de la figure 4.34 peut être traduit par la formule suivante où **sig(R,x,y)** représente le niveau **x** du signal de la remise dans la pièce **y**:

```
{chemin(1,2), chemin(1,4), chemin(4,5), chemin(5,6), etc.
  sig(R,3,3), sig(R,5,1), sig(R,6,2), sig(R,0,Remise),
  sig(R,3,Hangar), etc., ..., agent(Clotaire,2)}
```

Environnement généralisé

Les environnements centralisés peuvent être représentés en BRIC sous la forme d’un composant qui comprend deux entrées et trois sorties (fig. 4.44): l’entrée marquée *Influences* reçoit les influences produites par les agents. Celles-ci sont combinées avec l’état de l’environnement (place P_1) à l’aide de deux transitions qui représentent les réactions de l’environnement. T_1 correspond à des réactions en présence d’influences des agents et T_2 à ce qui se passe lorsque l’environnement est laissé à lui-même. Les sorties *Consommations* et *Perception* correspondent

respectivement à ce qui est consommé (prise d'un objet ou absorption de nourriture par exemple) et perçu par les agents lors de leurs actions. Des entrées/sorties de synchronisation (*Sync Réactions* et *Fin Réactions*) servent, comme pour les automates cellulaires (cf section 4.6.1), à synchroniser le comportement des agents et les réactions de l'environnement. La place P_2 sert uniquement de délai temporel pour faire en sorte que les agents perçoivent bien le nouvel état de l'environnement.

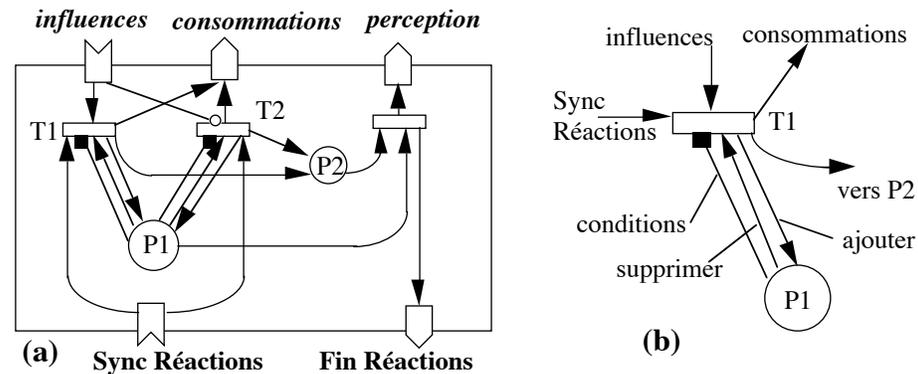


Figure 4.44: Représentation d'un environnement sous la forme d'un composant BRIC (a). La place P_1 représente l'état de l'environnement. Les influences des agents sont composées et traitées en fonction de l'état courant par les transitions (b) qui décrivent les réactions avec (T_1) et sans (T_2) influences. Les sorties *Consommation* et *Perception* représentent les éléments du monde que les agents peuvent percevoir ou consommer.

Les lois de l'univers, c'est-à-dire la définition des transitions T_1 et T_2 , décrivent les réactions de l'environnement avec et sans influences. Notons que ces lois peuvent être définies à l'aide d'un langage de description quelconque.

Environnement spécialisé

Il est possible, notamment, de représenter chaque loi sous la forme d'une transition, et d'associer une borne à chaque type d'influence. Dans ce cas, il est possible de décrire entièrement en BRIC le comportement d'un environnement, et on parle alors d'*environnement spécialisé*. Par exemple, supposons qu'un agent ne puisse que se déplacer, prendre ou déposer un objet dans l'environnement. Cela revient à décrire trois bornes d'influences, une pour le déplacement, une pour la prise et une pour le dépôt d'objet, les bornes de consommation et de perception restant inchangées, comme le montre la figure 4.45.

Implémentation d'un environnement centralisé spécialisé

Nous avons vu la représentation d'un environnement sous la forme d'un composant BRIC, mais qu'en est-il de son implémentation dans un langage de programmation? Il suffit de reprendre les modèles que nous avons donnés précédemment et de transformer chaque composant sous la forme d'un module (pour une implémentation dans un langage classique) ou d'une classe (pour une implémentation dans un langage objet).

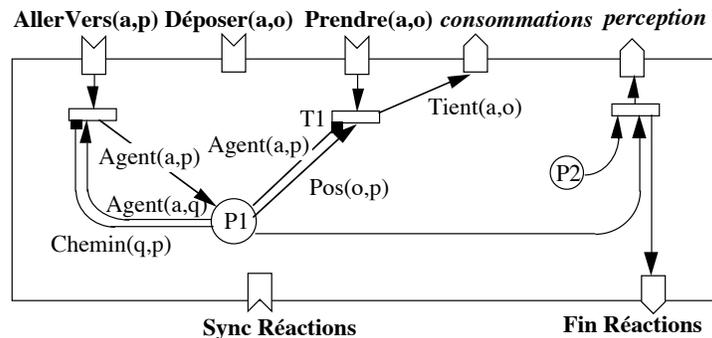


Figure 4.45: Modélisation d'un environnement spécialisé. Par soucis de clarté, on n'a pas fait figurer les mécanismes de synchronisation.

Dans l'approche centralisée, l'environnement est implémenté sous la forme d'une structure de donnée globale partagée par tous les agents et qui contient toutes les informations relatives au positionnement des agents et des autres objets dans l'espace. Eventuellement, d'autres propriétés pouvant modifier la perception des agents, telles que la visibilité des éléments de perception, peuvent y être associées. La figure 4.46 présente la structure d'un environnement centralisé, qui comporte deux parties: une partie comprend un dictionnaire qui contient la description et le positionnement de chacun des éléments de l'environnement, c'est-à-dire la position des agents et des objets, et la seconde contient les procédures ou méthodes d'accès à cet environnement.

On retrouve directement les principes de modélisation que nous avons utilisés précédemment: le dictionnaire des éléments représente en fait l'état de l'environnement, et les procédures d'accès correspondent, sur le plan de l'implémentation informatique, aux entrées du modèle spécialisé, utilisé par les agents pour envoyer des influences et modifier l'environnement. C'est donc ces procédures qui se chargeront de l'implémentation des réactions de l'environnement.

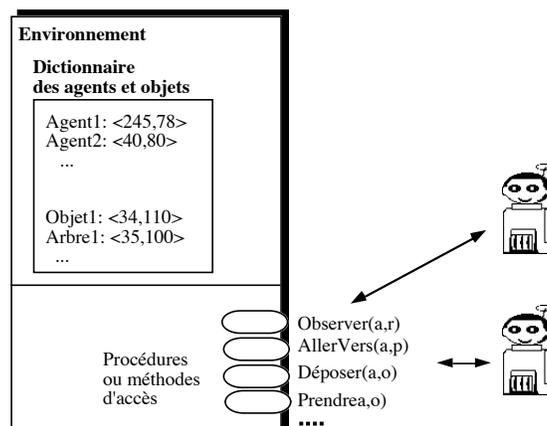


Figure 4.46: Implémentation d'un environnement centralisé

On retrouve les procédures définies précédemment pour notre modèle spécia-

lisé de l'environnement. Outre les procédures `allerVers`, `déposer` et `prendre` qui correspondent aux entrées déjà examinées, la procédure d'entrée `observer` sert pour les agents à obtenir des informations sur les éléments qui leur sont voisins, dans un rayon r autour d'eux. Cette procédure retourne la liste des objets et des agents ainsi que leurs positions à l'agent ayant effectué cette demande.

4.11.4 Modélisation d'un SMA situé

L'environnement étant décrit, il est maintenant possible de donner une description d'un système multi-agent purement situé. Commençons par les agents. De la même manière que pour les environnements, il est possible de donner un modèle généralisé des agents ou bien de donner un modèle spécialisé.

Un agent peut être décrit comme un composant comprenant deux entrées, les perceptions et les consommations. Un modèle d'agent général ne comprend qu'une sortie correspondant aux influences produites (fig. 4.47.a). Au contraire, dans une modélisation spécialisée, on représentera chaque type d'action sous la forme d'une borne de sortie, comme le montre la figure 4.47.b. Les perceptions sont filtrées par la transition de perception qui ne consomme pas ses marques d'entrées (en effet, on considère classiquement que percevoir ne modifie pas le monde perçu). Un agent ne perçoit le monde que lorsqu'il reçoit une synchronisation de perception. La réception de ces *percepts* déclenchent un traitement et produit des influences, lesquelles sont envoyées vers l'environnement. Lorsqu'il a terminé ce travail, l'agent envoie un signal de fin d'action vers le synchronisateur global du SMA.

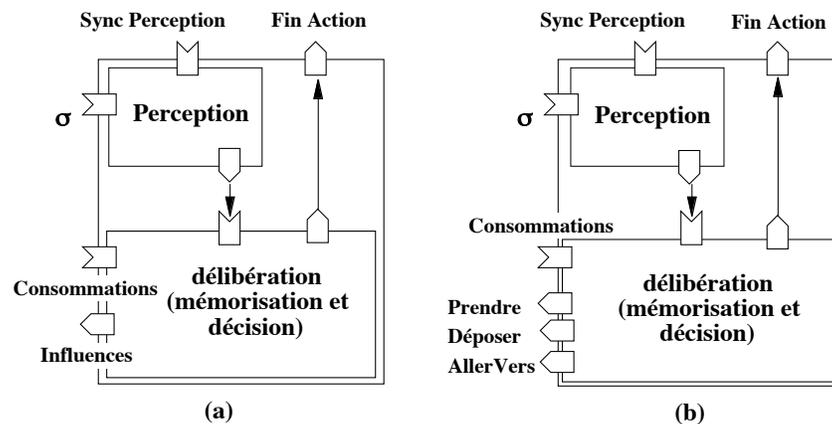


Figure 4.47: Représentation d'un agent situé sous la forme d'un composant BRIC. La forme généralisée (a) évite d'avoir à se prononcer sur l'ensemble des actions qu'un agent peut accomplir, à la différence de sa forme spécialisée (b). La borne σ représente les informations provenant de l'environnement.

On supposera que l'agent perçoit toujours quelque chose, au moins la marque de sa présence dans l'environnement, et donc qu'il aura toujours quelque chose à faire. De ce fait, un agent n'est jamais passif, puisqu'il traite toujours au moins un percept provenant de l'environnement.

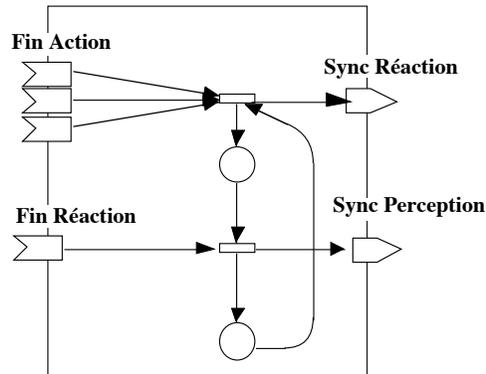


Figure 4.48: Le synchronisateur d'un SMA situé

Le synchronisateur global se charge de synchroniser les perceptions et les actions (fig. 4.48). Il est important de bien comprendre l'intérêt d'un synchronisateur. Il s'agit d'une horloge qui évite que des actions effectuées de manière simultanée soient interprétées comme étant totalement séparées dans le temps. Cette horloge ne donne pas l'heure mais assure qu'à un moment donné tous les agents agissent simultanément et que l'environnement ne réagisse qu'ensuite avant de redonner la main aux agents dans une boucle sans fin. La figure 4.49 donne le diagramme des temps d'un SMA comprenant deux agents, l'environnement et le synchronisateur.

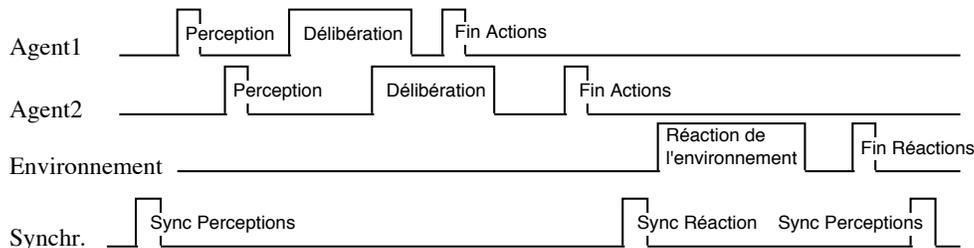


Figure 4.49: Diagramme de synchronisation pour deux agents et un environnement

Lorsque le synchronisateur envoie un signal **Sync Perceptions**, les agents **Agent1** et **Agent2** effectuent leur perception. Chaque agent délibère en traitant les perceptions qu'il reçoit en fonction de son état précédent, puis envoie ses influences à l'environnement. Finalement, il envoie un signal de fin d'actions. Lorsqu'il a reçu tous les signaux de fin d'actions provenant des agents, le synchronisateur envoie un signal **Sync Réaction** qui déclenche la réaction de l'environnement à la production des influences. Une fois terminé, l'environnement envoie un signal de fin de réaction au synchronisateur qui réagit en envoyant un signal **Sync Perceptions**, et le cycle recommence. Le schéma général d'un SMA situé est présenté à la figure 4.50. L'environnement comme les agents sont donnés sous leur forme généralisée. Il suffirait de remplacer la définition centralisée des agents et des environnements par des versions spécialisées pour disposer d'un modèle de SMA spécialisé.

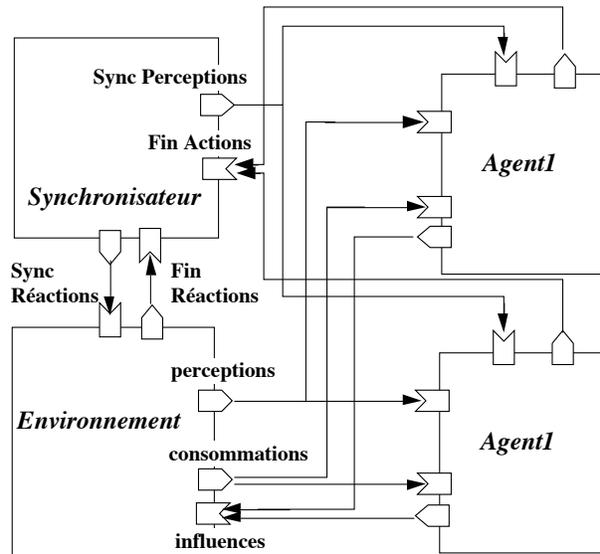


Figure 4.50: Schéma général d'un SMA situé à deux agents

4.11.5 Modélisation d'un SMA complet

La modélisation d'un système composé d'agents sachant à la fois communiquer et agir dans un environnement est relativement simple puisqu'il suffit de superposer les schémas d'agents situés et d'agents communicants. Mais cette apparente simplicité cache en fait un problème plus complexe qui porte sur le problème de la synchronisation des actions situées par rapport aux messages. Les communications ont été modélisées de manière asynchrone, notamment à cause du formalisme des réseaux de Petri qui est naturellement asynchrone, c'est-à-dire que les messages sont envoyés dans n'importe quel ordre et que l'on ne se soucie absolument pas de connaître le temps mis par un agent pour répondre à un message. Inversement, les agents situés sont totalement synchronisés par l'environnement, et ceci a nécessité l'introduction d'un synchronisateur. Ce problème de synchronisation peut être résolu de deux manières. La première consiste à l'ignorer purement et simplement en considérant que les communications possèdent leur rythme propre qui n'a rien à voir avec les actions dans un environnement. Mais cela pose néanmoins de grandes difficultés lorsque les agents doivent collaborer à une tâche commune dans laquelle le temps est un paramètre important (problèmes de temps réel par exemple). La seconde solution, qui est d'après moi la plus sage, consiste à synchroniser les messages avec les actions: les messages sont envoyés en même temps que les influences, et reçus avec les perceptions. Dans ce cas, les messages ont un temps de propagation considéré comme immédiat par le modèle (on peut néanmoins modifier le système d'acheminement des messages pour modéliser un système de délai de transmission si on le désire). Le synchronisateur ne comprend qu'une borne en plus, **Fin Message**, qui indique que tous les messages ont été envoyés et qui vient se combiner à l'indicateur **Fin Réactions** (fig. 4.51).

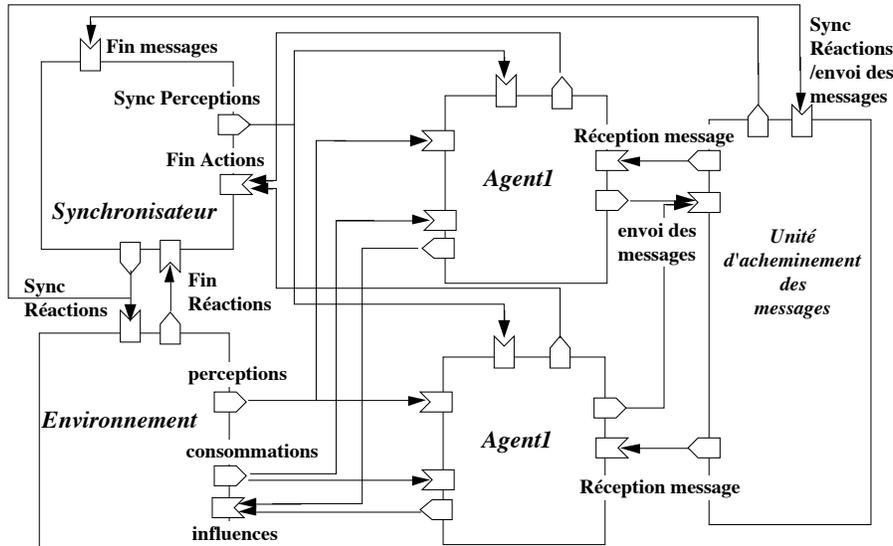


Figure 4.51: Modèle de système d'agents communicants et situés. Le synchronisateur synchronise les messages avec les actions.

4.11.6 Un exemple: les agents transporteurs

Nous allons maintenant montrer comment il est possible de construire un système multi-agent réactif simple à partir du modèle d'action fondé sur les influences. L'implémentation de l'environnement de manière globale est plus claire dans ce cas. L'exemple est celui de Clotaire (section 4.2.2). Clotaire doit aller dans la remise pour aller chercher un marteau et l'apporter dans le hangar. On suppose que la remise et le hangar émettent des signaux qui se propagent sous la forme de vagues en contournant les obstacles. L'environnement est initialement décrit par l'ensemble des marques suivantes:

```
{chemin(1,2), chemin(1,4), chemin(4,5), chemin(5,6), etc.
  sig(R,3,3), sig(R,5,1), sig(R,6,2), sig(R,0,Remise),
  sig(R,3,Hangar), etc.
  sig(H,0,Hangar), sig(H,5,2), sig(H,1,4), sig(H,1,6),
  sig(H,3,Remise), etc.
  agent(Clotaire,2), pince(Remise)}
```

Le comportement de l'agent se borne à aller vers la remise en suivant le gradient des signaux émis par cette pièce (signaux R), à prendre la pince, puis à aller dans le hangar pour l'y déposer en suivant les signaux H émis par le hangar. Il est possible de donner un modèle de comportement de cet agent sous la forme d'un agent situé, dont les seules actions sont d'aller vers une pièce adjacente, de prendre et de déposer quelque chose (fig. 4.52).

Le module de perception se contente de filtrer les informations qui intéressent l'agent et de construire les percepts obtenus à partir d'une combinaison des indices `agent(Clotaire, x)` qui donne la position de l'agent, `pince(y)` qui donne la position

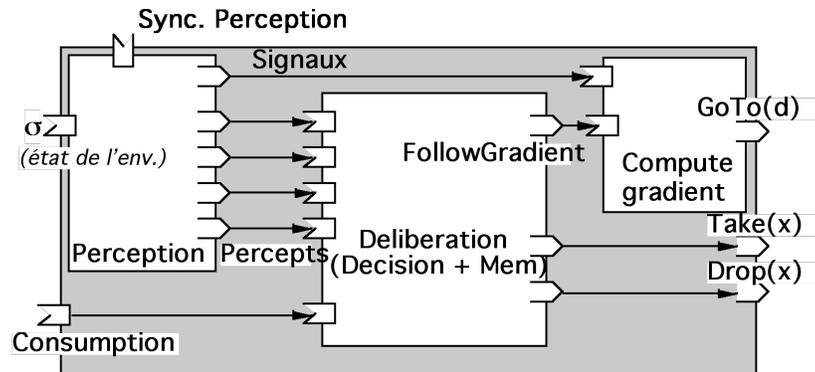


Figure 4.52: Un modèle d'agent réactif se basant uniquement sur sa perception

de la pince, et $\text{sig}(s, x, l)$ lequel donne l'intensité x d'un signal s en un lieu l (fig. 4.53).

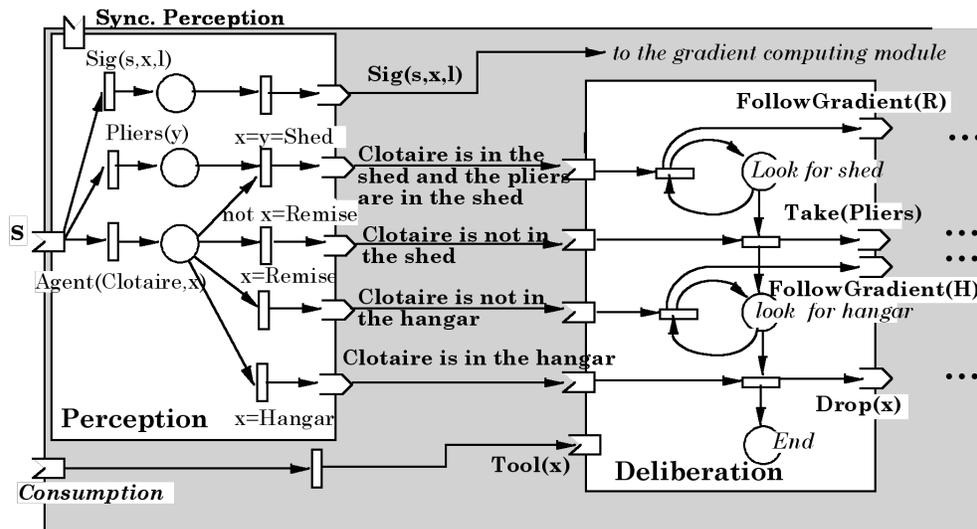


Figure 4.53: La structure interne des composants de perception et de délibération de l'agent transporteur

Le module de délibération se charge de la logique de l'agent. Celui-ci suit d'abord le gradient R qui le conduit vers la remise, puis, lorsqu'il se trouve dans la remise et si la pince s'y trouve aussi, il la prend (déclenchement de l'action "prendre", suivi de la "consommation" de l'objet), puis il se dirige vers le hangar en suivant les gradients des signaux H . Enfin, lorsqu'il arrive dans le hangar, il dépose ce qu'il tient (la pince en l'occurrence) et il s'arrête. Il s'agit d'un agent relativement simple, qui ne peut pas se rendre compte s'il perd la pince qu'il porte, et qui s'immobilise si la pince ne se trouve pas initialement dans la remise. Enfin, l'agent ne peut effectuer qu'une seule tâche extrêmement sommaire (aller chercher une pince et la porter d'une pièce à une autre), ce qui réduit bien évidemment son intérêt. Néanmoins, cet exemple montre qu'il est possible de décrire l'intégralité du comportement d'un SMA dans le

formalisme BRIC. Nous verrons aux chapitres suivants qu'il est possible de construire des agents beaucoup plus élaborés, dans lesquels ce formalisme prendra toute son importance.

Chapter 5

L'agent dans tous ses états

5.1 Etats mentaux et intentionnalité

5.1.1 Introduction

La psychologie ordinaire consiste à décrire et à expliquer le comportement des êtres humains à partir de leurs états mentaux, c'est-à-dire à partir de leurs croyances, désirs, intentions et engagements. Par exemple, si l'on dit que "Paul a envie de faire un cadeau à Marie, et qu'il croit que Marie aime les fleurs", on peut en déduire que Paul va certainement offrir des fleurs à Marie. De même si Pierre demande à quelqu'un s'il a l'heure, on en conclut d'une part qu'il lui manque l'information concernant l'heure au moment où il fait cette demande, et d'autre part qu'il désire qu'on lui fournisse cette information.

Ces raisonnements spontanés que nous effectuons quotidiennement et par lesquels nous expliquons le comportement des êtres humains peuvent donner lieu à l'élaboration de théories. On suppose alors que ce qui se passe dans la tête d'un agent (qu'il s'agisse d'un agent artificiel comme dans les systèmes multi-agents ou d'une personne humaine telle que peuvent l'appréhender les sciences cognitives) résulte de l'interaction d'un certain nombre de composants cognitifs, appelés généralement *états mentaux* ou *attitudes propositionnelles*.

Par exemple, croire qu'il pleut ou que le chat est sur la cheminée, c'est être dans une certaine disposition à *propos* de la pluie ou du chat. De même, en voulant aller au cinéma, je suis dans un certain état d'esprit à *propos* du fait d'aller au cinéma. Les désirs comme les croyances sont donc des états mentaux qui se réfèrent à quelque chose de concret comme dans les exemples précédents, voire à un autre état mental. Par exemple, si je crois que Paul désire un pain au chocolat, ma croyance se rapporte au désir de Paul, c'est-à-dire à un état mental qui se réfère lui-même à un pain au chocolat. Les états mentaux ont donc cette particularité d'être à *propos de quelque chose*, et c'est pourquoi on parle du caractère *intentionnel* des états mentaux.

Les termes *intentionnel* et *intentionnalité* ont été forgés par les philosophes scolastiques du Moyen Age. Ils proviennent du latin "intendo" qui signifie "diriger (ou tendre) vers". Ils ont ensuite été repris par les philosophes dans un sens cognitif: la conscience s'appliquant à quelque chose d'autre qu'elle-même, elle est donc par

nature tendue vers ce quelque chose, et donc intentionnelle. En effet, si je vois un objet, la croyance que cet objet se trouve bien devant moi est un phénomène mental qui se rapporte à cet objet. C'est d'ailleurs en cela que je peux dire que "je vois cet objet". De même, si je désire me rendre quelque part, ce désir s'applique à l'acte d'aller dans un certain lieu, et donc à quelque chose d'autre qu'à ce désir. Brentano et à sa suite Husserl (Husserl 1950) ont soutenu que l'intentionnalité représentait la distinction fondamentale permettant de définir le mental et le physique; tous les phénomènes mentaux, et eux seuls, font preuve d'intentionnalité. Depuis, les philosophes anglo-saxons, dans le cadre de ce que l'on appelle la *philosophie de l'esprit* (philosophy of mind), se sont emparés de cette question pour analyser ce qui se passe dans nos esprits et essayer de fournir des modèles du rapport qui existe entre les croyances, les engagements et les intentions pour ne parler que de quelques-uns de ces états mentaux.

La problématique de l'intentionnalité est extrêmement riche et on pourrait la formuler de la manière suivante: Qu'est ce qui fait qu'un agent fait ce qu'il fait? Quels sont les éléments internes qu'il met en œuvre pour nous donner l'illusion d'une autonomie et que son comportement suive une certaine cohérence dans les échanges qu'il entretient avec le monde et les autres agents. Malheureusement, en dépit de l'attention que reçoit actuellement ce concept, il existe un manque frappant de consensus à propos de l'analyse de l'intentionnalité. Le seul point d'accord concerne la nature du problème posé et les solutions peuvent prendre des formes effroyablement techniques¹.

Cette difficulté s'ajoute à une autre: les termes d'*intention* et d'*intentionnalité* peuvent prendre deux acceptions relativement distinctes. La première, utilisée dans le domaine philosophique, correspond à la définition que nous avons donné précédemment et se rapporte donc aux états mentaux. Le second se réfère au sens commun de la notion d'intention: on dit que quelqu'un a l'intention d'accomplir un acte (par exemple: sortir, prendre un verre) s'il désire effectivement exécuter cet acte. L'intention est alors, comme le précise Thomas d'Aquin, *un acte de volonté précédé d'un acte par lequel la raison ordonne quelque chose à sa fin*. Il est vrai que cette confusion est d'autant plus forte que la seconde acception n'est qu'un cas particulier de la première et ces deux significations sont souvent présentes dans un même ouvrage. La plupart du temps, on réserve le terme d'*intentionnalité* à la problématique générale des états mentaux et celui d'*intention* à la volonté d'agir.

Dans la suite de l'ouvrage et afin d'éviter cette confusion, nous n'utiliserons plus, sauf mention explicite du contraire, les termes *intention*, *intentionnel*, *intentionnellement* et *intentionnalité* que dans le second sens, c'est-à-dire dans celui de ce qui amène un agent à agir, préférant les expressions *états mentaux* ou *cognitions*

¹Pour avoir une bonne introduction à la problématique de l'intentionnalité en IA on pourra se référer à (Bachimont 1992) qui présente de manière relativement simple la relation qui existe entre les théories philosophiques de la connaissance et les solutions (partielles) apportées par l'IA. D'autre part, pour aborder la question de l'intentionnalité dans le débat cognitiviste actuel, on lira avec bonheur l'excellent ouvrage (Pacherie 1993) qui donne une vue très synthétique des différentes positions en présence. On pourra aussi lire (Searle 1985) et (Dennett 1987) qui détaillent des points de vue importants dans ce débat.

pour désigner le premier.

5.1.2 La notion de cogniton

Le terme *d'état mental* n'est pas non plus très heureux car il suppose une certaine idée de traitement séquentiel dans le mode cognitif de fonctionnement d'un agent. En effet, le mot *état* exprime généralement une caractéristique globale d'un système. Il donne l'illusion qu'un agent se trouve à un moment donné dans une configuration particulière aisément qualifiable. On a l'impression alors que si l'on dit qu'un agent se trouve dans *l'état de désirer* un objet, son "esprit" (mind) sera dans une configuration particulière caractéristique d'un désir pour un objet. Or cette impression est totalement fautive. On dira simplement qu'un agent présente l'état de désir, s'il existe quelque part dans son système cognitif des structures informationnelles caractéristiques du désir, et qui l'amène à mettre en œuvre des ressources de manière à, si cela s'avère possible, satisfaire son désir. L'état mental de désir (comme celui de croyance, d'intention, d'engagement) n'est en fait pas un état mais une structure calculatoire² représentant un désir pour quelque chose. Si par hasard un agent vient à désirer et à craindre un même objet, on ne dira pas que l'agent est dans un état particulier de désir ou de crainte, mais que les deux structures calculatoires de désir et de crainte sont présentes en même temps et agissent de manière conjointes (et opposées) pour la définition des comportements des agents.

De ce fait, je conserverai le terme d'état pour décrire toute configuration globale d'un système (c'est la notion d'état telle que nous l'avons vue avec la notion d'automate à états finis au chapitre 4), et j'utiliserai à la place le néologisme de *cogniton*³ pour décrire ces structures calculatoires qui sont à propos de quelque chose, et donc douées d'intentionnalité dans le premier sens du terme, c'est-à-dire celui des scolastiques et de Husserl.

Un cogniton est donc une unité cognitive élémentaire qui est reliée à d'autres cognitons pour former une configuration mentale dynamique qui évolue avec le temps. Des cognitons peuvent être créés ou disparaître comme conséquences de la création ou de la disparition d'autres cognitons dans l'appareil cognitif d'un agent. Par exemple, si je crois que le chat est sur la cheminée et que je le perçois sur la table, alors je réviserais mes croyances en considérant que je dois privilégier d'abord ce que je vois par rapport à ce que je crois. Le fait d'avoir une croyance à un moment donné et ensuite de réviser cette croyance à partir d'une perception fait partie de l'activité normale du module représentationnel, qui élabore des modèles du monde à partir des informations (percepts, messages informatifs) qu'il reçoit et des connaissances dont il dispose au préalable. Dans le même ordre d'idées, les intentions sont liées à

²De nombreux auteurs tels qu'E. Morin préfèrent le terme computationnel. Mais les informaticiens, qui savent bien que le calcul ne se résume pas au traitement de nombres, préfèrent généralement le terme calculatoire comme correspondant de l'anglais computational.

³Ce terme est construit à partir du vocable "cognitif" et de la terminaison "on" qui désigne quelque chose d'insécable, une particule élémentaire, à la manière de photon ou méson. Il a été élaboré par analogie avec "infon", l'unité élémentaire d'information développée par K. Devlin dans le cadre de la sémantique situationnelle (Devlin 1991).

la fois aux buts qui les motivent, mais aussi aux normes et aux engagements qui les contraignent. Si l'on propose un rendez-vous et que l'on est déjà occupé, il faudra soit modifier la date de ce rendez-vous, soit se décommander de ses engagements antérieurs, ce qui entraînera une révision de ses croyances et de ses intentions.

5.1.3 Une nomenclature des cognitons

Il existe évidemment un grand nombre de cognitons. Les trois principaux que l'on retrouve dans la plupart des théories sur les états mentaux sont les croyances, les intentions et les engagements. Mais il en existe beaucoup d'autres qui se rapportent aux différents éléments permettant de comprendre le fonctionnement des agents cognitifs: par exemple les percepts, les buts, les désirs, les pulsions, les hypothèses, etc. sont aussi essentiels à la définition du comportement des agents que peuvent l'être les engagements ou les croyances.

Il est possible de classer les cognitons en fonction du système organique auquel ils sont liés. Néanmoins, certains cognitons sont à cheval entre plusieurs sous-systèmes cognitifs. Par exemple, l'intention et l'engagement font aussi bien partie du sous-système conatif, puisqu'il s'agit d'éléments qui permettent la décision, que du sous-système organisationnel, puisqu'ils permettent la gestion de l'action. Voici la liste, non exhaustive, des cognitons et des sous-systèmes dans lesquels ils agissent. Lorsqu'un cogniton apparaît dans plusieurs, la liste en est donnée à la fin de sa description.

Les cognitons interactionnels

Percept. Les percepts représentent l'information sensible obtenue par perception de l'environnement. Cette information est généralement le résultat d'un travail d'élaboration effectué par des capteurs et des systèmes de prétraitements que nous n'étudierons pas.

Information. Une information est un cogniton qui représente une croyance véhiculée par un autre agent et transmise par message.

Commande (ou décision). Les commandes sont des cognitons provenant du système conatif et décrivant les opérations qui doivent être exécutées par le système interactionnel, qui se traduiront par des actions de l'agent sur son environnement.

Requête. Les requêtes sont des actions effectuées par un agent afin qu'un autre agent accomplisse une action (faire quelque chose, répondre à une question,...). Les requêtes sont ainsi des actions relationnelles puisqu'elles s'appliquent entre agents. Nous verrons au chapitre 6 que la théorie des actes de langage leur assure une base solide.

Normes. Les normes sont des cognitons qui représentent en fait les demandes et les contraintes sociales imposées par l'organisation sur un agent particulier. Les normes peuvent s'exprimer ainsi de nombreuses manières. La plus courante

consiste à attribuer à un agent un ensemble de tâches à accomplir en relation avec la fonction (ou le rôle) qu'il occupe dans son organisation. Ces normes sont alors appelées demandes sociales car elles jouent le même rôle que des demandes pour le déclenchement de l'action. La plupart des normes ne sont pas acquises par l'agent mais installées par le concepteur lors de la création d'un agent.

Les cognitons représentationnels

Croyance. Les croyances décrivent l'état du monde du point de vue d'un agent, et donc la manière dont il se représente son environnement, les autres agents ainsi que lui-même. En particulier il faut bien distinguer les croyances sur soi des croyances sur les autres, les premières pouvant être cohérentes avec l'état dans lequel se trouve l'agent⁴, alors que les secondes ne sont que partielles et éventuellement erronées.

Hypothèse. Les hypothèses sont des représentations possibles du monde et des autres agents qui ne sont pas encore "crués" par l'agent. Une hypothèse peut servir à déclencher des actions ayant pour but de la transformer en croyance.

Les cognitons conatifs

Tendance/But. Les tendances ou buts (nous ne ferons pas de différence ici) sont les résultats essentiellement de pulsions et de demandes. Qu'ils soient réalisables ou non n'intervient pas dans la définition d'un but. Par exemple, je peux avoir le but d'être milliardaire, même si je sais qu'il y a peu de chances que j'atteigne ce but. Mais j'essaierai néanmoins de satisfaire ce but, en jouant au Loto par exemple, et si jamais je gagne le gros lot, alors ce but sera satisfait. Les buts sont donc toujours présents tant que ce qui a conduit à la construction du but est lui-même valide. Les buts (ou tendances) peuvent s'exprimer aussi bien en termes d'attirance que de répulsion. Ils sont en fait le cœur du système conatif.

Pulsion. Les pulsions sont les sources internes de construction de buts. Elles correspondent à des demandes du système conservatif tel qu'assouvir sa faim, tenter de se reproduire et d'une manière générale tout ce qui permet la survie individuelle ou celle de l'espèce (sous-systèmes végétatif et conatif).

Demande. Les demandes sont les sources externes des buts. Elles interviennent lorsqu'un agent effectue une requête vers un autre agent.

Intention. Les intentions décrivent ce qui meut un agent, ce qui lui permet de passer à l'acte. En suivant Cohen et Levesque (cf. section 5.8), nous verrons qu'il est possible de se passer du cogniton de l'intention à condition d'avoir les

⁴On peut supposer que dans les systèmes artificiels, les agents aient des croyances vraies sur eux-mêmes et en particulier sur leurs compétences, ce qui n'est pas vrai pour des agents réels tels que les êtres humains qui se leurrent souvent sur eux-mêmes ou se mésestiment.

notions de buts et d'engagement, ainsi que la capacité d'évaluer l'importance des buts et leurs conséquences (sous-systèmes conatif et organisationnel).

Engagement Les engagements caractérisent les dépendances (devoirs, contraintes...) qui lient les agents cognitifs par rapport à eux-mêmes, mais surtout par rapport aux autres lorsqu'ils décident d'accomplir une action, de rendre un service et, d'une manière générale, lorsqu'ils ont l'intention de faire quelque chose (sous-systèmes conatif et organisationnel).

Les cognitons organisationnels

Ces cognitons assurent le suivi des actions des agents. Lorsqu'un agent possède un but, il évalue les différentes méthodes lui permettant de satisfaire ce but s'il en dispose, ou bien il essaye de décomposer son but en sous-buts de manière à trouver un plan qui lui permette d'accomplir ses objectifs. Après avoir pris sa décision, il se met au travail, c'est-à-dire qu'il exécute la tâche correspondant à la méthode choisie, cette tâche étant placée dans un agenda des choses à faire et à surveiller.

Méthode. Les méthodes sont l'ensemble des techniques, règles, plans et programmes qui peuvent être mis en œuvre par un agent pour accomplir une tâche et satisfaire un but. Une méthode décrit l'ensemble des étapes nécessaires à la résolution d'un problème ou à la réalisation d'un but. Par exemple, si l'on doit résoudre un problème, on peut utiliser la méthode que l'on pourrait appeler de "gros œuvre et détails" (propose and revise) dans laquelle on construit d'abord une solution partielle et peu détaillée que l'on s'applique ensuite à développer. Les méthodes prennent souvent l'aspect de plans, c'est-à-dire de suites d'actions à entreprendre, à différents niveaux de détails, pour accomplir un but, comme lorsque Clotaire doit déterminer le moyen de satisfaire le but "prendre la pince qui se trouve dans l'atelier".

Tâche. Les tâches se rencontrent à deux niveaux différents. Au niveau conceptuel, une tâche est la description de ce qu'il faut faire pour accomplir un but. Par exemple, pour que le marteau soit dans le hangar et plus dans l'atelier, il faut successivement aller chercher le marteau, le prendre et le transporter dans le hangar. Il s'agit uniquement des actions de haut niveau que l'on doit accomplir. Rien n'est dit sur leur mise en œuvre: ce sont les méthodes que l'on appliquera qui indiqueront comment aller dans l'atelier, comment prendre le marteau et comment transporter le marteau dans le hangar. Au niveau organisationnel, les tâches sont aussi les unités de travail en cours d'exécution, c'est-à-dire l'instanciation du déroulement d'une méthode.

Parfois, surtout pour des agents réactifs ou pour des agents cognitifs simples, les termes tâches et méthodes sont confondus: s'il n'existe qu'une seule méthode pour une tâche définie, alors il n'y a plus d'intérêt à dissocier ces deux concepts, et l'on emploiera alors l'un pour l'autre.

Autres cognitons

Il existe évidemment bien d'autres cognitons caractéristiques d'états mentaux, tels que les craintes, les désirs, les espoirs, les rêves, les affects (sentiments associés à un autre agent), etc. Nous n'en parlerons pas, estimant soit qu'ils font plus ou moins partie des états mentaux précédents (par exemple, les désirs sont liées aux pulsions et aux buts), soit qu'ils portent sur des caractéristiques qui ne nous intéressent pas pour des agents artificiels: les peurs, comme les rêves, les affects et les espoirs ne font pas (encore) partie de la structure des agents artificiels.

5.2 Le système interactionnel

C'est par la perception que l'agent acquiert des informations sur le monde pour lui permettre d'élaborer son action en poursuivant ses buts. Le système perceptif constitue ainsi une porte entre le monde et l'agent qui lui donne accès à une certaine "conception du monde" dans laquelle ses raisonnements et ses actions sont significatifs.

En effet toute perception est individuelle et contribue à une représentation personnelle du monde (Uexküll 1956). Pour la tique du chien, le monde se réduit à trois stimuli: la chaleur diffuse, l'odeur d'acide butyrique et la chaleur. Cela lui suffit pour ses besoins, son comportement étant entièrement déterminé par ces impressions élémentaires. Elle s'élève vers la lumière jusqu'à ce qu'elle ne puisse plus aller plus haut, et dès qu'elle perçoit l'odeur de l'acide butyrique, provenant des mammifères, elle se laisse tombée. Si elle tombe sur de la chaleur, cela signifie qu'elle a atteint son but. Elle trouve un endroit dépourvu de poils et elle perce la peau pour aspirer le sang. Puis le cycle recommence. Pour des êtres plus complexes, leur système perceptif leur permet de reconnaître leurs proies et leurs prédateurs, leur lieux de reproduction, les êtres de la même espèce, etc. Le système perceptif d'un agent est ainsi relié à ses compétences cognitives et comportementales. Un robot explorateur de minerai doit disposer de mécanismes de détection de minerai s'il veut accomplir sa tâche. Il pourra de plus percevoir les obstacles qui se présentent sur sa route (rochers, crevasses, arbres, etc.) afin de les éviter. S'il est capable de percevoir en plus la présence d'autres robots, il pourra élaborer des stratégies d'exploration lui permettant de mieux remplir sa fonction, en s'éloignant des autres robots qui recherchent du minerai et en s'approchant au contraire de ceux qui viennent d'en trouver. La gamme des réponses comportementales dépend donc de la richesse des perceptions et des capacités de reconnaissance d'un agent. Mimer le comportement d'un autre agent, par exemple, nécessite à la fois des capacités cognitives assez poussées, mais surtout des mécanismes de reconnaissance très élaborés lui permettant de reconnaître la posture et les gestes d'un autre agent afin de pouvoir les imiter.

Les philosophes, depuis Aristote, se sont toujours intéressés à la perception: pour ce dernier, qui avait une conception très réaliste, percevoir c'est séparer par abstraction la forme de la matière, l'être se trouvant ainsi "informé" par cette substance. L'objet perçu agit ainsi directement sur le système perceptif, l'agent se

contentant passivement de recevoir cette forme et de l'intégrer à son système cognitif (à son âme dirait Aristote). La perception est essentiellement une modification physique qui laisse des traces physiques, les "phantasmes", dans le corps (si l'on regarde un morceau de papier blanc, les yeux, ou une partie interne des yeux, deviennent blancs par une sorte de réflexe mimétique du percevant vis-à-vis du perçu). Pendant longtemps, et en fait jusqu'à Kant, les philosophes ont adhéré à une conception causale de la perception: les objets du monde agissent mécaniquement sur les divers organes des sens de manière à former des "idées de sensations", objets immédiats de la perception. Les explications modernes sont plus techniques mais partent souvent de la même idée: les sensations sont des données immédiates issues d'excitations provenant de capteurs externes ou internes qui transforment les stimuli physiques (vibration de l'air, énergie lumineuse, composés chimiques, etc.) en signaux internes, lesquels sont ensuite traités par l'ensemble des systèmes de perception primaires qui éliminent le bruit, segmentent les signaux et fournissent aux systèmes de traitement secondaires les données brutes à partir desquelles ils pourront effectivement reconnaître des formes signifiantes.

C'est encore cette conception, que l'on qualifie de "réaliste" qui prévaut de manière naïve dans notre activité quotidienne: nous supposons que nous percevons directement des propriétés "objectives" des corps qui nous entourent et qui viennent s'inscrire directement dans notre esprit par un procédé direct et immédiat. De ce fait, le monde est pour nous bien tel que nous nous le représentons: la neige est bien blanche, la rose bien rouge, la table bien plate, etc. Cette notion purement passive de la perception est tellement ancrée en nous que nous avons du mal à penser que nos facultés perceptives elles aussi sont le fruit d'une élaboration et que nous ne percevons bien que ce que nous avons d'une certaine manière déjà perçu.

Les perceptions sont les fruits de constructions qui dépendent de schémas perceptifs antérieurs, lesquels se sont construits progressivement par nos interactions avec l'environnement et les autres agents qui ont conforté nos points de vues. Un corps est "rouge" parce que nous sommes sensibles à certaines fréquences de la lumière, que des photons dans cette gamme de fréquence ont été émis par une source de lumière et n'ont pas été retenus par ce corps, et surtout parce que nous avons déjà appris à distinguer par notre culture, c'est-à-dire par des interactions antérieures avec d'autres membres de notre espèce, certaines gammes de couleurs en différenciant le rouge du rose et du violet par exemple (Bonnet et al. 1990).

Les théories logiques actuelles, comme la plupart des systèmes de reconnaissance de formes avant les années 90, reposent encore sur une conception de perception passive du monde. La sémantique situationnelle par exemple ne fait que reprendre les idées aristotéliennes en leur donnant une expression plus moderne: le monde est constitué d'objets et de situations déjà données, et l'agent ne fait qu'enregistrer et classer les informations provenant de son environnement. Le sens des phrases qu'il prononce se réfèrent ainsi directement (naïvement) aux objets du monde, et non à des constructions mentales élaborées indépendamment (Barwise et Perry 1983).

Au contraire, les résultats de la neurophysiologie de la perception montrent que la perception est le résultat d'une conceptualisation antérieure. Par exemple, dans les systèmes visuels périphériques, le nombre de voies nerveuses allant du cortex vers

le corps genouillé latéral (région du thalamus intermédiaire entre le nerf optique et le cortex visuel) est bien supérieur à celles qui proviennent directement de l'œil (Maturana et Varela 1987). L'individu "voit" plus ce qu'il reçoit du cerveau que ce qu'il reçoit de l'œil. D'autres mécanismes, tels que celui de la perception du mouvement chez les animaux et l'homme, utilisent le principe dit de "réafférence" qui lie le mouvement de l'individu et celui de la perception. Par réinjection de ce qui devrait être vu lors d'un de ses déplacements, l'individu analyse comme une image fixe un changement visuel provenant d'un de ses mouvements, et comme une image mobile le mouvement d'un objet. D'autres phénomènes du même ordre montrent que la perception résulte d'un processus actif, qui prend en compte les perceptions et conceptualisations précédentes, et qui implique une exploration active de l'environnement et de ce que l'agent cherche à percevoir.

La figure 5.1 montre les deux types de système. Dans les systèmes de *perception passive*, les signaux suivent une approche purement ascendante. Les signaux élémentaires sont prétraités puis segmentés pour obtenir des *traits élémentaires* conduisant à la reconnaissance d'objets, de scènes, de mots ou de phrases. Dans les systèmes actifs au contraire, le système perceptif reçoit à la fois les informations provenant des capteurs et les attentes et buts provenant du système cognitif. Il peut aussi contrôler ses capteurs de manière à maintenir une représentation cohérente de son environnement. L'un des premiers systèmes de perception combinant la génération d'hypothèses et leur validation peut être trouvé dans les travaux sur la reconnaissance de la parole par tableaux noirs (Erman et al. 1980). On pourra avoir une vision générale des problèmes de vision pour l'ordinateur et la machine dans (Marr 1982; Levine 1985). On trouvera dans (Ghallab et al. 1992) les principaux problèmes (et solutions) que pose la perception multi-sensorielle active, et dans (Lesser et Corkill 1983) et (Demazeau et al. 1994) l'utilisation de systèmes multi-agents pour la conception de systèmes de perception active.

Bien que nous soyons totalement en accord avec les théories qui supposent que la perception est elle-même une construction active du sujet, nous nous en tiendrons à une conception passive lorsqu'il s'agira d'intégrer le système perceptif dans l'ensemble des composants d'un agent cognitif pour des raisons à la fois de simplicité et aussi parce que la plupart des systèmes multi-agents supposent que leurs agents sont munis de systèmes perceptifs passifs.

5.3 Le système représentationnel

Nous avons vu que les agents cognitifs disposaient de connaissances explicites qu'ils pouvaient utiliser pour appréhender le monde et les autres agents. Mais que signifient "représentations", "croyances" et "connaissances" et comment les expliciter?

5.3.1 Qu'est-ce que la connaissance?

La notion de connaissance, telle qu'elle s'est développée en intelligence artificielle, peut-être définie comme l'ensemble des informations (savoir, savoir-faire, expérience,

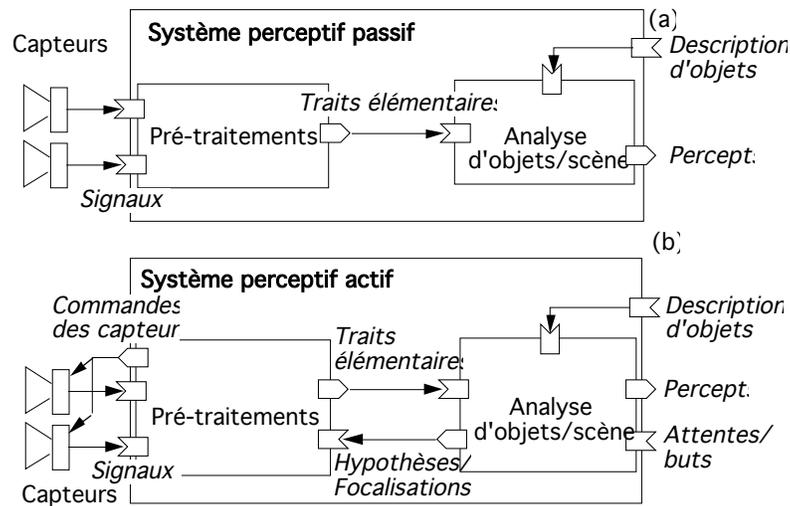


Figure 5.1: Le système perceptif peut-être soit passif (a) et, dans ce cas, l’analyse des informations s’effectue de manière purement ascendante, soit actif (b) en combinant l’approche ascendante à une démarche descendante tendant à la génération d’hypothèses et à la focalisation des perceptions.

souvenirs, concepts, faits...) nécessaires à un être humain (ou à une machine) organisées de manière à ce qu’il puisse accomplir une tâche considérée comme complexe. Par exemple, diagnostiquer une maladie, résoudre un problème de mathématique, réparer une machine, traiter un cas juridique, gérer un ensemble de comptes financiers, faire une campagne de marketing ou traduire un texte sont des activités, qui, lorsqu’elles sont réalisées par des êtres humains, réclament une grande quantité de compétences et d’informations organisées dans un but précis.

De ce fait, au terme de “connaissance” est associé une dimension pragmatique, c’est-à-dire de savoir spécialiste qui s’applique dans un domaine particulier pour effectuer une tâche précise. Cependant, la connaissance d’un agent ne se limite pas à celle des tâches qu’il accomplit: il faut aussi inclure les informations concernant le monde qui l’entoure, c’est-à-dire son environnement et les autres agents avec lesquels il communique et interagit.

La notion de connaissance pose un des problèmes centraux en philosophie et en psychologie cognitive, qui peut s’exprimer ainsi: Comment les concepts peuvent-ils être organisés de manière à permettre à un être humain ou à un agent d’accomplir des actes ou d’affirmer des propositions adéquates?

On distingue généralement entre *savoir quoi* et *savoir comment*. Le premier s’intéresse à la compréhension des êtres que l’on perçoit et à celle des phénomènes que l’on rencontre en permanence mais pour lesquels il est parfois difficile de donner une définition précise. Le second au contraire ne s’intéresse pas à la définition des objets, mais à l’analyse des relations qui existent entre les phénomènes, pour pouvoir modéliser les lois de l’univers afin d’en prédire leur évolution. Par exemple, essayer de définir ce qu’est le mouvement est un problème philosophique complexe, qui

n'avait été résolu que partiellement par les philosophes grecs et médiévaux⁵. Il fallut attendre Galilée d'abord, puis Newton pour se rendre compte qu'en changeant de problématique et en ne cherchant plus à définir ce qu'*est* le mouvement, il était possible de comprendre *comment* évoluaient les corps en mouvement à partir de relations entre des grandeurs mesurables telles que la distance, la vitesse et le temps.

Le savoir scientifique, comme le savoir pragmatique, est un savoir du *comment*, l'important étant de pouvoir agir dans le monde, en anticipant les conséquences des états actuels et des actions que nous pouvons entreprendre. Les grandes questions générales, telles que "Qu'est-ce que la vie?", si elles s'avèrent particulièrement essentielles pour nous situer et pouvoir répondre à nos interrogations existentielles, n'ont presque aucun intérêt pour le savoir scientifique. Celui-ci ne s'intéresse pas à ces questions. Il lui suffit, à partir de définitions initiales relativement vagues, d'élaborer un corpus de méthodes lui permettant de comprendre les phénomènes qui ont lieu lors de la vie des êtres vivants, et être ainsi à même d'expliquer *ce qui se passe*.

Il en est de même pour la connaissance. Il s'avère préférable, plutôt que d'essayer de définir précisément ce qu'est la "connaissance", de proposer des concepts et des modèles permettant de comprendre les phénomènes d'élaboration et d'utilisation de ce qui permet à un agent d'accomplir des tâches complexes.

Conceptualisation et raisonnement

Le problème de la connaissance est étroitement relié à ceux de la *conceptualisation*, du *raisonnement*: quelles sont les catégories conceptuelles *a priori* de la pensée et du raisonnement d'un agent, et quels sont, à partir de ces catégories fondamentales, les différents concepts construits ainsi que les mécanismes pratiques qu'un agent emploie de façon à pouvoir s'orienter dans son environnement, surmonter des difficultés, établir des jugements et résoudre des problèmes, prendre des décisions et parvenir à ses fins?

On ne peut en effet parler de connaissance sans parler du pouvoir inférentiel qui y est attaché: la connaissance n'existe pas sans les problèmes et les insatisfactions qui lui procurent ses motivations, sans les actions qui la mettent en pratique, sans la raison qui lui donne sa dynamique interne. Inversement, si toute conceptualisation reflète nécessairement une intention d'utilisation de cette connaissance⁶, aucun raisonnement ne peut s'effectuer sans une base conceptuelle solide. De ce fait, raisonnement et conceptualisation sont les deux facettes de la connaissance et de son utilisation.

Mais existe-t-il une bonne conceptualisation? l'Intelligence Artificielle s'est souvent posé ce problème. La recherche de la base de connaissances universelle qui contienne tout le savoir de l'humanité est l'un des mythes fondateurs de l'IA, où

⁵On peut d'ailleurs toujours se demander ce qu'est effectivement le mouvement. La position aristotélicienne assimilant le mouvement à un passage de la puissance, ou potentialité, à l'acte, ou réalisation de cette potentialité, est toujours actuelle avec les notions de champs de potentiel et d'attracteurs de la physique moderne (Thom 1988; Boutot 1993).

⁶Sauf peut-être dans le domaine des mathématiques pures.

l'on retrouve l'aspiration toujours renouvelée des encyclopédistes, le projet de pouvoir réduire toutes les connaissances en un système conceptuel fini (Lenat et Guha 1990). Malheureusement, il n'existe pas de définition simple et unique d'un concept: tout repose sur l'utilisation que l'on désire en faire, sur l'aspect qui nous intéresse.

Par exemple le concept "d'eau", bien qu'apparaissant pourtant comme l'un des plus simple, n'est pas si facile à dégager. Tout le monde peut parler d'eau, comprendre un raisonnement qui utilise ce concept, mais il est cependant bien difficile d'en donner une vision unifiée et claire. En effet parler de l'eau, c'est mettre en présence un grand nombre de concepts:

1. Pour celui qui a soif, l'eau est un liquide que l'on boit et qui permet d'étancher sa soif.
2. Pour le plombier, l'eau est quelque chose qui coule le long des tuyaux, et qui peut s'échapper des canalisations (les fuites).
3. Pour le chimiste, c'est la molécule H_2O .
4. Pour le peintre, c'est un solvant et un diluant.
5. Pour celui qui lave quelque chose, l'eau enlève la saleté et rince la lessive. Etc.

Chacune de ces définitions renvoie à un savoir-faire, à une utilisation du concept dans un contexte approprié, qui lui même suppose toute une conceptualisation. Par exemple, l'eau, en tant qu'entité chimique, renvoie à la notion de molécule, à leurs capacités d'agrégation et de séparation, leur composition, les effets globaux qui peuvent en être déduits, etc. Ces renvois sont sans fin, toute la connaissance étant liée en une totalité complexe constituée d'une agrégation de points de vue particuliers.

La connaissance n'est cependant pas une affaire individuelle. Le spécialiste isolé n'existe pas. Bien que les perspectives des sciences cognitives — de la psychologie cognitive et de la philosophie en particulier — aient surtout mis l'accent sur la connaissance que pouvait se former un individu pour comprendre le monde qui l'entoure et résoudre les problèmes qui l'environnent, la connaissance d'un individu est largement le fruit d'interactions avec d'autres êtres humains. Comme le montre l'exemple des enfants-loups, si un enfant est isolé depuis son plus jeune âge et qu'il est privé de ses interactions avec ses semblables, il ne pourra jamais disposer des facultés de son espèce et devenir un individu accompli. Le manque d'interaction, et donc des connaissances, des communications et de la possibilité d'imiter le comportement des autres dans son enfance prive l'humain de son intelligence. Mais même sans aller jusqu'à cette extrémité, il faut bien reconnaître qu'un spécialiste n'est jamais seul. Il est entouré d'abord de tous ceux qui sont du même domaine que lui: l'apprentissage d'une spécialité s'effectue au cours d'un long développement pendant lequel il interagit avec ses maîtres, mais aussi avec ceux qui deviendront ses compagnons, ses collègues. Il est au centre ensuite d'un ensemble d'interactions avec d'autres spécialistes de domaines voisins. Le maçon travaille avec le plombier et le menuisier, le spécialiste des fuselages d'avion œuvre avec le motoriste et l'aérodynamicien. La

connaissance s'élabore dans le travail en commun, dans la confrontation des points de vues et des savoir-faire. De nombreux philosophes et sociologues des sciences et de l'activité scientifique, tels que Kuhn (Kuhn 1962), Latour (Latour 1989) ou Lestel (Lestel 1986), ont ainsi montré l'importance des relations sociales pour la stabilisation de concepts scientifiques ainsi que l'ensemble des stratégies utilisées par les chercheurs pour tirer parti du moindre allié lors de la production de prétendus "faits scientifiques".

La connaissance dans l'interaction

Comment, en effet, imaginer un être humain sans l'ensemble des communications qu'il entretient tout au long de sa vie? Comment penser l'intelligence d'un individu sans penser l'intelligence de la famille, de la tribu ou de la société dans laquelle cet individu s'est développé?

Le problème de la connaissance ne se pose pas seulement dans les termes d'une relation entre un agent et le monde, mais aussi dans les interactions qui ont lieu entre les agents, c'est-à-dire dans la définition des catégories conceptuelles élémentaires et des concepts communs dont les agents disposent pour communiquer et interagir. Les travaux de psychologie sociale cognitive montrent l'importance des considérations sociales lors de l'élaboration de catégories conceptuelles. Le fait, pour un individu, de se trouver en harmonie avec ses considérations antérieures et avec celles du groupe entraîne des modifications de l'univers cognitif et en particulier des modifications dans ses catégorisations et ses jugements.

De ce fait, la connaissance est avant tout le fruit d'interactions entre agents cognitifs qui, par un processus de confrontation, d'objections, de preuves et de réfutations (Lakatos 1976), élaborent des concepts, construisent des théories et formulent des lois. Ce processus d'édification par interactions a lieu jusque dans des domaines que l'on pourrait croire peu enclins aux controverses et aux polémiques comme les mathématiques. Il est d'autant plus à l'œuvre dans des sciences portants sur des domaines à la fois plus complexes et plus appliqués comme peuvent l'être les sciences humaines, l'économie, la psychologie et les sciences cognitives en général. La kénétique peut donc faire sienne le slogan suivant:

Il n'y a pas de connaissance sans interaction

où l'on conçoit par "interaction" aussi bien les confrontations avec le monde que les communications avec les autres agents, la connaissance étant alors définie comme la résultante des savoirs et des savoir-faire spécifiques que possèdent les agents, ces savoirs et savoir-faire étant les fruits des confrontations de l'individu avec le monde et des interactions que l'individu peut entretenir avec d'autres individus.

Ces interactions sont non seulement inévitables mais aussi nécessaires: un spécialiste, bien que disposant d'un savoir approfondi dans un domaine spécifique, ne peut tout connaître: ses connaissances sont limitées d'une part par la difficulté matérielle d'appréhender des savoirs spécialisés recouvrant plusieurs domaines, mais aussi par le fait que l'ensemble de ces connaissances ne forme pas nécessairement un tout homogène et cohérent.

Carl Hewitt a ainsi décrit un type d'organisation, qu'il appelle Systèmes Ouverts d'Information (Open Information Systems) (Hewitt 1985), dans lequel la connaissance n'est pas la somme des connaissances de tous les agents, mais la résultante de l'interaction de plusieurs micro-théories, c'est-à-dire de savoirs et savoir-faire associés à des agents. Chaque micro-théorie, bien que cohérente en elle-même, peut éventuellement entrer en conflit, voire même être en contradiction totale avec d'autres micro-théories issues de la même organisation. Lors d'action conjointes, les agents porteurs de ces micro-théories doivent alors négocier pour déterminer quelle est la meilleure action commune à entreprendre. Par exemple, dans une entreprise, le responsable du marketing pourra posséder une expertise de la forme suivante:

R1: augmenter la publicité fera augmenter les ventes et donc apportera des bénéfices à l'entreprise.

qui viendra en conflit avec celle du responsable de la gestion:

R2: pour augmenter les bénéfices de l'entreprise, il faut diminuer les dépenses.

En effet, l'application de la règle R2 condamne à l'avance les frais supplémentaires réclamés par le marketing. Ces deux règles, issues de points de vue différents, sont l'expression de deux conceptions divergentes, l'une centrée sur le marché, la seconde sur la gestion interne de l'entreprise. Lorsqu'une décision doit être prise, ces deux micro-théories viennent à s'opposer, et la décision proviendra d'une résolution de ce conflit soit en choisissant délibérément pour l'une ou l'autre, soit en tentant une synthèse locale qui pourrait par exemple s'exprimer sous la forme:

R3: si le marché est porteur et les coûts supplémentaires dus à la publicité peuvent être amortis par les ventes, alors on peut lancer une campagne de publicité.

Dans le même ordre de difficulté, il est intéressant de noter que cette notion de micro-théorie, dont la combinaison ne conduit pas à une théorie cohérente, a été adoptée dans le cadre du projet CYC pour la représentation d'une grande masse de connaissances communes correspondant à peu près à celles d'un enfant de 6-10 ans. A l'origine du projet, les connaissances étaient considérées comme globales (Lenat et Guha 1990). Il s'agissait alors de fusionner des théories apparemment inconsistantes et, surtout, dont l'effet était local. Devant les difficultés à rendre cohérentes de telles théories, Lenat et Guha reprirent à leur compte les idées de micro-théories tirées du travail de McCarthy (McCarthy 1980) et de Hewitt.

Une micro-théorie est alors une entité particulière dont la portée est locale et l'application relative à un contexte. Par exemple, la connaissance de physique naïve correspondant au fait que *quelque chose qui n'est pas supporté tombe* est représentée ainsi:

Théorie: $\forall x, \neg \text{supporté}(x) \Rightarrow \text{tombe}(x)$

Evidemment, cette théorie est locale: elle ne s'applique pas aux montgolfières ni aux objets interstellaires et ne concerne que les objets que l'on rencontre dans notre vie quotidienne. Il s'agit donc d'une micro-théorie car sa portée est limitée, mais elle s'avère suffisante pour le raisonnement commun dans la plupart des situations que nous rencontrons tous les jours.

Toutes ces notions de micro-théories s'intègrent très bien dans une perspective multi-agent: de par leur caractère essentiellement limité et relatif à un contexte, chaque micro-théorie est représentative de la connaissance d'un agent. Ou plus exactement, toute connaissance associée à un agent possède la valeur et l'importance d'une micro-théorie. Dans un système multi-agent aucun agent ne possède une connaissance totale, ni donc un vérité globale sur le monde. Tout savoir est par conséquent local et toute vérité relative.

La connaissance ne peut pas non plus se réduire à un savoir. Même si l'intelligence artificielle s'est longtemps faite le champion de la connaissance déclarative, du savoir défini en soi indépendamment de son usage, elle a depuis réalisé, dans le cadre de l'intelligence artificielle distribuée, que l'on ne pouvait pas comprendre la connaissance vécue, c'est-à-dire le sens, sans faire référence au comportement d'un agent situé dans un environnement et en relation avec d'autres agents. De ce fait, l'attention s'est portée plus sur le savoir-faire, ce que l'on appelle aussi la compétence et l'adresse d'un agent, que sur les connaissances "livresques" dont il pourrait être porteur.

En effet, pour réparer une conduite d'eau, le savoir théorique portant sur la mécanique des fluides est de peu de secours. En revanche, avoir déjà effectué de telles réparations ou avoir vu quelqu'un à l'œuvre dans une telle situation peut s'avérer utile. En d'autres termes, ce que l'on demande à un spécialiste porte plus sur l'ensemble des expériences qu'il a pu accumuler dans des circonstances semblables que sur l'étendue de son savoir effectif. Comme le signale Dreyfus (Dreyfus 1992), la tradition rationaliste suppose que comprendre un domaine et savoir agir avec compétence dans ce domaine revient à posséder une théorie explicite sur celui-ci. Il montre que les "experts", c'est-à-dire les personnes les plus compétentes, sont celles qui ont le moins de savoir explicite, mais qui font le plus appel à des expériences compilées sous la forme de capacité à reconnaître une situation caractéristique. Ainsi, plus une connaissance est considérée de haut niveau, plus elle est en fait spécifique et intégrée aux actions de l'individu qui ne se rend d'ailleurs plus compte qu'il en est le porteur, cette connaissance lui étant devenue totalement évidente.

Dans ces conditions, qu'est-ce que la connaissance? Nous avons dans un premier temps distingué la connaissance théorique et la connaissance pratique, la première reposant sur un corpus explicite de règles, de modèles formels et de théories, la seconde dépendant de la somme des situations semblables et des actions accomplies. Cependant, il est manifeste que dans les domaines techniques un tant soit peu compliqués, c'est-à-dire ceux pour lesquels les modèles naïfs comme celui de la chute des corps que nous avons vu précédemment ne sont plus d'aucun support, il est nécessaire de faire appel à des modèles théoriques plus poussés.

5.3.2 Représenter les connaissances et les croyances

Parler de connaissances pose le problème épineux du mode de représentation de cette connaissance. Existe-t-il un niveau de représentation particulier indépendant de toute représentation physique, ou bien doit-on prendre en compte le substrat matériel utilisé pour ces représentations? Pour le point de vue de l'IA classique, la connaissance peut se représenter sous la forme d'un ensemble de symboles exprimant des unités de connaissance. Cette approche, qui représente ce que l'on appelle le paradigme *symbolico-cognitivist*, est définie par les trois propositions suivantes (Andler 1992):

1. Les représentations sont indépendantes du substrat physique sous-jacent.
2. Les états mentaux sont représentationnels ou doués d'intentionnalité (au sens philosophique), c'est-à-dire qu'ils renvoient à quelque chose d'extérieur aux agents.
3. Ces représentations sont constituées de symboles assemblés en phrases ou en structures informatiques quelconques et raisonner consiste à manipuler ces structures pour en tirer d'autres structures.

A cette école classique, il faut opposer le point de vue connexionniste qui suppose que la connaissance se trouve intégralement répartie dans un réseau d'automates cellulaires sous la forme de valeurs numériques attachées aux connexions. Raisonner consiste à propager des valeurs numériques dans ce réseau, c'est-à-dire à modifier les connexions établies entre les différents éléments de ce réseau. Nous avons vu au chapitre 3 que ces automates sont utilisés comme des neurones formels, c'est-à-dire comme de petites unités qui se bornent à propager des valeurs numériques. L'intérêt de cette approche réside essentiellement dans la réalisation de systèmes réactifs capables d'apprendre et d'avoir des comportements adaptatifs en reliant directement les perceptions aux actions, sans faire intervenir explicitement de cognitions. Les systèmes qui en découlent sont parfois capables de performances relativement évoluées (analyse grammaticale, résolution de problèmes simples) mais leur domaine de prédilection est la perception (reconnaissance d'objets, de visages, etc.) et l'articulation entre perception et action (prise d'objets, évolution de petits robots mobiles, etc.). Les présupposés sont très différents de ceux développés par l'école symbolico-cognitivist, puisqu'ils reposent sur une conception subsymbolique (Smolensky 1992), l'objectif étant de représenter (ou de simuler) ce qui se passe dans le système nerveux d'une créature vivante en utilisant les forces d'association d'automates connectés par des liaisons pondérées.

Bien que peu développée jusqu'à présent, il existe une troisième voie, que nous pourrions appeler l'hypothèse kénétique ou interactionniste, qui postule que la connaissance d'un individu peut être considérée comme un système multi-agent à part entière, les concepts, notions et représentations étant alors des agents de nature particulière qui "vivent" à l'intérieur des agents. Le raisonnement d'un agent peut alors être vu comme la résultante des interactions de ces agents noétiques (Morin 1991) à la manière dont Minsky le suggère dans sa *Société de l'esprit* (Minsky 1988).

Cette approche est aussi définie par Dennett (Dennett 1991), qui tente d'élaborer une théorie de la conscience en prenant comme point de départ l'idée que le système cognitif d'un être humain est fondamentalement constitué d'entités indépendantes qui entrent en compétition et coopèrent⁷ pour donner une signification au monde environnant. Comme dans le cas de l'école symbolico-cognitiviste, les éléments cognitifs sont indépendants du substrat physique, mais comme dans l'école connexionniste, on ne suppose pas qu'il existe un sens associé à un ensemble de symboles, mais que ce dernier émerge des interactions entre le milieu extérieur et intérieur de l'agent, ce dernier étant lui-même composé d'entités autonomes en interactions (Varela et al. 1993).

Dans cet ouvrage, les agents cognitifs seront supposés être réalisés à partir de modèles cognitivo-symboliques, car ce sont les plus développés en intelligence artificielle distribuée pour décrire des agents disposant d'états mentaux. Même s'il semble que la deuxième école et surtout la troisième pourraient apporter des résultats intéressants et des perspectives nouvelles quant à la compréhension de l'élaboration de la connaissance dans les systèmes multi-agents, ces modèles demeurent essentiellement appliqués à des agents réactifs pour lesquels le problème des états mentaux est réduit à sa plus simple expression: la sélection d'actions pertinentes par rapport aux perceptions locales des agents.

L'hypothèse symbolico-cognitiviste

L'hypothèse symbolico-cognitiviste considère donc que les représentations accessibles à un individu s'expriment sous la forme de symboles, lesquels se réfèrent directement à des entités du monde dans lequel est plongé cet individu⁸. Ces symboles sont articulés à l'aide d'un langage interne, une sorte de "mentalais", défini par une grammaire formelle et par les opérations qu'il est possible de leur appliquer. La syntaxe de ces langages formels (généralement dérivés de la logique des prédicats du premier ordre à laquelle on ajoute des opérateurs modaux) décrit l'ensemble des règles grammaticales par lesquelles les unités peuvent s'agréger pour former des assemblages symboliques en combinant des expressions ou des structures (graphes).

Raisonnement consiste alors à manipuler ces assemblages de symboles pour former d'autres assemblages lors d'un processus d'*inférence*. Les opérations permettant de définir de telles inférences doivent être *valides*: elles doivent respecter des contraintes sémantiques et faire en sorte que les inférences produites engendrent bien elles-mêmes des assemblages valides, c'est-à-dire en correspondance avec leurs interprétations sémantiques. Ce mode de raisonnement, qui est à la base de la conception classique de la représentation des connaissances, est fondé sur les cinq conditions

⁷Le terme de *coopération* vient de nous. Bien que Dennett ne parle pas beaucoup des mécanismes sous-jacents permettant à ces entités d'interagir, il est possible de lire son ouvrage avec un œil de kénétiicien et de le considérer comme un allié dans l'idée que la connaissance peut être considérée comme un système multi-agent à part entière.

⁸On peut supposer qu'il existe des entités intermédiaires entre les symboles et les entités du monde, comme dans les sémantiques fondées sur la notion de mondes possibles (cf. section 5.3.3) ou les logiques intensionnelles comme celle de Montague (Montague 1974), mais, à une indirection près, le schéma de pensée est le même.

suivantes:

1. L'existence d'un ensemble de symboles.
2. L'existence de règles grammaticales de composition de symboles pour former des propositions ou des structures symboliques.
3. L'existence d'un monde de référence composé d'entités bien séparées auxquelles les symboles peuvent se référer.
4. L'existence d'une fonction d'interprétation qui donne un sens au symbole, c'est-à-dire qui attribue à chaque symbole un référent (ou une classe de référents).
5. L'existence d'un système d'inférence permettant de produire de nouveaux assemblages (propositions ou structures).

Les symboles

Mais qu'est-ce qu'un symbole? Pour le courant symbolico-cognitivist, un symbole⁹ est un élément particulier qui renvoie à quelque chose d'autre, ce qu'il représente justement et que l'on appelle son *référent*. Par exemple, les mots sont des symboles qui renvoient à des contenus. Le mot **Jean** renvoie au vrai *Jean*¹⁰ que je connais, le symbole 2 renvoie au deuxième élément des nombres naturels, c'est-à-dire au successeur du successeur de zéro, etc. Les relations n-aires de la forme $p(a_1, \dots, a_n)$ renvoient à des contenus représentés sous la forme d'ensembles de n-uplets. Par exemple, dire que *Jean est une personne*, ce que l'on écrirait formellement en logique $\text{personne}(\text{Jean})$ revient à dire que *Jean* fait partie de l'ensemble *personne* composé de la manière suivante:

$$\text{personne} = \{ \textit{Jean}, \textit{Marie}, \textit{Paul}, \textit{Pierre}, \textit{Marc}, \dots \}$$

De même dire que Jean aime Marie, ce qui s'écrit $\text{aime}(\text{Jean}, \text{Marie})$ consiste à affirmer que le couple $\langle \textit{Jean}, \textit{Marie} \rangle$ appartient à l'ensemble *quiAimeQui* composé des couples $\langle a, b \rangle$ dans lesquels l'individu *a* aime l'individu *b*:

$$\text{quiAimeQui} = \{ \langle \textit{Jean}, \textit{Marie} \rangle, \langle \textit{Marie}, \textit{Pierre} \rangle, \dots \}$$

Dans l'hypothèse symbolico-cognitivist, l'intellect d'un agent cognitif est composé de ces symboles. Supposons que Jean ait certaines connaissances sur les chats. Il connaît par exemple Minou, Tom et Sylvestre, et il sait (ou il croit, nous reviendrons sur cette distinction plus bas) que les chats qui ronronnent sont heureux. Cela suppose donc que sa base de connaissance est constituée des expressions symboliques suivantes:

⁹On utilisera ici le mot symbole dans son acception cognitiviste, c'est-à-dire comme élément signifiant renvoyant à un contenu, ce qui le rend synonyme de "signe" en linguistique. Il ne faut pas le confondre avec la notion de "symbole" comme mécanisme et support d'accession à des significations spirituelles et subconscientes.

¹⁰Il est important de pouvoir distinguer le symbole de son référent. Nous utiliserons des formes typographiques différentes pour ces deux notions: les symboles (ainsi que les entités informatiques) seront inscrit en caractères **courier** alors que les référents seront écrits en *italique*.

$$\text{chat}(\text{Minou}), \text{chat}(\text{Tom}), \text{chat}(\text{Sylvestre}) \\ \forall x. [\text{chat}(x) \Rightarrow (\text{ronronne}(x) \wedge \text{heureux}(x))]$$

Sémantique et référence

Tant que l'on n'a pas décrit ce que ces symboles désignent, il ne s'agit que de simples agrégats syntaxiquement corrects. Pour les faire accéder au rang de symboles, il faut leur adjoindre une *sémantique*, c'est-à-dire la capacité d'interpréter ces expressions par rapport à ce que l'on appelle un modèle. Par exemple, il est impossible de dire si l'expression

$$\forall x \exists y P(x, y)$$

est vraie ou fausse tant que l'on ne sait pas à quoi se rattachent les symboles \forall , \exists , P , x et y et où ces symboles prennent leur valeur. En revanche, si l'on dit que le domaine de référence des variables est l'ensemble des entiers naturels, que P signifie "inférieur ou égal", alors il est possible de se déclarer sur la vérité ou la fausseté de la formule précédente en considérant qu'en effet on peut associer à tout entier x au moins un autre entier dont il est inférieur ou égal. Cette association entre les symboles et les choses d'un domaine D est définie par une fonction d'interprétation. Dans notre cas, la fonction d'interprétation I pourrait être la suivante:

$$\begin{aligned} I(\text{Minou}) &= \text{Minou} \\ I(\text{Tom}) &= \text{Tom} \\ I(\text{Sylvestre}) &= \text{Sylvestre} \\ I(\text{ronronne}) &= \text{Minou} \\ I(\text{heureux}) &= \text{Minou}, \text{Sylvestre}, \text{Jean} \\ I(\text{chat}) &= \text{Minou}, \text{Tom}, \text{Sylvestre} \\ I(\text{personne}) &= \text{Jean}, \text{Marie}, \text{Paul}, \text{Pierre}, \text{Marc}, \dots \\ I(\text{aime}) &= \langle \text{Jean}, \text{Marie} \rangle, \langle \text{Marie}, \text{Pierre} \rangle, \dots \\ I(\text{Jean}) &= \text{Jean} \end{aligned}$$

Savoir si une proposition est vraie pour une interprétation I , ce que l'on note \models_I consiste à vérifier, pour les formules atomiques, que les constantes apparaissant dans un prédicat (ou la liste des constantes) appartiennent bien au domaine d'interprétation du prédicat:

$$\begin{aligned} \models_I P(a) \text{ ssi } I(a) \in I(P) \\ \models_I R(a_1, \dots, a_n) \text{ ssi } I(\langle a_1, \dots, a_n \rangle) \in I(R) \end{aligned}$$

Il est possible de combiner les formules à l'aide d'opérateurs tels que \neg (non), \wedge (et), \vee (ou), \Rightarrow (implique), etc. et, dans ce cas, on vérifie la *satisfiabilité* d'une formule en analysant ses différents composants à partir des règles suivantes:

$$\begin{aligned} \models_I \neg \phi \text{ ssi } \not\models_I \phi \\ \models_I \phi \wedge \psi \text{ ssi } \models_I \phi \text{ et } \models_I \psi \\ \models_I \phi \vee \psi \text{ ssi } \models_I \phi \text{ ou } \models_I \psi \\ \models_I \phi \Rightarrow \psi \text{ ssi } \not\models_I \phi \text{ ou } \models_I \psi \end{aligned}$$

$$\begin{aligned} \models_I \forall x.\phi \text{ ssi } I_{x/d}(\phi) \text{ pour tout } d \in D \\ \models_I \exists x.\phi \text{ ssi il existe } I_{x/d}(\phi) \text{ pour au moins un } d \in D \end{aligned}$$

où $I_{x/d}(\phi)$ représente l'interprétation de la formule ϕ dans laquelle on a remplacé la variable x par l'élément d du domaine d'interprétation D .

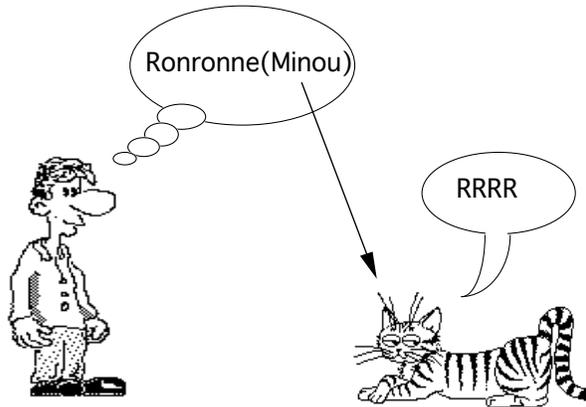


Figure 5.2: En logique, les symboles se réfèrent à des objets et à des situations du monde.

Par exemple, l'expression $\text{chat}(x) \wedge \text{heureux}(x)$ est vraie si effectivement l'interprétation de x appartient à la fois à l'ensemble des chats et à celui des heureux. De ce fait, la formule $\text{chat}(\text{Minou}) \wedge \text{heureux}(\text{Minou})$ est vraie puisque *Minou* appartient à la fois à *chat* et à *heureux*. De même la formule,

$$\forall x. [\text{chat}(x) \wedge \text{ronronne}(x) \Rightarrow \text{heureux}(x)]$$

est vraie puisque soit $\text{heureux}(x)$ est vrai ($x \in \text{heureux}$), soit $\text{chat}(x) \wedge \text{ronronne}(x)$ est faux, c'est-à-dire que x n'appartient pas à l'intersection de *chat* et *ronronne*.

La formalisation de la logique a permis d'avoir des propriétés générales sur des systèmes logiques indépendamment des interprétations que l'on peut avoir sur les symboles. En particulier, il est possible de montrer que, quelle que soit l'interprétation I adoptée, la formule

$$\models \forall x.P(x) \Rightarrow P(x)$$

est vraie, ce que l'on note en supprimant toute mention à une interprétation dans le signe \models . On dit alors qu'une telle formule est *valide*.

Système formel et inférence

Le raisonnement logique s'appuie sur la notion d'inférence et de système formel. La notion d'inférence est très ancienne (Aristote est généralement considéré comme l'inventeur de la logique) puisqu'elle a toujours été considérée comme une réflexion générale sur le raisonnement. Avec l'apparition des logiques modernes issues des travaux initiaux de Boole, Frege et Russell, le processus inférentiel est devenu

purement mécanique et “formel”, à partir de la notion de système formel qui détermine les lois de traitement syntaxique permettant d’assembler et de produire de manière cohérente des expressions à partir d’expressions plus élémentaires et considérées comme vraies. Un système formel est composé de:

1. Un ensemble de symboles dont le sens est purement arbitraire et qui n’ont par eux-mêmes *a priori* aucun lien avec ce qu’ils représentent.
2. Un ensemble de règles grammaticales permettant de construire des formules bien formées par assemblages de formules plus élémentaires.
3. Un ensemble de formules de départ, appelées axiomes.
4. Un mécanisme déductif permettant de produire de nouvelles formules, appelées théorèmes, à partir des axiomes de départ et des théorèmes déjà produits.

Le processus déductif suit un mécanisme itératif, c’est-à-dire que les conclusions sont obtenues après un certain nombre de “pas” de déduction. Chaque pas de ce processus doit suivre une *règle d’inférence* qui stipule la manière de produire de nouvelles formules. Une règle d’inférence s’écrit:

$$\phi_1, \dots, \phi_n \vdash \phi_n + 1$$

ce qui signifie qu’à partir des formules ϕ_1, \dots, ϕ_n , il est possible de produire la formule $\phi_n + 1$. La plus célèbre et la plus utilisée des règles d’inférence s’appelle le *modus ponens* et s’écrit ainsi:

$$\phi, \phi \Rightarrow \psi \vdash \psi$$

Elle stipule que, s’il existe une formule de la forme ϕ et une de la forme $\phi \Rightarrow \psi$, alors il est possible d’en inférer la formule ψ . Par exemple, de **il pleut** et de **il pleut \Rightarrow la chaussée est mouillée** il est possible de déduire **la chaussée est mouillée** en une seule étape.

L’un des systèmes logiques les plus connus en IA, que nous avons utilisé pour présenter la notion de système logique, est celui de la *logique des prédicats du premier ordre*. Conçue initialement pour l’analyse des fondements des mathématiques, cette logique permet de représenter de manière très élégante des connaissances certaines, universelles et intemporelles, c’est-à-dire qui expriment des vérités indépendantes de toute situation réelle d’énonciation. Comme nous l’avons vu, ce type de connaissance s’oppose à la notion de micro-théorie considérée précédemment. Du fait de son caractère trop simpliste et de son incapacité à traiter de notions aussi communes que le temps, les croyances, les actions, les doutes, etc., de nombreux autres systèmes logiques ont été élaborés. Cependant, elle reste très utilisée, en particulier par le fait qu’elle se trouve souvent à la base des autres logiques considérées ainsi comme des extensions de la logique des prédicats. En particulier, la logique modale des croyances qui est présentée plus bas s’appuie intégralement sur la logique des prédicats du premier ordre.

Généralement, les systèmes formels élaborés à partir de cette conception sont *compositionnels* et *réducteurs*. Ces deux termes, bien que généralement utilisés

ensembles, ne sont pas équivalents. On dit qu'une sémantique est compositionnelle si le sens d'une expression complexe est fonction du sens de ses constituants. Elle est réductrice s'il existe des primitives sémantiques, c'est-à-dire des symboles dont le sens n'est pas défini par composition du sens de ses constituants mais par un mécanisme de renvoi direct entre le symbole et la chose référencée. Les formalismes logiques, et en particulier la logique des prédicats du premier ordre et ses extensions modales, sont compositionnels (avec certaines restrictions pour les logiques modales comme nous le verrons plus loin) et réducteurs.

Il existe d'autres formalismes de représentation des connaissances. L'alternative la plus célèbre repose sur la notion de réseau sémantique, de frames ou d'objets, ces trois notions étant relativement imbriquées. Un réseau sémantique est un graphe étiqueté constitué de nœuds qui représentent des concepts et d'arcs qui relient ces concepts entre eux. Ce type de représentation des connaissances est très utilisé dans le domaine de la compréhension du langage naturel où des réseaux sémantiques rendent compte de la structure profonde des phrases. L'un des modèles les plus en vigueur actuellement est celui des graphes conceptuels de J. Sowa (Sowa 1984). D'autres langages de représentations des connaissances fondés sur la notion d'objets ou de "frames" sont couramment utilisés dans des applications d'intelligence artificielle. Un objet est une structure informatique décrite par un aspect statique qui qualifie son état et les liens qu'elle peut entretenir avec d'autres objets, et un aspect dynamique constitué de l'ensemble des opérations qui lui sont applicables. On pourra se reporter à (Ferber 1990) pour une introduction à la programmation et à la conception par objets, et à (Masini et al. 1989) pour une vue générale sur les langages à objets et les systèmes de "frames".

5.3.3 Logiques des savoirs et des croyances

Croire quelque chose, c'est admettre comme véritable une proposition, un fait, une histoire, sans que l'on sache effectivement son degré de vérité. Dans la langue quotidienne, on utilise souvent le terme "penser que" pour exprimer une croyance. Par exemple, si l'on dit "je pense que le facteur a distribué le courrier aujourd'hui", cela signifie que l'on a souscrit à l'affirmation "le facteur a distribué le courrier aujourd'hui" sans pour autant en être absolument sûr. Pour s'en assurer, il faut soit pouvoir confirmer ce fait par observation — en allant voir la boîte aux lettres — soit "croire" les personnes ou les choses qui ont affirmé cette proposition. Par exemple, si vous êtes en vacances, et qu'au téléphone on vous dise "le facteur n'a pas distribué le courrier aujourd'hui", vous n'avez aucun moyen d'aller le vérifier par vous-même et vous devez vous en remettre aux dires de cette personne. Mais il n'y a pas que les personnes que l'on doit croire dans la vie courante. Les objets inanimés aussi affirment des énoncés que l'on doit croire et que l'on ne peut pas (ou l'on ne veut pas) toujours vérifier. Par exemple, si la jauge d'essence de votre voiture est sur "plein", cela ne signifie pas effectivement que le réservoir est rempli d'essence. Si la jauge ne fonctionne pas, le réservoir est peut-être vide... Cependant, cette croyance peut avoir des conséquences sur vos actions futures. Si vous croyez que le réservoir est plein, vous pouvez envisager une longue conduite sans reprendre

d'essence, alors que si vous croyez qu'il est vide vous chercherez à faire le plein le plus rapidement possible. Si vous croyez que la jauge est cassée, vous utiliserez d'autres manières pour évaluer la quantité d'essence qui vous reste, peut-être en estimant votre consommation à partir de "croyances" sur la distance parcourue, en faisant confiance à d'autres instruments tels que le compteur kilométrique de votre voiture, ou les distances portées sur les cartes, etc. Mais vous n'échapperez pas aux croyances en certaines propositions et aux confiances que vous pouvez avoir quant aux choses.

Ces considérations précédentes n'ont qu'un seul but, montrer d'une part que tout notre raisonnement repose sur des croyances et d'autre part que la vérification de ces croyances ne peut-être effectuée que par des méthodes d'interaction avec d'autres entités (la personne au téléphone, la jauge d'essence) dont la fiabilité est elle-même l'œuvre d'autres croyances. En d'autres termes, il n'existe pas de "faits" en tant que tels: seulement des croyances qui n'ont pas été encore réfutées.

D'une manière plus philosophique, tout savoir est une croyance: il s'agit toujours d'une information rapportée par quelque chose ou quelqu'un d'autre, et cette information peut-être sujette à erreur. Ce que l'on voit n'est qu'une illusion donnée par nos sens, comme l'ont bien compris les empiristes Hume et Berkeley: les sens filtrent et traduisent la réalité, nous présentant les choses au travers d'une transformation parfois déformante. L'information est, elle aussi, l'objet d'une croyance. Ce que l'on apprend par les journaux est peut-être faux (en particulier si le journal est daté du 1er avril) ou tout simplement déjà plus vrai du fait de la périodicité du journal. De même, les lois et les résultats scientifiques, images même de la certitude doivent être considérées comme des croyances adéquates reposant sur des propositions scientifiques, c'est-à-dire réfutables, qui pourront être remises en question le moment venu (Popper 1963).

A la notion de croyance est associée celle de révision: toute information qui fait l'objet d'une croyance peut-être remise en question. Cette capacité de pouvoir remettre en question un fait, une déduction, une loi ou un jugement est à la base de notre capacité d'adaptation cognitive, c'est-à-dire la faculté que nous avons, êtres humains, d'accommoder notre système cognitif à un monde en perpétuelle évolution, cette évolution étant le fruit des actions des différents agents intervenants dans notre univers.

Il en est de même pour les agents intervenant dans le cadre de l'IAD, qui doivent être capable de remettre en question leurs informations et de réviser leur jugements. Par exemple, dans le problème de la poursuite (chap. 1), les prédateurs doivent réviser leurs croyances concernant la position de la proie et décider de l'action à entreprendre en fonction de cette croyance. Si le temps de réaction et de propagation des communications n'est pas négligeable, les prédateurs disposeront parfois d'informations erronées, la situation de la proie ayant changé entre-temps.

Enfin, les croyances ne sont pas universelles: elles sont relatives à un individu, un groupe, une société. Ce que croit Jean est différent de ce que croit Pierre: un prédateur peut disposer d'informations plus "fraîches" qu'un autre ou un agent peut disposer de compétences plus spécialisées lui permettant de mieux apprécier la validité d'un énoncé. Et comme nous l'avons vu plus haut dans le cadre des connaissances, rien n'indique que toutes ces croyances soient cohérentes. De ce fait,

la relativité des croyances aux individus prend une importance majeure dans le cadre de systèmes multi-agents, car elles mettent en valeur la localisation nécessaire des informations et la nécessité de communiquer pour pouvoir transmettre et confronter des croyances.

Formaliser les croyances

L'étude des croyances en philosophie puis en logique est le fruit d'une longue tradition qui remonte aux Grecs, quoique l'idée de pouvoir formaliser et donner une forme calculable aux croyances date du début des années 60, avec l'œuvre fondamentale de Hintikka (Hintikka 1962). Ces années virent un développement considérable de théories formelles concernant les croyances et les savoirs. Mais l'avènement des systèmes multi-agents et, en particulier, de l'essor des systèmes cognitifs coopératifs, a entraîné un regain d'intérêt pour toutes ces théories en tenant compte des limites imposées par leur mise en œuvre effective dans des environnements informatiques.

Le principe général de ces logiques est de définir un ensemble d'axiomes portant sur les croyances, puis ensuite de tenter de leur donner une sémantique en les faisant référer à des éléments d'une structure définie mathématiquement.

Mais a-t-on besoin d'une logique particulière pour les croyances? Pourquoi ne pas utiliser la logique des prédicats du premier ordre? En effet, une solution pratique serait d'utiliser une formule atomique spécifique telle que `croire(A,P)` pour indiquer que l'agent A croit la proposition P. Cependant, cette formulation se heurte à l'une des propriétés essentielles de la logique des prédicats qui est de pouvoir substituer les expressions identiques dans toutes les formules où elles apparaissent. Par exemple si Jean croit que l'auteur de *La Légende des siècles* est l'auteur de *Notre-Dame* et qu'il s'avère que Victor Hugo est bien l'auteur de la légende des siècles, ce que l'on peut écrire ainsi:

```
croire(Jean, auteur(auteurDe("Légende des siècles"),
                    "Notre-Dame"))
VictorHugo = auteurDe("Légende des siècles")
```

alors il s'ensuit logiquement que Jean croit que Victor Hugo est bien l'auteur de *Notre-Dame*, ce qui n'est peut-être pas une croyance de Jean. Ainsi, la propriété de substitution des identiques entraîne des déductions fausses en ce qui concerne les croyances. L'erreur vient du fait que la base de connaissances de Jean, dans laquelle se trouvent toutes ses croyances, constitue un contexte "opaque" à la référencement. De ce fait, si l'on veut pouvoir traiter la notion de croyance, il est nécessaire de pouvoir s'affranchir des limites de la logique des prédicats du premier ordre et d'utiliser une autre logique capable "d'opacifier" les contextes de croyances.

Ces *logiques épistémiques* qui traitent des savoirs et des croyances utilisent des formulations plus ésotériques que l'emploi d'un simple prédicat `croire` ou `savoir` et étendent la logique des prédicats du premier ordre pour prendre en compte toutes les subtilités des croyances et des savoirs. Croyances et savoirs ont de nombreux points communs. Traditionnellement, on les distingue en considérant que les savoir

sont des croyances vraies, c'est-à-dire des énoncés qui sont connus de l'agent et qui sont en conformité avec le monde. La notion de savoir peut donc se définir ainsi:

$$\text{savoir}(A, P) = \text{croire}(A, P) \wedge \text{vrai}(P)$$

signifie que si un agent sait P , alors P est naturellement vrai, ce que l'on peut formuler:

$$\text{savoir}(A, P) \Rightarrow P$$

Nous traiterons donc ensemble savoirs et croyances et ne distinguerons ces deux types de relation au monde que si cela s'avère nécessaire.

Les mondes possibles

La formulation classique d'une logique épistémique repose sur la notion de logique modale et leur analyse sémantique sur le modèle des *mondes possibles*. Ce dernier, initialement dû à Kripke (Kripke 1991) pour les logiques du possible et du nécessaire (les logiques *ontiques*), a été repris par Hintikka pour la formalisation des savoirs et des croyances.

L'idée intuitive est de considérer que chaque agent peut envisager un certain nombre de mondes qu'il imagine envisageables. On dit alors qu'un agent croit une proposition si elle est vraie dans tous les mondes qu'il imagine. Par exemple, si Jean croit que le réservoir d'essence de sa voiture est plein, cela signifie qu'il peut envisager plusieurs mondes possibles — dans certains Napoléon est mort à Sainte-Hélène, dans d'autres il est mort à l'île d'Elbe, dans d'autres encore Napoléon est mort aux Tuileries, Jean n'a pas de croyances sur la mort de Napoléon — mais que dans tous ces mondes le réservoir d'essence de sa voiture est plein.

Pour exprimer cette idée, il est nécessaire d'avoir un langage de représentation des connaissances. De manière classique, nous utiliserons un langage modal, c'est-à-dire un langage correspondant à celui de la logique propositionnelle ou à celui de la logique des prédicats, auquel on ajoutera un opérateur de croyance ou de savoir, ce que l'on appelle des modalités.

Par la suite nous noterons $\text{croire}(A, P)$ pour indiquer que l'agent A croit la proposition P , c'est-à-dire que l'opérateur modal croire est appliqué à la proposition P et indexé par l'agent A . Par exemple, la phrase *Jean croit que le réservoir de Titine est plein* s'exprime ainsi:

$$\text{croire}(\text{Jean}, \text{plein}(\text{réservoir}(\text{Titine})))$$

où Titine est la voiture de Jean. En terme de sémantique des mondes possibles, cela signifie qu'il existe un ensemble de mondes, appelé W , tels que, à partir du monde W_0 considéré comme le monde "réel", Jean ne voit (ou n'imagine) que des mondes dans lesquels le réservoir de sa voiture est plein. La figure 5.3 représente cette situation (on suppose de plus qu'il existe une règle de la forme $\forall x. \text{vide}(x) \Rightarrow \neg \text{plein}(x)$).

Dans certains des mondes que Jean imagine Titine est neuve, dans d'autres elle ne l'est plus. Parfois, Napoléon est mort à Sainte-Hélène, mais dans tous les cas

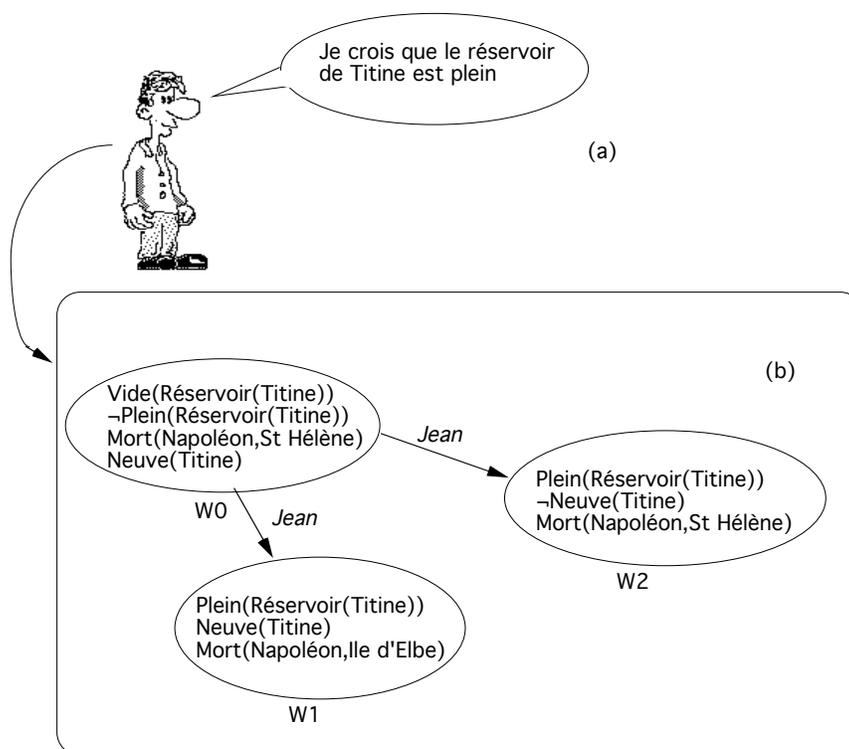


Figure 5.3: Un modèle de mondes possibles (b) correspondant aux croyances de Jean (a)

il n'imagine que des mondes dans lesquels le réservoir de Titine est plein, et c'est pour cela que l'on peut dire que Jean croit effectivement cette proposition. De plus, Jean croit quelque chose de faux. En effet, il est dit dans le monde originel W_0 que le réservoir n'est pas plein (il est même vide), mais cela Jean ne le voit pas. Plus formellement, on appelle interprétation en termes de mondes possibles, un modèle de Kripke M défini comme un n-uplet

$$M = \langle W, I, R_1, \dots, R_n \rangle$$

où W est un ensemble de mondes possibles (ou d'états possibles du monde si l'on préfère), où R_i est une relation d'accessibilité associée à l'agent i , et où I est une fonction de valuation qui associe une valeur de vérité à une proposition et à un monde, c'est-à-dire une fonction de type

$$I : Proposition \times W \rightarrow \{Vrai, Faux\}$$

L'interprétation d'une proposition contenant une croyance peut alors être donnée par une relation de *conséquence sémantique* notée \models où $M, w \models \phi$ signifie que ϕ est vrai dans le monde w du modèle M . Cette relation est définie récursivement ainsi dans le cadre d'une logique modale propositionnelle¹¹:

¹¹Nous ne définirons ici que la relation sémantique pour une logique modale dont les énoncés sont des propositions. L'extension de ce type de sémantique aux logiques modales prédicatives est immédiate. En particulier on pourra se référer au livre (Genesereth et Nilsson 1987) pour une présentation de cette extension.

$$\begin{aligned}
& M, w \models \phi \text{ si } \phi \text{ est une proposition primitive et } I(\phi, w) = \text{Vrai} \\
& M, w \models_I \neg\phi \text{ ssi } M, w \not\models_I \phi \\
& M, w \models_I \phi \wedge \psi \text{ ssi } M, w \models_I \phi \text{ et } M, w \models_I \psi \\
& M, w \models_I \phi \vee \psi \text{ ssi } \models_I \phi \text{ ou } \models_I \psi \\
& M, w \models_I \phi \Rightarrow \psi \text{ ssi } M, w \not\models_I \phi \text{ ou } M, w \models_I \psi \\
& M, w \models_I \text{croire}(a, \phi) \text{ ssi} \\
& \quad \text{pour tout monde } w_i \text{ tel que } w R_a w_i, M, w_i \models_I \phi
\end{aligned}$$

Il ne suffit pas de donner une sémantique à un opérateur modal. Encore faut-il être capable de caractériser les différentes propriétés que l'on attache à ces formes de connaissances. Celles-ci peuvent être définies par un ensemble d'axiomes qui caractérisent les aspects inférentiels des croyances.

Le premier de ces axiomes, l'*axiome de distribution*, porte sur le fait que les croyances ou les savoirs d'un agent suivent le principe du modus ponens: si un agent croit une implication entre deux termes, alors s'il croit le premier, il croit aussi le second. Cet axiome, qui est souvent considéré comme l'élément principal des logiques des savoirs et des croyances, peut avoir plusieurs expressions équivalentes:

$$\text{croire}(A, (P \Rightarrow Q)) \Rightarrow \text{croire}(A, P \Rightarrow \text{Croire}(A, Q)) \quad (K)$$

$$\text{croire}(A, P) \wedge \text{croire}(A, (P \Rightarrow Q)) \Rightarrow \text{croire}(A, Q)$$

L'*axiome de non contradiction* caractérise les facultés logiques d'un agent. S'il croit quelque chose, alors il ne croit pas son contraire, ce que l'on note:

$$\text{croire}(A, P) \Rightarrow \neg\text{croire}(A, \neg P) \quad (D)$$

par exemple, si Jean croit que son réservoir d'essence est plein, alors il ne croit pas que son réservoir est vide.

Les agents cognitifs peuvent être considérés comme des agents conscients de leur propres croyances. Cela indique que si un agent croit quelque chose, alors il croit qu'il le croit. Cette propriété est matérialisée par les axiomes d'*introspection positive* et *négative*:

$$\text{croire}(A, P) \Rightarrow \text{croire}(A, \text{croire}(A, P)) \quad (4)$$

$$\neg\text{croire}(A, P) \Rightarrow \text{croire}(A, \neg\text{croire}(A, P)) \quad (5)$$

L'axiome de savoir considère que tout ce que croit un agent est vrai, et est noté ainsi:

$$\text{croire}(A, P) \Rightarrow P \quad (T)$$

Cet axiome n'est pas adapté aux croyances: il est en effet possible d'avoir des croyances fausses, et c'est d'ailleurs ce que nous avons reconnu comme étant l'une des caractéristiques fondamentales des croyances. Lorsque Jean croit que son réservoir est plein, il se peut qu'il soit vide. En revanche, la notion de savoir, telle qu'elle est utilisée en logique, suppose que tout savoir est vrai, c'est-à-dire qu'un agent ne peut savoir quelque chose de faux. De ce fait, l'axiome de connaissance, qui est l'un des pivots des logiques des savoirs, ne sera pas utilisé pour la représentation des croyances.

Il est possible de panacher ces propriétés et de ne conserver que les axiomes que l'on désire. Dans ce cas on obtiendra à chaque fois un système logique différent, dans lequel certaines déductions pourront être faites et pas d'autres. La figure 5.4 (Konolige 1986) représente l'ensemble des systèmes modaux correspondant à ces axiomes ainsi que leurs liens de dépendance.

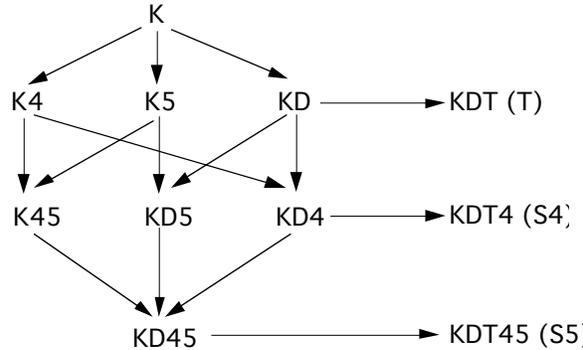


Figure 5.4: Les différentes logiques modales définissables à partir des axiomes K,D,T, 4 et 5. Une flèche $X \rightarrow Y$ indique que tous les théorèmes de X se retrouvent intégralement dans Y. Les noms entre () correspondent à leur dénomination usuelle.

L'intérêt de la sémantique des mondes possibles est de pouvoir relier les axiomes de ces systèmes logiques modaux aux propriétés des relations d'accessibilité. L'axiome K est indispensable parce qu'il permet d'interpréter les propositions modales en termes de mondes possibles. L'axiome T est caractéristique d'une relation d'accessibilité réflexive. Lorsqu'elle est en plus transitive on obtient le système S4. Lorsqu'il s'agit de relations d'équivalence, c'est-à-dire si l'accessibilité est réflexive, symétrique et transitive, le système d'axiome correspondant est S5. Les logiques des savoirs utilisent généralement les logiques S4 ou S5. En revanche les logiques des croyances, qui ne peuvent accepter l'axiome T, utilisent généralement KD45, la plus forte des logiques des croyances.

L'autre intérêt de la sémantique des mondes possibles pour les systèmes multi-agents est de pouvoir représenter facilement les croyances mutuelles que les agents peuvent avoir les uns envers les autres et de prendre en compte un ensemble d'assertions telles que:

```

croire(Jean,plein(réservoir(Titine)))
croire(Marie,vide(réservoir(Titine)))
croire(Marie,croire(Jean,vide(réservoir(Titine))))
  
```

dans laquelle Marie croit non seulement que le réservoir de Titine est vide, mais aussi que Jean croit que le réservoir est vide, alors que ce dernier croit que le réservoir est plein.

Cela signifie d'abord simplement que dans tous les mondes accessibles initialement par Jean (c'est-à-dire accessible depuis W_0) le réservoir de Titine est plein et que dans tous les mondes accessibles initialement par Marie, le réservoir est vide. La croyance mutuelle est précisée en disant que dans tous les mondes accessibles par Marie, la proposition

`croire(Jean, vide(réservoir(Titine)))`

est vraie, c'est-à-dire que dans tous les mondes accessibles par Jean à partir des mondes initialement accessibles par Marie, la proposition

`vide(réservoir(Titine))`

est vérifiée, comme le montre la figure 5.5.

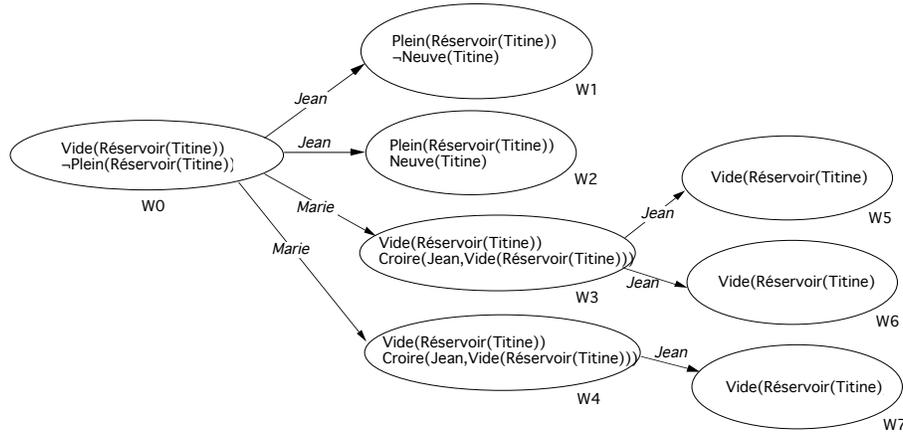


Figure 5.5: Un modèle de mondes possibles indiquant les croyances respectives de Jean et de Marie. Celle-ci croit que Jean croit que le réservoir est vide alors qu'il croit en fait qu'il est plein.

Limites de la sémantique des mondes possibles

Bien que très utilisée en IAD pour décrire les croyances et les savoirs des agents, la sémantique des mondes possibles et les logiques modales qui y sont associées sont très controversées. La première de ces controverses concerne l'*omniscience logique* des agents ainsi définis.

La sémantique des mondes possibles réclame absolument une règle, appelée règle de nécessité qui indique que toute formule valide (c'est-à-dire universellement vraie quel que soit le modèle) fait nécessairement partie des croyances de l'agent, ce que l'on exprime de la manière suivante:

$$\text{pour tout agent } a, \frac{\vdash \phi}{\vdash \text{croire}(a, \phi)}$$

L'inconvénient, c'est que cette règle, lorsqu'elle est associée à l'axiome K, impose à l'agent d'être capable de déduire toutes les conséquences logiques de ses croyances et en particulier celles qui proviennent des conséquences universellement valides. Par exemple si Jean croit que le réservoir de sa voiture est plein, et qu'il soit vrai que sa voiture puisse rouler 200 kilomètres sans avoir besoin de reprendre de l'essence, alors, d'après la sémantique des mondes possibles, Jean doit nécessairement croire qu'il pourra rouler 200 kilomètres. Mais cela n'est peut-être pas une croyance de

Jean, s'il croit que sa voiture peut rouler pendant 300 km sans faire le plein. Dans ce système, on ne peut donc pas croire quelque chose qui soit en contradiction avec un fait universellement vrai. Les croyances fausses sont donc interdites, ce qui est pourtant le propre des croyances. De plus cette sémantique ne prend pas en compte le fait que les agents disposent de mémoire et de processeurs limités.

On ne peut donc se sortir de ces difficultés qu'en supposant que cette sémantique définit des agents dont les performances de déduction sont "idéales". On fera aussi attention à limiter au maximum le nombre de formules universellement valides de telle manière que des déductions non voulues ne puissent pas être engendrées par le système.

Logiques opérationnelles

Le problème de l'omniscience logique ne constitue pas la seule difficulté que doivent affronter les logiques modales. Il existe aussi des problèmes métaphysiques concernant l'existence de ces "mondes possibles" et de leur construction. C'est pourquoi d'autres modèles de raisonnement sur les croyances et les savoirs ont été proposés, ces modèles ne reposant plus sur l'existence de ces mondes, mais sur les capacités mécaniques d'un agent de produire de nouvelles assertions. On parle alors de logique *opérationnelle* ou de sémantique *phrasistique* (sentential semantics) pour indiquer que la sémantique d'une expression est donnée en termes de manipulation formelles d'énoncé et non par référence à un modèle de monde, que ce modèle traite de mondes "réels" ou de mondes "possibles".

A chaque agent a on associe une base de connaissance Δ_a composée de formules de logique "ordinaire" (logique propositionnelle classique ou logique des prédicats du premier ordre) et d'un ensemble de règles d'inférence noté Γ_a . Combinées, formules et règles d'inférence permettent de définir l'ensemble Θ_a qui comprend l'ensemble des formules initiales plus celles qui peuvent être dérivées en utilisant les règles d'inférences de Γ_a et que l'on appelle la *théorie* de l'agent a . On note $\Delta_a \vdash_a \phi$ le fait que la proposition ϕ soit vraie dans la théorie Θ_a , c'est-à-dire que ϕ soit un théorème déduit à partir de la base de connaissance de l'agent a . Dans ce cadre, exprimer qu'un agent croit une proposition ϕ revient à dire que ϕ appartient à Θ_a , c'est-à-dire que l'on rattache une croyance à une possibilité déductive d'un agent. En particulier, établir qu'un agent croit que $\phi \Rightarrow \psi$ revient à essayer de prouver que $\phi \vdash_a \psi$ dans le cadre des connaissances et des capacités d'inférence de l'agent a .

Cette conception des croyances a l'avantage de beaucoup mieux rendre compte des mécanismes calculatoires qui sous-tendent les raisonnements des agents. Elles ne posent pas les problèmes rencontrés avec les sémantiques des mondes possibles et, en particulier, les problèmes de l'omniscience, puisqu'il est possible de définir des règles d'inférence limitées (par exemple en limitant le nombre de pas de déductions qu'un agent peut exécuter ou le type de déduction qu'il peut faire). Le sens est plus vu alors comme un usage (Wittgenstein 1951), c'est-à-dire comme l'ensemble des occurrences de cette expression dans le contexte de l'ensemble des preuves réalisées à l'occasion de l'interprétation de cette expression (Enjalbert 1989). Le modèle n'est alors plus donné a priori mais élaboré localement à partir d'interprétations partielles

locales à chacun des agents interagissant. C'est en utilisant cette approche que C. Lefèvre a proposé un modèle de croyances et de savoirs à partir d'un modèle logique issu du lambda calcul typé (Lefèvre 1995). Ce modèle résout les problèmes posés par la sémantique des mondes possibles tout en donnant une base formelle solide à des manières différentes de croire. Il est ainsi possible de distinguer la croyance comme preuve de la croyance comme argument et ainsi d'offrir au concepteur de SMA différentes manières d'appréhender le problème de la croyance.

5.3.4 Adéquation et révision des croyances

La première qualité d'un modèle est d'être adéquat, c'est-à-dire d'être dans un rapport de correspondance avec la réalité. Formellement l'adéquation revient à dire qu'il existe un morphisme entre la structure du modèle est celle du monde réel. Lorsque ce morphisme n'existe pas, le modèle est inadéquat: par exemple, une carte de France n'est d'aucune aide lorsqu'on se trouve aux Etats-Unis parce que les structures des deux pays sont trop différentes. De même, une carte routière n'est pas très utile lorsqu'on se trouve en bateau car les angles et les distances n'y sont pas respectés: les opérations d'addition vectorielles qui sont essentielles pour la navigation marine ne servent à rien pour se repérer sur un réseau de routes, dont la structure est essentiellement celle d'un graphe.

L'inadéquation des modèles est parfois responsable des difficultés que l'on éprouve à agir. Par exemple, beaucoup croyaient dur comme fer à la fin du siècle précédent qu'il n'était pas possible pour un "plus lourd que l'air" de voler. Le modèle que l'on avait alors et qui reposait sur la notion de poids sans faire intervenir de conception aérodynamique n'était pas adéquat, et l'on ne cherchait pas à faire des avions. Le fait même d'essayer de faire un aéroplane avec un moteur suffisamment puissant pour que la vitesse du vent produise une force portante ne venait pas à l'esprit de la plupart des gens, et seuls quelques pionniers croyaient aux plus lourds que l'air (Ferber 1906).

Un modèle est constitué de croyances, c'est-à-dire d'informations dont la validité peut-être remise en question si de nouvelles informations sont disponibles. Croire, c'est donc être capable de réviser son jugement si une information nouvelle vient infirmer ce que l'on pense. Lorsque le robot Clotaire se déplace, il doit en permanence remettre à jour l'ensemble des déductions qu'il peut faire sur le monde: la position des objets peut être modifiée et toutes les conséquences liées à ces objets doivent être mises à jour.

Logique des défauts

Par exemple, supposons que Clotaire, qui possède une pince préhensile, doive raisonner sur le fait que sa pince soit vide ou non pour savoir s'il peut prendre un objet immédiatement. La connaissance générale, valable pour tous les robots disposant de pinces, peut se formuler ainsi:

$$\text{pinceVide}(x) \Rightarrow \text{peutPrendreObjet}(x)$$

Cependant, cet énoncé pose problème. Supposons qu'initialement Clotaire ait la pince vide et qu'il possède donc le fait:

pinceVide(Clotaire)

dans sa base de connaissance. Il peut donc déduire le fait qu'il peut prendre un objet, c'est-à-dire que la formule

peutPrendreObjet(Clotaire)

est vraie. Mais dès qu'il prend un objet, cet énoncé ne l'est plus puisque sa pince n'est plus vide. D'autres phénomènes, par exemple si Clotaire a cassé sa pince, peuvent contredire la relation qui existe entre l'état de la pince et sa capacité à prendre des objets. L'affirmation que la pince est vide est relative à un état du monde et n'a donc aucune validité universelle.

Or la logique standard, telle que la logique des prédicats du premier ordre, ne peut pas prendre en compte ce type de situation. Dès qu'une proposition est affirmée, elle ne peut être remise en cause. On dit que ces logiques sont *monotones* parce que les raisonnements ne font qu'ajouter des faits à la base de connaissances dont la valeur de vérité ne peut être modifiée sous peine de rendre incohérente toute la base. Pour raisonner sur un monde qui évolue ou qui ne possède que des connaissances partielles, il est nécessaire de disposer de logiques différentes capable de traiter des connaissances révisables et ainsi de pouvoir remettre en cause des faits déduits au préalable et qui ne s'avèrent plus pertinents.

Réviser une base de connaissance, c'est corriger les expressions produites lorsque certains faits sont modifiés, c'est-à-dire définir la manière, progressive et incrémentale dont les croyances d'un agent changent lorsque des faits sont ajoutés ou retirés de sa base de connaissance et qui permettent de rendre compte du traitement d'informations *incomplètes*, *incertaines* et *évolutives*. Le développement de l'analyse du raisonnement révisable s'est effectué selon deux axes: sur le plan théorique d'une part en produisant des théories formelles capables de rendre compte des opérations de révisions, et sur le plan des mécanismes permettant de gérer les justifications des conséquences avec l'essor des systèmes de maintien de cohérence.

Les traitements logiques du raisonnement révisable sont fondés sur la définition de logiques dites *non monotones*, considérées comme des extensions des logiques classiques. Ces logiques n'ont pas été créées initialement pour caractériser le raisonnement révisable dû à un univers évolutif, mais pour formaliser certains aspects du *raisonnement de sens commun* et en particulier celui qui a trait au traitement des connaissances typiques et qui peut se paraphraser par "généralement tous les A sont B" (Sombé 1988). Pour reprendre l'exemple célèbre qui a donné ses lettres de noblesses aux recherches dans ce domaine, typiquement "les oiseaux volent". Donc, en l'absence d'informations contradictoires, il est possible d'en inférer que Titi qui est un oiseau, vole effectivement. Evidemment cette affirmation, qui est généralement correcte, n'est pas valide si Titi s'est cassé une aile ou s'il s'agit d'une autruche.

Les approches maintenant les plus développées dans ce domaine sont la *logique des défauts*, la *logique autoépistémique* de Moore (Moore 1988), et la *circonscription*

de McCarthy (McCarthy 1980). La logique des défauts, introduite par Reiter (Reiter 1980) présente l'avantage d'être à la fois simple à comprendre et à mettre en œuvre dans ses versions les plus élémentaires et d'avoir été bien étudiée.

Les formules de la logique des défauts sont les mêmes que celle de la logique classique. Les informations par défaut, c'est-à-dire de nature typiques, sont données par des règles d'inférences spécifiques appelées règles de défaut, qui s'écrivent ainsi:

$$\frac{\alpha : M\beta}{\gamma}$$

et qui signifient que si α est une croyance et que la formule β n'est pas incohérente avec les autres, alors γ peut être cru. Très souvent les formules β et γ sont les mêmes et on parle alors de défauts normaux. La règle des défauts portant sur les oiseaux et leur propriété de voler s'exprime ainsi:

$$\frac{\text{oiseau}(x) : M\text{vole}(x)}{\text{vole}(x)} \quad (RD1)$$

Il est possible d'utiliser la logique des défauts pour représenter le raisonnement révisable d'un agent. Par exemple, si l'on veut caractériser le fait que normalement un robot qui possède une pince vide peut prendre un objet sera formulé ainsi:

$$\frac{\text{pinceVide}(x) : M\text{peutPrendre}(x)}{\text{peutPrendre}(x)} \quad (RD2)$$

Cette règle exprime très précisément que "si la pince du robot est vide et qu'il n'est pas incohérent de penser qu'il peut prendre un objet, alors il peut le faire". De même la relation qui lie la jauge du réservoir de Titine (cf. section 5.3.3) au niveau d'essence peut s'exprimer ainsi:

$$\frac{\text{jauge}(x, \text{pleine}) : M\text{réservoirPlein}(x)}{\text{réservoirPlein}(x)} \quad (RD3)$$

L'intérêt d'une règle de défaut est de pouvoir définir des liens plus "lâches" entre les prémisses et les conclusions d'une règle qu'en logique des prédicats. Il suffit qu'il existe une contradiction entre la conclusion et d'autres faits de la base pour que la règle ne soit plus applicable et donc que l'inférence soit stoppée. Par exemple, si la pince de Clotaire est cassée et s'il existe une règle de la forme

$$\text{pinceCassée}(x) \Rightarrow \neg\text{peutPrendreObjet}(x)$$

et le fait

$$\text{pinceCassée}(\text{Clotaire})$$

alors le système en déduira

$$\neg\text{peutPrendreObjet}(x)$$

La règle RD2 ne pourra pas s'appliquer et on ne pourra pas en déduire que Clotaire peut prendre un objet même si sa pince est vide.

Les règles de défaut telles que RD2 ou RD3 indiquent toutes qu'un certain raisonnement peut être effectué en l'absence de contradictions considérées comme anormales. Par exemple, la règle RD3 pourrait être réécrite ainsi:

$$\text{jauge}(\mathbf{x}, \text{pleine}) \wedge \neg \text{anormal1}(\mathbf{x}) \Rightarrow \text{réservoirPlein}(\mathbf{x}) \quad (RD3')$$

où `anormal1` représente une caractéristique d'anormalité qui viendrait se mettre en travers de la relation existant entre la jauge et le réservoir. Si l'on considère qu'en l'absence d'informations supplémentaires, le prédicat `anormal1` est faux, ce que l'on obtient en appliquant l'hypothèse du monde clos où tout ce qui n'est pas connu comme vrai est faux, alors la formule précédente représente bien directement la règle de défaut RD3. La théorie de la *circumscription* de McCarthy est une généralisation de cette gestion de l'anormalité à n'importe quelle règle.

Justifications et TMS

L'intérêt de la logique des défauts et de la circonscription est de pouvoir être relativement facilement implémentée à l'aide de systèmes de maintien de la cohérence plus connus sous le nom de TMS (Truth Maintenance Systems). Les TMS sont des systèmes informatiques permettant de gérer les justifications des croyances. Lorsque Jean affirme que le réservoir de Titine est plein, c'est parce qu'il peut lire le niveau de la jauge d'essence qui agit comme une justification de sa croyance. Evidemment cette croyance peut s'avérer fautive si le lien de dépendance qui existe entre le niveau d'essence et la jauge s'avérait coupé. La justification est à la base des croyances et des modèles que l'on peut avoir sur le monde.

Les TMS sont des mécanismes informatiques très simples, inventés initialement par Doyle (Doyle 1979), qui gèrent les dépendances entre faits et indiquent si oui ou non un fait peut être cru, en fonction des justifications positives ou négatives qui peuvent lui être associées. Un TMS est un réseau de dépendances entre informations, les nœuds du réseau représentant les informations qui peuvent être vraies ou fausses, et les flèches caractérisant les dépendances entre les informations. Ces flèches sont soit positives soit négatives. Dans le premier cas, elles représentent des justifications positives entre les informations (ce que l'on appelle des flèches IN dans le jargon TMS), et dans le second des justifications négatives (des flèches OUT). Un nœud est dit valide, si tous les nœuds dont il dépend positivement sont valides, et tous les nœuds dont il dépend négativement sont invalides. Par exemple, le fait que Clotaire puisse prendre un objet s'il se trouve à côté de l'objet, que sa pince est vide et que l'objet ne soit pas trop lourd et que sa pince ne soit pas cassée peut s'exprimer sous la forme d'un schéma comme le montre la figure 5.6.

Cette figure illustre le caractère éminemment récursif du maintien de la cohérence. Les informations sont soit des primitives, soit dépendent d'autres informations. A chaque fait non primitif est associé un petit "processeur" qui calcule les validités des faits dont il dépend. Par exemple, le fait que la pince soit vide dépend de l'indication de son capteur et de l'état de santé de ce dernier. De même, une pince est cassée si elle ne peut plus effectuer de mouvements verticaux, horizontaux ou rotationnels.

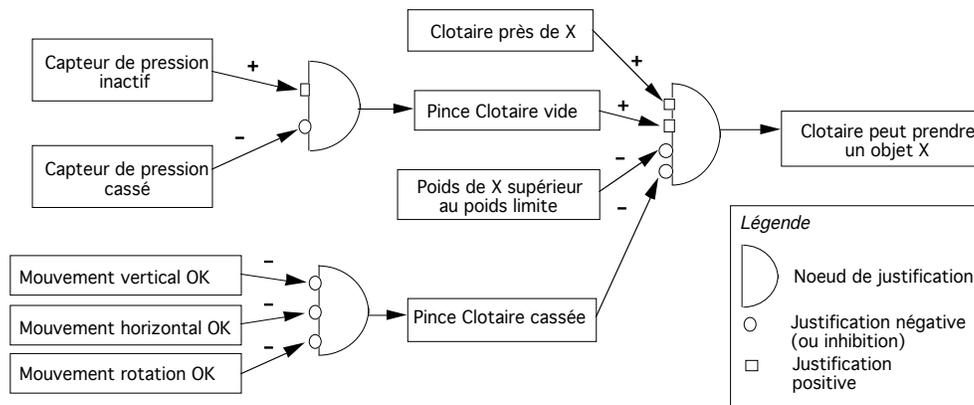


Figure 5.6: Une représentation schématique d'un ensemble de justifications. Une affirmation non primitive est vraie si toutes les justifications positives sont vraies et toutes les justifications négatives fausses.

Les nœuds qui ne dépendent d'aucun autre nœud sont soit des prémisses soit des hypothèses. Les prémisses sont des nœuds dont les informations sont toujours valides, alors que les hypothèses sont des nœuds dont les informations sont *supposées* valides. C'est au cours du fonctionnement du TMS que l'on déterminera si ces nœuds sont effectivement valides ou non.

Dans le cas idéal, où le réseau de dépendances est sans cycle, un simple algorithme de propagation suffit à gérer les justifications. Dès qu'un fait devient invalide, il invalide tous les faits qui dépendent de lui par un lien de justification positive, et il peut rendre valide un fait qui dépend de lui par un lien de justification négative, si tous les autres justifications négatives sont invalides elles aussi. Lorsqu'une contradiction est soulevée, un système de gestion des contradiction se met en œuvre en recherchant parmi l'ensemble des hypothèses celles qu'il faudrait remettre en cause pour lever la contradiction. On trouvera dans (Haton et al. 1991) une bonne introduction au raisonnement révisable et dans (Thibault 1993) une application des systèmes de maintien de cohérence à un système distribué de perception.

Prévisions et adéquation

Si un modèle doit être adéquat, il doit aussi permettre d'aider à la prévision de ce qui va se passer en anticipant sur un état futur du monde. A partir de représentations sur les lois de ce monde et de connaissances "psychologiques" sur le comportement des autres agents, un agent cognitif peut éventuellement déterminer les faits et les événements qui peuvent arriver. Cela consiste à pouvoir déterminer des cognitions interactionnelles d'entrée (percepts et informations) avant qu'ils ne soient disponibles. Par exemple, si Jean "sait" que Titine consomme 10 litres d'essence au 100 km, qu'il vient faire le plein et que son réservoir contient 30 L, il peut en déduire qu'il pourra faire 300 km avant de tomber en panne d'essence et qu'au bout d'environ 250 km la jauge d'essence indiquera que le réservoir est vide. Autrement dit, Jean et Clotaire peuvent anticiper sur ce qu'ils percevront dans le futur. La figure 5.7 donne le schéma d'un système prévisionnel.

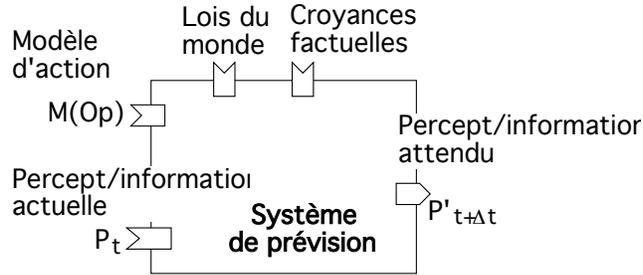


Figure 5.7: Un système de prévision produit des percepts “attendus”, que des perceptions ou des informations ultérieures viendront normalement corroborer.

A partir des perceptions ou des informations portant sur l'état actuel du monde, à partir des modèles des actions que l'agent peut accomplir et à partir des lois du monde auxquelles il croit et des croyances factuelles supplémentaires, le système prévisionnel fournit des *attentes*, c'est-à-dire des percepts ou informations qui devront “normalement” être corroborées par les percepts et informations provenant du système perceptif.

Formellement, on peut décrire le processus de prévision de la manière suivante. Soit P_1 un ensemble de percepts (ou d'informations) décrivant une partie de l'état s_1 actuel, soit $M(Op)$ le modèle de l'action que l'agent a est sur le point d'accomplir, soit L et C des ensembles de lois de l'univers et de croyances factuelles, alors on a:

$$\begin{aligned} P_1 &= \text{Percept}_a(\sigma_1) \\ P'_2 &= \text{Prévision}_a(P_1, M(Op), L, C) \end{aligned}$$

où P'_2 représente un ensemble de percepts (ou d'informations) attendues qui doivent être perçus ou obtenus lors de l'état σ_2 . D'un autre côté, l'agent accomplit des actions et le monde évolue. De ce fait, en reprenant le modèle de l'action comme production d'influences présenté au chapitre 4, on obtient un état σ_2 qui découle de l'exécution de l'action Op et des influences γ_{autres} issues d'autres actions (provenant d'autres agents):

$$\sigma_2 = \text{React}(\sigma_1, \text{Exec}(Op, \sigma_1) || \gamma_{autres})$$

Cet état peut être perçu par l'agent A (ou donner lieu à des informations que A reçoit) sous la forme d'un ensemble de percepts (ou d'informations) P_2 :

$$P_2 = \text{Percept}_a(\sigma_2)$$

Si P_2 est en accord avec P'_2 , pas de problèmes. On dira simplement que l'agent A a bien anticipé l'état σ_2 et que toutes ses lois du monde, ses croyances, ses mécanismes perceptifs ou informationnels ainsi que ses modèles d'actions sont adéquats. En revanche, si P'_2 et P_2 sont suffisamment différents pour être considérés comme incohérents (par exemple s'il existe des contradictions entre ces deux ensembles de cognitions), alors il faudra peut-être réviser les éléments représentationnels ayant entraîné cette fausse prévision. La figure 5.8 montre de manière diagrammatique ce processus de prévision. On peut constater qu'en effet si la prévision est adéquate, alors l'écart entre P'_2 et P_2 doit être nul.

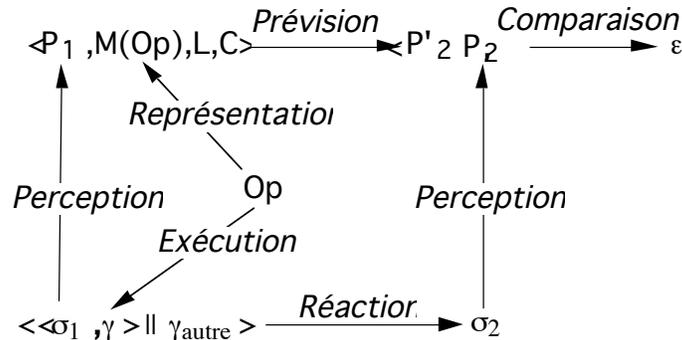


Figure 5.8: Diagramme de prévision. Lorsque ϵ est nul, les prévisions sont corroborées. Le diagramme commute et on peut dire que les représentations $M(Op)$, L , C ainsi que les systèmes perceptuels et informationnels sont adéquats.

Si l'écart n'est pas nul, cela signifie qu'il faut réviser certaines croyances. Il faut alors mettre en place des stratégies de révision et essayer de déterminer quels sont les fauteurs éventuels du trouble. Le problème consiste à choisir les éléments dont on doute le plus, c'est-à-dire les cognitions pour lesquels nous attribuons le moins de vraisemblance. Voici quelques solutions possibles :

1. *Les capteurs sont en mauvais état.* On s'assurera de leur bon fonctionnement en vérifiant que dans des situations connues on obtient bien les percepts attendus. Par exemple, si Jean trouve que sa voiture consomme trop, il pourra d'abord s'assurer de sa jauge en remettant de l'essence et en vérifiant que sa jauge fonctionne bien. Dans certains cas, un agent peut demander à un autre agent si ses perceptions sont bien les mêmes que les siennes en lui posant la question : "Est-ce que tu vois ce que je vois ?" La probabilité que les capteurs de plusieurs agents soient tombés en panne est tellement minime qu'une simple question de ce genre permet d'éliminer très rapidement ce facteur (l'illusion collective est très rare chez les agents artificiels).
2. *Les informations fournies par les autres agents ne sont plus bonnes ou insuffisantes.* Dans ce cas, on tentera de s'assurer de leur qualité en redemandant ces informations si cela est possible, ou bien en essayant de se procurer des compléments d'information. Si dans l'exemple de la poursuite, l'un des prédateurs utilise des informations fournies par un autre prédateur pour connaître la position d'une proie et qu'il ne la découvre pas au bon endroit, il pourra éventuellement demander confirmation de cette position à un autre prédateur pour savoir si c'est une information erronée (et ses croyances ne reposaient sur rien) ou périmée, la proie s'étant enfuie (et il faut simplement que l'agent mette à jour ses croyances en fonction des nouvelles informations).
3. *Les croyances factuelles dont l'agent dispose sont erronées ou insuffisantes.* Cela n'est pas toujours facile à réviser. Ces croyances dépendent souvent d'un grand nombre de percepts, d'informations et de lois du monde. Si l'on désire vérifier la qualité de ces croyances, il faudra vérifier tous ces éléments, ce

qui peut s'avérer fastidieux et peut-être inutile. En général on remettra plus facilement en cause les croyances dont les origines sont les plus douteuses, se réservant de ne réviser les croyances quasi certaines qu'en dernier ressort. Néanmoins, les autres agents peuvent être ici aussi d'un grand secours, en permettant à l'agent de confronter ses croyances à celle des autres afin de mieux localiser les causes de l'erreur de prévision.

4. *L'action ne correspond pas à son modèle.* Deux cas peuvent se présenter. Cela peut provenir d'un défaut de modélisation de l'action: il faut alors essayer de réviser ce modèle en testant l'action dans d'autres circonstances. Mais il peut aussi être tout simplement question d'un manque de compétence. Si je tire une flèche vers une cible et que je rate cette dernière, ce n'est pas tant le modèle de mon action qui est mauvais que mon manque de compétence dans le tir à l'arc. Il ne s'agit donc plus de révision de modèle, mais de pratique d'un exercice.

5. *Les lois de l'univers ou les modèles psychologiques des autres agents ne sont pas adéquats ou s'avèrent insuffisants.* Une manière de s'en assurer consiste à tester ces lois et ces modèles, c'est-à-dire à mettre en place des expériences (même très élémentaires) pour vérifier la validité de nos lois. Par exemple, si Jean a une théorie qui lui fait croire que tous les morceaux de bois flottent, il pourra être très perplexe devant un rondin de certains bois exotiques, tels que l'ébène, qui coule. Il pourra mettre à jour sa théorie, en y ajoutant des exceptions, ou mieux essayer de comprendre que derrière ce phénomène il existe des principes cachés qui sous-tendent cette théorie et qui portent sur la densité des corps et la loi d'Archimède. C'est le domaine de l'apprentissage par ordinateur que d'essayer à la fois de constituer des règles générales à partir de cas particuliers et de faire évoluer ces règles lorsque de nouveaux contre-exemples se présentent. Ce travail peut d'ailleurs aussi être l'objet d'interactions entre maîtres et élèves et d'une manière générale de dialogues entre agents. Il ne faut pas croire cependant qu'il soit très facile de mettre à jour des croyances portant sur des lois du monde ou des modèles psychologiques. On peut bien au contraire constater que dans nombre de domaines où les croyances sont en fait des marques de crédulité, la mise à l'épreuve des modèles et des conceptions du monde associées ne s'effectue pas d'elle-même. L'histoire des idées montre toutes les difficultés qu'ont encourues les scientifiques par rapport aux idéologies religieuses et magiques lorsque leurs méthodes et leurs réponses remettaient en cause des conceptions bien ancrées et nécessitaient une profonde remise en cause des modèles. "Et pourtant elle tourne" disait Galilée. Sa condamnation représente de manière emblématique tout le travail qu'il faut faire pour que des conceptions erronées disparaissent.

6. Enfin, il existe un dernier cas dans lequel aucun cogniton n'est en cause: ni les connaissances sur le monde, ni les croyances factuelles, ni les informations ni les percepts ne sont faux. Mais les motifs des erreurs de prévision résident tout simplement dans la complexité de la situation, la trop grande quantité de paramètres et la dynamique quasi imprévisible du système. Les

météorologistes, qui ont affaire à des phénomènes chaotiques, le savent bien: au-delà d'une certain laps de temps, toute prévision devient nécessairement fausse. Seules les boules de cristal peuvent prédire l'avenir lointain...

Beaucoup de travail reste encore à faire dans ce domaine afin de donner une description générale des stratégies de révision des croyances qui peuvent être mis en œuvre indépendamment des mécanismes particulier que l'on peut utiliser pour mettre à jour ses connaissances.

5.4 Que croire? Les contenus des représentations

Mais à quels contenus doivent se référer ces croyances? Sur quoi portent les connaissances et les croyances d'un agent et de quelles informations a-t-il besoin pour agir et accomplir ses tâches? Les croyances, qui peuvent être formalisées par des propositions, comme nous l'avons vu à la section précédente, représentent le monde, c'est-à-dire peuvent être interprétées en référant ces propositions à des modèles. La puissance d'un agent cognitif réside essentiellement dans sa faculté représentationnelle, c'est-à-dire dans sa capacité à élaborer des modèles du monde (de l'environnement et des autres agents) dans lequel il se trouve plongé permettant de comprendre, d'expliquer et de prévoir les événements et l'évolution des phénomènes. Il existe donc un grand nombre de modèles. Certains sont plus scientifiques, car il abordent l'adéquation du modèle de manière méthodique en cherchant à vérifier le modèle, ou plus exactement à montrer qu'il est à la fois réfutable mais résistant aux différentes épreuves que l'on peut lui faire subir. D'autres sont plus "intuitifs" et ne résistent pas à la moindre inquisition méthodique. Newton, Bohr, Einstein, Freud, Marx, Piaget, Darwin, Watson et Crick sont célèbres pour avoir développé des modèles permettant d'aller un peu plus loin dans la compréhension des phénomènes physiques, biologiques, psychologiques et sociaux. Ces modèles nous ont été transmis par la culture et nous "croyons" maintenant effectivement que les corps physiques se déplacent dans l'espace en suivant certaines trajectoires, que notre psychisme contient un certain nombre d'éléments inconscients dont les rêves sont révélateurs, que notre développement cognitif s'effectue par constructions successives, que nous descendons d'ancêtres communs aux singes, que nos caractères organiques individuels dépendent d'un cordon d'ADN. Tous ces modèles ont été élaborés lors de processus historiques mettant en présence un grand nombre de personnes dont certaines sont devenues célèbres en plaçant la pierre d'angle de ces édifices conceptuels.

On utilise des modèles dans la vie de tous les jours: une carte, par exemple, est un modèle du territoire qui nous permet de nous repérer et de déterminer notre chemin. De même un plan de maison est un modèle de la maison qui permet à des entrepreneurs et à des ouvriers de différents corps de coordonner leurs actions pour construire la maison. En d'autre termes, la puissance cognitive d'un être réside essentiellement dans la capacité à construire des modèles (théoriques ou pratiques) adéquats par rapport au monde, lui permettant d'agir et d'anticiper (dans une certaine mesure) sur le futur.

Ces modèles peuvent être classés en deux catégories: les *modèles factuels* et les *modèles prévisionnels*. Les premiers concernent les croyances qu'un agent possède sur l'état présent du monde et sur les caractéristiques des agents (y compris sur lui-même), et les seconds concernent les lois et théories permettant d'anticiper sur les états futurs du monde et sur le comportement des agents (et sur lui-même). Ces deux catégories de modèles peuvent ensuite se décliner dans les dimensions d'analyse d'un agent. On distinguera ainsi les croyances factuelles portant sur la société, l'environnement, les autres et soi-même, et les théories et lois portant sur la société, l'environnement (ce que nous avons appelé les lois de l'univers au chapitre 4), les autres et soi-même, ces deux dernières catégories introduisant de véritables *modèles psychologiques*, c'est-à-dire des représentations des modes de comportement que les agents peuvent s'attribuer les uns aux autres.

Les croyances portent sur la structure, l'état et les lois d'évolution du monde, sur l'état et le comportement d'agents. D'après la définition des dimensions fonctionnelles donnée au chapitre 3, ces croyances peuvent s'analyser selon les quatre dimensions environnementale (χ), sociale (σ), relationnelle (α) et personnelle (ω) (la dimension physique (ϕ) portant sur l'implémentation des autres dimensions). Les croyances factuelles d'un agent sont formées à partir de cognitions issues de son système perceptif et de ses croyances précédentes. Les perceptions et les informations sont les principales sources de construction de croyances factuelles: les premières servent à mettre à jour les croyances portant sur des caractéristiques physiques des objets et du monde, et les secondes permettent d'avoir une représentation du monde et des autres agents sans avoir besoin de les percevoir et d'être donc à proximité. Ces croyances sont ensuite combinées pour en produire de nouvelles à partir des concepts et des théories dont dispose un agent et de ses mécanismes de raisonnements. Les modèles prévisionnels sont donnés par le concepteur ou issus de processus d'apprentissage.

5.4.1 Croyances environnementales (χ)

Le long de la dimension environnementale, les croyances portent sur l'état de l'environnement, les caractéristiques des objets et des agents (considérés alors comme des objets, c'est-à-dire qu'on ne s'intéresse qu'à leurs caractéristiques externes telles que leur position, leur forme ou leur couleur), et d'une manière générale sur l'ensemble des caractéristiques qui pourraient être observées.

Dans l'exemple de la poursuite (cf. chap. 1), les prédateurs peuvent avoir une croyance sur la position des proies qu'ils poursuivent et sur celle des autres prédateurs. Ces croyances leur permettent de se diriger dans la bonne direction même s'ils ne perçoivent pas les proies ni ne reçoivent d'informations les concernant. Dans l'exemple des robots explorateurs conçu avec des agents cognitifs, ces croyances factuelles portent sur les positions du minerai, de la base et des robots foreurs et transporteurs ainsi que, d'une manière générale, sur l'état du monde. Les croyances de Clotaire, le robot transporteur d'objets, sont relatives à l'endroit où sont entreposés les outils et éventuellement à la topologie des lieux dans lesquels il doit se déplacer. Dans un système de contrôle distribué de processus, les croyances

portent sur l'état du processus et la valeur des paramètres de commande. Evidemment, ces croyances, par leur nature, peuvent être erronées et donc conduire à des interprétations fausses et à des décisions absurdes. De ce fait, il est fondamental d'y associer des mécanismes de révision des croyances qui puissent prendre en compte l'évolution naturelle de ces informations.

C'est le long de cette dimension que les lois de l'univers dans lequel est plongé l'agent sont éventuellement représentées. Si c'est le cas, l'agent dispose alors de véritables modèles prévisionnels, puisqu'il peut tenter de se projeter en avant dans le temps et d'anticiper l'état futur du monde. Par exemple, pour intercepter une balle, il faut anticiper sur sa position prochaine à partir de sa position actuelle et d'un modèle intériorisé de son mouvement.

5.4.2 Croyances sociales (σ)

La dimension sociale porte sur le rôle et les fonctions qui doivent être assurées à l'intérieur d'une société et, d'une manière générale, sur l'organisation et la structure du groupe de l'agent. Il s'agit donc d'une connaissance générale, à l'image des connaissances environnementales. Mais à la différence de ces dernières, elles ne sont pas tournées vers le monde et l'environnement de l'agent, mais en direction de la société dans lequel l'agent appartient. Comme nous l'avons vu préalablement, cette dimension a été encore peu explorée.

C'est sur ce plan que se situent aussi toutes les normes, contraintes et lois sociales applicables à tous les membres d'une organisation. Par exemple le code de la route est une loi sociale qui indique ce que doit faire un conducteur lorsqu'il se trouve dans certaines situations. Si je suis à un croisement, et que le feu est au vert pour moi, j'en déduis qu'il est au rouge pour les autres voies et que je peux passer sans dommage, en supposant que les autres conducteurs respectent cette consigne.

5.4.3 Croyances relationnelles (α)

La dimension relationnelle concerne la représentation des autres. Ce sont donc des croyances sur les compétences, les intentions, les engagements, les plans, les méthodes, les activités, les rôles et le modèle comportemental des autres agents. Ces représentations doivent être considérées comme des sortes de "mémoires caches" qui contiennent des informations sur les autres agents en permettant ainsi à un agent A d'avoir des informations sur des agents B ou C sans avoir à leur demander et ainsi de pouvoir élaborer des plans partiels et d'anticiper sur l'état futur du système. On peut aussi considérer que ces croyances sont des moyens d'attribuer à d'autres agents des caractéristiques cognitives et intentionnelles qu'ils n'ont peut-être pas, et ainsi faire cohabiter des agents hétérogènes disposant de caractéristiques structurales, cognitives et comportementales différentes.

Compétences

La compétence est certainement la croyance relationnelle qui présente le plus d'intérêt dans un système multi-agent composé d'agents communicants. Elle permet de répondre à la question: *qui peut faire quoi?* et donc, d'indiquer ce qu'un agent peut faire. Cette information est utilisée pour pouvoir déléguer du travail sans devoir recourir systématiquement à un appel d'offre. En effet, comme nous le verrons au chapitre 7, la répartition du travail entre les agents, qui constitue l'une des activités fondamentales des organisations, fait un grand usage de ce type d'information. Par exemple, la règle suivante:

Si un agent A doit faire une tâche T
et s'il ne sait pas faire T,
mais s'il connaît un agent B
qui a la compétence de réaliser T,
alors A pourra demander à B de faire T.

permet de déléguer des tâches en fonction des connaissances de compétences sur les autres agents, et donc de faire en sorte que le système ne "se plante pas" si un agent doit faire une tâche qu'il ne sait pas, ou ne peut pas, traiter lui-même. Il est possible de représenter les compétences de manière multiple, mais les deux représentations les plus classiques sont: les mots-clés, les prédicats et les objets.

La représentation par mots-clés suppose qu'il existe un vocabulaire de compétence, et que l'on associe à chaque agent les mots-clés correspondant à leur compétence. Par exemple, dans le cas de robots sur Mars, la spécialité des agents pourra être représentée par des mots-clés tels que **transport**, **extraction**, **prospection**, **relais**, **Energétique**, etc.

Mais ce mode de représentation n'est pas suffisant si l'on désire décrire plus finement des compétences. Nombre de systèmes utilisent alors, comme le système MACE (Gasser et al. 1987), des prédicats comportant éventuellement des arguments. Ainsi, pour dire qu'un agent est capable d'insérer un composant dans un dispositif, on utilisera le prédicat:

`capableInsérer(agent, composant, dispositif)`

et pour exprimer le fait qu'un agent est spécialiste en intégration symbolique d'une expression algébrique, on utilisera la description suivante:

`capableIntégrerSymbolique(agent, expression)`

Cependant, ce type de représentation est relativement limité, car il ne permet pas de décrire les liens qui existent entre les différentes compétences. Par exemple, l'intégration symbolique est une forme de calcul symbolique, qui en possède donc les caractéristiques générales et en particulier la faculté de manipuler des expressions symboliques comme arguments et comme résultats, à la différence des systèmes numériques qui fournissent des résultats uniquement numériques. Pour pouvoir décrire ce type d'information, on utilise des représentations par objets, en considérant que les compétences peuvent être décrites par des graphes symboliques

structurés, tels que des réseaux sémantiques (Lehmann 1992) ou les systèmes de Frames (Masini et al. 1989). Par exemple, la compétence relative à l'intégration symbolique peut s'exprimer sous la forme de la hiérarchie suivante:

CapableAction

```

...
  CapableCalculer
    CapableCalculerNumérique
      ...
    CapableCalculerSymbolique
      CapableDérivéeSymbolique
      CapableIntégrerSymbolique
      CapableSimplifier
      CapableFactoriser
      CapableRésoudreEquation
        CapableRésoudreEquationLinéaire
        CapableRésoudreEquationTrigo
          ...

```

On suppose alors que si un agent possède la compétence de “résoudre des équations linéaires”, il possède aussi celle de résoudre des équations et donc de faire du calcul formel. Cela ne signifie pas qu'il sache résoudre des équations trigonométriques, mais qu'il dispose a priori de qualités générales lui permettant d'effectuer certains calculs et donc qu'il semble compétent pour résoudre des équations au sens large, des équations linéaires très particulièrement, et éventuellement d'autres sortes d'équations. Toutes ces informations pourront être utilisées pour déléguer du travail aux autres agents et diminuer le nombre de messages lors de la phase d'allocation de tâches (cf. chap. 7).

Intentions et engagements

Les intentions et les buts indiquent ce qu'un agent a choisi d'accomplir. Cela permet de répondre à la question *qui va faire quoi*. Comme nous le verrons par la suite, les intentions conduisent, pour un agent, à la production d'engagements envers lui-même. De ce fait, il suffit de connaître les engagements que les agents ont contractés envers eux-mêmes et envers les autres pour avoir une bonne idée de leurs possibilités d'action. Par exemple, si l'on désire prendre un rendez-vous avec quelqu'un et que l'on connaisse les deux emplois du temps, il est plus facile de trouver une date qui convienne aux deux parties.

Plans/Méthodes

Cette information permet de répondre à la question “Quels sont les plans d'un autre agent?” Si un agent A connaît le plan d'un agent B, il pourra en tenir compte pour élaborer ses propres plans. On retrouvera cette notion dans le cadre de la planification distribuée que nous verrons au chapitre 8.

Activités

Cette information permet de répondre à la question *qui fait quoi?* Elle permet aux agents d'avoir une idée de la charge de travail de leurs accointances et donc d'en tenir compte lorsqu'ils cherchent à déléguer du travail.

Rôles

Cette indication renvoie à l'ensemble des activités qu'un agent est supposé savoir accomplir dans une organisation considérée. Comme nous l'avons vu au chapitre 3, les structures organisationnelles sont notamment définies par des rôles qui traduisent les différentes fonctions que les agents doivent savoir traiter au sein d'une organisation concrète. La connaissance des rôles pour soi et pour les autres permet à un agent de savoir à qui s'adresser s'il doit répondre à un problème.

Modèle comportemental

Les modèles comportementaux permettent de prédire l'état mental et les comportements futurs des autres agents, à partir d'une sorte de "psychologie naïve" (folk psychology) des autres agents. Ces théories sont fondées sur ce que l'on appelle *l'attribution d'intentionnalité*. On peut par exemple interpréter le comportement d'un agent situé comme s'il s'agissait d'un agent cognitif intentionnel et donc lui attribuer des états mentaux qu'il n'éprouve pas nécessairement (Dennett 1987). Ce problème renvoie à celui de l'intentionnalité tel que nous l'avons présenté plus haut.

5.4.4 Croyances personnelles (ω)

Les croyances factuelles personnelles sont des sortes de croyances relationnelles que l'agent possède sur lui-même et comprennent donc tous les types vus ci-dessus. La différence essentielle réside dans le fait que ces croyances peuvent être rendues toujours vraies et se comportent donc comme des savoirs (au sens de la théorie des croyances et des savoirs). En effet, il suffit de faire en sorte que ces croyances soient en prise directe avec les autres systèmes internes et soient automatiquement mises à jour lorsque les compétences, les intentions, les engagements, les croyances, etc. sont modifiés.

On peut donc dire que l'agent se représente comme s'il était un autre: il s'objective. Ainsi, toutes les représentations définies dans la dimension relationnelle pour les autres agents peuvent être explicitées au sein de l'agent même et combinées avec les autres croyances pour conduire un raisonnement.

5.5 Le système conatif

Comment un agent décide-t-il d'agir comme il le fait? Qu'est-ce qui lui fait accomplir certaines actions et pas d'autres? Quelles sont les causes de son action? Quels sont les dynamismes de l'activité mentale qui poussent à l'action? Quels sont les moteurs, les mécanismes et les freins du comportement?

Toutes ces questions ont été développées dans de nombreux domaines: qu'il s'agisse de la philosophie, de la psychologie du comportement, de la sociopsychologie, de la psychanalyse ou de l'éthologie, le problème des causes internes et externes de l'action a été débattu et rebattu et l'ensemble des théories avancées témoigne à la fois de la complexité du sujet et de l'étendue de notre ignorance quant aux mécanismes effectifs qui font qu'un animal et surtout un être humain se comportent comme ils le font. De ce fait, si les propositions sont nombreuses, aucune ne permet de rendre compte de l'étendue du problème. La réalisation d'agents artificiels repose encore une fois cette question plusieurs fois millénaires. Il n'est pas de l'objet de ce livre de récapituler toutes les théories qui ont pu être émises sur ce sujet, ni même d'en fournir un aperçu tant les propositions sont nombreuses lorsqu'on ajoute les théories du comportement animal à celle de la psychologie humaine.

On tentera au contraire de donner un modèle général du système conatif, c'est-à-dire de l'ensemble des structures et des processus permettant à un agent, tant réactif que cognitif, d'agir. Ce modèle se veut général en ce sens qu'il a pour ambition d'intégrer tous les facteurs psychiques permettant de concevoir des entités artificielles capables d'agir de manière autonome. Evidemment le système conatif d'un agent réactif sera plus frustré que celui d'un agent cognitif, mais ils seront conçus selon un même schéma général. Mais avant de présenter ce modèle en détail, il s'avère nécessaire de comprendre dans quel cadre ce modèle peut s'élaborer et, en particulier, comment il s'articule par rapport aux concepts de comportement rationnel et d'autonomie.

5.5.1 Rationalité et survie

La plupart des théories en intelligence artificielle supposent l'existence d'un *agent rationnel*, c'est-à-dire un agent dont les actions sont toujours le fruit d'une délibération raisonnée et qui servent directement à la satisfaction des buts de l'agent. Comme l'indique Newell, le principe de rationalité consiste à faire en sorte que "si un agent sait que l'une de ses actions lui permet d'atteindre ses buts, alors il sélectionnera cette action" (Newell 1982). Cette théorie de la rationalité doit être particulièrement circonstanciée dans le cadre de systèmes multi-agents. En effet, si la rationalité se résume simplement à une cohérence interne des mécanismes mis en œuvre par un agent pour qu'il atteigne ses objectifs, ce qui revient à affirmer que "*un agent qui dispose de buts se comporte de manière à satisfaire ses buts*", tout informaticien ne peut qu'adhérer à ces principes puisque cela suppose simplement que les concepteurs de ces agents ne sont pas totalement fous en développant des mécanismes qui tendent à répondre à des besoins. Mais dans ce cas, tous les agents, qu'ils soient cognitifs ou réactifs, sont rationnels, et l'intérêt de ce concept tombe alors du fait de sa nécessaire universalité.

Mais si on postule en plus que la rationalité suppose une élaboration et une délibération toutes deux raisonnées des buts que l'agent se propose consciemment d'atteindre, ce qui lui permet notamment d'expliquer et de justifier ses actions à qui lui demande, alors on aboutit à une conception beaucoup plus forte de la rationalité. On peut dire qu'un agent est rationnel s'il ajuste ses moyens aux fins qu'*il se propose*

de satisfaire, ce que l'on peut résumer en disant qu'un agent qui a l'intention de faire P le fera s'il en a la compétence:

$$\text{intention}(A,P) \text{ et peutFaire}(A,P) \Rightarrow \text{plusTard}(\text{Execute}(A,P))$$

Nous verrons (cf. section 5.8) que Cohen et Levesque ont développé tout un attirail théorique précisément pour comprendre et formaliser ce principe de rationalité lié à la notion d'intention et de passage à l'acte.

Si la rationalité sous-tend le travail d'une large communauté de chercheurs tant en IAD qu'en sciences cognitives, l'école réactive et, tout particulièrement, l'école réactive européenne et sud-américaine (je devrais dire franco-chilienne) — ce que l'on appelle parfois la seconde cybernétique (Dupuy 1994) — préfère, à la suite des travaux de von Foerster, expliquer le comportement d'un agent à partir des notions d'autonomie et de viabilité.

Étymologiquement, un système est dit *autonome* (auto-nomos) s'il est régi par les lois qu'il a lui-même édictées. Pour H. Maturana et F. Varela, l'autonomie a un sens précis: un système est autonome s'il est *autopoïétique*, c'est-à-dire s'il est constitué d'un ensemble de composants dont les transformations et les interactions ne cessent de reproduire l'organisation dont ils sont les composants (Maturana et Varela 1980; Varela 1989). Dans ce cas, le système définit une clôture opérationnelle entre lui et son environnement, entre ce qui reste invariant dans cette permanente opération de régénération (le système) et ce qui se transforme (l'environnement). Bien que cette définition nous séduise totalement et s'adapte idéalement au monde des organismes vivants, elle peut être trop restrictive dans le cas de systèmes multi-agents dans lesquels les agents sont conçus artificiellement et dont l'individualité est donnée par le concepteur. Par exemple, des robots mobiles peuvent être autonomes, du fait de leur indépendance en matière d'énergie et de prise de décision, sans être pour autant autopoïétiques.

Pour contourner cette difficulté, il est préférable, comme l'on fait F. Varela et P. Bourguine (Varela et Bourguine 1992), de considérer l'autonomie à partir du concept de viabilité. Est viable un système dont la dynamique du système reste à l'intérieur d'un domaine de tolérances. Le comportement d'un agent viable consiste donc à déterminer une trajectoire dans l'espace des configurations qui permettent à l'agent de demeurer dans sa zone de viabilité. S'il dépasse cette zone, si par exemple il ne récupère pas de l'énergie avant que son niveau minimum soit atteint, alors il perd ses capacités autopoïétiques: il meurt. Un agent doit donc tenter de rester dans cette zone de viabilité en se nourrissant, en fuyant (ou en attaquant) les prédateurs, en évitant les accidents et en accomplissant éventuellement les tâches fonctionnelles pour lesquelles il a été conçu, et tout cela dans un environnement inconnu et imprévisible. Nous verrons ci-dessous que toutes les stratégies comportementales qui sont mises en œuvre pour les agents réactifs tendent vers cet objectif de viabilité, indépendamment donc de toute hypothèse de rationalité.

5.5.2 Un modèle du système conatif

Le système conatif détermine l'action que l'agent doit entreprendre en fonction des informations et des connaissances dont il dispose pour assurer une fonction tout en conservant son intégrité structurelle et en faisant en sorte que l'organisation dont il fait partie conserve elle aussi son intégrité structurelle. Ce problème est parfois appelé dans la littérature de l'IA le problème du *contrôle* (Bachimont 1992; Hayes-Roth et Collinot 1993) ou celui de la *sélection d'action* (Tyrrell 1993b). Les concepts clés de notre modèle du système conatif sont les *tendances*, les *motivations* et les *commandes*, chacun ayant un cogniton associé.

En 1936, Albert Burloud (Burloud 1936) développe une théorie du comportement psychologique qui prend la tendance comme point de départ. Pour lui la tendance est une forme dynamique qui détermine un acte, soit qu'elle le conduise et s'inscrive en lui, soit qu'elle l'inspire et le finalise. En la distinguant de sa réalisation physique et biologique, il pose la notion de tendance comme centrale pour la psychologie et l'appréhende comme un objet scientifique. Pour notre modèle, nous reprendrons l'idée centrale de Burloud que tout comportement est le fruit d'une combinaison et d'une interaction de tendances, lesquelles sont les moteurs de l'activité psychique et du passage à l'acte. De plus, en tant que moteur de l'activité, toute tendance est l'expression d'un manque que l'agent cherche à réduire, ou comme le disent les philosophes comme une orientation psychique spontanée d'un individu vers une fin.

Par exemple, la faim est un manque de nourriture, le désir de l'autre s'exprime comme un manque de l'autre, réel ou fantasmé, la peur est l'anticipation d'un manque d'être, etc. Ces manques ne sont pas seulement individuels, mais aussi interindividuels et sociaux. Lorsqu'un agent ne peut pas accomplir une tâche qu'il s'est donnée à faire et qu'il demande à un autre de l'accomplir, alors il transmet par sa demande un manque que l'autre tentera de résoudre. Un agent satisfait, c'est-à-dire un agent sans manque, est un agent inactif, même si cette inaction s'exprime parfois par le mouvement. Par exemple dans le système MANTA (cf. chap. 7), les fourmis sont sensibles à des stimuli propagés par d'autres agents en manque. En particulier, les larves lorsqu'elles ont faim propagent des stimuli qui sont considérés comme des demandes par les ouvrières qui tentent de réduire la faim de ces larves. Ces demandes se combinent à d'autres motivations internes et externes des ouvrières pour constituer des tendances à agir, lesquelles entraîneront, après délibération, un comportement effectif des fourmis tendant à réduire ces insatisfactions. Mais une fourmi qui ne reçoit aucune demande et qui ne dispose d'aucune motivation interne pour agir accomplit un comportement élémentaire qui, dans le cas du système MANTA, consiste à ce que la fourmi déambule au hasard. Plus généralement la satisfaction se traduit par l'immobilité et l'insatisfaction conduit au mouvement dirigé vers un but, des déséquilibres dans l'homéostasie du milieu ou dans la physiologie interne de l'agent étant les moteurs internes des actions des agents. La source de l'action réside donc dans l'insatisfaction des agents et d'une manière générale trouve son origine dans des déséquilibres individuels ou sociaux¹².

¹²Cette théorie ne doit pas être généralisée, outre mesure, à des agents naturels (animaux, humains, etc.). Elle permet simplement d'avoir une perspective générale sur les origines de l'action

Outre les tendances qui poussent un agent à agir, il existe aussi d'autres tendances qui tendent au contraire à limiter l'action d'un agent. Par exemple, les tabous et les normes sociales tendent à limiter le répertoire des actions considérées comme convenables. Lorsqu'un policier indique à une voiture déjà stoppée de rester immobile, son chauffeur reconnaît ce signe comme une invite à ne pas agir, et il élabore donc une tendance à rester en place qui viendra s'opposer à la tendance de vouloir redémarrer. Mais cette tendance, quoique limitante peut elle aussi être considérée comme résultante d'un manque ou de l'anticipation d'un manque: si le chauffeur démarre, il risque d'être arrêté par le policier et donc de recevoir une amende, ce qui lui procurera un manque de liberté. Dans ce cas, la tendance à ne pas redémarrer est obtenue par combinaison d'un percept (le signe du policier) et d'une norme sociale (l'interprétation du signe).

Définition: *On appellera tendances les cognitions qui poussent ou contraignent un agent à agir ou qui l'empêchent d'agir.*

Les tendances elles-mêmes sont issues de combinaisons de *motivations*, c'est-à-dire de cognitions plus élémentaires (percepts, pulsions, normes, engagements, demandes sociales et interindividuelles) élaborées dans le *sous-système motivationnel* du système conatif. Les tendances deviennent ainsi des pivots du système conatif et du passage à l'acte. Par exemple, si un agent a faim et qu'il se trouve de la nourriture à proximité, il va développer la tendance de s'approprier cette nourriture. C'est donc la combinaison d'une pulsion interne, la faim, et d'une perception, la vue de la nourriture, qui provoquent son désir et produisent sa tendance à agir.

La forme de cette tendance résultante dépend alors de son organisation interne. S'il s'agit d'un agent réactif, cette tendance s'exprimera comme un couple <faim, valeur> où la valeur indique l'intensité de la tendance de type faim et donc la propension à déclencher la ou les actions associées à la quête de nourriture. S'il s'agit d'un agent cognitif, ce but pourra s'exprimer sous la forme d'une formule logique telle que **Rassasié(a)** où **a** est l'agent qui a faim. Dans ce cas, cette tendance s'exprime comme un but initial, ou objectif, c'est-à-dire comme le point de départ et le critère de satisfiabilité des plans que l'agent a pourra élaborer pour satisfaire cet objectif.

En fonction des croyances, des engagements, des plans et des contraintes dont il dispose, le *sous-système décisionnel* du système conatif évalue ces tendances, choisit les plus prioritaires et détermine les moyens à mettre en œuvre pour les satisfaire. Cette prise de décision dépend de la *personnalité* des agents, comme nous le verrons ci-dessous, et conduit à définir des priorités dynamiques entre tendances. Le résultat de cette délibération prend la forme d'un ensemble de décisions qui sont ensuite transmises au système organisationnel pour qu'il exécute ces décisions et accomplissent les actions et les tâches correspondantes. Pour un agent cognitif, dans lequel les tendances s'expriment comme des buts, la prise de décision peut donner lieu à l'élaboration de plans et à une décomposition de ces buts en sous-buts.

aussi bien pour des agents cognitifs que réactifs. Les exemples qui sont donnés ici sont destinés à illustrer mon propos. Ils ne constituent en aucune manière une théorie psychologique humaine ou animale.

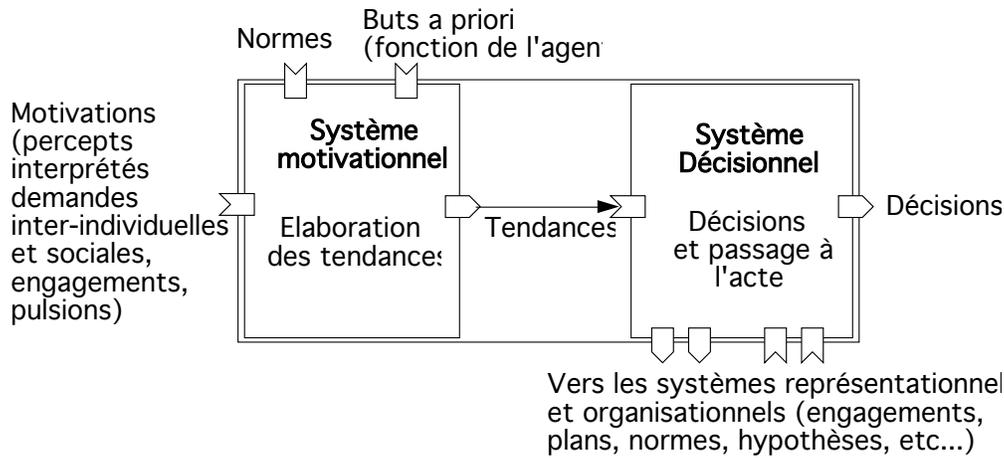


Figure 5.9: Le système conatif est composé de deux sous-systèmes: le système motivationnel qui élabore les tendances à partir de motivations et le système décisionnel qui prend les décisions et engage les actions à accomplir.

La figure 5.9 montre la structure générale du système conatif et sa décomposition en deux sous-systèmes motivationnel et décisionnel. On voit que les tendances constituent le point d'union de ces deux sous-systèmes.

5.6 Les motivations: les sources de l'action

Les raisons qui poussent un agent à l'action constituent ce que nous appellerons des *motivations*, lesquelles servent de matériau de base à la constitution des tendances. La faim qui pousse un agent à trouver de nouvelles ressources énergétiques ou une requête envoyée par un autre agent pour lui demander de l'aider sont des exemples de motivations. Ces motivations existent en grand nombre et il serait difficile d'en donner une liste exhaustive. Mais en reprenant les quatre dimensions conceptuelles¹³ d'un agent (personnelle, environnementale, sociale et relationnelle) décrites au chapitre 3, il est possible de les classer en quatre catégories: les *motivations personnelles*, qui ont pour objet l'agent et qui tendent à la satisfaction de ses besoins ou à l'exécution des engagements qu'il s'est donné à lui-même, les *motivations environnementales*, produites par ce que perçoit l'agent, les *motivations sociales* ou *déontiques*, qui ont trait aux demandes exprimées par l'organisation de niveau supérieur et par le concepteur, et enfin les *motivations relationnelles*, qui dépendent des tendances des autres agents.

Une source motivationnelle particulièrement active se trouve dans les engagements. En promettant d'accomplir une action pour autrui ou pour soi-même, l'agent s'engage à acquitter ses promesses, lesquelles deviennent des sources de motivation. Les engagements constituent ainsi des éléments de base pour la constitution des tendances au même titre que la faim ou le désir sexuel. Ces engagements sont envoyés

¹³Par définition, la dimension physique n'est pas conceptuelle et n'est pas directement source de motivations. En revanche, c'est le long de cette dimension que sont réalisées ces motivations.

par le système organisationnel et élaborés par le sous-système motivationnel sous forme de tendances à agir. L'agent est ainsi libre de suivre ou non ses engagements, puisque ceux-ci s'expriment comme des tendances sur lesquelles il est capable de délibérer. On appellera l'ensemble des motivations liées à des engagements, *motivations contractuelles*, et on les retrouvera dans les dimensions personnelles, sociales et relationnelles.

Toutes ces motivations sont liées et elles viennent se renforcer mutuellement. En vertu des principes hédonistes, si une action ne permet pas d'apporter une plus grande satisfaction ou d'éviter des désagréments, elle ne sera pas faite. De manière réciproque, une action ne sera accomplie que si elle se trouve en accord avec les normes de la société dans laquelle se trouve l'agent. On voit donc que les buts résultent d'un ensemble de motivations qui doivent trouver un compromis entre le principe de plaisir et celui de devoir, entre ce qui apporte du plaisir individuel et ce qui sert à l'organisation à laquelle il appartient.

Le rapport entre pulsion et tendance est d'autant plus complexe que l'agent est complexe. Pour un agent réactif simple, les tendances sont le résultat d'une combinaison de stimuli internes (que l'on appelle potentiels d'action spécifique en éthologie) et de stimuli externes, les percepts. Pour des agents cognitifs plus élaborés, des mécanismes de traitements peuvent intervenir pour combiner l'ensemble des motivations et produire des tendances "de haut niveau".

5.6.1 Les motivations personnelles: plaisir et contraintes

Les motivations personnelles regroupent à la fois celles qui tendent à procurer à l'agent un certain plaisir, et que l'on appelle *motivations hédonistes*, et les *motivations contractuelles personnelles* qui rappellent à l'agent les engagements qu'il a contractés avec lui-même et qui permettent la persistance des objectifs qu'il s'est assignés.

Les motivations hédonistes se trouvent à la base de tous les organismes vivants: la recherche de satisfaction, d'un plaisir ou tout simplement d'un certain bien-être est à la base du comportement. Par exemple, rechercher de la nourriture, faire des vêtements et des habitations, ou tenter de se reproduire sont des comportements liés à des besoins primaires qui s'expriment sous la forme de pulsions: faim, sensations de froid et d'inconfort ou désir sexuel. On appelle pulsion tout stimulus interne produit par le système végétatif et dont l'agent ne peut se soustraire. De plus, l'insatisfaction de ces pulsions peut avoir des conséquences fâcheuses pour l'agent et ses descendants. Par exemple, si les pulsions de faim ne sont pas satisfaites, l'agent dépérira et mourra, si les pulsions d'inconfort et de sécurité ne sont pas assurées, l'agent risque de décliner et de disparaître à cause des intempéries et de ses prédateurs, si les désir sexuels ne sont pas assouvis, sa descendance ne sera pas assurée.

Les pulsions sont donc associées à des besoins élémentaires et sont les représentants psychiques des excitations issues du système conservatif et d'une manière générale des besoins physiques de l'agent. Elles se transforment en tendances (et en objectifs pour les agents cognitifs) à partir des perceptions et des informations que l'agent

reçoit et des croyances dont il dispose. Les pulsions sont à la base de l'action individuelle mais elles entrent en conflit non seulement avec les motivations contractuelles, c'est-à-dire avec les engagements de l'agent, mais aussi entre elles, du fait que les tendances et les buts qu'elles engendrent sont souvent contradictoires. Par exemple, pour quelqu'un qui fait attention à son apparence, vouloir être mince est un but produit par la tendance à vouloir être désirable et à séduire d'autres personnes, lesquelles sont le résultat de pulsions sexuelles. Mais ces buts sont en contradiction avec la tendance à consommer un bon plat à l'ancienne (et hypercalorique), lequel est produit par des pulsions de plaisir gustatif immédiat. Le fait de satisfaire de préférence des tendances à plus ou moins long termes constitue ce que l'on pourrait appeler la "*personnalité*" de l'agent, et fait l'objet du sous-système décisionnel. Par exemple, un agent très "volontaire" aura plus tendance à privilégier les tendances permettant d'obtenir des satisfactions à long terme, alors qu'un agent plus "veule" satisfera plus volontiers les tendances permettant d'assouvir à court terme ses pulsions, même si le résultat global s'avère moins satisfaisant dans le second cas que dans le premier.

Les pulsions, en étant à la base du comportement des agents, rejoint certaines réflexions sur l'agir humain en psychologie et en anthropologie. Par exemple, les tenants du fonctionnalisme en anthropologie et sociologie, proposent de comprendre la structure des sociétés à partir des besoins élémentaires qui doivent être satisfaits (Malinowski 1944). Pour eux, le système de parenté ou la construction d'abris sont des réponses culturelles à des besoins élémentaires comme la reproduction ou le bien-être corporel. Cette théorie apparaît parfois un peu simpliste, mais nul doute qu'une analyse kénétique un peu plus fine permettrait d'élaborer une théorie plus circonstanciée et montrerait comment les actions des agents modifient les réponses à ces besoins élémentaires, et produisent des structures émergentes qui modifient entièrement le rapport entre ces besoins et les structures sociales qui en résultent.

5.6.2 Les motivations environnementales: le désir de l'objet

Les percepts sont d'abord les motivations essentielles des actions réflexes: par définition, un réflexe est une réaction immédiate à un stimulus externe. Ainsi, les actions situées que nous avons vu au chapitre 4 sont essentiellement des mises en œuvre de relations entre perceptions et actions sans faire intervenir de tendances explicites intermédiaires. Dans ce cas, le système conatif se réduit à une simple mise en correspondance de percepts et d'actions pré-programmées, comme le montre la figure 5.10, et les tendances sont confondues avec les percepts. Le sous-système décisionnel se borne alors à filtrer les percepts et à déclencher les actions associées.

Mais l'importance des percepts ne peut se réduire aux réflexes. Ils viennent en général prêter main fortes aux autres motivations et, en particulier, aux motivations hédonistes, en renforçant leur intensité. Par exemple, même si je n'ai pas très faim, la présence de nourriture peut néanmoins déclencher mon appétit et mon désir de l'absorber. C'est la présence des signes visuels et olfactifs de la nourriture qui déclenche en fait le désir d'absorption. Il y a donc renforcement entre les motivations internes de l'individu, ses pulsions, et les stimuli externes qu'il reçoit,

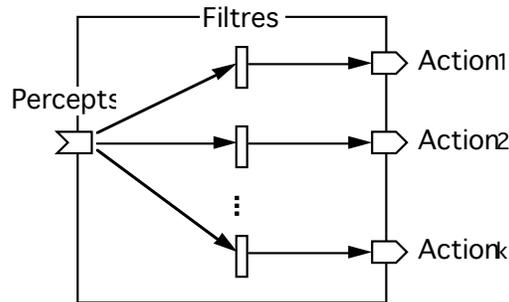


Figure 5.10: Un système conatif composé uniquement de réflexes se contente de relier les percepts aux actions. Les filtres sélectionnent les percepts qui les concernent et déclenchent les actions associées.

les percepts. En éthologie, on parle depuis Lorenz de *potentiel d'action spécifique* pour désigner la tendance à accomplir une action. Ce potentiel d'action spécifique doit être considéré comme une sorte de niveau qui tend à déclencher une chaîne d'action (fixed action pattern). Cette dernière, que nous avons appelée une tâche, est amplifiée à la fois par les pulsions (les excitations endogènes spécifiques dans la terminologie des éthologues) et les stimuli environnementaux déclencheurs, les percepts. La figure 5.11 représente le modèle d'activation de Lorenz (Lorenz 1984) tel qu'il l'a défini lui-même et tel qu'on peut le redéfinir à l'aide de notations modernes. L'intensité d'une tendance, c'est-à-dire la propension à accomplir une action, est une fonction (croissante) de la force des stimuli externes (percepts) et de celle des stimuli internes (pulsions). Plus les stimuli externes sont forts, moins la pulsion interne a besoin d'être importante, et vice versa. La fonction de combinaison est additive chez Lorenz, mais beaucoup de modèles considèrent qu'elle est multiplicative. Ce système motivationnel est à la base de la plupart des systèmes conatifs des agents réactifs, comme nous le verrons plus loin.

5.6.3 Les motivations sociales: le poids de la société

Dans les motivations déontiques, qui s'expriment le long de la dimension sociale sous la forme de normes, on peut distinguer les *demandes fonctionnelles* des *règles déontiques*. Les demandes fonctionnelles se trouvent être les moteurs de la plupart des actions que l'on effectue dans le cadre d'un travail. Pourquoi le commercial prospecte-t-il des clients? Pourquoi le juge prononce-t-il sa sentence? Pourquoi le bibliothécaire va-t-il ranger des livres? Pourquoi le boulanger pétrit-il sa pâte? Pourquoi la sage-femme aide-t-elle à mettre au monde des enfants? Pourquoi l'officiant célèbre-t-il les mystères de sa religion? Tout simplement pour assurer leur fonction au sein de la société, compte tenu de leur position sociale, et donc jouer au mieux leur rôle (professionnel ou autre) et accomplir l'ensemble des tâches qui y sont associées.

Ces motivations sont liées, dans le cadre des sociétés humaines, à des raisons hédonistes, que les avantages (ou les désagréments) soient immédiats ou lointains. Si on ne fait pas son travail, ou si on le fait mal, les désagréments suivront: diminution des profits, perte d'un emploi et de moyens de subsistance, déconsidération par

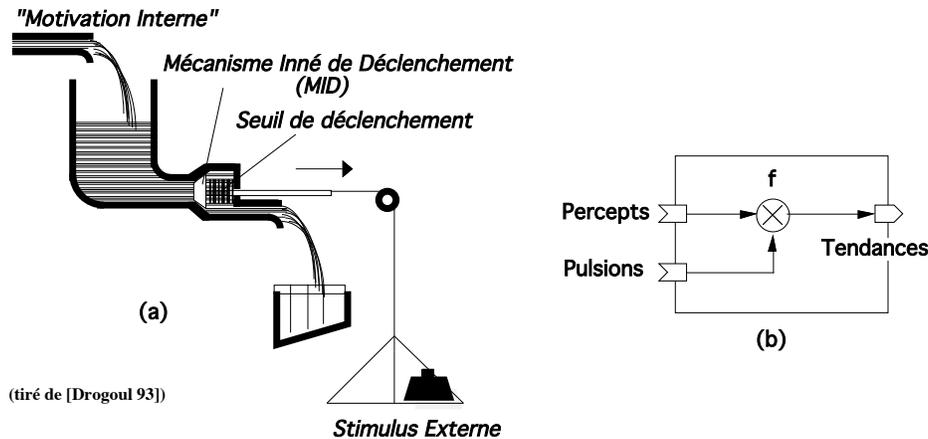


Figure 5.11: Le modèle d'activation de Lorenz dans sa version originale (a) (dessin extrait de (Drogoul 93)) et redéfini à l'aide de diagrammes systémiques (b). L'intensité de la tendance est fonction de la force des stimuli externes (les percepts) et de celle des stimuli internes (les pulsions).

les autres, etc. Cependant ces motivations fonctionnelles ne peuvent se ramener totalement à des caractères hédonistes, car elles prennent en compte les besoins de la société à laquelle l'agent se trouve. Par exemple, bien que les tâches de classement de courrier dans une entreprise ne remplissent pas de bonheur celui qui les accomplit, elles s'avèrent nécessaires au bon fonctionnement de la société et, indirectement, assurent à celui qui y travaille des moyens de subsistance.

Dans un système multi-agent, les motivations individuelles sont parfois plus limitées. Par exemple, une base de donnée ne prend pas de plaisir à donner des informations: elle effectue ce travail uniquement par motivation fonctionnelle, parce qu'on l'a construite expressément pour accomplir ce type de tâche, la finalité de son comportement se trouvant dans la tête du concepteur. Comme le montre cet exemple, le mode fonctionnel travaille uniquement à partir de buts préétablis par le concepteur du système ou par la fonction que doit remplir l'agent à l'intérieur d'une organisation.

Si les motivations fonctionnelles procèdent essentiellement en accomplissant des demandes provenant de l'organisation (ou du concepteur), les règles déontiques sont liées au devoir, c'est-à-dire aux interdits et aux idéaux qu'une société impose à ses membres. Elles prennent souvent l'aspect de lois ou de règlements qui précisent ce qui doit être fait ou interdit et agissent comme des contraintes qui s'opposent aux motivations hédonistes de l'agent, en inhibant des actions avant qu'elles ne soient débutées (ne pas voler) ou en cours de d'exécution (s'arrêter à un feu rouge).

Les raisons déontiques pour un agent peuvent être considérées comme des motivations hédonistes pour la société à laquelle il appartient. Par exemple, le fait de s'arrêter au feu rouge est une contrainte au niveau de l'individu qui trouve sa justification dans l'avantage social qui en résulte, telle que la diminution d'accidents. Ces normes peuvent comprendre la notion de peine, c'est-à-dire le fait qu'accomplir une action interdite puisse comporter des conséquences désagréables. Dans ce cas, ces

normes sont alors renforcées par des raisons hédonistes qui incitent l'agent à éviter les actions qui pourraient entraîner une pénalité.

Nous avons classé les raisons déontiques et fonctionnelles dans la même classe de motivation parce qu'elles sont intrinsèquement liées: toutes les deux sont issues de la pression sociale qui agit soit en demandant, soit en interdisant d'accomplir certaines actions. Le devoir s'exprime comme un chemin vers un objectif provenant de la société mais aussi comme une limite à l'ensemble des actions possibles. Mieux, il constitue un modèle qui transcende l'action individuelle: la raison d'un sacrifice effectué "par devoir" ne peut se trouver dans la satisfaction de l'agent, mais bien par l'ensemble des comportements idéalisés qui sont la marque de valeurs morales élaborées caractéristiques d'une société (ou d'un groupe d'individus) (Durkheim 1897).

5.6.4 Les motivations relationnelles: la raison c'est les autres

On appelle *motivations relationnelles* les motivations pour lesquelles le moteur de l'action n'est pas dans l'agent mais dans les demandes et les influences provenant des autres agents.

Par exemple, si un agent A demande à un agent B de faire une action P, le but pour B de réaliser P est une conséquence de la demande de A. B peut l'accomplir pour diverses raisons: parce qu'il ne peut pas faire autrement (il ne sait pas dire non), parce qu'il existe une notion d'autorité et que A lui est hiérarchiquement supérieur, parce que B a envie de faire plaisir à A, etc. Il s'agit certainement de la motivation la plus importante des systèmes multi-agents communicants, dans lesquels les agents coopèrent en s'envoyant des messages. Comme nous le verrons plus loin dans la section qui traite des intentions (cf. section 5.8) et au chapitre 6 portant sur les communications, communiquer c'est agir: la communication est assimilée à une extension de l'action individuelle, et demander à autrui de faire quelque chose n'est pas vraiment différent de se prier à soi-même de faire quelque chose. La différence repose essentiellement sur le caractère dynamique des liens qui existent entre les agents et sur les relations existantes entre l'agent demandeur et l'agent sollicité, ce dernier pouvant refuser de répondre aux demandes du premier, même s'il possède toutes les qualités requises pour les exécuter. Le problème de l'intentionnalité, lorsqu'il y a interaction, repose alors sur la relation qui existe entre les intentions individuelles du demandeur, les intentions individuelles des répondants, et les rapports qui les lient.

Inversement, recevoir une demande pour accomplir un service tend à provoquer une insatisfaction interne et donc une tendance à agir. Cette action consistera à accepter ou à refuser la demande (voire éventuellement à fuir), mais dans tous les cas à répondre à la demande originale. Les motivations relationnelles ne sont donc pas étrangères aux relations déontiques: si l'agent A est subordonné à l'agent B, il sera obligé d'effectuer ce que demande B du fait des normes et contraintes sociales impliquées par la relation de subordination entre A et B.

5.6.5 Les engagements: des motivations et contraintes relationnelles et sociales

L'engagement (commitment) se présente comme l'un des concepts clés de l'action collective dans le cas de systèmes multi-agents cognitifs. En s'engageant, un agent se lie par une promesse ou un "contrat" à effectuer une action dans le futur et donc à contraindre l'ensemble de ses comportements à venir. Par exemple, en prenant un rendez-vous, on promet d'une part de se rendre au lieu dit à l'heure indiquée par ce rendez-vous, mais on contraint aussi son emploi du temps en évitant de mettre un autre rendez-vous dans cette même plage horaire. Lorsqu'on s'engage, l'action promise est donc "programmée", et tout se passe comme si rien ne pouvait plus changer le comportement des agents.

D'une manière plus générale, l'engagement est un système de définition de liens sociaux par lesquels un groupe stabilise ses relations, pour favoriser l'organisation des actions. Tous les systèmes sociaux humains sont fondés sur des structures d'engagement relativement complexes. Qu'il s'agisse de contrats, de conventions, de serments ou de promesses, que l'on appelle cela responsabilité, fidélité ou devoir, qu'ils s'exercent auprès de sa famille, des amis, des relations professionnelles ou même de l'Etat tout entier, les engagements sont le fondement de l'action collective. Ils construisent, par tout un réseau de dépendances ténues, une certaine organisation sociale, dont même les sociétés apparemment les plus chaotiques et les plus violentes ne peuvent s'affranchir.

L'engagement est en effet ce qui permet de réduire l'imprévisible de l'action d'autrui, ce qui permet d'anticiper sur le futur et donc de planifier ses propres actions. Pour un commercial d'une entreprise par exemple, c'est parce qu'il sait que son collègue s'occupera d'un client qu'il pourra traiter une autre affaire.

Caractéristiques des engagements

Le concept d'engagement pour les systèmes multi-agents a été initialement analysé en détail par Alan Bond dans (Bond 1989) puis par Thierry Bouron dans (Bouron 1992). Tous deux définissent les caractéristiques communes aux engagements. Nous reprendrons leurs résultats d'analyse que nous intégrerons à notre définition penta-dimensionnelle d'un agent.

Les formes d'engagements

L'objet des engagements est variable car ils s'appliquent à tous les aspects de la vie sociale des agents. On peut néanmoins distinguer plusieurs cas de figure selon que les engagements portent sur la réalisation d'une action (je m'engage à vous payer), sur l'affirmation d'une croyance (je vous assure que c'est vrai), sur la réalisation d'un but (j'ai bien l'intention d'aller en Colombie cet hiver), sur des conventions sociales (je m'engage à accepter les règlements intérieurs de cette association) ou sur un rôle social (je m'engage à faire tout ce qui est en mon pouvoir en tant que premier ministre). Ces différentes sortes d'engagement peuvent être classées selon l'objet de

leur engagement et la dimension systémique dans laquelle s'effectue la relation entre agents.

α *Engagements relationnels.* Il s'agit du type d'engagement le plus fréquemment étudiés dans les systèmes multi-agents. Un agent s'engage envers autrui soit à accomplir une action (ou essayer de satisfaire un but sachant qu'un but peut être défini comme la condition d'accomplissement d'une action), soit sur la validité d'une information. L'engagement d'accomplir une action X par un agent A auprès d'un agent B s'exprime comme une relation du couple $\langle A, X \rangle$ vers B , ce que l'on peut exprimer par $\text{engagerFaire}(X, A, B)$. De même lorsqu'un agent A donne une information à un autre agent et si l'on suppose qu'il est sincère, l'agent A s'engage auprès de B sur la validité de cette information. De manière semblable à la précédente, on notera $\text{engagerInfo}(P, A, B)$ l'engagement de l'agent A auprès de B quant à la validité de l'information P , ce qui suppose que A croit à P et d'autre part qu'il a comme but le fait que B croit lui aussi à P . Tout ce que nous dirons par la suite sur les engagements portant sur des tâches pourra être transposé *mutas mutandis* aux engagements portant sur des transferts d'information.

χ *Engagements environnementaux.* Il s'agit de l'engagement envers les ressources. Par exemple, la prise de conscience écologique est une forme d'engagement environnemental, puisqu'il s'agit d'accomplir des actions qui ne nous lient plus directement aux autres, mais à notre milieu naturel.

σ *Engagements envers le groupe social.* Il existe deux formes d'engagement liées au groupe social. La première consiste à promettre d'accomplir une tâche pour le groupe. Ce type d'engagement est semblable à l'engagement relationnel, à la différence près que l'engagement se rapporte alors à un groupe et non plus à un agent. On notera par

$$\text{engagerFaireGr}(X, A, G)$$

le fait que l'agent A s'engage à accomplir l'action X auprès du groupe (ou de l'organisation) G . La deuxième forme porte sur l'engagement envers des conventions sociales et sur l'acceptation des contraintes associées à un rôle dans une organisation. Prendre une charge dans une association, ou obtenir un poste dans une entreprise, c'est s'engager envers le groupe social (l'association, l'entreprise) pour accomplir toutes les tâches associées à cette charge ou à ce poste. Soit R un rôle et G un groupe, l'engagement à un rôle se définit formellement ainsi:

$$\text{engagerRôle}(A, R) =_{def} \forall x \in \text{Tâches}(R), \text{engagerFaireGr}(X, A, G)$$

ϕ *Engagement des organisations envers ses membres.* Bien que l'on ne puisse parler que de manière métaphorique de l'engagement d'un agent envers ses composants physiques, il n'en est pas de même pour les organisations qui

s'engagent à satisfaire les besoins des agents qui en sont membres: une entreprise, par exemple, s'engage effectivement à payer ses salariés. Il s'agit donc d'un type d'engagement inverse du type précédent puisque c'est le groupe qui s'engage envers ses membres, ce que l'on peut noter formellement de la manière suivante:

$$\text{membreDe}(A,G) =_{def} \forall x \in \text{obligations}(G), \text{engagerFaireOrg}(X,G,A)$$

où $\text{engagerFaireOrg}(X,G,A)$ indique qu'une organisation G s'engage auprès de l'agent A à faire la tâche X .

ω *Engagements envers soi-même.* De même qu'il est possible de s'engager envers autrui, un agent peut s'engager envers soi-même à accomplir une action. Par exemple, on s'engage individuellement à apporter à son corps la nourriture dont il a besoin. De même une "bonne résolution" que l'on prend en début d'année correspond à un engagement envers soi-même. On notera par $\text{engagerFaire}(X,A,A)$ l'engagement que prend l'agent A envers soi-même à accomplir l'action X .

L'engagement comme structuration de l'action collective

Il s'agit peut-être de la caractéristique principale de cette notion: les engagements ne sont pas isolés mais forment une structure de dépendance qui peut s'exprimer ainsi: si A s'est engagé auprès de B pour effectuer l'action X , B s'est peut-être aussi engagé auprès de C pour l'action Y , en sachant que Y dépend de X . C'est ce qui arrive quotidiennement dans le monde industriel: la réalisation d'une voiture par exemple dépend du bon approvisionnement des pièces détachées et donc d'actions effectuées par les sous-traitants et les transporteurs. Les sous-traitants eux-mêmes ont des fournisseurs qui se sont engagés auprès d'eux pour leur livrer à temps la matière première dont ils ont besoin. Certaines tâches de B dépendent de l'engagement de A . Si l'on note

$$\langle B, Y \rangle \leftarrow \langle A, X \rangle$$

pour indiquer qu'il existe une tâche Y de B qui dépend de la réalisation de X par A , cette relation forme un graphe de dépendance comme le montre la figure 5.12.

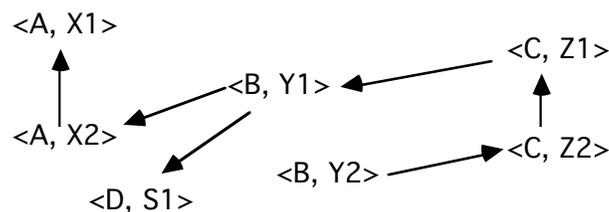


Figure 5.12: Graphe de dépendance entre tâches

Cette figure montre les différentes formes de dépendances entre tâches. Certaines tâches (par exemple $\langle A, X_2 \rangle$) sont dépendantes d'autres tâches réalisées par le

même agent (par exemple $\langle A, X_1 \rangle$). Une tâche peut dépendre directement de plusieurs autres, et on parle alors de dépendance multiple. Par exemple la tâche Y_1 de B dépend de la réalisation de la tâche X_2 de A et de S_1 de D. On voit aussi sur la figure qu'il existe des mécanismes de dépendance complexes. Par exemple la réalisation de la tâche Y_2 de B dépend directement de Z_2 de C et donc indirectement de Y_1 de B.

Cette structure est intéressante car elle montre qu'un agent peut prendre des engagements envers soi-même. Lorsque A désire accomplir X_2 il prend l'engagement de faire X_1 , ce qui impose à A d'avoir l'intention de faire X_1 . De la même manière, il est possible d'introduire les engagements concernant les informations sur ce graphe de dépendances, certaines tâches dépendant de la validité d'informations et, inversement, des informations dépendant de l'accomplissement de certaines tâches. Par exemple, comme nous l'avons vu pour les croyances, le fait d'aller faire du ski dépend de la validité de l'information "le niveau de neige est suffisant pour skier". Un autre exemple est donné par les tâches de calcul caractérisées par les informations qu'elles produisent comme résultats à partir des informations qui leur sont fournies en entrée. Souvent ces informations en sortie sont l'entrées d'autres tâches qui dépendent des premières. Par exemple, pour un gestionnaire, la prise de décision consistant à savoir s'il doit ou non investir dépend d'un ensemble de paramètres et en particulier de l'état de ses comptes obtenus comme résultat d'une tâche de calcul de son comptable.

Les graphes de dépendances peuvent contenir des cycles, mais dans ce cas, ils ne peuvent être exécutés: les tâches dépendant les unes des autres sont bloquées par une "étreinte fatale" (deadlock), problème bien connu en informatique des processus concurrents. Seuls les graphes corrects, c'est-à-dire les graphes de dépendance sans cycle, sont donc exécutables sans blocage. Ce problème des étreintes fatales a été très étudié pour la réalisation de systèmes distribués.

L'engagement comme moteur de l'action

L'engagement est à la source d'un grand nombre de conséquences pour un agent. S'engager à faire quelque chose, c'est à la fois faire la promesse à autrui que l'on va effectuer une tâche, mais c'est aussi s'engager envers soi-même à la faire, c'est-à-dire, si l'on est sincère, à avoir réellement l'intention de l'accomplir. On pourra noter cette dépendance entre engagement et intention de la manière suivante:

$$\text{engagerFaire}(X,A,B) \Rightarrow \text{engagerFaire}(X,A,A) \wedge \text{intention}(A,X)$$

De la même manière les engagements entre un agent et une organisation lient effectivement les intentions de l'agent.

$$\text{engagerFaireGr}(X,A,B) \Rightarrow \text{engagerFaire}(X,A,A) \wedge \text{intention}(A,X)$$

Cela signifie donc qu'à tout engagement envers autrui correspond un engagement envers soi-même.

Contraintes sur les ressources et cohérence des engagements

Les engagements imposent des contraintes non seulement sur l'intention des agents mais aussi sur les ressources utilisées pour accomplir leurs tâches. Si un agent s'engage à effectuer une tâche X qui consomme des ressources, il limite en même temps la possibilité pour les autres agents et pour lui-même d'effectuer des tâches utilisant les mêmes ressources au même moment.

Du fait de ces contraintes, certains engagements sont contradictoires. Par exemple, il est contradictoire de prendre l'engagement d'aller chez le dentiste à une date et une heure donnée et en même temps de prendre un rendez-vous d'affaire avec un client. L'un des intérêts des engagements étant de pouvoir prévoir et d'ordonner les tâches à effectuer, il est parfois possible de détecter à l'avance ces contradictions locales avant que la situation se produise (cf. chap. 8).

Dans les systèmes multi-agents, on suppose que les agents cognitifs sont cohérents localement en ne prenant pas d'engagements contradictoires. Cela ne signifie pas pour autant que les engagements soient irrévocables. Il est tout à fait naturel qu'un agent puisse revenir sur sa décision à condition qu'il en informe les agents auprès desquels il s'est engagé. Par exemple, si l'on estime que le fait d'aller chez le dentiste est prioritaire devant le rendez-vous d'affaire, on cherchera à remettre celui-ci en informant le client de cette modification de l'accord, ce qui revient à rompre le premier engagement pour en prendre un second.

Validité et persistance des engagements

Les engagements sont persistants, c'est-à-dire qu'ils perdurent jusqu'à ce qu'une des deux conditions suivantes soit remplie:

1. L'objet de la promesse a été accomplie.
2. L'agent rompt l'accord portant sur l'engagement.

La seconde figure intervient si l'agent ne peut plus remplir son engagement, soit parce qu'il n'en a plus les possibilités lui-même, soit parce qu'il dépend d'autres engagements et que les agents dont il dépend lui ont fait défaut. La possibilité qu'un engagement soit rompu fait partie de la notion d'engagement. De ce fait, s'engager à faire quelque chose c'est aussi s'engager à prévenir l'autre lorsque l'engagement n'est plus possible.

$$\begin{aligned} \text{engager}(\langle A, X \rangle, B) &\Rightarrow [\text{finalt}(\neg \text{peutFaire}(A, X) \\ &\Rightarrow \text{informer}(A, B, \neg \text{peutFaire}(A, X))] \end{aligned}$$

De la même manière, lorsqu'un agent A s'engage en donnant une information valide P à un autre agent, il promet en même temps de lui indiquer lorsque l'information P ne sera plus valide. On peut décrire formellement cette caractéristique de la manière suivante:

$$\begin{aligned} \text{engagerFait}(\langle A, P \rangle, B) &=_{def} \text{croire}(A, P) \wedge \\ &\text{but}(A, [\text{croire}(B, P) \wedge \\ &\text{finalt}(\text{croire}(A, \neg P) \Rightarrow \text{informer}(A, B, \neg P))]) \end{aligned}$$

Les ruptures d'engagements peuvent avoir un effet boule de neige et entraîner d'autres résiliation. La constitution d'une autre structure d'équilibre pose un problème complexe de détermination d'un consensus dans un système de contraintes distribuées.

Importance des engagements

Il faut retenir que les engagements sont des structures absolument indispensables pour le travail collectif dans le cas d'agents cognitifs capables d'anticiper sur l'avenir. Ils assurent une certaine stabilité au monde, en apportant aux agents une certaine confiance sur le comportement des autres agents et une représentation stable du monde. En effet, si les agents ne prenaient pas d'engagement, il leur serait impossible d'avoir une représentation adéquate d'un état futur du monde et donc de planifier leurs propres actions en anticipant sur l'avenir. Sans engagement, les agents cognitifs se retrouveraient dans le cas d'agents réactifs qui ne peuvent prendre en compte que la situation présente.

La notion d'engagement est donc le corrélat indispensable à la possibilité d'avoir des agents cognitifs disposant à la fois de capacités d'anticipation et d'un certain libre arbitre dans leur faculté de décisions. Les engagements donnent souvent lieu à l'établissement de contrats entre deux agents, c'est-à-dire de définition claire et précise pour chacune des deux parties des engagements auxquels elles se soumettent. Ces contrats résultent souvent d'un mécanisme d'allocation de tâche comme nous le verrons au chapitre 7.

5.7 Le passage à l'acte réactif

La deuxième partie d'un acte individuel réside dans la décision et le passage à l'acte. Cette phase de l'action, qui s'effectue dans ce que nous avons appelé le sous-système décisionnel, consiste à déterminer dans l'ensemble des tendances, lesquelles vont passer de la puissance à l'acte et donner lieu à la commande effective d'une action. De nombreux systèmes peuvent être imaginés. On peut distinguer ceux qui ne prennent en compte que l'état présent, en agissant à partir d'informations dont ils disposent dans l'ici et le maintenant, de ceux qui décident de l'action en se projetant dans le futur et en évaluant les conséquences possibles de l'action, quitte à réorganiser les buts. On appellera les premiers des *passages à l'acte réactifs*, et les seconds des *passages à l'acte intentionnels* (ou cognitifs). Nous étudierons ces derniers dans la section 5.8, nous consacrant pour l'instant à l'étude du passage à l'acte réactif.

La notion de passage à l'acte réactif se confond parfois avec celle de comportement réactif, les notions de motivations et de décision étant intrinsèquement liées pour ces agents. Néanmoins, le problème du passage à l'acte et de l'organisation des actions dans des systèmes réactifs est suffisamment importante pour que de nombreux chercheurs se soient penchés sur la question. On le rencontre souvent dans la littérature sous le terme de problème de la sélection d'action (action selection problem). Le problème est le suivant: ayant défini un répertoire d'actions, comment mettre en œuvre ces actions dans un ordre pertinent, compte tenu de l'ensemble

des stimuli externes et internes auxquels sont soumis les agents? Nous avons vu dans la section portant sur les motivations environnementales comment ces deux sortes de stimuli se renforcent mutuellement pour constituer des tendances à agir. Le problème de la sélection d'action se situe juste ensuite: comment utiliser ces tendances pour organiser le comportement de l'agent? La littérature sur ce sujet propose un grand nombre de modèles qui ne sont pas sans rapport avec les modèles présentés par des éthologues. Dans ce domaine d'ailleurs les collaborations entre informaticiens/roboticiens et éthologues/physiologistes, qui sont des plus fécondes, ne font que s'intensifier. Il suffit de lire les articles publiés lors des conférences sur la "Simulation of Adaptive Behavior: From Animals to Animats" (Meyer et Guillot 1989; Meyer et al. 1993; Meyer et Wilson 1991) pour s'en persuader.

5.7.1 Actes consommatoires et comportements appétitifs

Toutes les actions n'ont pas exactement le même impact sur un agent réactif. On distingue généralement les actions de consommation des comportements appétitifs. Les premières visent à décharger une tendance alors que les seconds s'appliquent à la phase active d'un comportement de recherche d'un but. Par exemple, dans le cas de la prise de nourriture, si l'acte consommatoire est **manger**, le comportement appétitif sera **chercher de la nourriture**. On constate de ce fait, que généralement l'acte consommatoire vient finir un comportement appétitif, et c'est pourquoi on l'appelle souvent *action finale*. L'intérêt de cette distinction vient de ce que la phase d'appétence peut se produire en l'absence de stimuli environnementaux alors que les actes consommatoires dépendent entièrement de la présence des stimuli correspondant dans l'environnement qui agissent comme des sortes de préconditions. En effet, on ne peut pas manger s'il n'y a pas de nourriture, alors qu'on peut aller chercher de la nourriture en son absence. Les comportements appétitifs ne sont donc dirigés que par des motivations internes (des pulsions dans notre terminologie).

A la limite des comportements appétitifs et des actes consommatoires se situent les taxies, c'est-à-dire les comportements qui orientent et déplacent un animal pour qu'il se dirige vers (ou fuit) une source de stimulation. Les taxies, lorsqu'il s'agit de se rapprocher d'un but, constituent souvent le chaînon entre le comportement appétitif et l'acte consommatoire final. Par exemple, si, dans sa phase d'appétence, l'agent perçoit de la nourriture, il se dirigera vers elle par un comportement de taxie avant de finalement la consommer. Les taxies utilisent souvent des mécanismes de suivi de gradient pour atteindre leur but (cf. section 4.5). La figure 5.13 représente un comportement type d'un agent réactif ayant pour but la décharge d'une pulsion interne. Chaque phase comportementale est conditionnée par un critère de déclenchement: pour que le comportement appétitif ait lieu, il faut que l'agent ait un niveau pulsionnel élevé. La taxie doit effectivement percevoir la nourriture et l'acte consommatoire pour être à même de prendre cette nourriture. Le résultat d'une telle chaîne comportementale conduit à la diminution de la motivation qui l'a engendrée.

Ce type de comportement n'est pas seulement l'apanage des animaux ou des agents simulant des animaux (les animats de Meyer et Wilson (Meyer et Guillot

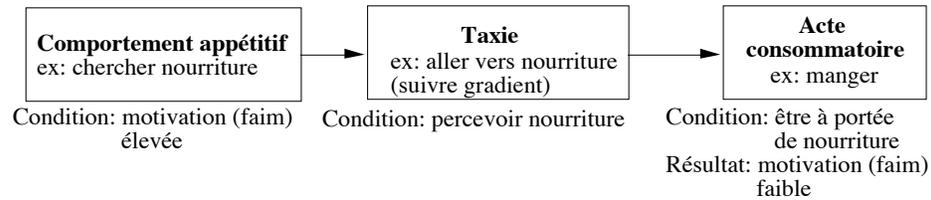


Figure 5.13: Une tâche ou chaîne comportementale type ayant pour but de décharger une motivation interne (par exemple la faim) chez un agent réactif.

1989). Ils sont la base même du comportement des agents réactifs.

Par exemple, les robots explorateurs suivent un tel schéma: d'abord une phase exploratoire pour chercher le minerai, puis une phase d'approche conduisant au minerai et enfin la phase consommatoire qui conduit l'agent à prendre le minerai et à retourner à la base.

5.7.2 Modes de sélection et de contrôle des actions

Le problème de la prise de décision pour un agent réactif, c'est-à-dire celui de l'action à sélectionner, se pose de plusieurs manières:

1. Que faire lorsque plusieurs chaînes comportementales comme celle que nous venons de voir sont en compétition? Par exemple, si l'agent doit choisir entre manger, boire ou copuler.
2. Que faire lorsque l'agent ressent plusieurs pulsions différentes et qu'il reçoit tout un ensemble de stimuli? En particulier, que faire si, alors qu'il recherche de la nourriture, il perçoit un congénère agréable avec lequel il pourrait s'accoupler?
3. Enfin, comment prendre en compte des conditions environnementales nouvelles qui sont susceptibles de mettre en cause l'intégrité de l'agent ou qui l'empêchent de satisfaire son but, telles que la présence d'un prédateur ou d'un obstacle?

Afin de répondre à ces questions, T. Tyrrell (Tyrrell 1993b) et G. Werner (Werner 1994) ont défini un certain nombre de critères qu'un "bon" modèle de sélection d'action doit pouvoir garantir, dont voici les principaux. Un bon modèle doit:

1. Être suffisamment général pour répondre de la même manière à tous les problèmes posés, et donc éviter les modèles *ad hoc*.
2. Montrer une certaine persistance dans les actes consommatoires pour éviter à l'agent de tergiverser sans arrêt, l'empêcher par exemple de prendre un peu de nourriture, puis un peu d'eau, puis un peu de nourriture, etc. dépensant alors plus d'énergie à aller d'un point à un autre qu'à se rassasier. Pour éviter ce problème, il suffit de donner aux stimuli externes une intensité qui dépende de la distance de l'agent à l'objet qu'il convoite.

3. Trier en fonction des motivations: plus un agent a faim, plus il doit avoir tendance à déclencher une chaîne d'actions lui permettant d'assouvir cette faim.
4. Préférer les actions consommatoires aux appétitives. Ce n'est pas la peine d'aller chercher ailleurs ce que l'on a déjà sous la main.
5. Favoriser l'accomplissement de chaînes d'actions et éviter des comportements trop lunatiques dans lesquels un agent n'arrête pas de commencer une chaîne d'actions et n'arrive jamais à en terminer une.
6. Interrompre une action si une situation nouvelle met en péril l'agent (prédateur ou danger) ou si l'action s'avère manifestement impossible à accomplir (obstacle).
7. Montrer un certain opportunisme et pouvoir accomplir des actes consommatoires qui ne sont pas nécessairement reliés à la chaîne d'actions (à la tâche) courante.
8. Utiliser toutes les informations disponibles pour exhiber le comportement le plus adapté à la situation. Par exemple, l'agent doit éviter d'aller vers une source de nourriture si un prédateur se trouve à côté.
9. Permettre l'accomplissement de plusieurs actions en parallèle, tel que fuir en envoyant un signal de détresse.
10. Être facilement extensible pour permettre au concepteur d'ajouter de nouvelles actions et de nouvelles chaînes d'actions à son répertoire.
11. Evoluer par apprentissage, en modifiant éventuellement les choix de manière plus appropriée avec l'expérience.

Certains de ces critères peuvent sembler contradictoires: vouloir à la fois que l'agent soit capable de persister dans ses actions, de s'interrompre et de faire montre d'opportunisme semble difficile à réaliser dans un même système. Mais de nombreux modèles permettent, dans une certaine mesure, de satisfaire la plupart d'entre eux.

Actions situées et champs de potentiels

Le premier de ces modèles est celui des actions situées que nous avons étudié au chapitre 4. Il se compose d'un ensemble de règles qui relie une action à une situation. Bien que ce modèle ait surtout tendance à valoriser les conditions environnementales, il est possible d'y adjoindre des paramètres correspondant à des états internes et ainsi de définir la situation courante comme un couple

<conditions environnementales, état interne>

Le système ABLE de D. Connah et P. Wavish fonctionne de cette manière (Wavish et Connah 1990). Il est suffisamment souple pour faire opérer plusieurs agents ensembles. Il peut traiter des situations relativement complexes comme celle qui consiste, pour un chien de berger (simulé), à faire entrer des moutons dans un enclos (Wavish 1992). Néanmoins, les actions situées souffrent de nombreux maux: il leur est difficile de faire montre d'une véritable persévérance (sauf à programmer quelques mécanismes de manière *ad hoc*, mais cela condamne alors l'extensibilité du système), et s'ils savent se montrer très opportunistes, il leur est difficile de savoir prendre des décisions fondées sur des poids. Il est très compliqué de programmer un agent à l'aide d'actions situées pour lui faire effectuer plusieurs tâches différentes. De ce fait, les actions situées ne sont pas considérées comme de bons modèles de sélection d'action.

L'usage de champs de potentiels, comme ceux que nous avons examinés au chapitre 4, peut résoudre un certain nombre de problèmes. En tenant compte de la distance de l'agent à la source des stimuli, ces champs donnent à l'agent une certaine persistance dans ses activités consommatoires. De plus, ils lui permettent de tenir compte de nouvelles informations, d'être opportuniste et de traiter plusieurs actions en parallèle. Par exemple, dans le système PENGU que nous avons présenté au chapitre 4, le pingouin peut à la fois fuir les abeilles et aller chercher des diamants, tout en gobant de manière très opportuniste les nouveaux diamants qui se trouvent sur son chemin.

Pour tenir compte des motivations internes, il suffit de considérer que l'intensité d'un signal est multiplié par un facteur dépendant de ces motivations. Néanmoins, les champs de potentiels, en étant très liés à l'environnement ne peuvent pas toujours traiter tous les problèmes de sélection de chaînes d'actions, et il est parfois nécessaire d'introduire d'autres mécanismes pour conduire un comportement efficace, tels que ceux que nous verrons par la suite. Néanmoins, ils présentent de nombreuses qualités intéressantes pour la conduite de comportements dynamiques multiples, et nous les reverrons associés à d'autres systèmes de sélection de tâches.

Modèles hiérarchiques

De nombreux modèles se sont inspirés des travaux en éthologie pour implémenter des modèles efficaces de sélection d'action. Les plus étudiés reposent sur une approche hiérarchique du comportement, dans laquelle des comportements de haut niveau sont décomposés en actions de bas niveau. Le plus célèbre de ces modèles est dû à un éthologue éminent, prix Nobel de surcroît, N. Tinbergen (Tinbergen 1951).

Dans celui-ci, l'organisation du comportement est donnée sous la forme d'une hiérarchie de nœuds. A chacun de ces nœuds, qui se comportent comme des sortes des neurones formels, est lié un mécanisme inné de déclenchement (MID), sorte de précondition pour l'activation du nœud. Le déclenchement des nœuds s'effectue par la combinaison de stimuli provenant de l'extérieur et des nœuds hiérarchiquement supérieurs. Lorsque la somme des stimuli afférents aux nœuds dépasse un certain seuil, le nœud est "libéré" et son énergie d'activation est transmise aux nœuds de niveau inférieur. Les nœuds supérieurs correspondent à des

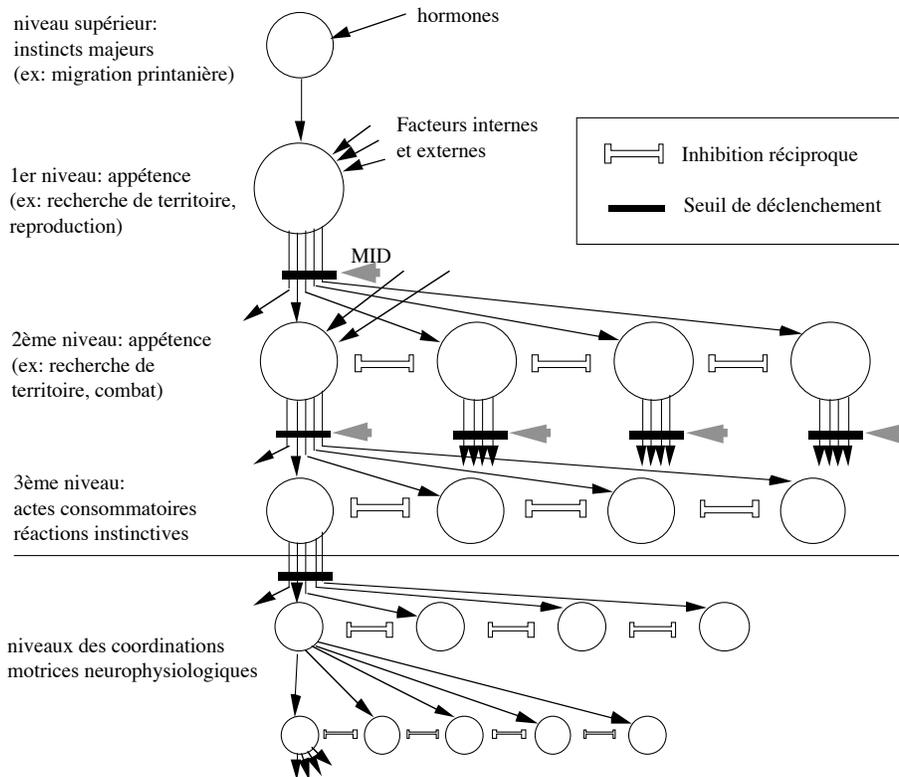


Figure 5.14: Le système hiérarchique des instincts de N. Tinbergen

activités générales telles que **reproduction** ou **nettoyage**, et plus on descend, plus les nœuds s'occupent d'actions de bas niveau, allant même jusqu'à déclencher les muscles (fig. 5.14). Chaque nœud d'un niveau est en conflit avec les autres nœuds du même niveau: si l'un d'eux est activé, alors les autres sont inhibés. Ce modèle est intéressant par son aspect pédagogique et explicatif mais il n'a jamais été conçu en vue d'une implémentation informatique ou robotique. De toute façon, les modèles hiérarchiques, en donnant trop d'importance aux pulsions (aux stimuli internes), empêchent les réactions opportunistes et ne permettent pratiquement pas à une chaîne d'actions de s'interrompre. En d'autres termes, l'agent est alors trop "concentré" sur ses buts, trop "sourd" aux stimuli environnementaux.

C'est pourquoi certains auteurs, tels que Rosenblatt et Payton (Rosenblatt et Payton 1989) et Tyrrell (Tyrrell 1993b), ont proposé d'autres modèles qui sont à la fois hiérarchiques dans la structure, mais non hiérarchiques dans la prise de décision. Ces modèles sont fondés sur ce qu'ils appellent des hiérarchies à flux libre (*free-flow hierarchies*), dans lesquelles les nœuds supérieurs ne décident pas des nœuds inférieurs à activer, mais définissent simplement des préférences sur un ensemble de candidats de plus bas niveau. A partir du modèle initial de Rosenblatt et Payton, Tyrrell en a adapté un autre qu'il a mis en compétition avec d'autres modèles d'action qu'il a testés dans un monde dans lequel un agent doit accomplir un certain nombre de tâches pour survivre, telles que manger, boire, échapper aux prédateurs et se reproduire (Tyrrell 1993a). La figure 5.15 montre la structure d'un

tel modèle pour la tâche de recherche de nourriture. On peut voir que le flux des activations descend des niveaux supérieurs vers les niveaux inférieurs, mais sans que ces derniers ne soient en opposition. Ce n'est qu'au niveau inférieur qu'un mécanisme de sélection choisit l'action ayant la valeur d'activation la plus grande.

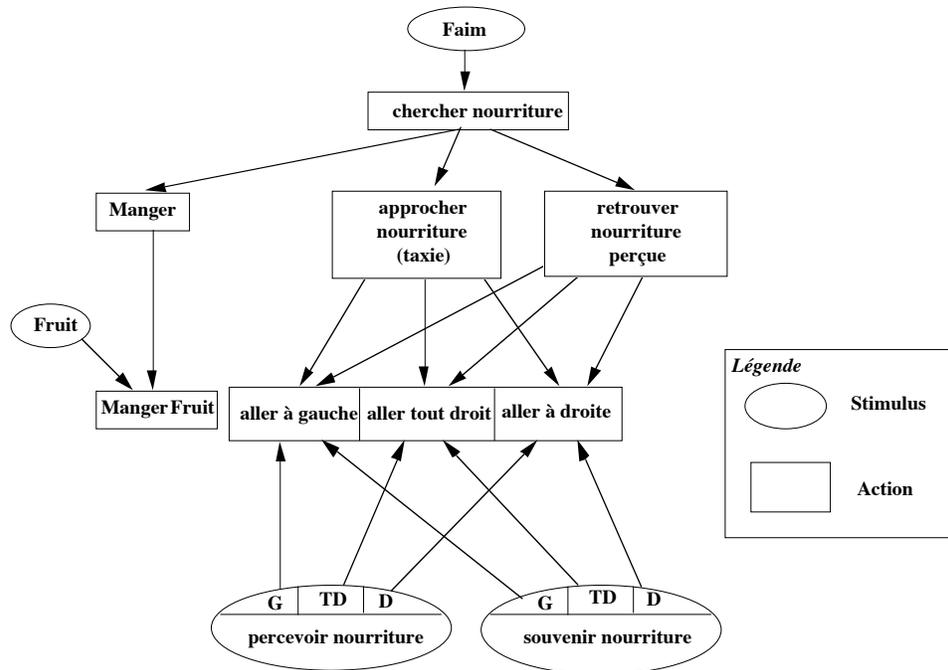


Figure 5.15: Le modèle de Rosenblatt et Payton pour la tâche de recherche de nourriture

Même s'il s'agit d'un modèle qui demande de très importantes mises au point pour faire en sorte que les comportements appétitifs ne viennent pas totalement dominer les actions consommatoires, il est, d'après Tyrrell, celui qui présente les meilleures qualités par rapport à des modèles purement hiérarchiques ou même des modèles ascendant comme celui de P. Maes.

Modèles ascendants

En prenant totalement le contre-pied des systèmes hiérarchiques, d'autres auteurs tentent de trouver une voie qui passe par l'élimination totale de toute organisation dans laquelle toute forme de dominance puisse exister. C'est le cas en particulier de l'approche choisie par P. Maes avec son système ANA (Maes 1991; Maes 1992) qui tient à la fois du réseau de neurones, de l'approche multi-agent de Minsky et du modèle des tâches compétitives. Le modèle ANA est composé d'un réseau de modules, chacun correspondant à une action élémentaire, telle que *aller-vers-nourriture*, *boire-eau*, *fuir-creature*, etc. Chaque action est décrite par ses conditions d'application, son niveau d'activation, les liens qu'elle a établis avec ses prédécesseurs, ses successeurs et les actions avec lesquelles elle se trouve en opposition:

Tâche aller-vers-créature
 niveau d'activation: 56.3
 conditions: (créature-perçue)
 successeurs: combattre
 prédécesseurs: explorer
 opposées: fuir-créature
 code: le code des commandes à exécuter

Le réseau d'un tel système est représenté à la figure 5.16. Les stimuli environnementaux et les motivations internes (peur, fatigue, agression, faim, soif, curiosité) tendent à activer certaines actions en élevant leur niveau d'activation. L'action ayant le plus fort niveau d'activation est sélectionnée, et ses successeurs voient leur niveau d'activation légèrement augmenté, ce qui augmente leur chance de sélection.

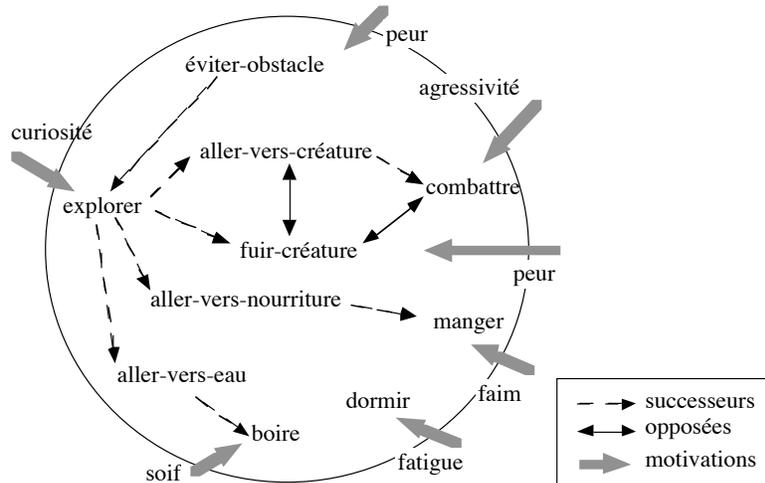


Figure 5.16: Le réseau d'activation des tâches du système ANA de P. Maes

Ce système tend à dégager les actions appétitives des actions consommatoires: les premières ne sont liées à aucun stimuli interne, alors que les secondes le sont. Les liens successeurs qui existent entre elles sont constitués et renforcés chaque fois qu'un acte consommatoire suit une action appétitive. Tyrrell a testé le système de P. Maes dans son environnement simulé (Tyrrell 1993a) et il a constaté qu'il n'avait pas de très bonnes performances lorsque le réseau était très important. Ce phénomène est dû essentiellement à des problèmes de choix entre actions consommatoires ou appétitives, à la difficulté de bien prendre en compte les buts multiples et tout simplement à l'embaras qu'il y a à choisir entre de nombreuses actions candidates. Il semblerait donc qu'une certaine structure soit nécessaire pour assurer une bonne sélection d'action.

Néanmoins, en reprenant une technique assez neuronale, G. Werner (Werner 1994) propose une démarche qui, d'après lui, possède tous les avantages de la méthode de Rosenblatt et Payton, sans reposer sur une approche hiérarchique, et qui, en plus, donne la possibilité d'exécuter plusieurs actions simultanées indépendantes. Son modèle repose en fait sur une simple combinaison multiplicative des motivations internes et externes, en reprenant donc le mécanisme motivationnel que

nous avons vu plus haut (section 5.6.2). Il comprend quatre ensembles de nœuds: les nœuds d'entrée qui sont connectés à des types de percepts relatifs à l'agent (**nourriture-à-droite**, **prédateur-devant**, etc.) ainsi qu'aux pulsions (faim, soif,...), les nœuds de sortie qui commandent des actions élémentaires (**aller-à-droite**, **aller-àgauche**,...), des nœuds correspondant à une couche cachée (mais qui n'est pas utilisée dans les exemples présentés), et surtout un ensemble de nœuds qui représentent ce que Werner appelle des commandes et qui ressemblent en fait à ce que nous avons appelé des tendances, c'est-à-dire des propensions à effectuer une action plutôt qu'une autre. Ces commandes sont utilisées comme des coefficients qui interviennent de manière multiplicatives dans les connexions qui relient les neurones d'entrée aux neurones de sortie (fig. 5.17). De ce fait, plus un agent a faim, plus il a tendance à se diriger vers la nourriture. Néanmoins, s'il y a de l'eau très proche et qu'il a un peu soif, il se dirigera d'abord vers l'eau si le produit du stimulus externe provenant de la présence d'eau avec la pulsion de soif est supérieure à celle du produit du stimulus de la nourriture (qui est située plus loin) avec celle de l'envie d'avoir de la nourriture. Cela montre que le comportement de l'agent est à la fois persévérant (il ne change pas d'avis si sa pulsion est suffisamment forte) et opportuniste (si un stimulus externe est suffisamment puissant, il essaiera de le satisfaire).

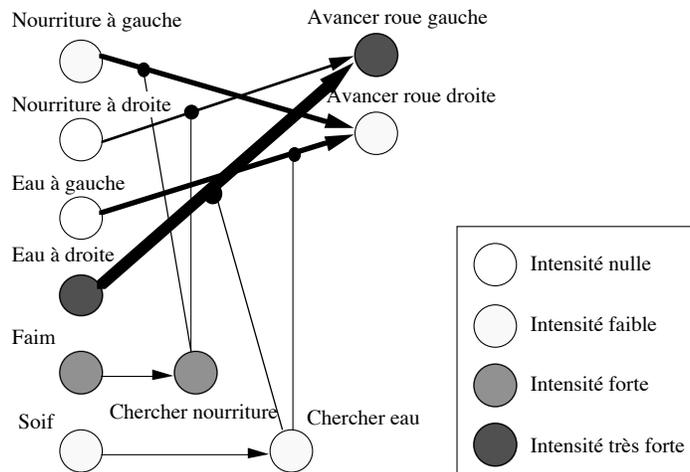


Figure 5.17: Le modèle de système décisionnel de Werner pour un agent réactif. La force des connexions est indiquée par l'épaisseur des flèches.

Bien que les exemples présentés montrent en effet qu'un agent peut survivre dans un environnement dans lequel il doit trouver sa nourriture, se désaltérer et fuir les prédateurs, G. Werner n'a pas testé son modèle dans un environnement suffisamment complexe pour que l'on sache si son modèle sait correctement choisir entre comportements appétitifs et actes consommatoires, étant donné que l'exemple qu'il présente semble pouvoir être résolu uniquement avec des taxies et des actes consommatoires. Néanmoins, il présente l'avantage de la simplicité et montre l'importance des facteurs multiplicatifs dans la combinaison des percepts et des pulsions.

Il existe une autre approche ascendante, celle choisie par A. Drogoul pour le modèle EMF (Etho Modelling Framework) (Drogoul 1993) qui se trouve à la base

du système MANTA (Drogoul et Ferber 1994) de simulation de l'activité d'un nid de fourmis (cf. chap. 7). Sa problématique est un peu différente des autres, car il s'agit d'un des seuls modèles présentés ici (avec le modèle de L. Steels présenté section 5.7.3 ci-dessous) qui considèrent l'agent non pas isolément mais socialement en interaction avec d'autres agents. L'accent de ce modèle, fondé sur une architecture à base de tâches compétitives (cf. chap. 3), a été mis sur ses capacités d'adaptation sociales et, en particulier, sur sa capacité à spécialiser les compétences d'un agent au cours du temps. A la différence des autres modèles, il n'existe pas dans EMF de différence entre action appétitive, taxie et action consommatoire, mais une distinction de niveau entre tâches et actions. Une action est ici une primitive comportementale telle que suivre un gradient, prendre un objet ou consommer de la nourriture. Une tâche est un ensemble de telles primitives qui s'exécutent en séquence à l'image d'un programme classique. Néanmoins, une tâche peut être arrêtée à tout moment si les conditions environnementales et l'état de l'agent le nécessitent. La figure 5.18 présente une tâche composée de deux actions primitives dans le langage graphique de l'éditeur de tâches du système MANTA.

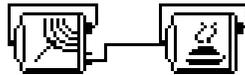


Figure 5.18: Représentation graphique de la composition d'une tâche de prise de nourriture dans le modèle EMF. L'agent suit d'abord un gradient d'odeur jusqu'à la source de la nourriture (action de taxie), avant de consommer la nourriture lorsqu'il peut le faire directement.

Comme dans les autres systèmes ascendants, une tâche est déclenchée par un composé des stimuli internes et externes. La particularité du modèle EMF par rapport aux autres modèles vient de ce que le système tend à se spécialiser dans le déclenchement préférentiel de certaines tâches ou, en d'autres termes, plus une tâche est déclenchée, plus elle aura de chances de se déclencher dans le futur. La figure 5.19 montre le diagramme de contrôle d'une tâche EMF. Les stimuli internes et externes sont multipliés comme dans le système de Werner, mais le poids de la tâche joue aussi comme effet multiplicateur pour augmenter la sensibilité du système à ces types de stimuli. Le comparateur prend toutes les tâches dont le niveau d'activation est supérieur à un seuil S . La tâche sélectionnée voit alors son poids augmenté d'un incrément.

De plus (mais cela n'est pas montré dans le diagramme) une tâche qui s'exécute voit son poids diminué afin d'implémenter un effet de "fatigue" et de faire en sorte que les autres tâches aient une chance d'être sélectionnées. Ce modèle, indépendamment de ses caractéristiques adaptatives pour un ensemble d'agents, est particulièrement simple à mettre en œuvre et à utiliser pour programmer d'autres comportements.

Néanmoins, l'un de ses problèmes majeurs réside dans sa difficulté à revenir sur une tâche lorsque celle-ci est interrompue par effet d'opportunisme. L'agent "oublie" ce qu'il était en train de faire. Par exemple si un agent affamé et assoiffé est interrompu dans sa quête de nourriture, il ne reprendra pas nécessairement cette activité. Ce qu'il fera dépendra entièrement des facteurs environnementaux et de

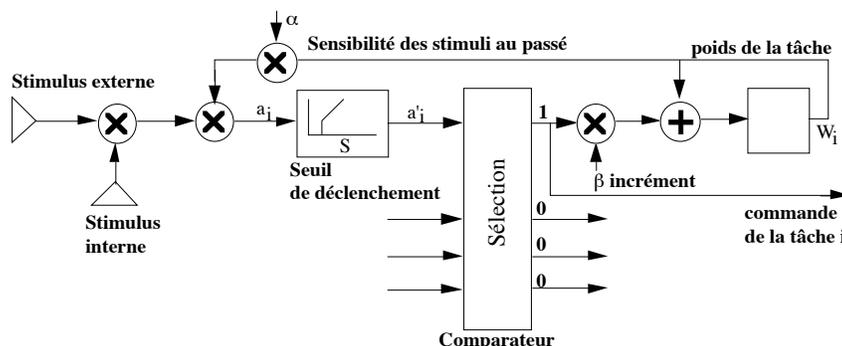


Figure 5.19: Diagramme du mécanisme de contrôle d'une tâche dans EMF

ses pulsions du moment. Ce modèle partage cette caractéristique avec la plupart des autres systèmes et il s'apparente notamment à celui de Werner par le fait qu'il conjugue de manière multiplicative les pulsions et les stimuli externes. Mais à l'encontre de ce dernier, les tâches, qui correspondent à des comportements de haut niveau (niveau 2 dans la hiérarchie de Tinbergen), sont élues par un mécanisme de sélection compétitive. Du fait de ses qualités intrinsèques, il est dommage que Tyrrell n'ait pas pu comparer ce modèle à celui de P. Maes ou à celui de Rosenblatt et Payton.

5.7.3 Sélection d'action ou combinaison dynamique

Le mécanisme de passage à l'acte par sélection d'action, bien que partagé par tous les systèmes considérés jusqu'à présent (hormis celui de Werner), est très controversé par ceux qui préconisent la définition de systèmes dynamiques. En effet, la sélection compétitive, ce que les Anglo-saxons appelle le *winner-takes-all* (le gagnant rafle toute la mise), les actions sont nécessairement incompatibles les unes avec les autres. Lorsqu'une action est sélectionnée, les autres sont *de facto* mises en sommeil, et il est impossible non seulement d'accomplir deux actions de manière simultanée, mais aussi d'exécuter une combinaison de ces actions: chacune d'elle est totalement indépendante des autres.

Dans les systèmes hiérarchiques, l'ordre est donné par en haut, alors que dans les systèmes ascendant ou distribués, le choix dépend des conditions environnementales et des liens que les actions ont instituées entre elles. Mais, dans tous les cas, une action et une seule est sélectionnée à un moment donné. De ce fait, les agents utilisant ce modèle ne savent pas à la fois fuir un prédateur et aller vers une source de nourriture. Ils font l'un ou l'autre en fonction de l'état dans lequel ils se trouvent. Seuls, parmi les modèles vus plus haut, celui des champs des potentiels et celui de Werner sont capables de combiner des actions parce qu'ils combinent en fait les tendances qui les poussent à agir. C'est contre cette limitation des mécanismes de sélection compétitive que L. Steels s'est insurgé en définissant son modèle de système dynamique (Steels 1994) que nous avons présenté dans le cadre des architectures au chapitre 3. Ce modèle peut être illustré par le diagramme de la figure 5.20.

Les comportements primitifs sont des fonctions (généralement linéaires) des

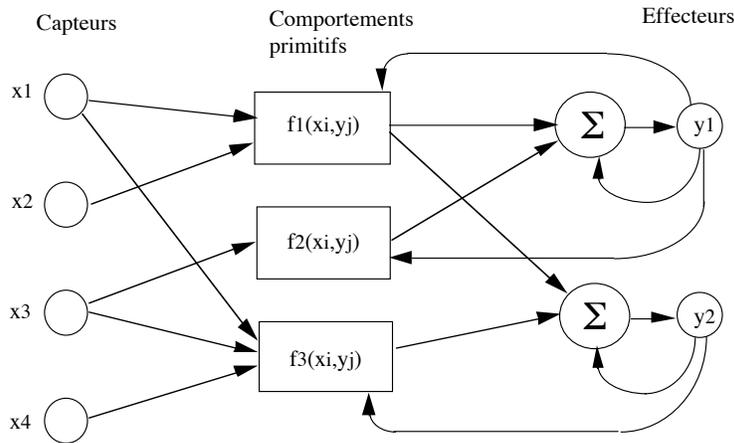


Figure 5.20: Modèle d'un système dynamique de comportement. Les résultats des comportements primitifs sont sommés avant de commander les effecteurs.

stimuli internes et externes et des valeurs des effecteurs (vitesse des roues gauche ou droite, état de la pince, etc.). Les résultats de ces fonctions sont ensuite sommées avant d'être envoyées comme valeur de commande aux effecteurs correspondant. La définition formelle de la valeur d'une commande y_j peut donc être donnée par l'équation dynamique discrète suivante:

$$y_i(t + 1) = \sum_{k=1}^p f_k(x_1(t), \dots, x_n(t), y_1(t), \dots, y_m(t))$$

où les $x_i(t)$ et les $y_j(t)$ donnent respectivement la valeur des stimuli i et des effecteurs j au temps t . L'intérêt de cette approche est de pouvoir combiner des comportements définis de manière indépendante et d'obtenir un comportement résultant qui est le fruit de la composition de ces comportements. Mais l'inconvénient du modèle de L. Steels est, nous l'avons dit, d'être encore trop lié à une architecture matérielle définie. Que l'on enlève un capteur ou que l'on ajoute une roue et il faut totalement reprendre la définition de ces fonctions. Il n'y a pas de véritable modularité entre les niveaux décisionnels définissant le passage à l'acte et les niveaux d'exécution qui mettent en œuvre ces comportements. En d'autres termes, c'est l'architecture qui est responsable du comportement, et il n'y a pas, à proprement parler, de modèle comportemental d'un agent indépendamment de son organisation matérielle.

Il est pourtant possible dans certains cas de s'abstraire de la représentation matérielle, tout en conservant la faculté de combiner les actions pour définir un comportement. Lorsque les actions peuvent s'exprimer dans un espace métrique, ce qui est souvent le cas, il est possible de leur appliquer les lois du calcul vectoriel et de combiner les comportements comme s'il s'agissait de vecteurs. Par exemple, si l'agent doit à la fois éviter un prédateur et aller chercher de la nourriture, alors il est possible de définir trois comportements: aller vers un but, éviter un danger en le contournant et fuir si le danger devient trop important (par exemple si le prédateur se dirige vers l'agent). Il suffit pour cela de considérer que les buts et les dangers produisent des champs de potentiels et donc des forces qui sont parallèles

aux gradients de ces champs (fig. 5.21). De plus, on peut définir le comportement de contournement comme un déplacement se situant le long d'une ligne de même potentiel ou, si l'on préfère, comme un déplacement perpendiculaire au gradient (cf. chap. 8).

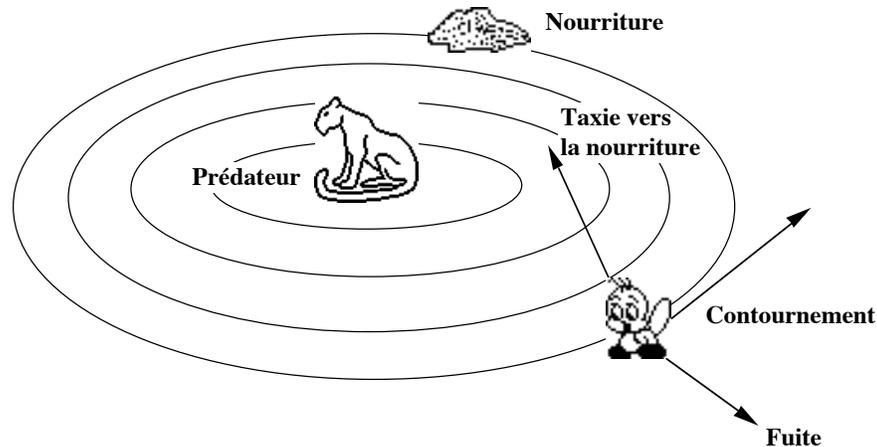


Figure 5.21: Le comportement d'un agent peut être donné comme la combinaison vectorielle de comportements élémentaires donnés sous la forme de vecteurs.

Ces comportements peuvent être ensuite combinés pour décrire tout un ensemble de comportements tels qu'aller vers une nourriture en contournant un danger, mais fuir si le danger est trop important. C'est la technique qu'a adopté K. Zeghal dans son système *Craash* d'évitement de collisions entre avions (Zeghal 1993). Chaque appareil considère les autres comme un danger potentiel: si deux avions risquent d'entrer en collision, ils cherchent à s'éviter en se contournant tout en essayant de conserver leur trajectoire de route, mais si par hasard la distance devient vraiment trop faible alors il vaut mieux qu'ils se fient quitte à prendre une autre direction. Le comportement global, décrit par le vecteur \vec{X} , peut être alors défini par l'équation:

$$\vec{X} = \frac{\alpha \vec{B} + \beta \vec{F} + \delta \vec{C}}{\alpha + \beta + \delta}$$

où les vecteurs \vec{B} , \vec{F} et \vec{C} représentent respectivement l'attraction vers un but, la fuite et le contournement d'un danger. L'intensité de ces vecteurs dépend du potentiel produit par les buts et les dangers. Les paramètres α , β et δ correspondent à des facteurs critiques qui dépendent de la position respective de l'agent, du but et surtout du danger. Si la distance entre le danger et l'agent devient trop importante, alors la valeur de β augmente au détriment de celle de α et de δ . La combinaison de ces comportements ne se réduit pas au contournement et à la fuite d'agents, et il est possible d'ajouter facilement de nouveaux comportements aux anciens. Par exemple, il est possible de définir des comportements de troupeaux (ou de patrouilles) dans lesquels les agents cherchent à se grouper (cf. chap. 8), lesquels viennent se superposer aux comportement d'attraction, de fuite et de contournement. L'avantage de cette technique, du fait de sa formalisation, est de pouvoir définir de manière

totallement abstraite le comportement d'un agent sans devoir faire référence à des capteurs ou à des effecteurs précis. De plus, la combinaison de ces actions est totalement modulaire, ce qui rend le comportement d'un agent aisément extensible. Enfin, il suffit d'utiliser les techniques multiplicatives des stimuli internes et externes pour bénéficier à la fois de persévérance et d'opportunisme. La combinaison vectorielle des actions constitue donc a priori la meilleure des solutions pour la définition de comportements réactifs dans l'état des connaissances actuelles.

5.8 Passages à l'acte intentionnels

Après les modes d'actions réactifs, nous pouvons commencer à étudier la notion d'intention telle qu'elle apparaît dans les travaux d'IAD et dans les sciences cognitives. Le problème est à la fois très simple et très ardu: Comment un agent parvient-il à faire ce qu'il fait dans un univers plus ou moins stable, où les liens qu'il entretient avec les autres agents peuvent à tout moment disparaître? La réponse à cette question, disent certains spécialistes en sciences cognitives, se trouve dans l'intention, c'est-à-dire dans la volonté consciente d'effectuer un acte, dans cette disposition mentale qui pousse à agir dans un certain sens en se proposant un but pour des actions futures.

Nous expliquons communément ce que font les autres en supposant qu'ils le font avec une certaine intention. Les ambitions, les désirs, les volontés et les motivations servent ainsi à expliquer le comportement des êtres humains. Pourtant il ne s'agit pas nécessairement d'une attitude scientifique. En effet, expliquer les conduites humaines par la notion d'intention est souvent considérée comme de la psychologie naïve, c'est-à-dire comme une pseudo-théorie constituée de concepts et de schémas de raisonnement que l'on emploie quotidiennement pour expliquer le comportement des êtres qui nous entourent, sans qu'ils soient acceptés par le milieu scientifique. Par exemple, si Marie me dit "Je crois que Paula aime les fleurs", je pourrais en inférer qu'elle a une croyance sur les préférences de Paula et qu'elle a l'intention de lui offrir des fleurs. J'attribue ainsi des états mentaux à Marie, tels que des croyances ou des intentions, sans que je sache précisément si elle possède effectivement ces états mentaux, et si son comportement est motivé par ces derniers.

Le fait que l'on parle à cet égard de "psychologie naïve" (folk psychology) ne signifie pas que l'approche pour comprendre ces phénomènes soit naïve elle-même. Bien au contraire, ces théories font l'objet d'un grand nombre de travaux et plusieurs modèles formels ont été proposés pour théoriser le passage à l'acte chez les agents, qu'il s'agisse d'individus humains ou artificiels (Pacherie 1993). Le terme clé de ces théories réside dans le concept d'intention, c'est-à-dire dans ce qui fait qu'un agent cognitif s'engage à accomplir une action.

Le concept d'intention a été (et demeure) l'un des plus controversés de l'histoire de la psychologie. Certains, les éliminativistes, refusent purement et simplement d'introduire ce concept dans leurs théories, considérant non seulement qu'il est inutile, mais aussi qu'il brouille inconsidérément les questions. D'autres, au contraire, pensent qu'il s'agit d'un des concepts essentiels de la psychologie et qu'on devrait lui donner un rôle central, car il constitue la clé de voûte de l'explication du comportement humain en termes d'états mentaux. Enfin, l'école psychanalytique n'y voit qu'un vague concept commode dans certains cas, mais qu'il convient généralement de substituer au profit du désir et des pulsions, seuls capables de rendre compte du comportement global de l'être humain dans ses aspirations et ses souffrances.

La notion d'intention n'épuise pas l'ensemble de la problématique du passage à l'acte pour les agents cognitifs. En effet, bien que les philosophes du XVIII^e siècle considéraient le passage à l'acte comme une simple résultante de conflits

entre tendances pondérées, la plus forte remportant la mise et définissant l'action à accomplir comme pour les agents réactifs, l'intention ne peut se résumer à un simple conflit entre tendances issues de pulsions, de demandes et de contraintes. Si les tendances prennent leur source dans des éléments extrêmement simple du psychisme de l'agent, les intentions supposent une analyse raisonnée de la situation présente et future et des possibilités d'action. Les agents cognitifs sont capables de prévoir dans une certaine mesure les conséquences de leurs choix, puisqu'ils décident en tenant justement compte des rapports de causalité existant entre les événements, c'est-à-dire en anticipant sur le futur. C'est cette capacité à prévoir les événements qui permet de dire qu'un agent "intentionnel" est nécessairement cognitif puisqu'il base son comportement sur une projection des situations futures, c'est-à-dire sur un modèle du monde qu'il est capable de manipuler pour évaluer les conséquences de ses actes et choisir l'action à entreprendre. De plus cette prise de décision est supposée être "rationnelle" comme nous l'avions indiqué plus haut, c'est-à-dire subordonnée aux objectifs de l'agent. C'est en effet par un calcul conscient que l'agent peut comparer les situations finales, qu'il peut calculer et comparer les avantages correspondant aux différentes situations et donc faire le bilan des différentes possibilités qui s'ouvrent à lui afin de toujours choisir la solution qui s'avère la "meilleure" en fonction des buts de l'agent et de ses possibilités.

De nombreux auteurs en IA et en philosophie ont proposé de considérer l'intention qu'un agent a d'accomplir un acte comme une des caractéristiques majeures des agents cognitifs. L'IAD n'est pas en reste, car l'intention est un concept fort important pour les systèmes multi-agents car il permet de relier les buts aux croyances et aux engagements en disposant d'une théorie du passage à l'acte des agents cognitifs. La plus grande partie des théories portant sur la notion d'intention dans le cadre d'agents cognitifs est issue des travaux effectués d'abord en philosophie analytique par Searle (Searle 1985), Castañeda (Castañeda 1975) et surtout Bratman (Bratman 1987) et reprise par des chercheurs en intelligence artificielle tels que Allen et Perrault (Allen et Perrault 1980) et surtout Cohen et Levesque (Cohen et Levesque 1990a) qui ont donné la base d'une théorie formelle des comportements intentionnels.

5.8.1 Théories logiques de l'intention

On distingue généralement entre deux types d'intentions (Searle 1985): l'intention d'agir dans le futur et l'intention de faire quelque chose maintenant. Par exemple, si je dis "j'ai l'intention d'aller aux Antilles cet hiver", l'intention est dirigée vers le futur (surtout si le moment de l'énonciation est l'été), et se trouve à la source d'un comportement, en particulier des autres intentions que je pourrais formuler pour atteindre mon but initial: aller prendre un billet d'avion, se renseigner pour trouver un endroit où séjourner (hôtel, club, maison à louer), etc. L'intention de faire maintenant est relative à l'action immédiate: si je dis "j'ai l'intention de prendre cette pomme", l'action qui s'ensuit doit être effectuée maintenant, et n'implique pas d'autres intentions secondaires, car elle préside à la réalisation de mon geste.

En fait ces deux types d'intentions ne coïncident pas totalement et renvoient à deux formes d'intentionnalité différente. La première suppose un mécanisme de

planification, alors que la seconde renvoie seulement au mécanisme de l'exécution des actions. La première est la plus intéressante du point de vue des agents intentionnels, car elle suppose un mécanisme d'anticipation et de planification des événements futurs caractéristique de ce que l'on appelle les agents intentionnels. Nous ne nous intéresserons qu'à celle-là et nous appellerons intention ce mécanisme de projection dans le futur d'un état du monde non réalisé actuellement.

L'intention s'inscrit dans le temps, l'action et les croyances. Un agent désire accomplir une action parce qu'il croit que ce qu'il va faire va permettre de satisfaire un but. De ce fait, toute théorie de l'intention nécessite une bonne définition de ce qu'est une action, un événement, un état du monde et comment ces événements et états du monde sont liés dans le temps. C'est d'ailleurs la partie généralement la plus complexe des théories des intentions, et aussi ce qui les rend les plus fragiles, du fait de la double difficulté à manipuler des opérateurs temporels et à rendre compte de l'ensemble des aspects de la notion d'action.

Bratman (Bratman 1990) introduit ce qu'il pense être les caractéristiques nécessaires à une théorie de l'action intentionnelle: en premier lieu, les intentions, pour aboutir, sont à l'origine de problèmes que les agents devront résoudre. Les intentions doivent aussi être cohérentes entre elles, ce qui implique qu'elles fournissent un *filtre d'acceptabilité* (screen of admissibility) pour adopter d'autres intentions. Par exemple, si un agent a l'intention de faire un œuf sur le plat et qu'il ne dispose que d'un œuf, il ne peut en même temps avoir l'intention de faire une "omelette". En outre, les agents suivent les résultats de leurs actions tout en étant prêts à replanifier leurs actions pour satisfaire leurs intentions. Enfin, il est nécessaire de distinguer entre les résultats intentionnellement voulus par un agent de ceux qui ne sont que les conséquences involontaires de l'action.

Définition: *on dit qu'un agent x a l'intention de faire une action a , ce que l'on note $\text{intention}(x, a)$ si x a comme but qu'une proposition P portant sur un état du monde soit vraie (noté $\text{but}(x, P)$) et que les conditions suivantes soient satisfaites:*

1. x croit que P appartient aux conséquences de a ;
2. x croit que P n'est pas vrai actuellement;
3. x croit qu'il est capable de faire a ;
4. x croit que a sera possible et donc que P pourra être satisfait.

De plus, on suppose que la persistance de l'intention durera tant que les conditions précédentes sont vérifiées et que P fait partie des buts de x , c'est-à-dire qu'un agent n'abandonnera pas son intention "pour rien". Par exemple, si un agent désire aller chercher du bois et pense que l'action est possible, c'est-à-dire qu'il pense qu'il existe un tas de bois dont il connaît le lieu et dont il connaît un chemin pour y aller, et sachant qu'il a la capacité de se déplacer, alors il aura l'intention d'y aller, cette intention étant à l'origine de son action d'aller chercher le bois.

Cependant cette définition repose sur la notion de "croire qu'une action est possible". Dans le cas d'un univers mono-agent statique, c'est-à-dire dont les seules

modifications de l'état du monde dépendent de l'agent, une action est *réalisable* (ou possible) si les conditions pour accomplir cette action sont réalisées à l'instant présent, ou s'il existe un plan, c'est-à-dire une suite d'actions exécutables par l'agent qui permette de vérifier les conditions d'application de cette action que l'agent a l'intention de faire.

Que le monde évolue ou que plusieurs agents puissent modifier le monde par leurs propres actions, et il n'est pas sûr que l'agent pourra effectivement aller jusqu'au bout de son intention. Dans ce cas, les conditions de réalisation d'une action sont nettement moins simples, car elles reposent sur les possibilités d'anticiper sur l'évolution du monde. Par exemple, si j'ai l'intention d'aller faire du ski, cela signifie que je pense que les conditions nécessaires sont vérifiées et en particulier qu'il y a de la neige. Or, si j'ai l'intention non pas d'aller faire du ski maintenant mais dans quelques mois, cela signifie que je dois prévoir qu'il y aura de la neige au bon moment, mais rien ne m'assure de cette situation. Tout dépend des connaissances générales que je peux avoir sur la question. Je peux en effet penser que le fait d'aller faire du ski en février dans une station savoyarde est possible (en tout cas par défaut ou de manière statistique). Rien ne me garantit totalement cette possibilité. Si la saison est particulièrement mauvaise, la neige pourra ne pas être au rendez-vous.

Il en est de même dans le cas d'univers multi-agents dans lesquels les actions des autres peuvent modifier l'état du monde à tout moment. Pour savoir si une action est possible, je dois prendre en compte non seulement mes compétences et l'évolution naturelle du monde, mais aussi les actions et surtout les intentions des autres agents. Par exemple, si le robot Clotaire sait que le robot Chilpéric a l'intention d'aller chercher un outil, il peut le prendre en compte dans l'élaboration de ses propres intentions et donc se déterminer en utilisant les actions de Chilpéric comme s'il s'agissait du résultat de ses propres actions et en supposant que ce dernier n'abandonne pas ses actions de manière inconsidérée.

De ce fait, l'élaboration des intentions s'appuie sur les régularités du monde, c'est-à-dire sur le fait que l'environnement et les autres agents *s'engagent* sur les actions futures qu'ils envisagent d'accomplir. Toute définition d'intention dans des systèmes multi-agents cognitifs repose sur la possibilité de planifier des actions conjointes entre agents multiples dans un univers évolutif.

Néanmoins, cette définition pose tout de même un problème: rien n'est dit sur le fait que l'agent qui a l'intention de faire l'action a poursuit effectivement cette intention et va mettre tout ce qui est en son pouvoir pour réaliser cette action. Avoir l'intention de faire quelque chose, c'est aussi s'engager, au moins envers soi-même, à essayer de l'accomplir. De ce fait, les théories formelles sur l'intention prennent la notion d'engagement comme un élément fondamental du passage à l'acte d'un agent cognitif. Les travaux les plus connus dans ce sens sont ceux de Cohen et Levesque et la tentative de mise en œuvre de Yoav Shoham.

5.8.2 La théorie de l'action rationnelle de Cohen et Levesque

Philip Cohen et Hector Levesque ont défini une logique de l'action rationnelle qui a eu beaucoup de retentissements dans le monde des univers multi-agents et qui a vu

un nombre importants de recherches reprendre le formalisme et les fondements de cette théorie. On trouvera dans (Cohen et Levesque 1990a; Cohen et Levesque 1990b) les détails de cette théorie et en particulier les définitions proprement formelles. Nous ne présenterons ici que ses caractéristiques les plus générales, ainsi que ses conséquences sur la définition de mécanismes intentionnels.

La théorie de Cohen et Levesque est particulièrement intéressante pour deux raisons: elle fournit une notation uniforme et pratique pour lier des notions telles que les intentions, les croyances et les actions et, surtout, elle propose une conception relativement intégrée de l'action rationnelle.

La notation

Le formalisme de Cohen et Levesque repose sur une logique modale, c'est-à-dire sur un système logique des prédicats du premier ordre (avec égalité) augmenté d'un certain nombre d'opérateurs représentant des attitudes propositionnelles ou des séquences d'événements. Les deux modalités de base de leur logique sont la croyance et le but. Le premier $\text{croire}(x, p)$ signifie que l'agent x croit p , et le second, $\text{but}(x, p)$ indique que x a pour but que p soit satisfait. D'autres prédicats servent à gérer la notion d'action ou plus exactement celle d'événement, car ici l'action est considérée simplement comme un événement qui rend possible la satisfaction d'une proposition. Ils ne prennent donc pas en compte les "effets de bords" ou les conséquences des actions, ni ne considèrent les problèmes de simultanéité comme le chapitre 4 l'avait montré. Ces prédicats supplémentaires sont $\text{agent}(x, e)$ pour signifier que l'agent x est le seul agent initiateur de la séquence d'événements e , $e1 \leq e2$ qui stipule que $e1$ est une séquence d'événements précédant $e2$, et $\text{vaSePasser}(e)$ et $\text{achevée}(e)$ qui indiquent qu'une séquence d'événements est juste sur le point de se passer ou vient d'être achevée.

Dans leur formalisme, la notion d'action n'est pas distincte de celle de séquence d'événements. On parlera ainsi indistinctement d'action ou d'événements. Ces séquences peuvent être structurées à l'aide d'opérateurs semblables à ceux de la logique dynamique (Harel 1979):

- $a; b$ est la composition séquentielle d'actions. Cela signifie que l'on fait donc d'abord a , puis b lorsque a est terminé.
- $a|b$ est le choix non déterministe d'actions. Cela signifie que l'on fait soit a soit b .
- $p?$ est une action qui teste si p est vérifiée. Si elle n'est pas vérifiée, cette action "bloque" les autres actions empêchant la suite de la séquence de s'accomplir.
- a^* est la répétition. L'action a se répète indéfiniment, sauf si elle est interrompue par un test ne vérifiant pas une certaine propriété.

A partir de ces opérateurs de base, il est possible de construire des structures de contrôle plus élaborées de la manière suivante:

- **si p alors a sinon b** $=_{def} p?; a | \neg p?; b$, ce qui signifie que la conditionnelle est une action dans laquelle on effectue **a** dès que **p** est vérifiée et **b** dès que **p** est fausse.
- **tantque p faire a** $=_{def} (p?; a)*; \neg p?$, ce qui signifie que l'on répète **a** tant que **p** est vraie et qu'ensuite **p** est fausse.
- **finalt(p)** $=_{def} \exists e \text{ vaSePasser}(e; p?)$, décrit l'opérateur finalement (eventually) qui indique qu'il existe une séquence d'action **e** après laquelle **p** sera vraie.
- **tjrs(p)** $=_{def} \neg \text{finalt}(\neg p)$, décrit l'opérateur toujours, qui signifie que **p** est toujours vraie (littéralement **p** ne sera jamais fausse).
- **plusTard(p)** $=_{def} \neg p \wedge \text{finalt}(p)$, qui signifie que **p** sera vraie plus tard, c'est-à-dire que **p** n'est pas vraie maintenant mais sera finalement vraie.
- **avant(p, q)** $=_{def} \forall c \text{ vaSePasser}(c; q?) \Rightarrow \exists a(a \leq c) \wedge \text{vaSePasser}(a; p?)$, c'est-à-dire que **p** vient avant **q** si quand **q** est vraie **p** a été vraie précédemment.

et d'introduire des abréviations telles que

$$\begin{aligned} \text{achevé}(x, a) &=_{def} \text{achevée}(a) \wedge \text{agent}(x, a) \\ \text{vaFaire}(x, a) &=_{def} \text{agent}(x, a) \wedge \text{vaSePasser}(a) \end{aligned}$$

qui permet de représenter la notion d'action achevée et accomplie par un agent **a**. La définition de tous ces opérateurs est donnée par une axiomatique modale (en particulier l'opérateur de croyance correspond à une axiomatique S5, cf. section 5.3.3) et par une sémantique des mondes possibles sur l'ensemble des opérateurs de croyances, de buts et de constructions de séquences. A partir de ces définitions, il est possible de tirer tout un ensemble de conséquences, telles que celle-ci:

$$\text{finalt}(q) \wedge \text{avant}(p, q) \Rightarrow \text{finalt}(p)$$

c'est-à-dire que si **q** finalement est vérifiée et que **p** est avant **q** alors **p** sera nécessairement vérifiée aussi.

L'intention et l'action rationnelle

Pour Cohen et Levesque, la théorie de l'action rationnelle reprend les idées de Bratman telles que nous les avons résumées ci-dessus. L'intention y est conçue comme un choix et un engagement envers son choix. Leur concept de but est assez particulier: ils n'utilisent pas une formalisation de la notion intuitive du mot "but", mais considèrent que la formule **but(x, p)** signifie que **p** sera vraie dans tous les mondes possibles dans lesquels leurs buts seront achevés. Ainsi ils peuvent faire l'économie du concept de but et ne considérer que les conséquences des buts.

L'intention est alors une sorte de *but persistant*, c'est-à-dire un but qu'un agent conserve tant que certaines conditions demeurent. Dans leur notation, un but persistant s'exprime ainsi:

$$\begin{aligned} \text{but-p}(x,p) =_{def} & \text{but}(x,\text{plusTard}(p)) \wedge \\ & \text{croire}(x,\neg p) \wedge \\ & [\text{avant}(\text{croire}(x,p) \vee \text{croire}(x,\text{tjrs}(\neg p))), \\ & \neg\text{but}(x,\text{plusTard}(p))] \end{aligned}$$

ce qui se lit ainsi: on dit qu'un agent x a un but persistant p s'il a le but que p soit vérifié plus tard, qu'il croit que p n'est pas vérifié pour l'instant, et qu'il abandonnera son but si soit il croit que p est vrai soit il croit que p ne pourra jamais être vrai. De ce fait, les buts persistants introduisent la notion d'engagement quoique de manière implicite. En effet, l'agent persistera dans son but sauf si une condition particulière est vérifiée.

De ce fait, si l'on suppose que les agents un jour abandonneront leurs buts (il ne peuvent rester indéfiniment prostrés et tenter d'atteindre indéfiniment leurs buts). On peut donc en déduire que:

$$\text{but-p}(x,p) \Rightarrow \text{finalt}[\text{croire}(x,p) \vee \text{croire}(x,\text{tjrs}(\neg p))]$$

c'est-à-dire que si un agent possède un but persistant il lui arrivera finalement soit de croire que p est vrai soit que p ne sera jamais vrai. A partir de ces premières définitions, Cohen et Levesque définissent l'intention pour un agent x de faire une action a comme un engagement "fanatique" d'accomplir l'action a dès qu'elle peut être effectuée (en fait dès que l'agent croit qu'elle peut s'effectuer), ce qui s'écrit ainsi:

$$\text{intention-fan}(x,a) =_{def} \text{but-p}(x,\text{achevée}(x, \\ \text{croire}(x,\text{vaSePasser}(a))?) ; a))$$

Cette définition permet d'avoir une idée de la notion d'intention, mais elle est trop stricte. Elle concerne ce que Cohen et Levesque appellent des agents "fanatiques", qui ne peuvent supprimer leur but une fois que ces derniers se sont engagés à les satisfaire. Un agent implémenté avec une telle notion d'intention se comportera comme ces robots tueurs des films de science-fiction que l'on ne peut arrêter une fois qu'ils ont reçu leurs consignes et qui deviennent ainsi dangereux même pour ceux qui les ont programmés.

Cohen et Levesque ont donc défini une autre sorte de but qualifié de *persistant-relatif*, c'est-à-dire un but persistant lié à une condition de satisfaction supplémentaire. Cette dernière fonctionne comme une garde dans la réalisation de l'action, ce que ces auteurs considèrent comme les *raisons* pour un agent d'accomplir une action.

$$\begin{aligned} \text{but-p-rel}(x,p,q) =_{def} & \text{but}(x,\text{plusTard}(p)) \wedge \text{croire}(x,\neg p) \wedge \\ & [\text{avant}([\text{croire}(x,p) \vee \text{croire}(x,\text{tjrs}(\neg p)) \vee \text{croire}(x,\neg q)], \\ & \neg\text{but}(x,\text{plusTard}(p))] \end{aligned}$$

A partir de cette définition il est possible de caractériser une nouvelle notion d'intention relative à une condition q qui s'exprime ainsi:

$$\begin{aligned} \text{intention-rel}(x,a,q) =_{def} \\ \text{but-p-rel}(x,\text{achevé}(x,\text{croire}(x,\text{vaSePasser}(a))?) ; a), q) \end{aligned}$$

c'est-à-dire que x a l'intention d'accomplir a relativement à la condition q , s'il a le but persistant relatif à q d'accomplir a dès qu'il croit que cette action peut être effectuée. Ces auteurs ont aussi défini la notion d'*intention vague* ou d'*intention abstraite*. Par exemple, l'intention de "devenir riche" est vague parce que l'agent ne dispose pas nécessairement d'une action immédiate qui lui permettrait de satisfaire ce but. L'intention se rapporte alors à la satisfaction d'une proposition et non à la réalisation d'une action. Avoir l'intention dans ce sens consiste alors à croire qu'il existe une séquence d'action e' telle qu'il soit possible ensuite d'accomplir l'action e qui satisfasse p :

$$\begin{aligned} \text{intention-abs}(x,p) =_{def} & \text{but-p}(x, \exists e \text{ achevé}(x, \\ & [\text{croire}(x, \exists e' \text{ vaFaire}(x, e'; p?) \wedge \\ & \neg \text{but}(x, \neg \text{vaFaire}(x, e; p?))]?; e; p?)) \end{aligned}$$

La théorie en question

Le principal intérêt de cette théorie est de fournir un langage relativement simple et concis et de montrer qu'à partir d'un tel langage il est possible de définir une théorie de l'action rationnelle qui puisse se formaliser et ainsi prêter à discussion de manière rigoureuse. On peut aussi voir que de nombreuses situations humaines ou artificielles peuvent se traduire sous cette forme. Par exemple, pour un agent calculateur qui a besoin de certains paramètres, son intention consiste à chercher s'il existe un agent qui possède ces paramètres, tant que le problème initial n'a pas été résolu, ce que l'on peut exprimer de cette manière:

$$\begin{aligned} \text{intention-rel}(\text{Agent23}, \text{trouverParamètres}(\{p1, p2\}), \\ \neg \text{trouvé}(\text{problème1})) \end{aligned}$$

Plusieurs chercheurs ont élaboré des théories multi-agents à partir de ce formalisme. Par exemple, J. Galliers a décrit une théorie des actions de coopération en utilisant cette technique (Galliers 1988) et C. Castelfranchi et R. Conte ont élaboré une théorie des intentions sociales et des influences (Castelfranchi et Conte 1992; Conte et al. 1991). Ce type de formalisation présente cependant quelques lacunes:

1. Les actions sont considérées comme des constructions d'événements, et non comme des opérations applicables sur le monde. De ce fait, ce type de théorie ne peut prendre en compte les modifications apportées par les actions des agents et donc s'avère inadéquat pour résoudre des problèmes de planification (mono ou multi-agents) ou plus généralement les phénomènes d'interactions entre agents. Néanmoins, avec des agents purement communicants, il est possible (et Cohen et Levesque l'on fait (Cohen et Levesque 1990c)) d'introduire des opérateurs de communications et ainsi de réaliser une extension de leur formalisme pour traiter les interactions entre agents. Cette extension passe par la théorie des actes de langage que nous verrons au chapitre 6.
2. Il est difficile de prouver mécaniquement de telles théories: les opérateurs modaux multiples ainsi que les constructions de logique dynamique ne rendent

pas facile l'automatisation d'inférence permettant de juger de la validité ou non d'une formule. Tout doit se faire "à la main" en utilisant des considérations sémantiques sur les mondes possibles, ce qui est assez mal commode.

3. Quoi qu'en disent Cohen et Levesque, il ne s'agit nullement d'une description opératoire des mécanismes mis en œuvre pour qu'un agent rationnel passe à l'acte, mais seulement des conditions minimales qui doivent être satisfaites pour que l'on puisse parler d'intention chez un agent.

Par exemple, la notion d'engagement n'existe que par défaut et le passage à l'acte apparaît nulle part. On a l'impression que l'agent a toujours le désir d'accomplir l'action mais on ne voit pas quand il accomplit cette action ni s'il l'accomplit explicitement. De ce fait, à partir des spécifications de Cohen et Levesque, on pourrait réaliser un agent, peut-être un peu irrationnel il faut bien le dire, qui passerait son temps à avoir le but d'accomplir des actions sans les exécuter effectivement. Mais il ne faut pas en porter grief à Cohen et Levesque, mais plutôt au formalisme qui ne permet pas de prendre en compte l'opérationnalité de l'intention et se révèle donc incapable de décrire effectivement le passage à l'acte d'un agent.

D'une manière générale, ce type de théorie présente l'avantage de bien dégager quelques-uns des concepts clés de l'intention. Cependant, il a tendance à simplifier à l'extrême les problèmes de l'action et en particulier à ne pas prendre en compte les révisions de l'action ni à intégrer le mécanisme d'élaboration de ces intentions et donc des conflits qui peuvent survenir entre les motivations de diverses natures et les contraintes. Cette théorie suppose qu'il existe une relation 1 : 1 entre les buts et la séquence d'actions qui peut satisfaire les buts. Elle est prise en défaut dès qu'il existe plusieurs moyens d'atteindre un même objectif, ce qui est pourtant le cas le plus courant.

Par exemple, si une firme veut monter une revue de communication d'entreprise, elle peut soit essayer de l'éditer en interne sur un système de micro-édition, soit utiliser les services d'une entreprise externe spécialisée en édition. Ces deux possibilités atteignent le même but bien que les moyens soient différents. Or ce type de théorie ne dit rien sur cette prise de décision. Elle permet seulement, en codant explicitement les raisons d'abandon, de pouvoir remettre en cause un but si celui-ci ne convient plus. De ce fait, de telles théories de l'action rationnelle devraient être étendues par des considérations d'évaluation de buts et de moyens, en utilisant des techniques semblables à celle de la théorie des jeux ou de la décision.

Il serait préférable de rendre compte de la notion d'intention à partir de mécanismes réactifs comme ceux que nous avons examinés section 5.7. Cela permettrait de faire le pont entre les agents réactifs et les agents cognitifs en montrant que les intentions, les choix et les engagements sont le produits de mécanismes élémentaires fondés sur des considérations de poids, de seuils et de renforcement. Mais ce travail reste à faire.

Toute bonne théorie de l'action rationnelle devrait pouvoir intégrer et coordonner les notions de but, de croyance, de décision, de passage à l'acte, d'engagement, de communication et de résultat des actions de manière cohérente. Il serait de plus

intéressant que ces théories puissent aider à l'implémentation d'agents fonctionnant sur ce principe.

La théorie de Yoav Shoham, élaborée dans son article “Agent Oriented Programming” (Shoham 1993), correspond à cet objectif en proposant un langage de programmation “orienté agent” de la même manière qu'il existe des langages “orientés objets”. Il s'agit de considérer que l'on programme un agent à l'aide d'éléments de haut niveau tels que des buts, des choix, des compétences, des croyances, etc. et que les types de message que les agents s'échangent se réfèrent à mécanismes de communication de haut niveau eux aussi, en définissant des messages d'information, de requête, d'offre, de promesse, de refus, d'acceptation, etc.

Son modèle de langage est fondé sur des logiques et une sémantique en termes de théorie des mondes possibles. Ses deux modalités de base sont la croyances et l'engagement, représentées respectivement par $\text{croire}(x, p)^t$ pour dire que l'agent x croit la proposition p au temps t , et $\text{engagement}(x, y, a)^t$ pour signifier que l'agent x s'est engagé au temps t auprès de y pour accomplir l'action a . Toutes les autres modalités sont dérivées de ces deux primitives. Choisir d'effectuer une action est définie comme un engagement envers soi-même et la capacité à exécuter une action comme l'implication que le choix de p entraîne p .

$$\begin{aligned}\text{choix}(x, p)^t &=_{def} \text{engagement}(x, x, p)^t \\ \text{capable}(x, p)^t &=_{def} \text{choix}(x, p)^t \Rightarrow p\end{aligned}$$

A partir de ces conceptions initiales, il a développé AGENT0, un modèle d'agents capables d'exécuter des actions, d'informer d'autres agents et de leur envoyer des requêtes. Bien que très intéressante parce que présentant une tentative d'opérationnaliser des spécifications du type de celle de Cohen et Levesque, cette théorie demeure encore un peu simpliste et montre de ce fait la difficulté d'introduire un véritable langage “orienté agent” et surtout une théorie du passage à l'acte pour les agents cognitifs qui puisse s'opérationnaliser. D'autres tentatives ont été faites dans ce sens. Signalons en particulier celle de Chaib-Draa (Chaib-Draa 1989) qui a développé, à partir des travaux de Cohen et Levesque (ceux que nous avons présentés ici et ceux qu'ils ont effectués sur la communication entre agents à partir des actes de langages que nous développerons dans le chapitre suivant), un système permettant à plusieurs agents de coopérer pour résoudre des problèmes de planification relativement simples. Récemment, un petit groupe de chercheurs, autour d'H. Levesque, d'Y. Lespérance et de R. Reiter, ont repris l'idée de Y. Shoham et ont développé, sur des bases personnelles, une autre approche logique plus complète de la programmation d'agents qui décrit directement le fonctionnement interne des agents (Lespérance et al. 1995).

Chapter 6

Communications

6.1 Aspects de la communication

Les communications, dans les systèmes multi-agents comme chez les humains, sont à la base des interactions et de l'organisation sociale. Sans communication, l'agent n'est qu'un individu isolé, sourd et muet aux autres agents, refermé sur sa boucle perception-délibération-action. C'est parce que les agents communiquent qu'ils peuvent coopérer, coordonner leurs actions, réaliser des tâches en commun et devenir ainsi de véritables êtres sociaux. La communication s'exprime comme une forme d'interaction dans laquelle la relation dynamique entre les agents s'exprime par l'intermédiaire de médiateurs, les *signaux*, qui, une fois interprétés, vont produire des effets sur ces agents. Il existe un grand nombre d'approches de la communication: les sciences humaines et en particulier la linguistique et la philosophie du langage ont développé tout un ensemble de concepts dont je m'inspirerai pour présenter les principales approches de la communication dans les systèmes multi-agents.

6.1.1 Signes, indices et signaux

Les communications s'effectuent par l'intermédiaire de signaux, chaque signal caractérisant un indice ou un signe. A la différence du langage courant où ces termes sont presque synonymes (on dit qu'un médecin diagnostique une maladie à partir de signes ou d'indices cliniques), la linguistique et la sémiologie, c'est-à-dire la science des signes, distinguent très précisément les signes des indices et des signaux. Un signal est l'élément le plus primitif. On peut le définir comme une marque, une trace, une modification de l'environnement sur le plan mécanique (audition, toucher), électromagnétique (vision) ou chimique (système olfactif ou gustatif), éventuellement porteur d'une information pour celui qui est capable de percevoir ce signal. Les signaux se propagent en vertu de leurs propres lois de propagation et en tant que tels ne signifient rien. Pour signifier, il leur faut deux conditions: un agent ayant des capacités de perception suffisantes pour être sensible à ce signal (par exemple, en tant qu'être humain notre ouïe n'est sensible à des vibrations de l'air que dans la bande passante 30-16000 Hz) et un système interprétatif capable de transformer ce signal en un sens ou tout du moins en un comportement. Lorsque

le signal ne produit qu'un comportement, on dit qu'il s'agit d'un *stimulus*. C'est le cas pour la plupart des animaux et, pour ce qui nous concerne, pour l'ensemble des agents réactifs. Comme nous l'avons vu au chapitre 5, un signal, lorsqu'il est reçu par un tel agent, est capable de renforcer une tendance et, si cette dernière est suffisante, de le faire passer à l'acte. Lorsque le signal ne produit pas seulement un comportement, mais s'insère dans un système cognitif lui permettant de le rendre intelligible, on dit alors que le signal est *porteur de signification* et prend le statut de signe. Comme le remarque U. Eco (Eco 1988), le signe est défini avant tout comme *aliquid stat pro aliquo*, c'est-à-dire quelque chose (une expression concrète, un son, un mouvement, une figure, etc.) qui renvoie à quelque chose d'autre (une situation, un objet physique, une action, une procédure, une autre expression, etc.). Les signaux inclus dans les signes n'ont par eux-mêmes rien de commun avec le sens qu'ils sont destinés à transmettre. On dit avec Saussure que les signes linguistiques sont arbitraires, c'est-à-dire qu'ils dérivent d'une convention et que leur forme n'est pas liée à ce qu'ils réfèrent (Baylon et Mignot 1991).

Il existe plusieurs catégories de signes. Il y a d'abord les signes linguistiques qui sont issus d'un langage, c'est-à-dire d'une structure de communication possédant à la fois une première articulation phonétique, les sons se combinant pour composer des mots, et une seconde articulation grammaticale dans laquelle les mots sont assemblés pour construire des phrases. Les signes linguistiques, qui sont les constituants essentiels des communications entre agents cognitifs, associent à une forme signifiante, qui prend généralement la forme d'une chaîne de caractères, un signifié, c'est-à-dire un concept, une idée, la représentation de quelque chose ou d'un état mental. Mais les signes linguistiques ne constituent qu'une partie de l'univers des signes. Les indices notamment forment une catégorie importante de signes qui ne sont pas construits à partir d'un langage doté d'une double articulation, mais renvoient à la construction d'hypothèses sur une situation ou une chose. Par exemple, des traces d'animaux constituent des indices de leur passage, les symptômes d'un malade sont les indices d'une maladie, la fumée est l'indice d'un feu. Le procédé cognitif par lequel un agent passe d'un indice à une hypothèse est appelé *abduction*. Il est caractéristique de tous les raisonnements d'investigation (pensons en particulier à Sherlock Holmes capable de reconstruire toute la personnalité d'un individu à partir de quelques indices) et de diagnostic¹.

6.1.2 Définition et modèles de communication

Il existe de nombreuses théories de la communication mais elles reposent essentiellement sur des variantes de la *théorie de la communication* issue des recherches en télécommunications des années 40 développées par Shannon et Weaver (Shannon et Weaver 1948). Dans ce modèle, l'acte de communication consiste en une transmission d'information d'un *émetteur* vers un *récepteur* (ou *destinataire*), cette information étant *codée* à l'aide d'un *langage* et *décodée* à l'arrivée par le destinataire. Cette

¹Pour avoir une bonne idée des théories du signe et du sens en linguistique et en sémiologie à partir d'une lecture provenant des sciences cognitives, on pourra essayer de se procurer (Enjalbert 1993).

information est transmise dans un *canal* (ou *médium*) de transmission qui peut être bruité. Le *contexte* est la situation dans laquelle les interlocuteurs sont placés (ce contexte est lui-même décomposé en contexte de l'émetteur et contexte du destinataire). La figure 6.1 illustre ce modèle.

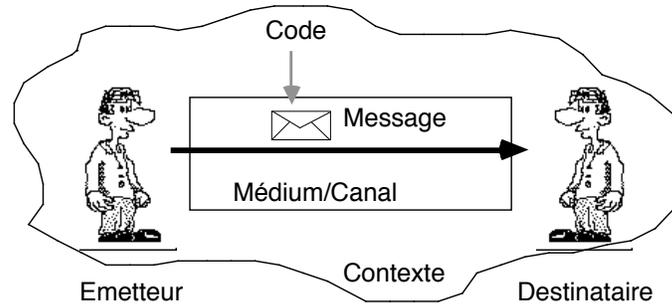


Figure 6.1: Le modèle classique de la théorie de la communication

Le modèle initial de cette théorie était très simpliste et totalement lié à des motifs techniques, le problème étant alors de transmettre des données de manière qu'elles soient le moins perturbées par le canal de transmission. Depuis, ce modèle s'est beaucoup amélioré en prenant en compte les états mentaux des interlocuteurs d'une part et le contenu des messages d'autre part, en distinguant différentes situations de communication et en classant les fonctions assurées par les messages, comme nous le verrons ci-dessous. La notion de communication s'est depuis structurée et, de la simple transmission d'information initiale, on est passé à des formes plus élaborées, telles que les actes de langage et les structures conversationnelles qui mettent l'accent sur la notion d'interaction dans les communications.

Mais la communication va au-delà de l'échange verbal. En effet, cette vision de la communication comme un processus d'émission/réception de messages ne constitue pas la seule théorie en vigueur. Pour des chercheurs en sciences humaines (ethnologie, sociologie et psychologie) réunis dans une école de pensée connue sous l'appellation d'*école de Palo Alto*, cette conception est très liée à son origine technique. Elle porte essentiellement sur la communication verbale, consciente et volontaire qui s'établit entre deux êtres humains et ne permet pas de comprendre le fait de communication dans son intégralité, c'est-à-dire l'ensemble des interactions signifiantes s'exprimant entre plusieurs êtres. *La communication est alors considérée comme un processus social permanent intégrant de multiples modes de comportement: la parole, le geste, le regard, la mimique, l'espace interindividuel, etc. La communication est un tout intégré* (Bateson et al. 1981). Dans ce cas, les messages n'ont plus de signification intrinsèque: seul le contexte est capable d'apporter une lumière sur la signification des modes d'interaction.

Enfin, dans le contexte des sciences cognitives, certains remettent en cause le modèle classique de communication définie à partir de considérations cognitivo-linguistiques en constatant que, dans ce mécanisme de codage/décodage, le code constitue une connaissance partagée par les interlocuteurs. Mais cette connaissance commune existe-t-elle vraiment dans tous les cas? Est-ce bien la même chose qui est émise et interprétée? Le code n'est-il pas un leurre de notre perception naïve des

formes de communication? C'est en répondant par la négative à toutes ces réponses et en mettant l'accent sur les mécanismes inférentiels des interlocuteurs que Sperber et Wilson (Sperber et Wilson 1986) proposent un schéma de communication qui repose sur l'aspect interprétatif des communications: un énoncé qui a du sens est un énoncé pertinent, c'est-à-dire l'énoncé "minimal" nécessaire pour que l'émetteur parvienne à se faire comprendre et donc à faire en sorte que le destinataire ait le moins de mal possible à le comprendre. Chacun des interlocuteurs se construit un ensemble d'hypothèses sur les représentations et les objectifs de l'autre: le locuteur ne produit que les énoncés qui lui paraissent nécessaire pour que le destinataire ait effectivement compris ses intentions communicationnelles. Cette conception de la communication, qui est encore très peu utilisée dans le monde de l'intelligence artificielle distribuée, est pourtant très intéressante car elle repose entièrement sur des mécanismes inférentiels et sur la construction de modèles de l'autre. De plus, elle suppose que le destinataire dispose d'un ensemble de capacités de traitement qui présentent de grandes analogies avec les différents systèmes de compréhension de langage naturel développés par l'intelligence artificielle (Sabah 1989).

6.1.3 Catégories de communication

Malgré toutes les critiques prononcées à son égard, le schéma classique de la communication est apparu comme suffisamment pertinent par nombre de linguistes pour qu'ils l'intègrent dans leur propre description des faits de langues en classant les manières de communiquer par rapport aux relations qui s'expriment entre les différentes entités concernées par l'acte de communication. Ces relations portent sur:

1. la liaison émetteur-destinataire
2. la nature du médium
3. l'intention de communiquer

Liaison émetteur-destinataire

Quelle est la relation qui unit le receveur d'une communication à son émetteur? Lorsque le destinataire est connu de l'émetteur, ce dernier peut lui adresser des messages en particulier et ainsi instaurer une communication individuelle. On dit alors que les communications sont effectuées selon un mode *point à point*. C'est ce type de communication qui est généralement le plus employé dans les agents cognitifs. On le notera ainsi:

(id) <émetteur> : <destinataire> << énoncé>

où <id> est l'identifiant (optionnel) du message, <émetteur> est l'agent émetteur du message, <destinataire> est l'agent à qui est destiné le message et <énoncé> correspond au contenu du message. Par exemple, le message suivant:

(M1) A : B << bonjour>

est une communication point à point qui se résume pour A à dire “bonjour” à B. Si le destinataire n’est pas connu de l’émetteur, le message est envoyé sous un mode appelé *diffusion* (broadcasting) à tout un ensemble d’agents, ces derniers étant liés à l’émetteur par une relation de voisinage (proximité dans l’environnement, connexité dans un réseau, etc.). Ce type de communication est très utilisé dans les systèmes dynamiques dans lesquels des agents peuvent apparaître ou disparaître. Il s’emploie en particulier dans des protocoles d’allocation de tâches comme celui du réseau contractuel dont nous parlons au chapitre 7. On notera à la place du destinataire le mot clé **Tous** ou la relation de voisinage reliant l’émetteur au destinataire. Par exemple, la communication

(M2) A : Tous << bonjour

consiste à dire “bonjour” à tous les agents du système, et

(M2') A : {x | dist(A,X)} < d} << bonjour}

à diffuser le même message à tous les agents dont la distance avec A est inférieure à une valeur d. A la différence des messages comportant un destinataire explicite, les messages diffus ne sont pas nécessairement des primitives du système. Ils peuvent en effet être décrits par combinaisons de messages directs à condition qu’il existe une entité responsable de la diffusion. Par exemple, le message diffus

(M3) A : {x | P(x)} << M

peut s’exprimer comme la combinaison des message M3' et M3" suivants:

(M3') A : C << diffuser(M)

(M3") pour tout x de Receveurs(C), C : x << M

où C est une entité de diffusion, un médiateur, qui se contente de déléguer un message à tous ses receveurs. Par exemple, si une entreprise décide d’envoyer un “mailing”, elle s’adressera à une entreprise spécialiste en “marketing direct” qui transmettra son message publicitaire à l’ensemble des personnes présentes dans son fichier de mailing, l’entreprise spécialisée jouant alors le rôle de l’entité de diffusion.

Nature du médium

La nature du canal de communication joue aussi un rôle important dans la communication. On peut distinguer trois sortes de mécanismes d’acheminement de message: l’acheminement direct, l’acheminement par propagation d’un signal et l’acheminement par voie d’affiche.

1. L’*acheminement direct* est le mode le plus simple: lorsqu’un agent désire envoyer un message, celui-ci est pris par le canal de communication et apporte le message directement à son destinataire (ou à tous les destinataires potentiels dans le cas d’un message diffus) sans tenir compte de la distance (sauf si elle est explicitement mentionnée dans l’en-tête d’un message diffus) et sans donner la possibilité à d’autres agents de le recevoir. L’exemple dans notre vie courante est celui du courrier postal ou électronique. C’est ce type d’acheminement qui est le plus souvent pratiqué dans les systèmes multi-agents cognitifs.

2. L'acheminement par propagation d'un signal est caractéristique du mode de communication des agents réactifs. Un agent émet un signal qui se diffuse dans l'environnement et dont l'intensité décroît en fonction de la distance. Cette décroissance varie généralement soit linéairement soit en fonction du carré de la distance. Ainsi au point x , le signal possède l'une des intensités $V(x)$ suivantes:

$$V(x) = \frac{V(x_0)}{\text{dist}(x, x_0)} \quad V(x) = \frac{V(x_0)}{\text{dist}(x, x_0)^2}$$

où x_0 correspond au point de la source du signal. De ce fait, plus un agent se trouve près de la source, plus le signal émis sera fort. Si ce signal est considéré comme un stimulus, les agents se trouvant près de la source auront plus tendance à tenir compte de ce signal que les agents situés plus loin. Ce type de communication rend

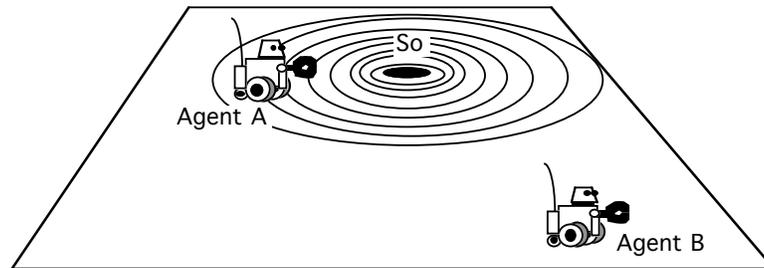


Figure 6.2: L'agent A reçoit un signal plus fort que celui de l'agent B.

difficile toute communication directe: tout agent situé près de la source peut recevoir le signal et l'interpréter comme il l'entend. Pour adresser un message à un destinataire particulier, il faut ajouter des informations supplémentaires. Par exemple dans le monde réel, la communication parlée nécessite tout un ensemble d'indices supplémentaires (posture du corps, direction du regard, appel préalable, etc.) pour que deux personnes puissent communiquer directement et faire en sorte qu'un canal de communication privilégié s'instaure entre l'émetteur et le destinataire.

La décroissance des signaux au cours du temps présente un impact majeur sur l'organisation d'une société d'agents en définissant des différences topologiques entre les agents. Les différences spatiales sont alors transformées en différences sociales par les agents. Si les agents sont sensibles à l'intensité des signaux, leur comportement sera affecté par leur position respective dans l'environnement. Par exemple, supposons que l'on dispose d'un système multi-agent réactif et qu'un stimulus S produit par une source au point x_0 peut déclencher le comportement P dans l'un des deux agents A et B . Si A est plus proche de la source que B , puisque l'intensité du stimulus décroît en fonction de la distance, le niveau du stimulus sera plus élevé pour A que pour B , ce qui aura plus tendance (toutes choses égales par ailleurs) à déclencher le comportement P chez A que chez B . En d'autres termes, on peut dire que l'appel du stimulus se fait plus pressant pour A que pour B (fig. 6.2). Si de plus il existe un mécanisme tendant à renforcer les comportements déjà déclenchés (par exemple si les agents utilisent une architecture de tâches en compétition avec renforcement, cf. chap. 3), le simple fait que A se trouve plus près d'un signal que

B produira une spécialisation des comportements des deux agents. C'est ce que l'on exprime en disant que *les différences topologiques conduisent à des différences sociales*. Cette caractéristique est très utilisée dans les systèmes multi-agents réactifs comme mécanisme à la fois de différenciation et de régulation sociale comme nous le verrons au chapitre 8.

3. Le mode d'*acheminement par voie d'affiche* est moins utilisé dans les systèmes multi-agents, sans grande raison d'ailleurs. C'est typiquement le mécanisme de communication des "petites annonces". Un agent, s'il désire communiquer, place son message dans un espace commun, appelé tableau d'affichage, tableau noir ou "ether" (Kornfeld 1979), visible par tous les agents (ou tous ceux d'une classe particulière). Ce mode de transport combine les caractéristiques des messages directs (rien n'empêche de mettre un destinataire en en-tête) et ceux des messages diffus.

Il est cependant très utilisé dans les architectures de types "tableaux noirs" (cf. chap. 3) où les sources de connaissances sont déclenchées par les événements survenant sur le tableau noir, mais il n'a pas été véritablement envisagé comme mécanisme d'acheminement de communications entre agents, sauf dans le cas du système ETHER (Kornfeld et Hewitt 1980) où les messages de demandes de résolution, d'hypothèses, de preuves et de réfutation jouent un rôle essentiel pour la résolution de problèmes distribués.

L'intention de communiquer

L'agent veut-il communiquer ou non? L'action de communication est-elle liée à une intention de l'émetteur qui désire ainsi obtenir quelque chose de son destinataire, ou bien s'agit-il d'une action incidente, d'un processus indépendant de la volonté de l'émetteur? Ces deux formes de communication qui existent dans la nature se retrouvent dans les systèmes multi-agents. Lorsque deux personnes conversent, lorsqu'un herbivore signale l'apparition d'un prédateur aux autres membres de son groupe ou lorsqu'une personne envoie une lettre à un tiers, la communication peut être qualifiée d'intentionnelle, en ce sens que l'émetteur décide résolument de communiquer quelque chose à son (ou ses) destinataire(s). Les *communications intentionnelles* procèdent d'un choix. Elles expriment une volonté, ou tout du moins une décision d'action. Au contraire, les *communications incidentes* sont effectuées sans que l'émetteur y prenne une part active. Les traces d'odeur ou les empreintes de pas, les postures du corps chez un être humain sont de véritables messages pour ceux qui veulent (ou peuvent) les interpréter, mais ils sont souvent communiqués à l'insu de celui qui les envoie. Ces messages sont des épiphénomènes de la présence ou de l'état dans lequel se trouvent les agents et peuvent être utilisés par ceux qui y sont sensibles. On ne peut pas parler d'intention dans ce cas. L'herbivore ne désire pas spécialement exprimer sa présence à un prédateur, mais son odeur diffusée dans l'atmosphère ou ses empreintes de pas sont le simple résultat de sa présence.

Les communications incidentes présentent deux caractéristiques notables: leur sémantique est liée uniquement à l'état de l'émetteur (sans que l'émetteur soit nécessairement capable de maîtriser cette communication), et leur interprétation, qui ne résulte pas d'un code fixé à l'avance est totalement variable et ne dépend que

du receveur et de la signification qu'il lui donne. En effet, bien que l'on puisse parler de langage des traces ou des postures du corps, le sens de ces signes réfère en dernière analyse à l'état de celui qui les a produits d'une part et des capacités inférentielles de celui qui les reçoit d'autre part. En ce sens, les communications incidentes sont totalement liées aux inférences interprétationnelles du receveur (Sperber et Wilson 1986). De ce fait, ces messages peuvent être employés de manières très diverses en fonction du receveur du message. Par exemple, des traces peuvent être utilisées par un animal pour retrouver ses congénères comme elles peuvent servir à le pister par un chasseur. Bien qu'il existe une sémantique (ce signe réfère à un état de l'émetteur), il n'existe pas de pragmatique intrinsèque au message (le signe peut engendrer des comportements différents chez les receveurs). Ce type de message doit donc être nettement différencié des messages intentionnels dont la pragmatique peut être analysée par la théorie des *actes de langage* comme nous le verrons plus loin.

L'intention de communiquer n'est pas une affaire de tout ou rien, mais un système gradué qui dépend des capacités cognitives de l'émetteur. Les études des communications animales notamment ont permis de dégager les rapports complexes existant entre les capacités de représentation d'un individu et les formes de communications qu'il utilise (Vauclair 1992). J. Vauclair rappelle ainsi que D. Dennett (Dennett 1983) a proposé un ensemble de niveaux d'intentionnalité hiérarchisés selon une échelle de complexité qui part de 0 pour aller jusqu'à un niveau théorique n .

L'intentionnalité d'ordre 0 décrit les situations dans lesquelles l'émetteur envoie un signal parce qu'il présente un certain état interne (par exemple, crier s'il a faim) ou parce qu'il reçoit un certain stimulus (perception d'un prédateur par exemple) même s'il n'existe aucun récepteur pour ce signal. La production de signal repose alors uniquement sur un rapport stimulus/réponse et ne dépend aucunement d'une quelconque délibération de la part de l'émetteur, ce que l'on peut exprimer en disant que:

X envoie le message M parce que X perçoit S

où S est un signal. Dans ce cas, on dit que M est un *message incident*. L'intentionnalité d'ordre 1 suppose l'existence d'une volonté directe de l'émetteur d'obtenir un effet du récepteur. Par exemple, poser une question pour obtenir effectivement une réponse ou crier pour faire quitter les lieux à un animal constituent des intentionnalités d'ordre 1. On peut exprimer la relation entre le message et l'intention de la manière suivante:

X envoie le message M à Y parce que X veut que Y fasse P

La plupart des communications multi-agents dites "intentionnelles" se situent à ce niveau. Les niveaux supérieurs s'occupent des croyances produites par les messages. En particulier l'ordre 2 suppose que l'émetteur ne veut plus directement obtenir une réponse du récepteur mais une croyance portant sur un état du monde. Sa forme logique est la suivante:

X envoie le message M à Y parce que X veut que Y croie P

Par exemple, un individu X peut crier pour que son agresseur croie qu'il va attaquer. L'ordre 3 peut être décrit par la forme:

X envoie M à Y parce que X veut que Y croie que Z croie P

où Z désigne un tiers. Les formes suivantes se ramènent naturellement à des croyances imbriquées supplémentaires.

Ces différents niveaux d'intentionnalité ne sont pas des caractéristiques externes qu'il est possible de déterminer par observation. En effet, si l'on voit un agent X envoyer un signal à Y et que l'on observe une réaction de Y , il n'est pas possible d'en inférer un quelconque niveau d'intentionnalité. Il se peut tout simplement que X possède une intentionnalité d'ordre 0 et que Y soit un agent réactif directement déclenché par le message de X , ou que X possède une intentionnalité d'ordre 2 et que Y a réagi à la croyance induite par le message de X . Il est impossible de trancher simplement entre les deux situations. Pour en avoir le cœur net, il faut soit avoir accès à l'architecture de X et Y et savoir quels sont leurs états mentaux respectifs, soit effectuer des expériences complexes comme le font les spécialistes de la cognition animale (Vauclair 1992).

On ne trouve généralement pas dans les systèmes multi-agents toute la gamme des communications possibles. Le tableau 6.1 montre les principaux types de langages en indiquant leurs caractéristiques en termes de modes de diffusion, d'acheminement et d'intentionnalité.

Types de message	Mode de communication	Acheminement	Intentionnalité
Message symbolique point à point	Point à point	Direct	Généralement intentionnel
Message symbolique diffusé	Diffusion générale	Direct	Généralement intentionnel
Annonce	Point à point/ diffusion générale	Tableau d'affichage	Généralement intentionnel
Signal	Diffusion	Propagation	Incident

Table 6.1: Principaux modes de communication dans les systèmes multi-agents

Il est bien entendu possible d'imaginer d'autres types de messages, tels que des messages point à point acheminés par propagation, mais ils présentent a priori moins d'intérêt et ne sont généralement pas implémentés.

6.1.4 A quoi sert la communication?

Nous avons classé les communications par rapport aux éléments de communication et non par rapport à ce qui fait l'essence de la communication, c'est-à-dire la fonction qu'elle assure. Selon Roman Jakobson (Jakobson 1963), qui adopte sur le langage un point de vue fonctionnaliste, il est possible de classer les modes de communication selon leurs fonctions. Il en distingue six: les fonctions expressives,

conatives, dénotatives, phatiques, poétiques et métalinguistiques. Cette dernière sera décomposée par nos soins en fonction paralinguistique et métaconceptuelle.

- La *fonction expressive* caractérise l’attitude de l’émetteur du message: elle indique l’état dans lequel se trouve cet agent et quelles sont ses intentions, ses buts et comment il juge et voit les choses. Cette fonction permet en particulier de synchroniser des agents et donc de coordonner leur tâches, mais aussi de mettre en commun leurs croyances. Cette fonction peut se résumer par la phrase: “Voilà mon état, voilà ce que je pense, voilà mes croyances.”
- La *fonction conative* (du latin *conari*, entreprendre, essayer) renvoie au destinataire du message et correspond à la notion d’ordre ou de requête qu’un émetteur adresse à un destinataire. Il faut remarquer que la plupart des messages dans les systèmes multi-agents les plus frustrés assurent cette fonction. Elle est résumée par l’une des phrases: “Fais ceci” ou “Répond à ma question”. On lui associe aussi les réponses d’acceptation et de refus. Il s’agit certainement de la fonction la plus manifeste et la plus importante dans les systèmes multi-agents. Nous l’étudierons plus particulièrement dans la section réservée à la théorie des actes de langage.
- La *fonction référentielle* est centrée sur le contexte. Cette fonction est garante de la transmission d’informations portant sur des faits du monde. Un message qui assure cette fonction porte alors sur un état du monde ou d’une tierce personne. “Voilà ce qui est” est une phrase qui caractérise cette fonction.
- La *fonction phatique* sert essentiellement à établir, prolonger ou interrompre une communication. Elle est utilisée notamment pour vérifier que le canal fonctionne. Les messages d’acquiescement chers aux protocoles des systèmes distribués relèvent essentiellement de cette fonction. Elle peut se résumer par l’expression: “Je veux communiquer avec toi, je reçois bien tes messages.”
- La *fonction poétique* ou esthétique s’attache à la mise en valeur du message lui-même. Dans les systèmes multi-agents où l’aspect esthétique n’a pas encore été traité, la fonction poétique n’est pas (ou peu) mise en évidence.
- La *fonction métalinguistique* porte sur tout ce qui se rapporte aux messages, à la langue et à la situation de communication. Cette fonction est très importante dans les systèmes multi-agents, car elle permet à des messages de parler d’autres messages, de la situation de communication et aussi de mettre à l’unisson les concepts, le vocabulaire et la syntaxe employés par les interlocuteurs. On distinguera la *fonction paralinguistique* qui se rapporte aux messages qui vont ou viennent d’être émis de la *fonction métaconceptuelle* qui s’attache à la définition d’un moule linguistique et conceptuel commun. La première porte sur les communications antérieures et postérieures en s’attachant aux objectifs de la communication et à l’importance des messages. Des phrases du type “Pourquoi me demandes-tu cela?”, “Je ne comprend pas, peux-tu préciser?”, “Ceci est très important” sont caractéristiques de cette fonction.

La seconde porte sur le code du message. Les interlocuteurs s'assurent de la bonne compréhension de ce qui vient d'être exprimé et de la définition des concepts employés. Cette dernière est souvent la plus complexe à mettre en œuvre dans les systèmes multi-agents car elle porte en particulier sur l'élaboration de concepts communs, la définition de notions communes. On peut résumer cette fonction par la phrase "par X je veux dire Y".

Un message assure plusieurs fonctions en même temps. En particulier tout message possède à la fois une fonction expressive et phatique. Par exemple, un message de la forme:

(SM1) B: A << Quels sont les X tels que P(x)

qui assure une fonction bien évidemment conative, comporte aussi une fonction expressive en indiquant que l'état de l'émetteur B se trouve dans un état où il cherche à savoir quelque chose. De plus, comme à tout message, on peut lui associer une fonction phatique: en effet, par sa seule présence, SM1 indique que B veut continuer le dialogue. On appellera *fonction principale* la fonction associée directement au type de message utilisé et *fonctions secondaires* les autres fonctions véhiculées par ce message. Dans l'exemple précédent, la fonction principale de M1 est conative, les fonctions expressives et phatiques étant secondaires. En revanche, un message de la forme:

(SM2) B: A << Je crois que P(c)

assure une fonction essentiellement expressive. Outre la fonction phatique portée par chaque message, M2 entretient une fonction référentielle, puisqu'on y parle de quelque chose (l'objet C) auquel on attribue la propriété P. Par exemple un message de la forme:

(M2.1) B: A << Je crois que C est en position <x1,y1>

est un exemple possible du schéma de message SM2. Il exprime non seulement la croyance que B place dans le positionnement de C en <x1,y1>, mais aussi le fait que C se trouve peut être *effectivement* dans cette position si B ne s'est pas mépris. Il existe en plus une fonction conative secondaire, si l'on interprète ce message comme une tentative de la part de B de convaincre A de croire à la véracité de P(C), c'est-à-dire s'il est interprété par B comme un message de la forme:

(M2.2) B: A << Je veux te convaincre que
C est bien en position <x1,y1>

La communication s'exprime aussi comme un *contrat cognitif* (Ghiglione 1986) dans lequel l'émetteur suppose connu certaines informations et donc en avance de nouvelles sans rappeler les anciennes. Grice a donné quelques conseils, sous la forme de maximes (Grice 1975), sur la manière d'intervenir dans une conversation de manière la plus concise et pertinente possible. Par exemple, la maxime de quantité précise qu'un message doit être aussi informatif que possible, mais pas plus

que nécessaire. La maxime de qualité prescrit de ne donner que des informations vraies, ou tout du moins que l'émetteur croit vraies. Ces maximes peuvent être utilisées dans le cadre de systèmes multi-agents (Gleizes et al. 1994), non pas comme Grice l'entendait, pour essayer d'avoir la meilleure communication possible, mais *a contrario* comme point de départ pour concevoir un système communicationnel. Ainsi, on supposera que tous les agents ne communiquent que des informations qu'ils croient vraies (postulat d'honnêteté) et répondent systématiquement aux demandes qui leur sont adressées (postulat de politesse). La notion de contrat cognitif est aussi à rapprocher de celle d'engagement que nous avons vu au chapitre 5 et qui porte sur le fait qu'il existe nécessairement un substrat commun pour que deux agents puissent communiquer. Ce substrat n'est pas forcément fixe et déterminé une fois pour toutes. Bien au contraire les recherches actuelles en psychologie sociale de la communication mettent l'accent sur la faculté qu'a la communication de produire et d'entretenir des univers cognitifs communs, c'est-à-dire une "co-construction de la référence" (Ghiglione 1989). Malheureusement, cette conception de la communication comme source de construction du modèle cognitif des interlocuteurs et du langage lui-même semble parfois un peu trop en avance pour les possibilités techniques dont nous disposons à l'heure actuelle dans la réalisation de systèmes multi-agents, car elle suppose une certaine plasticité du langage, la possibilité de jouer sur la polysémie des termes et sur les connotations engendrées par les mots et les énoncés, et pose le problème central du symbole et de sa signification, un problème crucial que les systèmes multi-agents ne peuvent prétendre résoudre seuls.

6.2 Les actes de langage

Il est des théories qui semblent à la fois être une évidence et en même temps constituer une rupture majeure avec les conceptions précédentes. Ceci est particulièrement vrai pour ce que l'on appelle maintenant la pragmatique du discours, ou plus simplement les actes de langage. Il s'agit d'une théorie majeure de la philosophie du langage qui présente un grand intérêt pour l'analyse des communications symboliques point à point dans les systèmes multi-agents.

6.2.1 Dire, c'est faire

La parution, en 1962, du livre d'Austin *How To Do Things With Words* (Quand dire, c'est faire) fit un peu l'effet d'une bombe. Pour tous les linguistes, descendants de Ferdinand de Saussure, et pour tous les philosophes de la langue issus du positivisme logique, étudier le langage, c'était essayer de comprendre le sens des phrases en indiquant comment à partir d'une combinaison de mots il est possible de donner un énoncé signifiant. Pour les théoriciens de la sémantique du langage, c'est-à-dire de l'étude du sens des énoncés, l'important était de comprendre comment une phrase pouvait rendre compte de manière satisfaisante de la réalité. Pour cela, ils s'intéressaient essentiellement à savoir quel était le degré de vérité (ou de fausseté) d'une phrase *constative*, c'est-à-dire qui affirme quelque chose du monde. De ce fait,

les théoriciens ne s'intéressaient qu'à la fonction dénotative du langage, c'est-à-dire au rôle de transfert d'informations, négligeant par là même les autres aspects de la communication et en particulier la fonction conative qui traite de l'action sur le destinataire du message. Dans ce cadre, nul ne se posait véritablement la question de savoir à quoi sert le langage, ou plus exactement, comme l'indique le titre anglais du livre d'Austin, de savoir si l'on pouvait faire des choses avec des mots. Pour Austin et ses successeurs, dont le plus important est sans conteste John Searle, l'énonciation, c'est-à-dire le fait de produire un énoncé, est un acte qui sert avant tout à produire des effets sur son destinataire. Lorsqu'on dit "passe moi le sel" à table, on ne s'intéresse pas directement à la vérité ou à la fausseté de la phrase, mais surtout à l'effet que cette demande peut produire sur son interlocuteur et donc à ce que ce dernier passe effectivement le sel. Plus exactement cette phrase n'est ni vraie ni fausse, mais elle renvoie à une action qui peut réussir ou échouer. De ce fait le problème de la vérité perd sa place centrale et laisse le champ libre aux critères de succès de la communication, c'est-à-dire à la pragmatique des communications.

Mais qu'est-ce qu'un acte de langage? Les actes de langage désignent l'ensemble des actions intentionnelles (au sens opératoire de "perform") effectuées au cours d'une communication. Il existe plusieurs types d'actes de langage. D'après Searle (Searle 1979) et Vanderveken (Vanderveken 1988) on peut distinguer les principaux types d'actes suivants:

1. Les *assertifs* servent à donner une information sur le monde en affirmant quelque chose (exemple: il fait beau, Jean a 21 ans, les rectangles ont quatre angles droits).
2. Les *directifs* sont utilisés pour donner des directives au destinataire (exemple: donne moi ta montre, lave-toi les mains, quelle est la valeur de la troisième décimale de π).
3. Les *promissifs* engagent le locuteur à accomplir certains actes dans l'avenir (exemple: je viendrai à la réunion de 5 heures, je te promets de t'envoyer des cartes postales).
4. Les *expressifs* servent à donner au destinataire des indications concernant l'état mental du locuteur (exemple: je suis heureux, je m'excuse pour hier, je te remercie).
5. Les *déclaratifs* accomplissent un acte par le fait même de prononcer l'énoncé (exemple: je déclare la séance ouverte, je vous condamne à 2 années de prison, je te maudis, je vous engage).

Cette classification, malgré son intérêt pratique, n'est pas toujours considérée comme totalement définitive. Elle pose encore de nombreux problèmes qui font l'objet de recherches dans le domaine de la pragmatique linguistique (Récanati 1981; Moeschler 1989). En particulier comment classer les expressions conditionnelles (exemple: si tu n'es pas sage, tu n'auras pas de dessert) ou les injures. De plus, elle est trop générale et ne peut être utilisée telle quelle pour réaliser des systèmes multi-agents. C'est

pourquoi nous serons amené à proposer une classification plus fine que la précédente. Cette opération sera effectuée en deux temps. D’une part, nous décomposerons les directifs en *interrogatifs* qui servent à poser une question pour obtenir des informations et en *exercitifs* qui demandent à un agent d’accomplir une action dans le monde. D’autre part, nous affinerons notre analyse en définissant des “sous-actes” de langage en détaillant l’objet (ou l’action) sur lequel porte l’acte et, surtout, comme nous le verrons par la suite, en caractérisant la structure conversationnelle dans laquelle il s’insère.

6.2.2 Actes locutoires, illocutoires et perlocutoires

Depuis Austin (Austin 1962) et surtout Searle (Searle 1969), les actes de langage sont définis comme des structures complexes formées de trois composantes, considérées comme des actes élémentaires:

- La composante *locutoire* concerne la génération matérielle des énoncés par émission d’ondes sonores ou par écriture de caractères, c’est-à-dire le mode de production de phrases à l’aide d’une grammaire et d’un lexique donnés.
- La composante *illocutoire* se rapporte à la réalisation de l’acte effectué par le locuteur sur le destinataire de l’énoncé. Les actes illocutoires, les plus étudiés en pragmatique du langage, sont caractérisés par une force illocutoire (exemple: affirmer, questionner, demander de faire, promettre, ordonner, prévenir, etc.) et par un contenu propositionnel qui est l’objet de la force illocutoire. Il est ainsi possible de représenter ces actes sous la forme $F(P)$ où F est la force illocutoire et P le contenu propositionnel. Par exemple, affirmer qu’*il pleut* se mettra sous la forme:

Affirmer(il pleut)

alors que l’interrogation correspondante, “Est-ce qu’il pleut?”, sera représentée ainsi:

Questionner(il pleut)

cette expression indiquant qu’il s’agit d’une question. Souvent les actes illocutoires sont indiqués par la présence d’un verbe, appelé *performatif*, qui marque le type d’acte. Par exemple, la phrase “Je te demande de venir demain matin” marque un acte illocutoire directif introduit par le performatif **demander**. La notion de performatif est utile pour les systèmes multi-agents, puisque ce sont les performatifs qui indiqueront les différents types d’actes de langage que peuvent émettre et interpréter les agents. De ce fait, les énoncés des messages sont exprimés sous la forme

<performatif>(<contenu>)

où <performatif> est un mot clé qui désigne le type d'acte illocutoire associé au message, et <contenu> est la forme propositionnelle.

- La composante *perlocutoire* enfin porte sur les effets que les actes illocutoires peuvent avoir sur l'état du destinataire, ses actions, ses croyances et ses jugements. Par exemple, convaincre, inspirer, effrayer, persuader, etc. sont des actes perlocutoires. Ceux-ci ne sont pas introduits par des performatifs mais doivent être compris comme des conséquences des actes illocutoires. Par exemple, la phrase “Je t'affirme que la Terre tourne autour du Soleil” est un acte illocutoire d'affirmation, qui peut être pris, dans un certain contexte d'énonciation, comme une tentative de convaincre le destinataire que la Terre tourne et donc de modifier l'ensemble de ses croyances. De même lorsqu'on demande à quelqu'un de faire quelque chose, le fait qu'il ait compris qu'il s'agisse d'un exercitif montre qu'il a reconnu la composante illocutoire de l'acte, et s'il accomplit effectivement ce qui lui a été demandé, alors on peut dire que la composante perlocutoire a été satisfaite.

Tous ces actes se trouvent présents dans la même expression à différents niveaux. C'est pourquoi on peut aussi parler des aspects locutoires, illocutoires et perlocutoires des expressions. Lorsque nous parlerons d'actes de langage, nous nous référerons implicitement à la totalité de ces trois aspects et pas seulement aux actes illocutoires comme le font Searle et un certain nombre d'auteurs, même s'il s'agit d'un des aspects les plus importants.

6.2.3 Succès et satisfaction

Comme nous l'avons vu, un énoncé n'est pas vrai ou faux, il réussit ou échoue. Par exemple, si l'agent A demande à B de résoudre une équation différentielle, et que B ne sache pas le faire, la demande échouera. Il en est de même si B n'a pas compris la demande. Dans les deux cas, l'acte de langage n'a pas réussi. Ainsi, un acte de langage peut manquer son but de plusieurs façons. Il peut le faire:

1. *Dans l'énonciation de l'acte*: Que la transmission ne s'effectue pas bien, que le locuteur balbutie, qu'il y ait du bruit sur la ligne ou que le destinataire ne comprenne pas la langue utilisée par l'émetteur et l'acte échouera soit parce qu'il ne sera pas compris, soit parce qu'il entraînera une méprise de la part du destinataire.
2. *Dans l'interprétation de l'acte*: L'émetteur s'exprime correctement et le message est arrivé à bon port, mais le destinataire n'interprète pas correctement la force illocutoire de l'émetteur. Par exemple, si à la question:

(M4) A: B << Questionner(il pleut)>>

le destinataire B comprend en fait

(M4') A: B << Affirmer(il pleut)>>

il confondra une question et une affirmation, ce qui peut l'amener à produire une réponse que A considérera peut être comme bizarre:

(M5) B: A << Affirmer(je vais donc prendre
mon parapluie)}

3. *Dans la réalisation effective de l'acte entraîné par l'énonciation*: Les causes d'échec sont multiples. Il suffit par exemple que l'un des interlocuteurs n'ait pas la compétence pour réaliser cet acte. Si A demande à B de résoudre un problème d'équations différentielles et que B ne sache pas le faire, cela constituera un échec. D'une manière générale un refus du destinataire conduit à faire échouer tous les actes directifs, c'est-à-dire qui portent sur une question ou sur une demande d'accomplir une action. De même, les promesses sont caduques si l'émetteur n'est pas capable de remplir ses promesses. La promesse "Je te rendrai ton argent demain" risque d'échouer si l'émetteur est totalement sans le sou et qu'il n'a aucun moyen de rendre cet argent. De même un prisonnier qui promet à une personne hors de prison qu'il viendra la voir demain aura toutes les chances de faire échouer sa promesse s'il n'est pas libéré entre-temps.

Vanderveken (Vanderveken 1988) propose une autre classification concernant l'aspect pragmatique des actes de langage en différenciant le *succès* de la *satisfaction*. Les conditions de succès sont celles qui doivent être remplies dans un contexte d'énonciation pour que l'émetteur réussisse à accomplir cet acte. Une promesse a comme condition de succès que le locuteur s'engage effectivement à accomplir l'acte correspondant à la promesse. De même une déclaration, telle que "je déclare la séance ouverte" a comme condition de succès que le locuteur ait bien l'autorité lui permettant cette déclaration. Il y a donc succès si le locuteur accomplit l'acte illocutoire impliqué par l'énoncé. Par exemple, comme le signale C. Brassac (Brassac 1992), le message

(M5) A: B << DemanderFaire(P)}

est accompli avec succès et sans défaut si:

- a) le locuteur tente de faire en sorte que son interlocuteur ajuste le monde aux mots (c'est-à-dire que le monde soit dans l'état que décrit P),
- b) avec une certaine position d'autorité,
- c) en lui laissant la possibilité de refuser,
- d) l'interlocuteur est dans la capacité de le faire et
- e) le locuteur désire que ce dernier le fasse (le désir est le mode psychologique relatif aux conditions de sincérité des directifs).

En revanche, les conditions de satisfaction sont relatives à la composante perlocutoire et prennent en compte l'état du monde résultant de cet acte. De ce fait, l'acte de langage associé au message M5 est satisfait si B accomplit P. De même une question est

satisfaite si le destinataire répond à la question, une affirmation si elle est vraie, une promesse si elle est tenue. La condition de satisfaction est donc plus forte, puisqu'elle prend en compte le résultat de l'accomplissement de P. Ainsi la satisfaction entraîne le succès, mais non l'inverse.

6.2.4 Composantes des actes illocutoires

Afin de mieux comprendre la structure des actes illocutoires, Searle (Searle 1969) décrit les différents types d'actes de langage par l'ensemble des conditions qu'il estime nécessaires et suffisantes pour leur accomplissement:

1. Conditions de *départ et d'arrivée* ou d'entrée/sortie: elles portent sur le fait que le message peut arriver de l'émetteur, qui n'est pas muet, au destinataire, qui n'est pas sourd, et donc qu'il existe un canal de communication en état de fonctionnement entre les deux interlocuteurs. En d'autres termes, la fonction phatique de la communication est assurée.
2. Conditions sur le *contenu propositionnel*: les actes de langage supposent généralement une structure particulière de la syntaxe des contenus propositionnels qui leur sont associés. Ces conditions portent donc sur les restrictions grammaticales et conceptuelles concernant le contenu de ces propositions.
3. Conditions *préparatoires*: elles portent sur ce qui doit être vrai dans le monde pour qu'un locuteur puisse accomplir un acte de langage. Dans le cas des *exercitifs*, c'est-à-dire dans le cas d'un énoncé tel que M5, ces conditions sont les suivantes:
 - B est capable de faire P
 - A croit que B est capable de faire P
 - il n'est pas certain, ni à A ni à B que B fera P

On peut remarquer que ces conditions préparatoires ressemblent fort à la définition logique de l'intention du chapitre 5. En effet, communiquer c'est agir, et il n'est pas curieux que les conditions de réalisation d'une communication intentionnelle soient très voisines des conditions d'exécution d'une action intentionnelle. La différence essentielle vient de la présence ici de deux agents (pour communiquer, il faut être au moins deux), alors que dans l'action intentionnelle on ne mettait en œuvre qu'un seul agent (on peut agir seul).

4. Conditions de *sincérité*: un acte ne peut réussir que si le locuteur est sincère, c'est-à-dire s'il veut réaliser ce qu'il prétend faire en énonçant sa phrase. Dans le cas d'une demande d'action, cela signifie que A veut réellement que B accomplisse l'action P. De la même manière, si A fait une promesse à B de faire P, cela signifie qu'il a vraiment la volonté d'effectuer P dans le futur. Enfin, s'il s'agit d'une affirmation, A est supposé croire en ce qu'il affirme. Ces conditions, qui ne sont pas toujours réalisées lors d'agents naturels, sont évidentes pour des agents artificiels.

5. Condition *essentielle*: il s'agit en fait de ce que veut réellement faire le locuteur en effectuant un acte de langage. Si le locuteur pose une question, c'est parce qu'il désire obtenir une information. S'il demande de faire quelque chose, c'est parce qu'il veut que l'action soit accomplie, etc. A cette condition essentielle, nous préférons la notion de but illocutoire de Vanderveken, qui correspond à l'intention principale amenant la réalisation de l'acte. Ces buts, d'après lui, sont au nombre de 5 et correspondent aux différents types d'actes de langage: buts *assertif*, *promissif*, *directif*, *déclaratif* et *expressif*.
6. Le *degré de puissance* de Vanderveken (Vanderveken 1988) correspond à l'intensité avec laquelle l'acte est accompli. Par exemple, s'il s'agit d'un acte exercitif, la supplication est plus forte que la simple demande car elle suppose un désir d'accomplissement plus fort. De même, la conjecture est plus faible que l'assertion qui est elle-même plus faible qu'une affirmation solennelle. Cependant, nous préférons par la suite classer les actes de langage relatifs à tous ces degrés de puissance en une taxonomie portant sur leur rôle dans des schémas de communication, plutôt que de les caractériser par une simple valeur numérique, ce qui généralement est la marque d'une analyse superficielle.

Cohen et Perrault (Cohen et Perrault 1979) ont remarqué que toutes ces conditions ne se situaient pas du tout sur le même plan. Alors que certaines se rapportent à l'aspect locutoire du message, d'autres se rapportent à l'intentionnalité du locuteur, c'est-à-dire à ce qu'il désire accomplir lorsqu'il émet un message. Par exemple, le but illocutoire est directement relatif à l'intention de l'émetteur et les conditions préparatoires aux connaissances qu'un agent possède sur un autre agent.

6.3 Les conversations

La théorie des actes de langage telle qu'elle a été développée par Austin, Searle, Vanderveken, Récanati et d'autres ne prend en compte que l'acte isolé, l'énonciation initiale avec ses conditions d'application et les effets locaux qu'elle peut avoir sur les interlocuteurs. Elle ne traite absolument pas de la séquence d'interactions qui s'établit entre les interlocuteurs lors de leurs communications ni de leurs attentes réciproques. Par exemple, un agent qui pose une question s'attend à une réponse ou à un refus, une promesse conduit par la suite à la réalisation de la promesse ou au dédit de cette promesse par le locuteur, de même qu'une affirmation entraîne une acceptation du genre "Je le savais déjà" ou à une dénégalation "Cela n'est pas possible". De ce fait, la notion d'interaction dialogique est parfois oubliée par les actes de langage. Néanmoins, une école de pensée, essentiellement européenne, autour de Moeschler (Moeschler 1985) à Genève et de Trognon et Brassac à Nancy (Trognon et Brassac 1988), a tenté de fournir une théorie des enchaînements conversationnels à partir d'une extension de la logique illocutoire formalisée par Searle et Vanderveken (Searle et Vanderveken 1985; Vanderveken 1992). Dans ce cadre, les actes de langage sont les unités élémentaires permettant d'analyser les conversations. En particulier,

Brassac et Trognon ont montré qu'il était possible d'assigner la force illocutoire d'un énoncé à partir des énonciations futures et que l'attribution d'un acte de langage résulte d'un processus de co-interprétation des énoncés (Trognon et Brassac 1992). Cette approche semble particulièrement féconde dans le cadre des conversations humaines car elle ne suppose pas qu'il existe a priori une et une seule interprétation possible d'un énoncé. Au contraire, une phrase ne prend son sens que par l'ensemble des réactions qu'elle suscite chez les interlocuteurs, la signification pragmatique, et donc la force illocutoire, étant reconstruite a posteriori. Néanmoins, cette théorie est encore en avance par rapport aux possibilités des systèmes multi-agents. Nous proposerons donc une autre extension de la théorie des actes de langage qui s'appuie sur la définition de *protocoles de communication*, en considérant que tout acte de langage suppose un certain enchaînement possible des énonciations et qu'il engendre certaines modifications dans l'état mental des interlocuteurs. En effet, un acte de langage n'est pratiquement jamais accompli isolément, et il se trouve souvent à l'origine d'autres actes. Par exemple, une promesse telle que "Je te promets de venir demain" constitue un engagement du locuteur pour effectuer un acte particulier, celui de venir, à un moment précis, le lendemain. De même, la demande

(M5) A: B << DemanderFaire(P)

est un acte de demande d'action et se trouve à l'origine de toute une série d'actions ultérieures: l'acceptation ou le refus de B d'effectuer P, éventuellement la réalisation de la tâche P et sa signalisation à A. Ces conséquences sont importantes car elles entraînent des attentes de la part du locuteur. Ce dernier, en fonction des messages de B, pourra prendre des dispositions concernant ses propres actes à accomplir et sera ainsi à même d'anticiper sur le futur. Par exemple, si l'agent B accepte de faire P, A peut supposer que l'action P sera terminée à la date $D+T$ où D est la date de début de l'action P et T est le temps normal de réalisation de P. Par exemple, si A est le propriétaire d'une maison, B un peintre et P l'action de peindre le salon de la maison de A, le message M3, s'il est accepté par B peut amener A à penser que son salon sera repeint après un certain temps. La définition de ce temps peut lui même faire l'objet de transactions ou bien se référer au temps normal d'exécution, c'est-à-dire faire référence à une conception standard, partagée par les interlocuteurs, des caractéristiques normales de la réalisation de la tâche en question. En effet, un agent A qui pose une question à un agent B, c'est-à-dire qui émet un message de la forme:

(M6) A: B << Questionner(quel est l'ensemble $\{x \mid P(x)\}$)

attend l'une des trois réactions suivantes:

1. La réponse à la question:

(M7) B: A << Réponse(M6, a_1, \dots, a_n) où les a_1, \dots, a_n sont les réponses à la question précédente, c'est-à-dire les valeurs de x qui satisfont $P(x)$;

2. Un refus de répondre à la question, éventuellement adjoint d'une explication:

(M7') B: A << RefuserIncompétentDemande(M6)

3. Une demande de renseignements complémentaires (en passant au niveau méta, puisqu'il est nécessaire de parler de la communication):

(M7") B: A << MétaQuestionner(M6, Arguments(P))

La modélisation de ces conversations passe notamment par la définition de *protocoles*, c'est-à-dire de séquences valides de messages. Il existe plusieurs manières de décrire des protocoles, mais les plus courantes utilisent des automates à états finis ou des réseaux de Petri.

6.3.1 Conversations et automates à états finis

Une conversation peut se décrire comme une suite d'états liés par des transitions, les communications que s'échangent les agents et donc se modéliser sous la forme d'un automate à états finis (Winograd et Flores 1986). La figure 6.3 montre l'automate correspondant à une conversation initialisée par un exercitif.

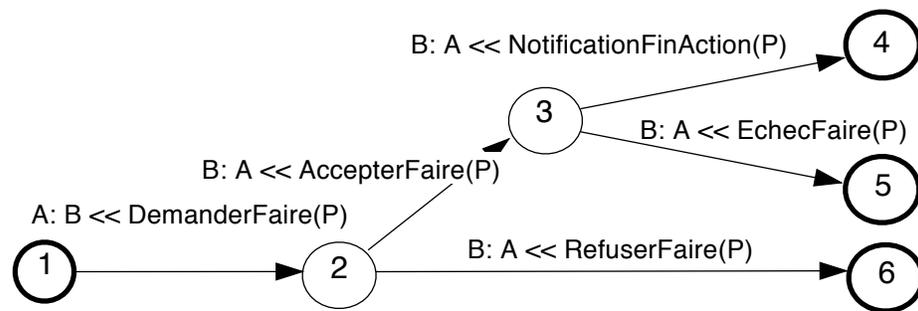


Figure 6.3: Automate de description d'une conversation commençant par un exercitif

Initialement la conversation se trouve dans l'état 1. Puis A, en demandant à B d'accomplir une action P, entame la conversation, qui passe alors dans l'état 2. Plusieurs possibilités s'ouvrent alors: B peut accepter, ce qui l'engage auprès de A ou B peut refuser (par exemple en se considérant incompetent pour accomplir cette action). Il est aussi possible pour B de passer au niveau méta en demandant des précisions supplémentaires, mais ceci n'est pas pris en compte dans le diagramme. En fonction des réponses de B, la conversation passera dans l'état 3 ou 6, déterminant ainsi les suites de communications possibles. La conversation est considérée comme terminée lorsque l'automate atteint l'un de des états terminaux 4, 5 et 6.

A partir des remarques de Winograd et Flores (Winograd et Flores 1986) sur la nature des conversations, il est possible d'indiquer les propriétés principales que doivent remplir ces structures communicationnelles:

1. Les conversations débutent par un acte de langage majeur (assertif, promissif, exercitif, interrogatif, déclaratif et expressif) et s'expriment comme le résultat d'une intention d'un agent envers un autre agent.
2. A chaque étape de la conversation, il existe un ensemble réduit d'actions possibles. Ces actions sont des actes de langage majeurs ou mineurs (exemple: refus, acceptation, etc.).

3. Il existe des états terminaux qui déterminent la fin d'une conversation. Lorsque l'un de ces états est atteint, la conversation est considérée comme achevée.
4. La réalisation d'un acte de langage modifie non seulement l'état du discours, mais aussi l'état des agents (croyances, engagements, intentions) impliqués dans la conversation. Par exemple, lorsqu'un agent B indique qu'il ne peut pas accomplir une action P parce qu'il lui manque la compétence pour la faire, l'agent A peut mettre à jour ses accointances en modifiant la représentation qu'il a de B. De même lorsque B s'engage auprès de A pour accomplir P, ce dernier peut ajouter cette promesse dans sa liste des engagements.

6.3.2 Conversations et réseaux de Petri

Les automates à états finis sont très pratiques pour préciser la structure des conversations lorsqu'elles apparaissent de manière isolée, c'est-à-dire lorsqu'on peut ramener une conversation à un seul processus. Cependant, les agents sont parfois engagés dans plusieurs conversations à la fois et il leur faut gérer ces conversations multiples. Il est alors plus facile de décrire la nature de ces interactions en utilisant les réseaux de Petri introduits au chapitre 4. Ces derniers sont d'ailleurs très utilisés pour modéliser des protocoles dans des systèmes distribués (par exemple (Estraillier et Girault 1992)) et il n'est pas étonnant qu'on puisse les utiliser pour décrire des conversations entre agents. Cette approche, qui n'a été pour l'instant que très peu développée dans le cadre des systèmes multi-agents (sauf par (Coria 1993) et (Ferber et Magnin 1994)), semble pourtant promise à un bel avenir.

La figure 6.4 montre un réseau de Petri correspondant au traitement d'un exercitif identique à celui de la section précédente. Chaque agent est décrit par un sous-réseau de Petri, dont les places correspondent aux états internes de l'agent (ou de la tâche en cours, comme nous le verrons par la suite). Les transitions correspondent soit à des synchronisations dues à la réception de messages, soit à des conditions d'applications des actions, telles que les conditions essentielles ou préparatoires des actes de langage associés. Les messages en cours d'acheminement sont représentés par des places supplémentaires qui raccordent les agents A et B de manière à ne plus former qu'un seul réseau.

Les places DA et DB décrivent les états initiaux dans lesquels se trouvent les agents avant le début de la conversation et les places FA1, FA2 et FB représentent les états de fin de conversation. A partir de l'état DA, l'agent A envoie une requête à B et passe dans l'état AR1 qui représente une attente de réponse. Si B ne sait pas faire P, il renvoie un refus à A, qui passe alors dans l'état FA1, ce qui indique que A doit chercher ailleurs la réalisation de P. Si B peut faire P, il envoie un message d'acceptation à A, qui se place de nouveau en attente de réponse. Pendant ce temps, B se trouve dans l'état BR en essayant d'accomplir l'action P. S'il peut la faire, alors il envoie une notification de fin d'accomplissement, ce qui place l'agent A en état FA2, sinon il indique qu'il ne lui est pas possible de faire P, mais dans les deux cas il passe dans l'état terminal FB.

L'intérêt de cette approche est de pouvoir non seulement expliquer les échanges

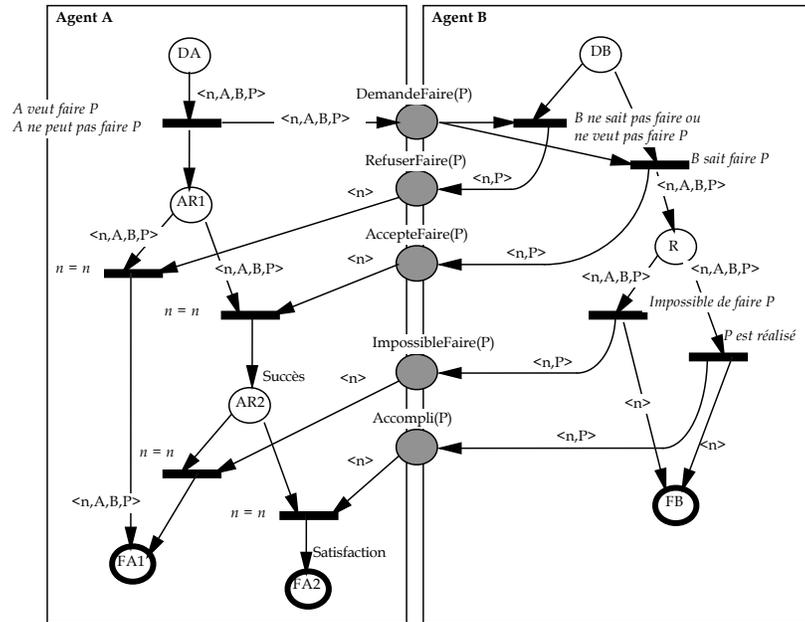


Figure 6.4: Modélisation d'une conversation par réseau de Petri

entre les agents, mais aussi de préciser les états internes dans lesquels ils se trouvent, et donc de décrire plus finement ce qui se passe au cours d'une conversation. On notera d'ailleurs que toutes les caractéristiques liées aux actes de langage se retrouvent ici. Par exemple les états de succès et de satisfaction de la demande initiale sont figurés respectivement aux places AR2 et FA2.

Les réseaux de Petri permettent même de modéliser plusieurs communications simultanées avec plusieurs correspondants en utilisant les réseaux de Petri colorés. Dans ce cas, ce ne sont plus des jetons qui circulent sur les places et franchissent les transitions, mais des structures qui représentent les conversations en cours. Pour que les agents puissent s'y retrouver et ne confondent pas les messages de plusieurs conversations différentes, on numérote chaque message par un numéro d'ordre correspondant à la conversation à laquelle il appartient. C'est ce que nous verrons à la section suivante, avec une modélisation des actes de langage à l'aide de BRIC.

6.3.3 Une classification des actes de langage pour des structures conversationnelles multi-agents

Il est bien difficile de donner une taxonomie générale de tous les actes de langage pouvant être utilisés par des agents intentionnels, et cela pour plusieurs raisons. En premier lieu, l'étendue des performatifs nécessaires à la réalisation d'un système multi-agent donné dépend bien évidemment des capacités des agents eux-mêmes. Si ces derniers ne sont pas capables de s'engager par exemple, les promissifs ne serviront à rien. En deuxième lieu, il n'est pas toujours facile de savoir si un acte particulier doit faire l'objet d'un nouveau performatif ou s'il faut au contraire utiliser

des modalités dans le contenu propositionnel. Par exemple, pour qu'un agent puisse demander à un autre agent de le prévenir chaque fois qu'il recevra une information portant sur un sujet donné, faut-il définir un nouveau performatif et envoyer des messages de la forme

(M20) A: B << DemanderPrevenirNouvellesInformationsSur(S)

ou bien au contraire utiliser un acte exercitif plus général et décrire l'action consistant à prévenir dans le contenu propositionnel, quitte à ce que ce langage puisse contenir lui-même des mentions aux performatifs, en particulier pour décrire des actes conditionnels?

(M21) A: B << DemanderFaire(si(info(x) & type(x,S)
alors Informer(A,x))

Il est évident qu'un choix dans un sens ou un autre aura un impact important sur l'ensemble des performatifs, mais aussi sur le langage de communication utilisé dans la forme propositionnelle, et que ces deux "langages" sont complémentaires l'un de l'autre: plus l'ensemble des performatifs est important, plus le langage de contenu propositionnel peut être simplifié. Au contraire, un ensemble de performatifs réduit renverra toute la difficulté sur le langage de description.

Malgré toutes ces difficultés, nous présentons ici une classification d'un ensemble d'actes de langage destinés à être utilisés dans un système multi-agent. On définit ces types par genre et espèce, c'est-à-dire en les classant hiérarchiquement. Au niveau le plus général se trouvent les genres, c'est-à-dire les actes principaux définis dans leurs caractères génériques. Au niveau inférieur se situent les sous-types correspondant à des espèces, c'est-à-dire à des spécialisations de ces actes principaux. Le tableau 6.2 répertorie quelques actes de langage élémentaires, avec leur catégorie pragmatique et des exemples de performatifs caractéristiques. Les sous-types sont

Type de performatif	Exemple de performatif	Catégorie
Demande	DemanderFaire	Exercitif
Interrogation	Questionner	Interrogatif
Affirmation	Affirmer	Assertif
Offre de service	OffrirService	Promissif
Indication de compétences	SaitFaire	Expressif
Proposition d'hypothèse	ProposerHypothèse	Expressif

Table 6.2: Quelques actes de langage élémentaires

construits par différenciation de ce type principal en associant l'objet (ou l'action) sur lequel porte l'acte de langage. Les performatifs des sous-types sont obtenus par concaténation du performatif du type et des objets ou des actions qui précisent le sens de l'acte. Par exemple, les performatifs des sous-types de *Demande* seront formés par concaténation de *Demander* et des arguments supplémentaires, tels que

DemanderFaire pour qu'une action soit accomplie, ou **DemanderRésoudre** pour solliciter un calcul. Evidemment ce procédé sera réitéré jusqu'au niveau de détail voulu. Cependant, pour des raisons de simplification, certains performatifs seront définis à partir de mots plus courts et plus faciles à comprendre. Par exemple, le performatif **Aviser** devrait en fait s'écrire **InformerDemanderPrévenir** d'après notre système de dénomination.

On distingue deux grands groupes d'actes de langage: les actes *initiateurs* qui sont à l'origine d'une conversation et les actes *réponses* qui n'interviennent qu'en réponse à un acte initiateur. A ces actes élémentaires, il faudrait ajouter d'autres types d'actes plus complexes qui s'appuient sur des structures conversationnelles plus élaborées, telles que l'appel d'offre ou les différentes formes de collaboration, de coordination d'action et de négociation, que nous verrons aux chapitres suivants.

Éléments de description des actes de langage

Les actes de langage seront donnés sous une forme double: une fiche regroupe les informations de l'acte et un diagramme BRIC décrit la structure conversationnelle associée. Les fiches exposent précisément les conditions d'application d'un acte de langage du point de vue de l'initiateur de l'acte, reprennent les composantes des actes illocutoires en indiquant le but illocutoire, les conditions essentielles et préparatoires décrites par Searle et Vanderveken, ainsi que les conditions de succès et de satisfaction. Elles comprennent aussi les échecs possibles et leurs conséquences normales. Une fiche est composée des parties suivantes:

Format : décrit la syntaxe du message associé à l'acte. Il est de la forme $A : B \ll F(P_1, \dots, P_n)$ où **A** est l'agent émetteur, **B** le destinataire, **F** est le performatif indiquant l'acte de langage, et les P_i sont les arguments du performatif représentant le contenu propositionnel de l'acte.

Sous-actes : liste l'ensemble des variantes de l'acte et leurs performatifs.

Conditions : donne les conditions de réalisation de l'acte, c'est-à-dire ses conditions préparatoires et essentielles. Elles portent généralement sur les intentions et les croyances de l'émetteur et sur les compétences des deux interlocuteurs.

Succès : décrit les conséquences normales du succès de l'acte et le message qui réalise cette condition.

Satisfaction : liste les post-conditions de satisfaction associées à la satisfaction de l'acte, ainsi que le message signalant la validité de cette condition.

Echecs : donne les conditions d'échecs "normales", c'est-à-dire entièrement prévisibles si l'une des conditions de réalisation de l'acte n'est pas satisfaite. Ces échecs, signalés par des messages particuliers, entraînent généralement des conséquences sur les croyances de l'émetteur et d'une manière générale sur son état. Cependant, il n'est pas fait mention des comportements de demande d'explications qui sont souvent associés aux causes d'échecs.

Les conditions et les conséquences de ces actes sont spécifiées à l'aide de prédicats et de cognitons de base. En voici la liste:

Cognitons

croire(A,P) : l'agent A croit l'expression P. Les croyances sont supposées être révisables. **croire(A,P)** signifie en position prédicative que l'agent A croit P, et en position d'action que l'agent A remet en cause ses croyances de manière à croire désormais que P est vrai, quelle que soit la valeur de sa croyance antérieure sur P.

but(A,P) : l'agent A a le but que la condition P soit vraie.

intention(A,P) : l'agent A a l'intention de faire l'action P.

engager(A,B,P) : l'agent A est engagé auprès de B pour faire P.

compétent(A,P) : signifie que l'agent A est compétent pour faire l'action P. **croire(A,compétent(B,P))** précise que l'agent A croit que l'agent B est compétent pour faire P.

Prédicats

calculer(E) : représente l'action de calculer une expression E. Par exemple,

- **calculer(2x+3 avec x=5)** correspond à l'action de calculer l'expression 2x+3 dans laquelle la valeur x vaut 5,
- **calculer({x|descendants(LouisXIV,x)})** décrit l'action de calculer tous les descendants de Louis XIV.

percevoir(A,E) : indique si l'agent A perçoit l'objet ou la situation E.

exec(P) : indique si l'action P est exécutée ou non.

vouloirFaire(A,P) : signifie que l'agent A voudra bien faire P dans le futur. Ce cogniton est équivalent à **finalt(intention(A,P))**, en utilisant l'opérateur **finalt** présenté à la section 5.8, dans le cadre de la théorie des intentions de Cohen et Levesque.

Les exercitifs

Les exercitifs comprennent essentiellement deux actes de langage principaux: les délégations (demandes d'action et demandes de résolution) et les demandes d'abonnement. Ces deux actes de langage sont initiateurs de conversation.

La délégation

La conversation associée à l'acte de délégation est certainement, avec la question, le type de conversation le plus utilisé dans les systèmes multi-agents. Un modèle en termes de réseau de Petri a d'ailleurs déjà été présenté précédemment (fig. 6.4). Une demande d'action ou de résolution d'un problème est envoyée à un agent qui indique s'il s'engage ou non à la réalisation de cette tâche. S'il s'engage, alors

- a) soit il retourne ensuite une notification de fin d'action ou le résultat s'il s'agit d'une résolution de problèmes,
- b) soit il indique qu'il lui a été impossible d'accomplir son engagement, malgré sa promesse.

On notera dans le diagramme BRIC correspondant (fig. 6.5) que les conditions d'initiation de l'acte ne sont pas traitées dans ce diagramme, car elles appartiennent au système de décision et d'allocation de tâche de l'agent A qui choisit l'agent auquel il désire demander ce service. De même, les échecs et les satisfactions produisent des modifications de l'état mental d'un agent non décrites dans ce diagramme.

L'abonnement

L'abonnement, ou plus exactement la demande d'abonnement, est une forme de conversation très importante et malheureusement peu étudiée dans les travaux sur les systèmes multi-agents. Le principe en est simple: un agent A désire être prévenu lorsqu'un événement se passe ou obtenir toutes les informations qu'un agent B pourrait acquérir sur un sujet particulier. Par exemple, le travail d'exploration, de surveillance ou d'enquête à plusieurs est fondé sur ce type de protocole dans lequel des agents ont pour mission de référer de toutes les informations concernant tel type d'informations. L'abonnement peut être cyclique, A demande à B de faire un rapport systématiquement à intervalles réguliers, ou sur événement, dès que quelque chose de nouveau est apparu. La figure 6.6 montre un modèle de conversation de demande d'abonnement sur événement. On peut constater que ce type de conversation produit des échanges réguliers, jusqu'à ce que:

- a) soit l'initiateur de la conversation (l'agent A) émette une fin d'abonnement,
- b) soit le destinataire (l'agent B) considère qu'il ne peut plus faire son travail et arrête ses émissions.

Les interrogatifs

Les interrogatifs servent à poser des questions dont les réponses ne doivent pas donner matière à l'exécution d'un calcul complexe. Les questions peuvent porter aussi bien sur le monde, en demandant à quoi croit un agent, ou sur l'état mental du destinataire: quels sont ses buts, ses engagements, ses accointances, etc. La figure 6.7 montre un modèle d'acte de langage interrogatif.

Poser une question est relativement simple puisqu'il n'existe pas d'engagement possible du destinataire: si A questionne B, soit B connaît la réponse, veut bien la donner et la donne, soit il ne sait pas ou refuse de répondre.

Les assertifs

Les assertifs servent à donner des renseignements sur le monde et donc à décrire des objets ou des situations. On peut considérer deux niveaux d'assertifs. Ceux qui sont envoyés à la demande d'une information, et qui sont donc soumis à la logique d'autres

Demander : acte de langage principal consistant à demander à un autre agent d'effectuer une tâche: accomplir une action ou résoudre un problème

Format : A: B << Demander(P) avec P une tâche

Sous-actes :

- DemanderFaire(P) avec P une action
- DemanderRésoudre(P) avec P un problème

Conditions :

- $\text{but}(A,C) \wedge \text{croire}(A, \text{exec}(P) \Rightarrow C) \wedge$
 $\text{croire}(A, \text{compétent}(B,P)) \wedge$
 $\text{croire}(A, \text{vouloirFaire}(A,P)) \wedge$
 $[\text{croire}(A, \neg \text{compétent}(A,P)) \vee \neg \text{vouloirFaire}(A,P)]$
- $\text{croire}(A, \neg \text{finalt}(C))$

Succès :

- B: A << AccepterFaire(P) \rightarrow engager(B, A, P)

Satisfaction :

- B: A << NotificationFinAction(P) \rightarrow croire(A, C) si P est une action
- B: A << Résultat(P,R) \rightarrow croire(A,R) si P est un problème

Echecs :

- B: A << RefuserFaire(P) \rightarrow croire(A, $\neg \text{compétent}(B,P) \vee \neg \text{vouloirFaire}(B,P)$)
- A << EchecFaire(P) \rightarrow demander explications

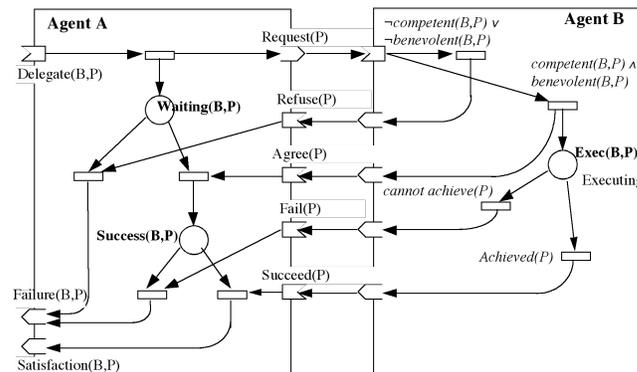


Figure 6.5: Modèle conversationnel de délégation d'une tâche à un autre agent

Abonner : Acte de langage principal consistant à demander à un autre agent qu'il le prévienne selon l'une des modalités suivantes: à intervalles réguliers, lorsqu'un événement intervient ou lorsque le destinataire possède de nouvelles informations d'un certain type.

Format : $A: B \ll \text{Abonner}(P,R)$ avec P une condition et R une expression

Conditions :

- $\text{but}(A, \text{croire}(B,P) \Rightarrow \text{croire}(A,R))$
- $\text{croire}(A, \text{finalt}(\text{croire}(B,P))) \wedge \text{croire}(A, \text{croire}(B,P) \Rightarrow \text{vouloirFaire}(B, B: A \ll \text{Aviser}(R)))$

Succès :

- $B: A \ll \text{AccepterAbonner}(P,R) \rightarrow \text{engager}(B, A, \text{croire}(B,P)) \Rightarrow \neg \text{vouloirFaire}(B: A \ll \text{Aviser}(R))$

Satisfaction :

- $B: A \ll \text{Aviser}(R) \rightarrow \text{croire}(A, R)$
- $B: A \ll \text{RefuserAbonner}(P,R) \rightarrow \text{croire}(A, \neg \text{vouloirFaire}(B, \text{croire}(B,P) \Rightarrow \text{vouloirFaire}(B: A \ll \text{Aviser}(R)))$

Echecs :

- $B: A \ll \text{RefuserAbonner}(P,R) \rightarrow \text{croire}(A, \neg \text{vouloirFaire}(B, \text{croire}(B,P) \Rightarrow \text{vouloirFaire}(B \ll A, \text{Aviser}(R)))$

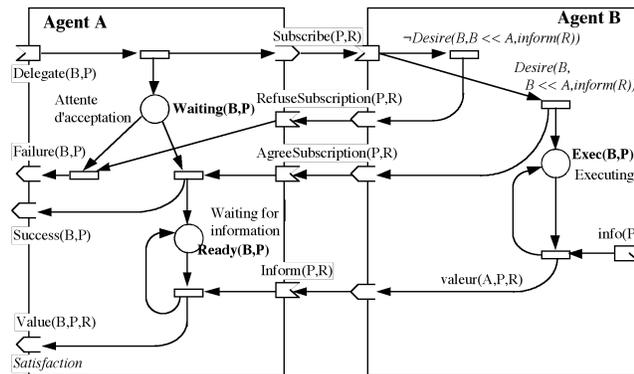


Figure 6.6: Modèle conversationnel d'un abonnement. La résiliation de l'abonnement n'est pas indiqué.

actes de langage, et ceux qui cherchent à faire partager une croyance à un autre agent. Dans le premier cas, le succès et la satisfaction sont liés: il suffit de répondre à la requête pour que l'acte soit réussi, c'est le cas par exemple du performatif **Répondre** qui est envoyé à la suite d'une demande d'information. Dans le second, en revanche, l'acte se décompose en une phase de transmission de l'information (ce qui constitue le succès) et une phase d'acceptation de l'information (la satisfaction), qui prouve que cette information est compatible avec les croyances du destinataire. Par exemple, un avertissement (exemple: il y a un virus qui se promène) conduit à donner une information sans que cela découle d'un acte initiateur antérieur. Ce type de conversation, dont le performatif **Affirmer** est caractéristique (fig. 6.8), suit un protocole assez simple: A informe B, lequel accepte (**Acquiescer**) ou exprime son désaccord (**RefuserCroire**). Dans une forme plus complexe, B pourrait même réfuter ce que lui dit A et il peut s'ensuivre un dialogue de type *argumentation* où les deux agents cherchent mutuellement à se convaincre de la véracité (ou de la fausseté) d'un énoncé. On trouvera dans les travaux de J. Quinqueton, P. Reitz et J. Sallentin des définitions d'agents rationnels engagés dans des conversations complexes portant en particulier sur l'argumentation (suite d'affirmation, d'acceptation et de réfutations, voire de silence) comme forme supérieure d'affirmation (Quinqueton et al. 1991).

Outre le performatif **Affirmer**, les assertifs comprennent le performatif **Inform** qui est plus faible que le précédent, puisqu'il ne suppose pas qu'un agent A qui informe un agent B désire que B croie effectivement l'information qu'il lui envoie.

Quelques autres performatifs

A côté des performatifs principaux que nous venons de voir, il est possible de définir d'autres performatifs. En voici quelques uns que nous ne ferons que lister sans en donner les spécifications détaillées:

Les promissifs

OffrirService : un agent A propose ses services (exemple: réaliser une tâche) à un agent B qui peut refuser ou non. Cette proposition engage A auprès de B, qui normalement doit exécuter les demandes de B correspondant à ses compétences.

Proposer/Confirmer/Réfuter Hypothèse : cet ensemble de performatifs peuvent être utilisés lors d'une résolution distribuée de problèmes.

Les expressifs

SaitFaire : un agent A indique à un autre agent B ses compétences. Il existe aussi le performatif inverse, dans lequel un agent précise qu'il ne sait pas faire quelque chose.

Croire : un agent A indique à un autre agent B qu'il pense qu'une expression est vraie.

Cette liste ne se veut ni définitive, ni exhaustive. Nous verrons tout un ensemble d'autres performatifs aux chapitres suivants.

6.4 KQML

Aux Etats-Unis, un projet de recherche fondé par la DARPA a pour but de développer un standard de communication de haut niveau fondé sur les actes de langage pour permettre à des agents cognitifs de coopérer. Le tableau 6.3 donne la liste des performatifs proposés pour ce langage qui s'appelle KQML pour "Knowledge Query and Manipulation Language". Il s'agit d'un projet qui n'est pas achevé et les descriptions qui sont données ici pourront changer avec le temps. Malgré l'intérêt certain d'un tel travail, le projet KQML présente de nombreuses lacunes. Ainsi, pour Cohen et Levesque, il présente les défauts suivants (Cohen et Levesque 1995):

- *Ambiguïté et imprécision.* La signification des performatifs est donnée sous la forme de simples descriptions en langage naturel, ce qui les rend souvent vagues et confus.
- *Performatifs inutiles et incohérents.* Certains performatifs ne sont en fait pas des actes de langage, car ils n'ont pas le pouvoir de satisfaire les buts de l'agent émetteur. C'est particulièrement vrai des performatifs **achieve**, **broker** et **stream-all**, qui ne peuvent être compris comme des actes de langage indirects puisqu'ils satisfont en fait des buts appartenant à d'autres agents. Néanmoins, il est clair que tout système de communication doit pouvoir faire intervenir des médiateurs et il est donc indispensable de prendre en compte de tels actes.
- *Performatifs manquants.* Malgré le nombre imposant de performatifs, toute une catégorie de performatifs manque: les *promissifs*. Par exemple, il n'est pas possible en KQML de dire que l'on s'engage, auprès d'un tiers, à accomplir une action.

Toutes ces raisons montrent que KQML manque cruellement de spécifications et de formalisation. De ce fait, n'importe qui peut prétendre qu'il emploie KQML simplement parce que ses agents sont capables de s'envoyer des requêtes et des informations. A partir de ces critiques, Cohen et Levesque proposent de définir un ensemble minimum de performatifs en leur donnant des propriétés de compositionnalité, afin qu'il soit possible de définir de nouveaux performatifs comme combinaison d'actes de langage plus primitifs. Pour cela, et à partir de leur théorie de l'action rationnelle étudiée au chapitre 5, ils ont développé une sémantique pour quelques-uns des performatifs les plus fondamentaux de KQML. Malgré l'intérêt de ce travail, je pense que les spécifications en termes d'états mentaux posent un problème majeur: si l'on suppose que des standards de communication comme KQML ont par essence le projet de faire communiquer entre eux des agents de natures diverses, alors toute théorie reposant sur des états mentaux imposent aux agents de se conformer à une architecture spécifique qui implémente précisément cette théorie. Par exemple, la théorie des cognitions que nous avons développée au chapitre 5 ne s'accorde pas nécessairement à celle de Cohen et Levesque, laquelle ne reprend pas les mêmes primitives que celle de Y. Shoham. De ce fait, tout standard de communication, qui repose sur la manière dont un agent se comporte, limite les possibilités d'intégration

Nom	Description
achieve	A veut que Y accomplisse une action
advertise	A indique qu'il peut faire P
ask-about	A veut toutes les expressions sur P de la BC de B
ask-all	A veut toutes les réponses de B à une question
ask-one	A veut une réponse de B à une question
break	A veut supprimer une communication
broadcast	A veut que B envoie un performatif à toutes ses accointances
broker-all	A veut que B récupère toutes les réponses à un performatif
broker-one	A veut de l'aide pour récupérer la réponse à un performatif
deny	A indique à B que le performatif ne s'applique plus à B
delete	A veut que B supprime une expression de sa BC
delete-all	A veut que B supprime toutes les P correspondantes de sa BC
delete-one	A veut que B supprime une expression P de sa BC
discard	A ne veut pas les réponses suivantes de B
eos	Fin d'une suite de réponse à une requête préalable
error	A considère que le message de B est mal formé
evaluate	A veut que B simplifie l'expression associée
forward	A veut que B délègue un performatif
generator	Identique à standby pour un 'stream-all'
insert	A veut que R insère une expression dans sa BC
monitor	A veut que R mette à jour ses réponses à un 'stream-all'
next	A veut la réponse suivante à une requête préalable
pipe	A veut que R redirige tous les performatifs suivants à un agent
ready	A est prêt à répondre à un performatif préalable de B
recommend-all	A veut tous les agents qui répondent à un performatif donné
recommend-one	A veut un agent qui répond à un performatif donné
recruit-all	A veut que tous les agents capables répondent à un performatif
recruit-one	A veut qu'un agent capable répond à un performatif
register	A indique à B qu'il peut accomplir une action
reply	A répond à une demande attendue
rest	A veut toutes les réponses suivantes
sorry	A ne peut fournir une réponse plus informative
standby	A veut que B soit prêt à répondre à un performatif
stream-about	Version à réponses multiples de 'ask-about'
stream-all	Version à réponses multiples de 'ask-all'
subscribe	A veut que B mette à jour ses réponses à un performatif
tell	Indique qu'une expression fait partie de la BC de A
transport-address	A associe un nom symbolique à une adresse de transport
unregister	Un 'deny' de 'register'
untell	indique qu'une expression ne fait pas partie de la BC de A

Table 6.3: liste des performatifs définis dans KQML

d'agents véritablement hétérogènes. Il apparaît donc qu'une spécification des communications, non plus en termes d'états mentaux, mais en termes de protocoles, présente l'avantage de pouvoir faire abstraction de la nature propre des agents en se focalisant sur les relations qui existent entre les communications. C'est d'ailleurs cette approche qui a été présentée dans ce chapitre et qui sera développée dans les chapitres suivants.

Questionner : acte de langage principal consistant à demander un renseignement à un autre agent.

Format : $A: B \ll \text{Questionner}(f,P)$ où P est une donnée et f une fonction

Sous-actes :

- **QuestionnerBool**(P) avec P une expression propositionnelle
- **QuestionnerRaisons**(E) avec E une expression sans variables

Conditions :

- **but**($A, \text{croire}(A,R)$) avec:
 $R = f(P) \wedge \text{croire}(A, \text{compétent}(B,f)) \wedge$
 $\text{croire}(A, \neg \text{compétent}(A,f)) \wedge$
 $\text{croire}(A, \neg \text{croire}(A,R)) \wedge$
 $\text{croire}(A, \text{vouloirRépondre}(B,f))$

Satisfaction :

- $B: A \ll \text{Répondre}(R) \rightarrow \text{croire}(A,R)$

Echecs :

- $B: A \ll \text{RefuserRépondre}(R) \rightarrow$
 $\text{croire}(A, \neg \text{vouloirRépondre}(f) \vee \neg \text{compétent}(B,f))$

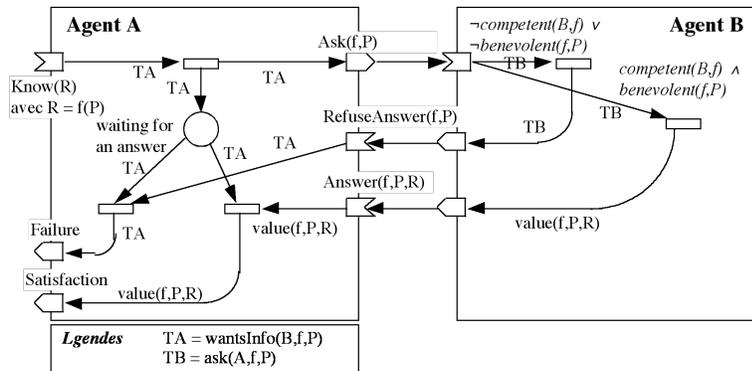


Figure 6.7: Modèle conversationnel d'une question

Affirmer : acte de langage principal consistant à faire partager une croyance à un autre agent, c'est-à-dire à convaincre un tiers de la véracité d'un fait.

Format : $A: B \ll \text{Affirmer}(R)$

Conditions :

- $\text{But}(A, \text{croire}(B, R)) \wedge \text{croire}(A, \neg \text{croire}(B, R))$

Satisfaction :

- $B: A, \text{Acquiescer}(R) \rightarrow \text{croire}(A, \text{croire}(B, R))$

Echecs :

- $B: A, \text{RefuserCroire}(R) \rightarrow \text{croire}(A, \text{croire}(B, \neg R))$

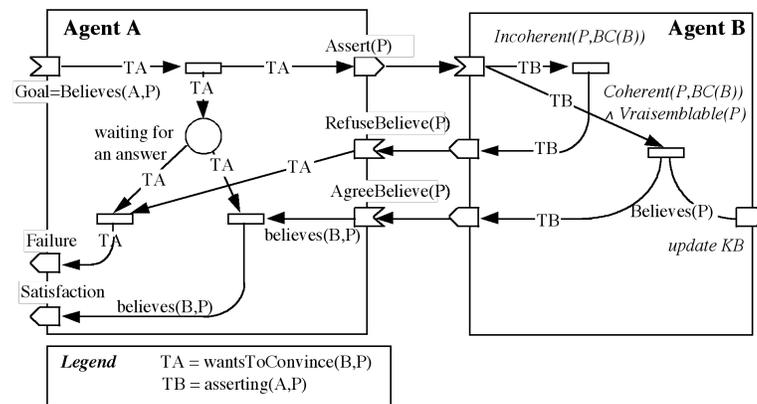


Figure 6.8: Modèle conversationnel d'une affirmation

Chapter 7

Collaboration et répartition des tâches

Si la coopération permet de procurer des avantages en termes d'efficacité quantitative et d'émergence qualitative, comme nous l'avons vu au chapitre 2, elle pose aussi des problèmes de répartition du travail entre les agents. Ces problèmes sont difficiles, car ils font intervenir de nombreux paramètres: capacités cognitives et d'engagement des agents, compétences des individus concernés, nature des tâches, efficacité, coûts de transmission, structures sociales dans lesquelles évoluent les agents, etc.

De ce fait, la répartition des tâches et des ressources constitue à la fois l'un des domaines majeurs des systèmes multi-agents et l'une de leurs principales contributions à l'informatique en général. En mettant l'accent sur l'allocation distribuée du travail, sur la notion de contrat et d'engagement, les systèmes multi-agents posent le problème de l'activité dans des termes à la fois sociaux et calculatoires, ce qui les distingue des formes plus classiques utilisées auparavant, qu'il s'agisse de problème d'affectation de machines ou de répartition des processus sur des processeurs. Néanmoins, comme nous le verrons, les techniques d'allocation de tâches reprennent certains des résultats obtenus dans le domaine de la recherche opérationnelle et des systèmes distribués, en les adaptant à leurs besoins.

Répartir des tâches, des informations et des ressources revient à répondre à la question: *qui doit faire quoi et avec quels moyens, en fonction des buts et des compétences des agents et des contraintes contextuelles (types et quantités de ressources, nature de l'environnement, etc.)*. Il s'agit là d'un des points essentiels de la fonction organisationnelle d'une société d'agents (cf. chap. 3) avec la coordination d'actions que nous examinerons au chapitre 8.

7.1 Modes d'allocation des tâches

L'allocation (ou répartition) des tâches passe par la définition des mécanismes organisationnels par lesquels des agents peuvent mettre leurs compétences en commun afin de réaliser un travail collectif. Il s'agit donc de décrire la manière d'allouer les tâches, en sachant que les capacités d'un agent dépendent à fois de ses aptitudes

intrinsèques (architectures dont ils disposent, capacités cognitives, types de communications envisagées), des moyens énergétiques dont il dispose (temps de calcul et autonomie énergétique), que des ressources externes (outils, sources d'énergies) et des contraintes environnementales.

Les tâches qui réclament plus de moyens, de travail ou de savoir-faire qu'un seul agent n'est capable de fournir, doivent être *décomposées* d'abord en plusieurs sous-tâches puis être *réparties* (ou allouées) parmi les différents agents. Ces deux opérations sont évidemment liées car la décomposition des tâches doit souvent prendre en compte les compétences des agents présents et faciliter ainsi la répartition qui s'ensuit.

7.1.1 Critères de décomposition des tâches

Bond, Gasser et Hill (Bond & Gasser 1988) (Gasser & Hill 1990) ont présenté quelques critères de décomposition des problèmes. Ils indiquent notamment que si les problèmes peuvent être naturellement analysés par niveau d'abstraction (en travaillant du plus général au plus détaillé), il est aussi nécessaire de prendre en compte d'autres contraintes telles que le contrôle, les données ou les ressources. Il est en effet nécessaire de rendre les tâches aussi indépendantes que possibles les unes des autres de manière à diminuer la coordination. En particulier, on cherchera à minimiser la quantité d'informations que les tâches doivent se transmettre et à faire en sorte que les tâches puissent utiliser les ressources locales afin de diminuer les conflits liés aux ressources.

Mais, bien que les activités de décomposition soient aussi importantes que celles de répartition, elles sont généralement effectuées par des êtres humains et il n'existe pas à notre connaissance de système informatique de décomposition de tâches. De ce fait, l'essentiel de la recherche s'est porté sur la répartition automatique des tâches, laissant un peu de côté le problème de leurs décomposition en sous-tâches. Nous avons cependant parlé des différentes manières d'aborder la décomposition des tâches (à l'aide d'une perspective objet ou fonctionnelle) au chapitre 3.

7.1.2 Rôles

Un système d'allocation automatique des tâches doit être capable de mettre en rapport des agents qui ont besoin d'une information ou désirent que l'on réalise une tâche, les *clients* ou *demandeurs*, et des agents capables de fournir un service, les *fournisseurs* ou *serveurs*. Souvent les mêmes agents peuvent être à la fois clients et serveurs, la qualité de client ou de fournisseur étant déterminée généralement de manière dynamique lors du fonctionnement d'un système multi-agents. On dit alors que les agents prennent le *rôle* de client ou le *rôle* de fournisseur, sans qu'il s'agisse d'une de leurs caractéristiques intrinsèques. D'autres rôles peuvent être définis, tels que celui de *médiateur* (trader), qui s'occupe de la mise en contact des serveurs et des clients.

7.1.3 Formes d'allocation

La gestion de la répartition des tâches peut s'effectuer soit en centralisant le processus d'allocation soit en le distribuant à l'ensemble des agents concernés. Dans un mode d'*allocation centralisé*, deux cas se présentent :

1. Si la structure de subordination est hiérarchique, c'est le supérieur qui demande précisément à un subordonné d'accomplir sa tâche. On parlera alors d'allocation rigide ou définie. Ce mode d'allocation, caractéristique d'un appel de procédure en programmation classique, ne sera pas étudié ici. On ne rencontre ce type de répartition que dans les organisations fixes (cf. chap. 3).
2. Au contraire, si la structure est égalitaire, la répartition passe alors par la définition d'agents spéciaux, les médiateurs, qui gèrent l'ensemble du processus d'allocation en centralisant les demandes des clients et les offres de services des serveurs, afin de mettre en correspondance ces deux catégories d'agents, ce qui permet d'appliquer des modes centralisés à des organisations variables.

Dans un mode d'allocation distribué, chaque agent s'occupe individuellement d'obtenir les services des fournisseurs qui peuvent lui être utiles pour la réalisation de ses projets. Ces modes ne s'appliquent qu'aux organisations variables, c'est-à-dire qui supposent que les liens entre les clients et les fournisseurs peuvent évoluer dans le temps, sans remettre en cause la structure générale de l'organisation. On distingue deux mécanismes d'allocation distribuée dans les organisations prédéfinies :

1. Le mode d'*allocation par réseau d'acointances* suppose que les clients possèdent une représentation des autres agents et des capacités dont ils disposent. Nous verrons qu'il n'est pas nécessaire que les agents connaissent les capacités de tous les autres agents pour pouvoir résoudre leur problème, mais seulement que le réseau formé par l'ensemble des acointances soit fortement connexe (c'est-à-dire qu'il existe un chemin allant de n'importe quel agent vers n'importe quel agent) et évidemment cohérent (c'est-à-dire que les représentations sur les compétences des acointances d'un agent correspondent bien aux compétences effectives des agents).
2. Le mode d'*allocation par appel d'offre* est plus connu en intelligence artificielle distribuée sous le nom de *réseau contractuel*. Il présente l'avantage d'une très grande dynamique et s'avère particulièrement facile à mettre en œuvre. Nous verrons cependant qu'il faut faire attention lors de sa mise en œuvre à un certain nombre de problèmes inhérents à la nature particulièrement dynamique de ce mode.

A ces modes s'appliquant à des organisations prédéfinies, il faut ajouter le mode d'*allocation émergente*, moins connu généralement et plus caractéristique des systèmes réactifs dans lesquels les communications s'effectuent de manière incidente (cf. chap. 6) par propagation de stimuli. La figure 7.1 résume les principaux modes d'allocation de tâches. Il peut exister évidemment un certain nombre de variantes de ces approches et, en particulier, des approches hybrides qui tentent de synthétiser les avantages de ces différentes approches.

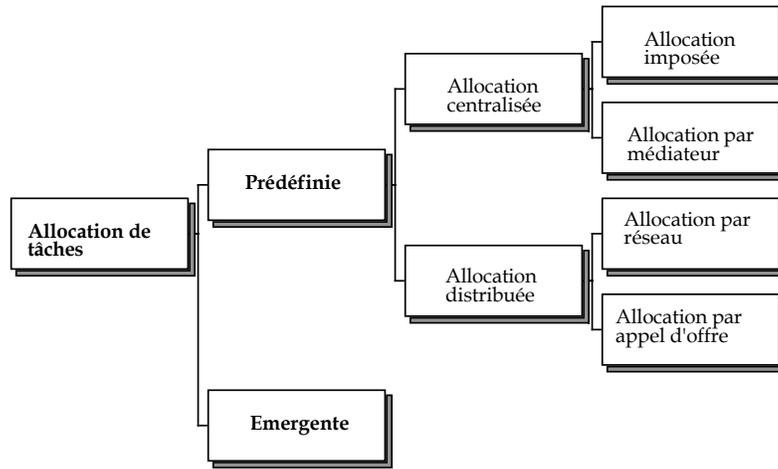


Figure 7.1: Les principaux modes de répartition des tâches

7.2 Allocation centralisée des tâches par médiateur

Le cas le plus simple est celui où il n'existe qu'un seul médiateur qui dispose d'une table d'accointance lui indiquant l'ensemble des agents capables d'accomplir une tâche particulière T . Cette table peut être mise à jour directement par les fournisseurs qui entrent dans le système et qui indiquent ainsi qu'elles sont leurs compétences.

La figure 7.2 illustre le déroulement du processus. Lorsque l'agent A a besoin d'effectuer une tâche T qu'il ne peut (ou ne veut pas) faire lui-même, il demande au médiateur de trouver quelqu'un pour réaliser T . Le médiateur s'adresse alors aux agents dont il sait qu'ils possèdent la capacité de faire T . Si l'un d'eux accepte, le médiateur envoie l'accusé d'acceptation au client, sinon il l'informe qu'il n'a trouvé personne pour accomplir la tâche.

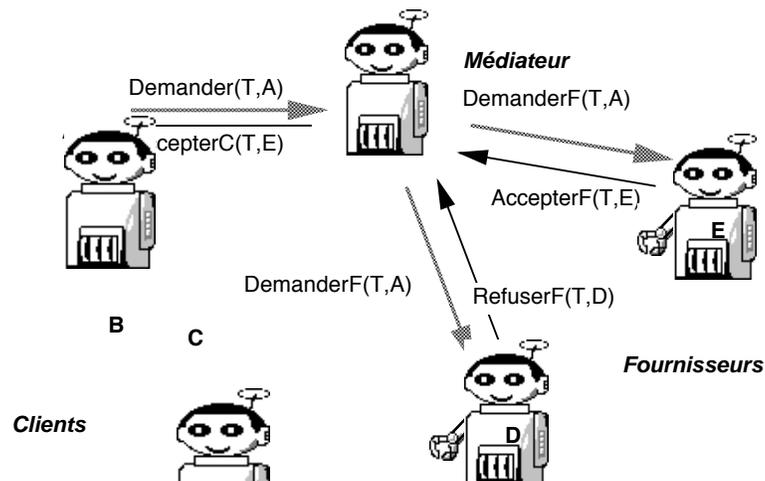


Figure 7.2: Allocation centralisée de tâches

On suppose dans ce traitement qu'un fournisseur, qui accepte d'accomplir une tâche, s'engage effectivement à la faire et ne se dédiera pas lors de la passation de contrat avec un client ou au cours de la réalisation de la tâche. On fera aussi l'hypothèse que le médiateur possède une table des compétences des agents qui soit à la fois complète et cohérente, tous les agents A qui possèdent une compétence étant bien référencés dans cette table, et que toute référence dans cette table existe bien, c'est-à-dire que pour tout couple $\langle A, C \rangle$ de la table, l'agent A possède bien la compétence C.

Avec ces éléments, le protocole d'allocation peut être défini sous la forme d'un composant BRIC. On ne modélisera que le médiateur, puisque le comportement des clients et des fournisseurs reprend les structures de requêtes (**DemanderFaire**) et de réponses décrites au chapitre 6 sur les actes de langages. Ce module, qui définit entièrement le comportement du médiateur, prend en entrée les requêtes du client et les réponses du fournisseur et produit en sortie les demandes vers les fournisseurs susceptibles de supporter cette requête ainsi que les réponses vers les clients. Les communications entre le médiateur, les clients et les fournisseurs utilisent un ensemble de messages correspondant aux actes de langages suivants:

Demander(T,X) est une demande du client X au médiateur pour que la tâche T soit réalisée.

Accepter(T,Y) est la réponse du médiateur au client pour lui indiquer que l'agent Y s'engage à accomplir la tâche T.

Impossible(T) est une réponse du médiateur au client pour lui indiquer qu'il n'existe aucun agent qui accepte de faire la tâche T.

DemanderF(T,X) est une demande du médiateur afin qu'un fournisseur accomplisse la tâche P pour le client X.

AccepterF(T,Y,X) est la réponse positive du fournisseur Y à la demande du médiateur. Le fournisseur indique au client qu'il accepte d'accomplir la tâche T pour le client X.

RefuserF(T,Y,X) est la réponse négative du fournisseur Y à la demande d'allocation de la tâche T pour le client X.

La figure 7.3 montre la représentation du médiateur centralisé en BRIC.

Lorsque le médiateur reçoit de la part d'un client X une demande pour faire T, il cherche dans sa table des compétences une liste d'agents (éventuellement triée par ordre de préférence) capables d'effectuer cette tâche. Si cet ensemble E est vide, alors le médiateur répond par le message **ImpossibleFaire**. Sinon, il demande itérativement à tous les éléments de E s'il leur est possible de faire T, en utilisant la place P_1 comme mémoire de tous les fournisseurs encore à examiner. Si l'un d'entre eux accepte, la place P_1 est vidée de cette demande de tâche et le médiateur répond au client X que le fournisseur Y accepte d'accomplir T. Ce module utilise les trois structures de données suivantes:

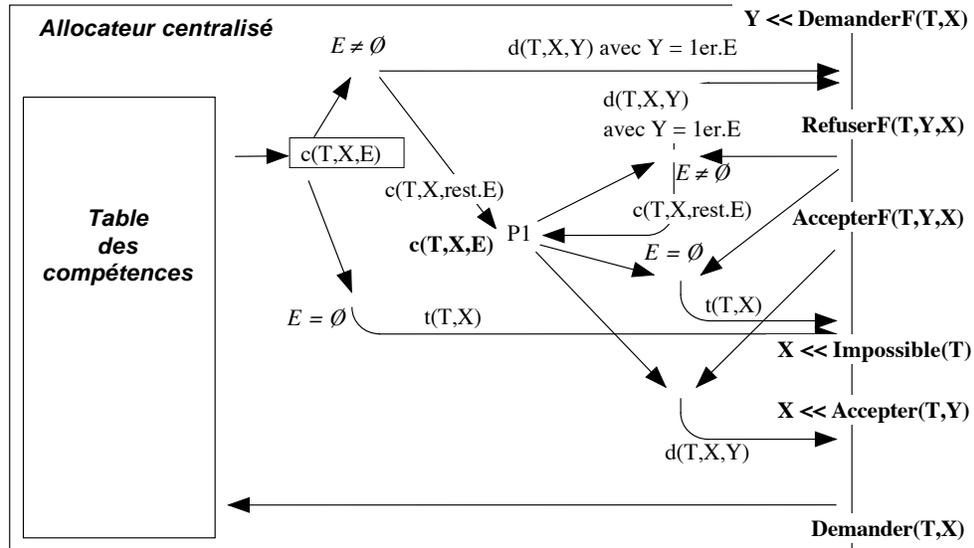


Figure 7.3: La description du médiateur centralisé en BRIC

$c(T,X,E)$ donne la liste des agents E ayant la compétence pour effectuer la tâche T pour le client X . $1er.E$ retourne le premier élément de la liste (identique à `car` en LISP, et `rest.E` retourne la liste E privée de son premier élément, identique à `cdr` en LISP).

$d(T,X,Y)$ demande du médiateur auprès du fournisseur Y pour qu'il effectue la tâche T pour X .

$t(T,X)$ tâche T demandée par X .

Il est possible d'améliorer ce mécanisme de plusieurs manières. La première consiste à optimiser l'allocation en sélectionnant en priorité les agents les plus compétents ou ceux qui minimisent une fonction de coût. Par exemple, dans le cas des robots récupérateurs de minerai comprenant à la fois des robots foreurs et des transporteurs, il est préférable de répartir les tâches de transports en fonction de la distance qui sépare les transporteurs des foreurs, de manière à minimiser les déplacements à effectuer. On peut envisager deux cas:

1. Si le médiateur connaît les fonctions d'évaluation utilisées par les fournisseurs, il peut trier les agents compétents en fonction des résultats de l'évaluation et effectuer ses demandes dans cet ordre. Le protocole n'est pas modifié: il suffit simplement de supposer que le gestionnaire d'accointances du médiateur retourne l'ensemble des fournisseurs triés par ordre de préférence.
2. Si le médiateur ne connaît pas les fonctions d'évaluation, il doit d'abord demander à chaque agent de formuler sa proposition avant de choisir effectivement celui qui sera considéré comme le meilleur. Cette technique s'apparente alors à un mécanisme d'appel d'offre (cf. section 7.3.2).

Dans notre algorithme, le médiateur parcourt l'ensemble des agents de manière séquentielle, n'envoyant une requête à l'agent suivant que si le précédent a répondu qu'il refusait d'accomplir la tâche. Il est aussi possible de définir des algorithmes parallèles. Dans ce cas, dès qu'un fournisseur indique qu'il est d'accord pour accomplir la tâche, on avertit les autres fournisseurs afin qu'ils sachent qu'il n'est plus nécessaire de répondre. Ce type de gestion est plus fin, car il faut tenir compte du fait que des agents peuvent envoyer leurs acceptations entre-temps et qu'ils doivent donc être capable de mettre à jour leurs engagements.

L'intérêt de l'allocation centralisée est de pouvoir tenir à jour l'ensemble des agents et leurs compétences respectives et donc de favoriser la cohérence du système. De plus, les besoins en optimisation sont plus facilement satisfaits, le médiateur connaissant l'ensemble des agents disponibles, et il lui est plus facile de choisir le "meilleur" des agents par rapport à une demande de tâche donnée.

Mais l'inconvénient majeur de ce type de système est de constituer un goulet d'étranglement et donc de diminuer considérablement les performances du système dès que le nombre des agents et des demandes augmente. En effet, le nombre de messages que le médiateur doit gérer croît comme le carré de N , où N est le nombre d'agents. Si l'on considère qu'il existe un taux α de clients potentiels (α est le rapport nombre de clients/ N) qui font k demandes par unité de temps et si l'on appelle β le taux des fournisseurs potentiels (c'est-à-dire le rapport nombre de fournisseurs/ N) pour chaque demande, alors le nombre M de messages est égal à :

$$M = \alpha k N (2 + 2\beta N)$$

c'est-à-dire au produit du nombre de demandes par la somme des messages qui existent, d'une part, entre les clients et le médiateur et, d'autre part, entre le médiateur et les fournisseurs. Par exemple, si α vaut 0,5 (la moitié des agents sont demandeurs), β vaut 0,2 (20% des agents sont des fournisseurs potentiels pour chaque demande, ce qui signifie que le système est relativement redondant) et k vaut 1 (on a une demande par unité de temps, on obtient la courbe de la figure 7.4).

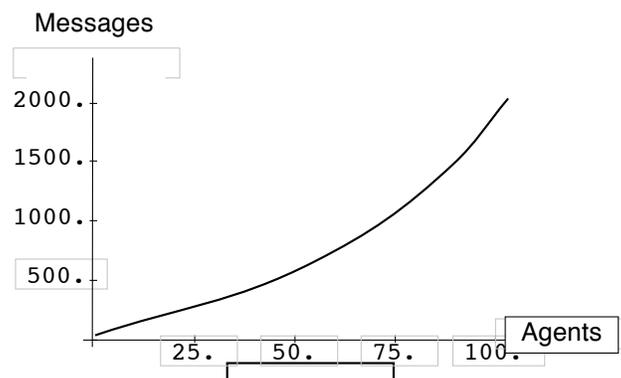


Figure 7.4: Nombre de messages traités par le médiateur en fonction du nombre d'agents du système

On peut constater que si pour 50 agents, le nombre de messages à traiter par le médiateur est de l'ordre de 500, il passe à 2 000 pour 100 agents, ce qui indique que

cette méthode n'est pas à conseiller pour un grand nombre d'agents. De plus, si le système est distribué, tous les messages doivent passer par le médiateur sans tenir compte de la topologie du réseau.

Enfin, un système centralisé est très sensible aux défaillances. En effet, si le médiateur tombe en panne, tout le système s'écroule. Il faut alors prévoir des mécanismes de secours complexes pour essayer de pallier ces problèmes, mais ceci est hors de notre propos. Il est possible d'améliorer cette situation en utilisant plusieurs médiateurs. Mais cela pose d'autres problèmes, car il faut pouvoir gérer la cohérence entre les différents médiateurs, ce qui devient alors très compliqué. Lorsqu'on souhaite que le système puisse répartir ses mécanismes d'allocation, il est préférable de passer à un mode d'allocation véritablement distribué des tâches.

7.3 Allocation distribuée des tâches

7.3.1 Allocation par réseau d'accointances

Dans les systèmes d'allocation par réseau d'accointances, on suppose que chaque agent dispose d'une table de compétences des agents qu'il connaît, sous la forme d'un dictionnaire de la forme

$$\{C_1 : [A_{11}, \dots, A_{1k}], \dots, C_n : [A_{n1}, \dots, A_{nl}]\}$$

où les C_i sont les compétences requises pour réaliser une tâche T_i et les A_{ij} sont les agents qui savent effectuer ce type de tâches. Il est possible de représenter cette table sous la forme d'un tableau local à chaque agent comportant en ligne les compétences et en colonne ses accointances, c'est-à-dire les agents qu'il connaît. Chaque case du tableau comporte alors 1 ou 0 selon que l'agent de la colonne possède ou non la compétence de la ligne correspondante. Par exemple, un agent A, qui sait qu'il possède la compétence C_3 , que la compétence C_1 est disponible chez B et C et que D sait faire C_3 , possède la table de compétence suivante:

	A	B	C	D
C1	0	1	1	0
C2	0	0	1	0
C3	1	0	0	1

On ne suppose pas que chaque agent connaisse l'ensemble des compétences des autres agents, car cela violerait l'hypothèse de représentation partielle des systèmes multi-agents et poserait une beaucoup trop grande contrainte sur la réalisation et la mise à jour de tels systèmes. On suppose simplement que les tables d'accointances sont correctes, c'est-à-dire que leurs indications correspondent bien aux réelles compétences des agents. D'autre part nous ferons l'hypothèse que les tables d'accointances de chacun des agents sont données une fois pour toutes et ne sont pas remises à jour. Nous verrons section 7.3.1 comment traiter le problème des remises à jour dans des réseaux d'accointances.

Un réseau d'accointances peut être représenté sous la forme d'un graphe. Les agents sont les sommets du nœud et les arcs représentent les compétences que

les agents connaissent des autres agents. Ce graphe superpose ainsi de manière graphique les différentes tables d'accointances locales aux agents. Par exemple, la figure 7.5 représente la superposition des tables suivantes:

	A	B	C	D		A	B	C	D		A	B	C	D		A	B	C	D			
C1	0	1	1	0		C1	0	1	0	0		C1	0	1	1	0		C1	0	0	0	0
C2	0	0	1	0		C2	0	0	0	1		C2	0	0	1	0		C2	0	0	1	1
C3	1	0	0	1		C3	1	0	0	0		C3	0	0	0	0		C3	0	0	0	1
	A					B					C					D						

Il existe essentiellement deux modes d'allocation par réseau d'accointances selon que l'on autorise les agents à déléguer leurs demandes à d'autres agents ou non. On parlera alors d'*allocation par délégation* ou d'*allocation directe*.

Allocation directe

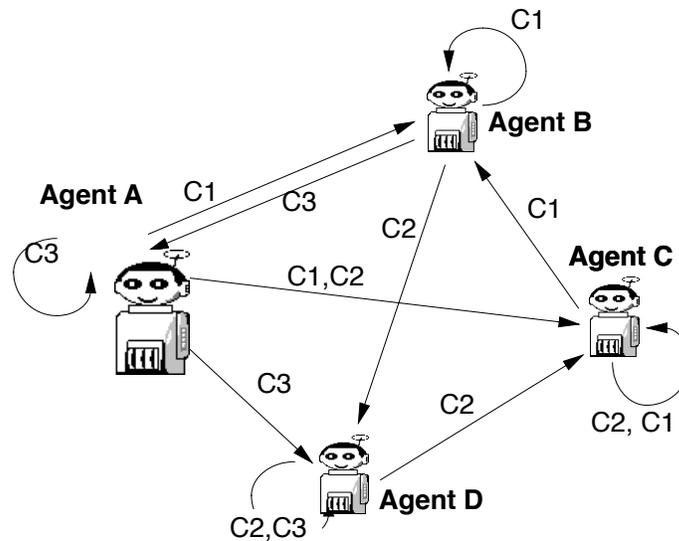


Figure 7.5: Un réseau d'accointances superpose l'ensemble des tables d'accointances locales en un graphe unique, qui représente ce que chaque agent connaît des compétences des autres agents.

Dans le mode direct, un agent ne peut faire exécuter une tâche qu'à un agent qu'il connaît directement. Le mécanisme de répartition est très simple et reprend le comportement du médiateur unique dans le cas des répartitions centralisées. L'agent qui veut faire accomplir une tâche essaye à tour de rôle chacun des agents qu'il connaît et qui dispose de la compétence désirée jusqu'à ce que l'un d'eux accepte. Si aucun n'accepte, on atteint un état spécial qu'il faut traiter. Soit on considère qu'il s'agit d'un défaut du système, mais il faut alors s'assurer que dans tous les cas il y aura au moins un agent qui acceptera d'effectuer la tâche, soit on prend des mesures tendant à passer outre, en affectant autoritairement une tâche à un agent ou en relançant les opérations par un mécanisme d'appel d'offre permettant de choisir l'agent qui a le moins de raisons de refuser cette tâche. Si cela échoue, on peut aussi faire appel à un système centralisé pour allouer la tâche. Les réserves associées

aux systèmes centralisés ne sont plus de mises dans ce cas, puisque seul un petit nombre de demandes seront envoyées au système centralisé, les demandes banales empruntant la voie du réseau d'acointances. Dans ce cas, le système centralisé, qui joue plus le rôle d'un agent de maintenance et de réparation que celui d'un véritable médiateur, s'avère très utile car il peut aussi servir à mettre à jour les tables d'acointances des agents et réorganiser ainsi le réseau d'acointances afin d'accélérer le processus d'allocation.

Ici l'implémentation de ce mécanisme d'allocation n'est pas réalisé par un agent distinct des demandeurs, mais par un module intégré directement dans le système organisationnel des clients potentiels. Il y a donc deux types de connexions: celles qui lient le module d'allocation aux systèmes motivationnel et organisationnel et celles qui représentent les messages que le client échange avec les autres agents. Les connexions internes sont définies par les messages suivants:

Allouer(T) est une demande du module motivationnel vers le module d'allocation pour que ce dernier trouve quelqu'un pour faire la tâche T.

ImpossibleAll(P) est une réponse du gestionnaire d'allocation au système motivationnel de l'agent qui indique qu'il n'existe aucun agent qui accepte de faire la tâche T.

EngagéFaire(Y,T) est une réponse du module d'allocation au module organisationnel indiquant que l'agent Y s'est engagé à faire la tâche T.

Les messages externes sont définis par les actes de langage suivants:

Demander(T,self) est une demande de l'agent à un agent fournisseur potentiel que la tâche T soit réalisée.

Accepter(T,Y) est la réponse positive du fournisseur Y à la demande du client.

Refuser(T,Y) est la réponse négative du fournisseur Y à la demande d'allocation de tâche T.

La structure du module d'allocation directe est présentée figure 7.6. Il reprend les grandes lignes de la structure d'allocation centralisée de la section précédente. Il diffère essentiellement sur deux points:

1. Le paramètre X qui représentait précédemment le client a disparu des structures de données internes du médiateur.
2. Les résultats d'acceptation ou d'impossibilité à trouver un agent sont renvoyés aux systèmes motivationnels et organisationnels du client.

Comme dans le cas du système d'allocation centralisé, l'ordre dans lequel les agents sont questionnés n'est pas nécessairement neutre. On peut envisager plusieurs possibilités, en donnant des priorités aux agents ou en effectuant un calcul qui tienne compte de la charge de traitement d'un agent et des caractéristiques de la tâche. L'avantage de ce mode d'allocation réside évidemment dans sa simplicité. Il suppose

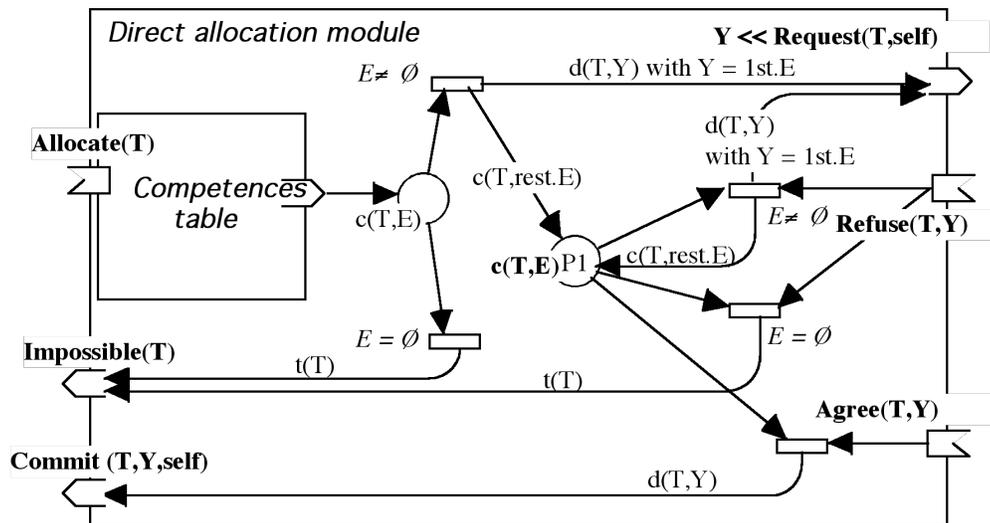


Figure 7.6: Représentation du module de demande du médiateur par accointances directes

simplement que les clients ont la capacité de connaître suffisamment de spécialistes pour que les tâches qu'ils ne peuvent pas effectuer eux-mêmes puissent être réalisées par d'autres. Cependant, il s'avère beaucoup trop limité, car il ne peut mettre en contact des fournisseurs qui ne seraient pas en contact direct avec les clients. Il est donc nécessaire d'introduire un mécanisme qui permette à plusieurs agents de réaliser un contact indirect.

Allocation par délégation

La technique d'allocation par délégation permet de relier entre eux des clients et des fournisseurs qui ne se connaissent pas directement. Ce mode d'allocation permet en effet à un fournisseur à qui on demande d'effectuer une tâche de la renvoyer à un autre agent s'il s'avère qu'il n'est pas capable de le faire. Par exemple, dans la figure 7.7, l'agent A ne connaît personne capable d'accomplir une tâche qui réclame la compétence C2. Il doit de ce fait demander à tous les agents qu'il connaît de demander eux-mêmes à leurs accointances possédant la compétence C2 d'effectuer la tâche T demandée, ce processus devant être répété jusqu'à ce que l'on trouve un agent qui accepte d'accomplir cette tâche ou que l'ensemble du réseau ait été parcouru.

La réalisation d'un tel processus de recherche repose sur des algorithmes de parcours de graphes en parallèles. Il existe essentiellement deux approches: le parcours en profondeur et le parcours en largeur. Il s'agit d'algorithmes très connus en algorithmique distribuée (on pourra par exemple se référer à (Raynal 1987)), mais il faut les adapter à l'allocation de tâches.

Nous ne décrivons ici que la technique reposant sur un parcours en parallèle, laissant au lecteur le soin d'en traduire les principes dans le cas du parcours en profondeur. La technique générale des parcours en parallèle repose sur des algorithmes de diffusion qui propagent récursivement les demandes à tous les agents

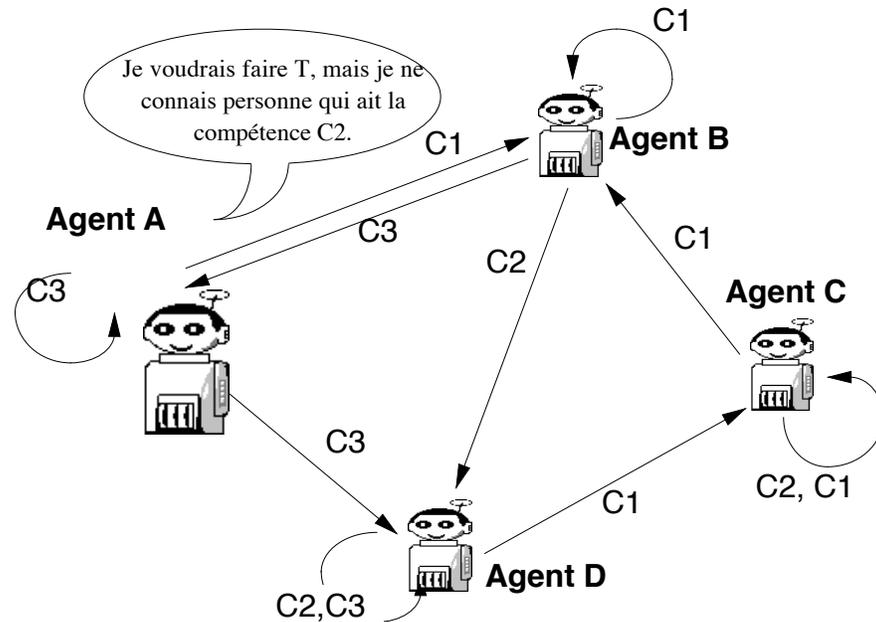


Figure 7.7: Le problème de l'allocation par délégation dans un réseau d'accointances

connus, excepté l'agent demandeur lui-même. Cependant, il faut tenir compte de deux contraintes: d'une part, vérifier que l'on ne demande qu'une seule fois à un agent d'accomplir une tâche T et, d'autre part, s'assurer que le processus termine, afin d'en tenir éventuellement compte pour affecter la tâche T de manière autoritaire ou passer en mode d'appel d'offre. La première difficulté est résolue en marquant les agents lors du parcours et la seconde en utilisant un message supplémentaire, appelé *Acquitter*, qui sert à retourner vers les agents en amont du processus le fait que toutes les demandes situées en aval ont bien été effectuées.

Chaque agent Z , lorsqu'il reçoit une première demande d'un agent Y concernant la réalisation d'une tâche T pour un agent X et réclamant une compétence C , évalue d'abord la tâche. S'il possède la compétence et qu'il l'accepte, il envoie alors à l'agent X un message d'acceptation puis retourne à l'agent Y un message d'acquiescement de manière à indiquer la fin du processus. Sinon, il se marque comme ayant déjà analysé la demande et relance la demande à tous les agents qu'il connaît. Lorsque tous ont répondu par un message d'acquiescement, il en renvoie un lui-même. Lorsque de nouvelles demandes lui sont proposées pour la même tâche T , il se contente alors d'envoyer un message d'acquiescement.

Il faut de plus gérer la possibilité d'avoir plusieurs acceptations simultanées. Du fait du parallélisme, deux agents peuvent en effet accepter la demande et s'engager vis-à-vis de l'agent X . De ce fait, le fournisseur qui accepte ne doit pas s'engager, mais simplement réserver son engagement jusqu'à l'établissement du contrat avec le client. D'autre part le client doit tenir compte du fait qu'il peut recevoir plusieurs propositions et on considère qu'il ne doit accepter que la première. La figure 7.8 illustre le fonctionnement d'un tel système.

Lorsque l'agent A a besoin de faire effectuer la tâche T , il demande à toutes ses

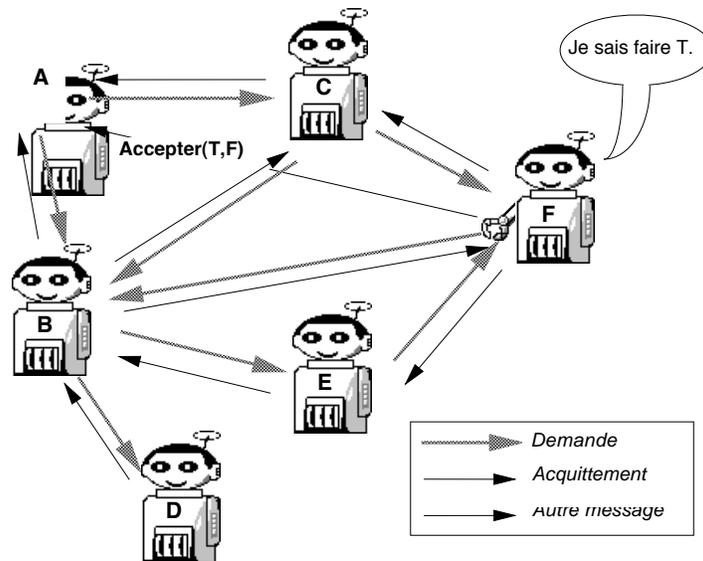


Figure 7.8: Un système d'allocation par délégation

accointances, B et C, d'accomplir T. Comme ils ne peuvent pas le faire eux-mêmes, ils délèguent la demande à D et E pour B et à B et F pour C. De ce fait, B reçoit une nouvelle demande et il se contente d'envoyer un message **Acquitter** à l'agent C. Puisque D ne peut pas le faire et qu'il n'a aucune accointance, il renvoie simplement le message **Acquitter** à B. En revanche E peut déléguer la demande à F. Supposons que F sache effectuer la tâche T et qu'il reçoive la première demande de B. Il envoie à A son acceptation de faire T et à B un message d'acquiescement. Lorsque F reçoit une nouvelle demande de C, il répond uniquement par un message d'acquiescement. Si plusieurs messages d'acceptation arrivent à A, il ne prend en compte que le premier en marquant que la tâche qu'il a demandée est en contrat, c'est-à-dire que F s'est engagé à l'accomplir. Les actes de langage nécessaires aux communications entre agents sont les suivants:

Demander(P,X,Y) est une demande du client X au fournisseur potentiel pour que la tâche P soit réalisée, cette demande provenant du demandeur Y.

Proposer(T,Y) est la réponse positive du fournisseur Y à la demande du client. Il indique qu'il réserve T et attend soit un message d'accord, soit un message de refus.

Acquitter(T,Y) indique soit que l'agent Y accepte de faire P, soit qu'il a reçu un message d'acquiescement de tous les agents à qui il a demandé de faire T.

OffreAcceptée(T,X) est une réponse positive du client X à la proposition d'acceptation du fournisseur. Le fournisseur peut alors s'engager à faire P pour X.

OffreRefusée(T,X) est une réponse négative du client X à la proposition d'acceptation du fournisseur, lequel peut retirer P de ses engagements en réserve.

A ces messages, il faut ajouter comme toujours les connexions avec les systèmes organisationnel et motivationnel de l'agent. On retrouvera les messages internes entre modules décrits à la section précédente, à savoir:

Allouer(T) est une demande du système motivationnel pour essayer de trouver quelqu'un à qui allouer la tâche T.

ImpossibleFaire(T) est une réponse au système motivationnel indiquant que la tâche T n'a pas pu être allouée.

EngagéFaire(T,Y,X) est un engagement de l'agent X à accomplir la tâche T pour le compte de Y. Chaque fois qu'il y a entente entre un client et un fournisseur, les deux agents mémorisent cet engagement de manière symétrique.

Le mécanisme d'allocation par délégation peut s'écrire sous la forme d'un ensemble de quatre modules (fig. 7.9) qui gèrent aussi bien la partie client que fournisseur de chaque agent:

1. Un module d'*évaluation des demandes* qui soit propose une offre, soit transmet la demande au module de délégation. Si l'agent a déjà reçu une demande, il se contente d'envoyer un message d'acquiescement à l'agent émetteur de la demande.
2. Un module de *délégation* qui se charge d'envoyer un message de demande à toutes ses accointances et de recevoir tous les acquiescements. Ce module est activé soit par une commande du module précédent, soit par une demande d'allocation de tâche provenant du système motivationnel de l'agent.
3. Un module d'*évaluation des propositions*, qui se charge simplement d'accepter la première offre reçue d'un fournisseur et de refuser toutes les autres.
4. Un module de *réception des décisions* qui engage l'agent à accomplir une tâche si l'offre a été acceptée et à mettre à la poubelle les offres en attentes refusées.

Le module de délégation (fig. 7.10) se contente d'envoyer le message **DemanderF** à toutes ses accointances et à attendre leur acquiescement. Il compte les acquiescements reçus grâce à la place P_1 . Lorsqu'il a reçu autant d'acquiescements qu'il a envoyé de demandes, il renvoie lui aussi un message d'acquiescement sauf s'il est le client, c'est-à-dire l'agent initiateur de la diffusion. Dans ce cas, il renvoie le message **Fin** au module d'évaluation des propositions.

Le module d'évaluation des demandes (fig. 7.11) se charge de traiter une demande. S'il peut (et veut bien) faire la tâche T, il propose ses services et se met en attente. Sinon il délègue cette demande à toutes ses accointances, sauf s'il a déjà reçu une demande semblable, auquel cas, il renvoie simplement un message d'acquiescement. La place P_2 sert simplement à mémoriser le fait qu'il a déjà répondu à une demande d'effectuer T pour un client X.

Les deux derniers modules traitent respectivement de la réception des décisions provenant du client (fig. 7.12.a) et de l'évaluation des propositions (fig. 7.12.b).

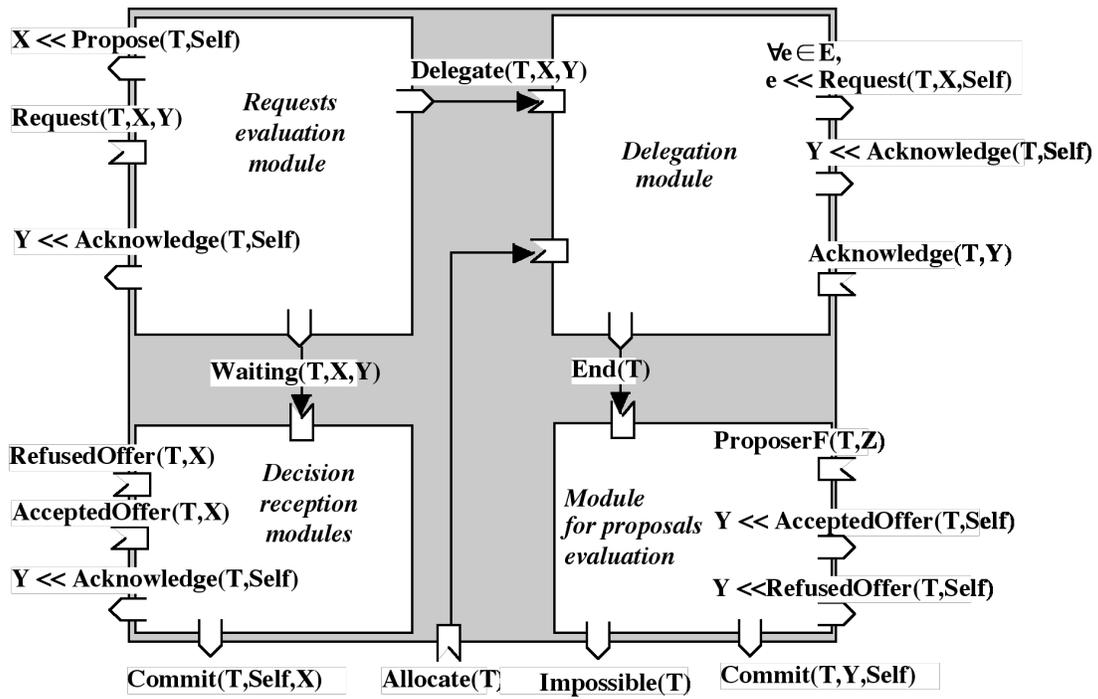


Figure 7.9: Le système d'allocation par délégation décomposé en quatre modules

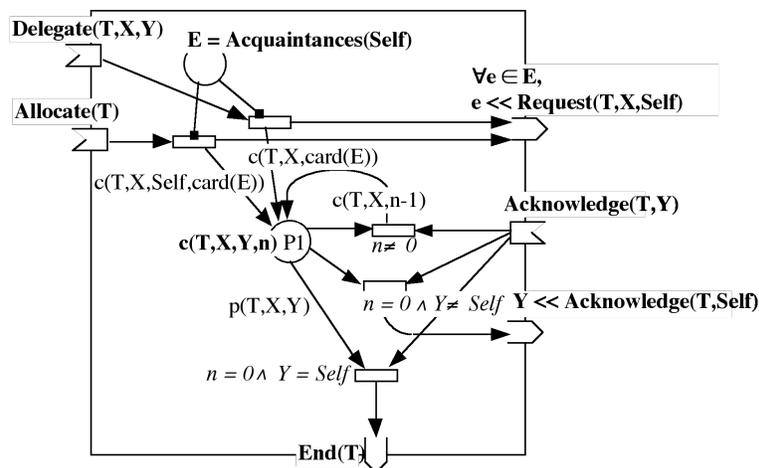


Figure 7.10: Le module de délégation

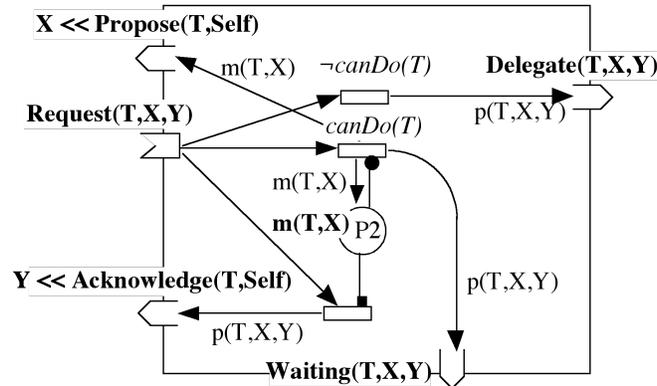


Figure 7.11: Le module d'évaluation des demandes

Ce dernier, lorsqu'il reçoit une offre, indique au système organisationnel que le fournisseur va bien s'engager, mémorise l'offre à l'aide de la place P_5 et refuse toutes les autres. Le premier reçoit ces acceptations ou refus et envoie à l'agent demandeur un acquittement pour lui signaler qu'il a bien reçu une demande. Enfin, si l'offre est acceptée, il s'engage auprès de son client. La place P_4 mémorise les propositions en attente d'acceptation et de refus, et la place P_3 , qui sert de puits, consomme les marques si l'offre est refusée.

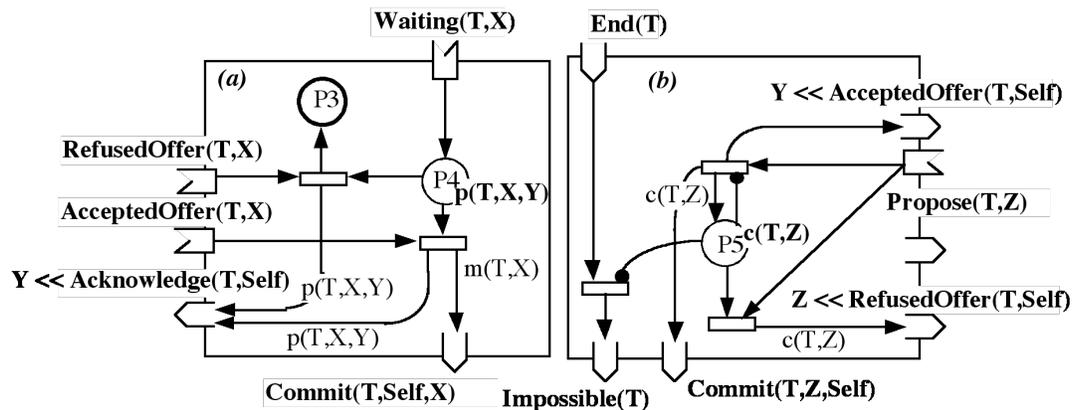


Figure 7.12: Les modules de réception des décisions (a) et d'évaluation des propositions (b)

Cet algorithme présente néanmoins quelques imperfections.

1. En premier lieu, on n'a pas tenu compte des informations portant sur les compétences des accointances pour optimiser le parcours. Chaque agent envoie les demandes à toutes ses accointances, alors qu'il pourrait envoyer la demande en priorité aux agents qu'il sait disposer de cette compétence. Mais, cette optimisation, qui paraît bien anodine cache cependant plusieurs petites difficultés. Il faudra d'abord que les demandes soient différenciées: on fera d'abord une requête directe aux seules accointances disposant de la compétence pour leur

demander si elles acceptent d'accomplir P, que l'on fera suivre par une demande générale qui se bornera à la délégation des messages. Pour que cet algorithme fonctionne, il faut être sûr que les représentations des agents soient complètes, c'est-à-dire que si un agent B fait partie des accointances de A, alors A sait effectivement quelles sont les compétences de B.

2. Lorsqu'un agent indique qu'il accepte une tâche, il serait tentant de vouloir arrêter l'ensemble du processus et de dire à toutes les autres branches qui sont en train de parcourir le réseau qu'il n'est plus nécessaire de chercher à obtenir un candidat. Il suffit alors que l'agent qui accepte envoie un message `AcquitterSuccès` vers le demandeur et que ce demandeur envoie un message `Arrêter` à tous les agents à qui il a envoyé une demande et qui n'ont pas encore acquitté une réponse. Cette modification, qui peut augmenter le nombre de messages à envoyer entre les agents, n'est efficace que si les messages `Arrêter` sont plus rapides et plus prioritaires que les autres. Autrement, cela ne fera que compliquer inutilement le mécanisme.
3. Cet algorithme, comme les précédents, ne prend pas du tout en compte les différents niveaux de compétences des agents, en ne considérant que la première acceptation. L'agent ne demande pas de "devis" à partir desquels il serait capable d'exprimer un choix. L'introduction d'un tel mécanisme est cependant relativement facile. Il suffit de mémoriser la liste des agents qui acceptent de faire le travail, de ne passer un contrat qu'avec le "mieux disant" et d'indiquer aux autres qu'ils ne sont pas acceptés. Cette technique s'apparente alors à celles des appels d'offres que nous verrons à la section suivante.
4. Enfin, il faut faire attention à la manière de libeller une demande de tâche. Si plusieurs demandes correspondant à des tâches identiques sont envoyées les unes derrière les autres, elles risquent de se mélanger, un agent croyant avoir déjà répondu à la demande de cette tâche alors qu'il s'agit en fait d'une nouvelle. En effet, la place qui se trouve dans le module d'évaluation des demandes conserve systématiquement toutes les demandes qu'elle a reçu. Pour éviter ces confusions, il suffit de libeller les demandes de tâches en concaténant la description de la tâche, l'agent demandeur initial et un numéro d'ordre, les demandes étant alors toutes systématiquement différentes.

Réorganisation d'un réseau d'accointances

Les accointances d'un agent peuvent être considérées comme une sorte de mémoire tampon (mémoire "cache") par lesquelles un agent est à même de connaître directement les caractéristiques des autres agents dont il peut avoir besoin. De ce fait, on retrouve dans les systèmes d'accointances les problèmes classiques des mémoires tampon et, en particulier, la difficulté à maintenir une cohérence entre les accointances et l'état des agents qu'elles représentent, notamment lorsque des agents changent de spécificité, quand un nouvel agent apparaît dans le système, ou lorsque l'un d'entre eux vient à le quitter. Dans ce cas il est nécessaire de mettre à jour la base des

accointances des agents de telle manière qu'elle redevienne cohérente, ce qui pose le difficile problème de la réorganisation d'un réseau d'accointances.

Par exemple, supposons qu'un agent A sache qu'un agent B dispose des compétences nécessaires pour accomplir une tâche T et que, pour des raisons quelconques (un module de B tombe en panne, B s'est spécialisé dans une autre tâche...), B voit ses compétences modifiées et n'ait plus la capacité de faire T. La prochaine fois que A demandera à B de faire T, il tombera sur un refus, alors que jusqu'à maintenant il tenait pour acquis le fait que B fasse T chaque fois qu'il lui demandait. Il est possible de résoudre ce problème de cohérence distribuée de deux manières:

- Soit en faisant en sorte que B transmette son changement d'état à tous les agents qui connaissent B. Mais cela soulève deux difficultés: (1) il faudrait que B sache que A connaît les capacités de B, ce qui nécessite une gestion de relations bilatérales entre toutes les accointances et (2) il faut faire attention aux problèmes de parallélisme qui peuvent faire en sorte qu'une autre demande parvienne à B avant qu'il n'ait eu le temps de prévenir tous les agents dont il est une accointance.
- Soit en modifiant la table d'accointance de A. Lorsque A demande à B de faire T et que cette requête échoue, alors A mémorise le fait que B ne sait plus faire T et relance le processus de recherche d'agent.

Il existe évidemment un grand nombre de variantes possibles du réseau d'accointances. Par exemple, on peut vouloir réorganiser le réseau de manière dynamique afin que les agents qui ne se connaissaient pas initialement soient mis en contact. On peut aussi améliorer les performances du système en gérant de manière statistique les accointances des agents de manière à optimiser les requêtes. Les agents avec lesquels un agent est souvent en contact se trouveront dans sa table d'accointances, alors qu'il laissera tomber les accointances qui ne sont jamais utilisées. Il est alors possible de faire en sorte que le réseau "apprenne" à allouer les tâches qu'il doit accomplir au fur et à mesure de son fonctionnement. On donne des "bonnes" notes aux agents qui donnent satisfaction et de "mauvaises" à ceux qui répondent moins bien aux demandes. Le système CASCADE réalisé par H.V.D. Parunak (Parunak 1990) qui contrôle un système de gestion de manutention fonctionne sur ce principe.

Que faire lorsque de nouveaux agents sont mis en service dans un univers multi-agent? De nombreuses techniques sont disponibles et presque toutes proviennent de la technologie des réseaux (par exemple le réseau Internet) ou de celle des réseaux distribués. Avec la technique du "check-in" notamment, les nouveaux agents se présentent à un "administrateur", mais cela nécessite que tous les agents soient connus par l'administrateur. On peut éliminer presque totalement cette difficulté en définissant un système hiérarchique d'administrateur, comme dans les réseaux d'ordinateur, chaque administrateur sachant contrôler sa région et son ensemble d'agents. On peut aussi faire en sorte que les nouveaux agents ne s'adressent qu'à quelques agents du système, puis, par l'utilisation du réseau d'accointances (et en particulier par son système de réorganisation d'accointances s'il en possède un) l'information se propage le long du réseau. Si le graphe des accointances est bien

fortement connexe, tous les agents du système peuvent bénéficier des caractéristiques des nouveaux entrants.

7.3.2 Allocation par appel d'offre: le réseau contractuel

Le réseau contractuel est un mécanisme d'allocation de tâche fondé sur la notion d'appel d'offre. Introduit par R.G. Smith (Smith 1979), cette technique a reçu un accueil enthousiaste par la communauté des chercheurs en intelligence artificielle distribuée et de nombreux systèmes l'ont utilisée. C'est en effet une structure de contrôle très simple à comprendre et à utiliser. Nous verrons, cependant, que sa simplicité cache quelques difficultés qu'il faut résoudre pour employer le réseau contractuel dans un contexte réellement distribué. Le réseau contractuel est l'une de ces techniques de l'informatique où l'aspect conceptuel s'associe à l'implémentation pour présenter des mécanismes utilisables, faciles à appréhender et à programmer, dont les résultats correspondent relativement bien aux attentes.

Le réseau contractuel reprend le protocole de l'élaboration de contrats dans les marchés publics. La relation entre le client, ou *administrateur* (manager), et les fournisseurs, ou *offrants* (bidders), passe par l'intermédiaire d'un appel d'offre et d'une évaluation des propositions envoyées par les fournisseurs. L'appel d'offre s'effectue en quatre étapes:

1. La première est dévolue à l'appel d'offre proprement dit. L'administrateur envoie une description de la tâche qu'il voudrait voir effectuée à tous ceux qu'il estime pouvoir répondre ou à tous les agents du système (fig. 7.13.a).
2. A partir de cette description, dans une deuxième étape, les offrants élaborent une proposition qu'ils envoient à l'administrateur (fig. 7.13.b).
3. L'administrateur reçoit et évalue les propositions, attribue le marché au meilleur offrant au cours de la troisième étape (fig. 7.13.c).
4. Enfin, dans la dernière et quatrième étape, l'offrant, qui a reçu le marché et qui devient ainsi le *contractant* (contractor), envoie un message à l'administrateur lui indiquant qu'il est toujours d'accord pour accomplir la tâche requise (fig. 7.13.d) et qu'il s'engage de ce fait à la réaliser ou bien qu'il ne peut s'acquitter du contrat, ce qui relance l'évaluation des offres et l'attribution du marché à un autre agent (et on revient à l'étape 3).

Algorithme avec un seul administrateur

On analysera dans un premier temps les protocoles du réseau contractuel lorsqu'il n'existe qu'un seul administrateur à un instant donné et l'on verra ensuite les problèmes de généralisation à un nombre quelconque d'administrateurs simultanés. Les types de messages nécessaires à la réalisation de ce système d'allocation sont les suivants:

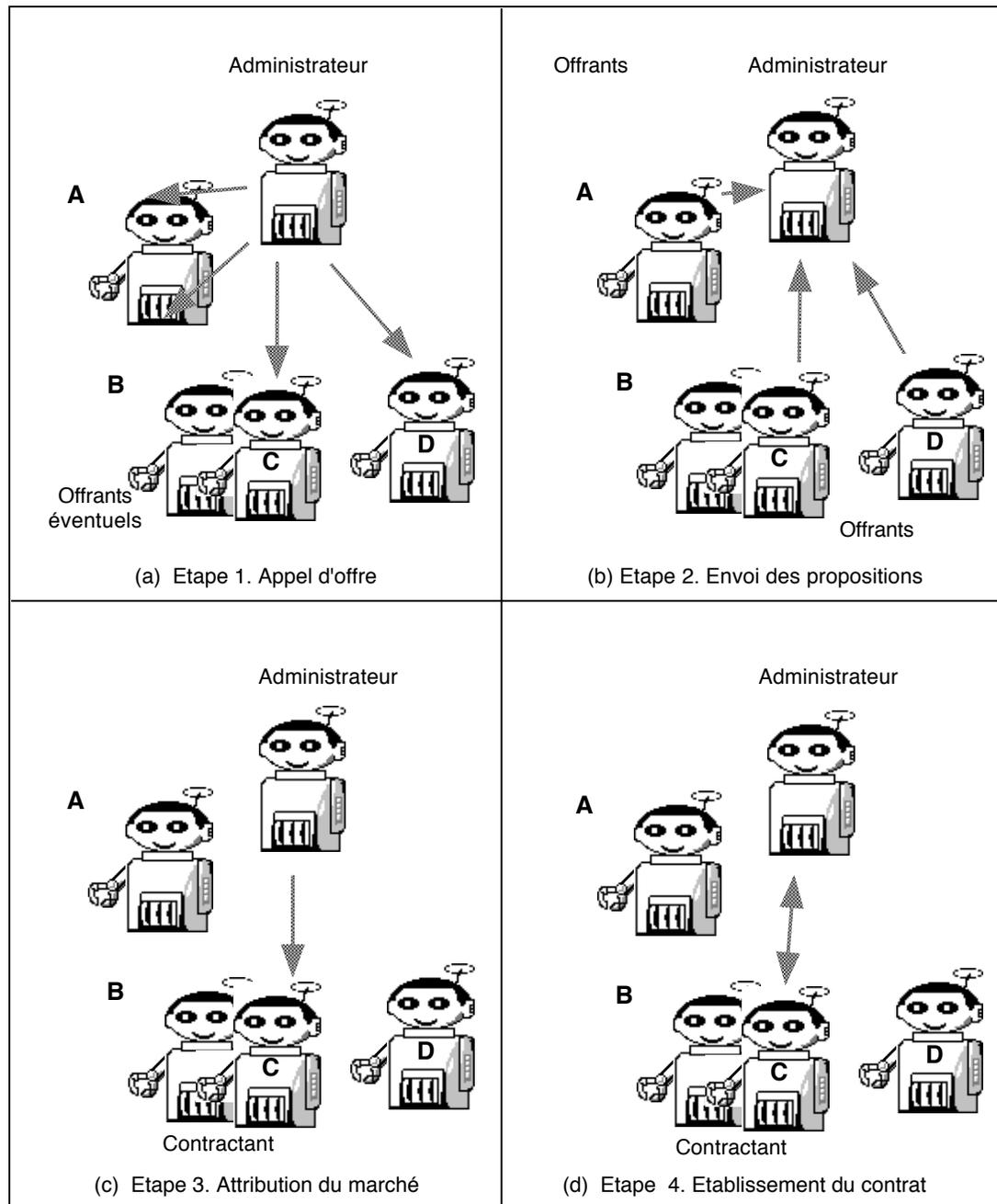


Figure 7.13: Les quatre étapes du réseau contractuel

AppelOffre(T,X) est un message de l'administrateur X à un offrant potentiel pour lui demander d'envoyer une proposition pour la réalisation de la tâche T s'il est intéressé.

Proposition(T,O) est la réponse positive d'un offrant Y à un appel d'offre concernant la tâche T. Il retourne alors une offre O, qui comprend entre autres informations le nom de l'offrant.

PasIntéressé(T,Y) est la réponse négative d'un offrant à l'administrateur indiquant à ce dernier que Y n'est pas intéressé pour accomplir T.

Attribuer(T,C,X) est le message de l'administrateur X à un offrant pour lui signifier qu'il lui attribue le contrat C portant sur la tâche T.

Accepter(T,Y) est une réponse positive de l'offrant Y à l'attribution du contrat par l'administrateur.

Refuser(T,Y) est une réponse négative de l'offrant Y à l'attribution du contrat par l'administrateur, ce qui relance l'évaluation et le processus d'attribution.

De plus, comme dans les autres systèmes d'allocation distribués vus précédemment, l'administrateur (qui est aussi le client), présente les connexions classiques avec son système motivationnel et organisationnel: la demande d'allocation **Allouer**, les réponses (**ImpossibleFaire**, **EngagéFaire**) et l'engagement de l'offrant (**EngagéFaire**).

Un agent peut être à la fois administrateur et offrant. Il s'agit de deux rôles différents qu'un même agent peut assumer indifféremment. Cependant, pour être plus clair, j'ai préféré décomposer cet algorithme en distinguant la définition du comportement de l'administrateur de celui de l'offrant. Les appels d'offre seront reconnus dans le réseau de manière unique par une clé formée de la description de la tâche, de l'administrateur responsable de la diffusion de l'appel et d'un numéro d'ordre local à ce dernier. Par soucis de simplification, on confondra la tâche et le numéro de l'appel d'offre. La figure 7.14 présente l'architecture générale des modules nécessaires à l'établissement d'un protocole de réseau contractuel.

L'administrateur comprend deux modules (fig. 7.15): le module de définition de l'appel d'offre et de réception des propositions (A). Celles-ci sont ensuite sélectionnées par un second module (B) qui se charge d'attribuer les contrats et de recevoir les acceptations ou refus des offrants. Le module A comporte trois places. La première, P_1 , contient simplement l'identificateur de tous les agents du système. La place P_2 sert à compter les réponses à l'appel d'offre pour savoir quand il peut passer à l'étape suivante et la place P_3 mémorise toutes les propositions reçues. Le module B ne comporte qu'une seule place qui est utilisée pour la définition de l'attribution itérative des contrats aux offrants. On a supposé, dans cette implémentation du protocole du réseau contractuel, que les offrants pouvaient refuser le contrat envoyé par l'administrateur. Dans de nombreux cas, on peut supposer que les offrants acceptent toujours ces contrats, ce qui simplifie l'écriture de ce module, puisqu'on peut se passer des réponses d'acceptation et de refus des offrants ainsi que du parcours

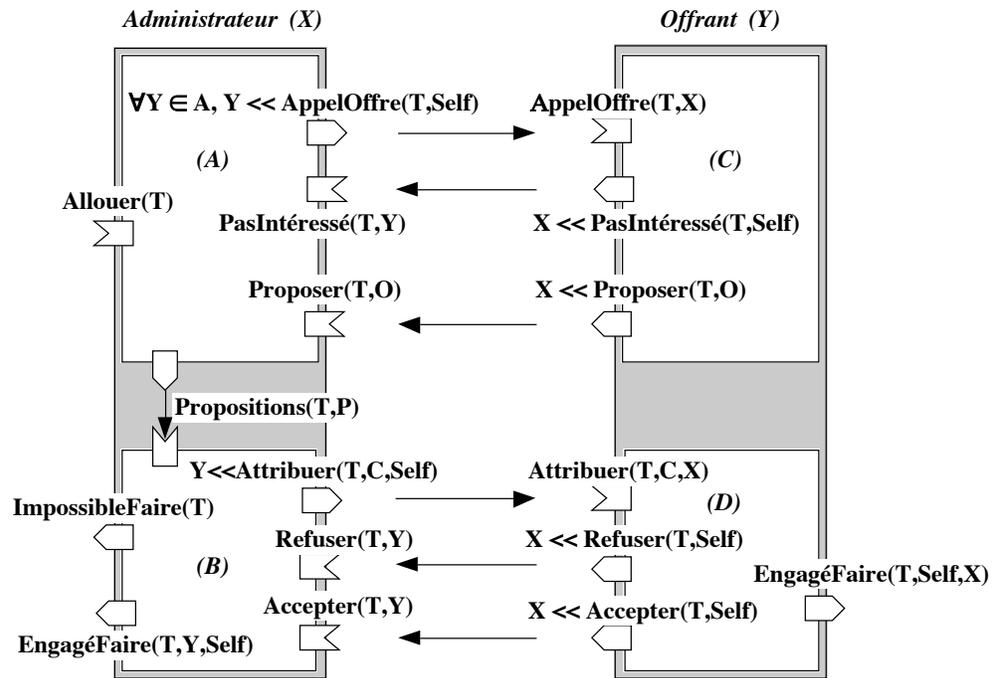


Figure 7.14: Diagramme synoptique d'un protocole de réseau contractuel. L'administrateur se charge de la définition de l'appel d'offre et de la réception des propositions (module A), puis sélectionne les propositions et attribue un contrat au "mieux disant" (module B). L'offrant élabore une proposition en fonction de l'appel d'offre (module C) et décide ou non d'accepter le contrat élaboré par l'administrateur (module D).

itératif de la liste des propositions retenues. Les offrants comportent eux aussi deux modules (fig. 7.16). Le premier (C) est consacré à l'évaluation de l'appel d'offre et à l'élaboration d'une proposition, le second à l'évaluation des contrats attribués par l'administrateur.

Langage d'élaboration de contrats

Les primitives de communication que nous avons présentées ci-dessus sont assez élémentaires, mais suffisantes pour implémenter l'essentiel du réseau contractuel. Cependant, R. Smith propose un langage plus complet qui se situe à un niveau d'abstraction plus élevé (Smith 1980) dans lequel l'annonce de l'appel d'offre comporte, outre la description de la tâche et le numéro du contrat, une *définition des qualités requises* pour la tâche, la *forme de la proposition* ainsi qu'une *date d'expiration*. La définition des qualités requises permet d'éliminer rapidement les agents qui n'ont pas ces qualités sans avoir besoin d'effectuer une analyse minutieuse de la description de la tâche.

La forme de la proposition sert à indiquer à l'offrant quels sont les critères qui seront pris en compte lors de l'évaluation et donc de faciliter ensuite le travail d'évaluation de l'administrateur. Enfin, la date d'expiration donne la date limite après laquelle

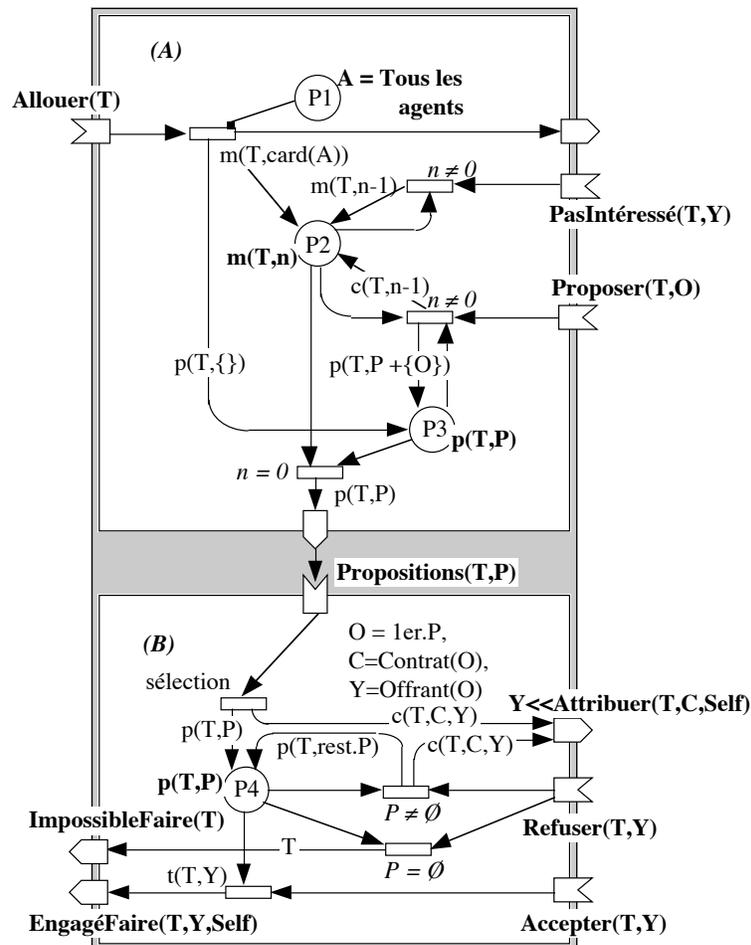


Figure 7.15: Diagramme du comportement de l'administrateur d'un appel d'offre. Le module (A) sert à l'émission des appels d'offres et à la réception des propositions qui sont ensuite transmises au module (B) de sélection des propositions et d'attribution du contrat.

les propositions ne seront plus retenues.

Prenons l'exemple des robots récupérateurs de minerai. Un robot explorateur, ayant trouvé un filon, joue le rôle d'un administrateur et propose à tous les robots capables de creuser dans du sol meuble de forer à l'endroit où il se trouve. Il leur demande leur position, de manière à pouvoir sélectionner le foreur le plus proche de lui, mais aussi la liste de leurs performances en fonction des différents types de sols, en sachant que seuls les robots capables de creuser dans des sols meubles seront sélectionnés. En utilisant la notation de R. Smith pour définir des messages entre administrateurs et offrants, voici la manière dont un appel d'offre peut être diffusé:

```

Message: AppelOffre
A: * // indique un message diffusé
De: Explorateur-25
Contrat: Forer-Explorateur-25-234
DescriptionDeTâche:

```

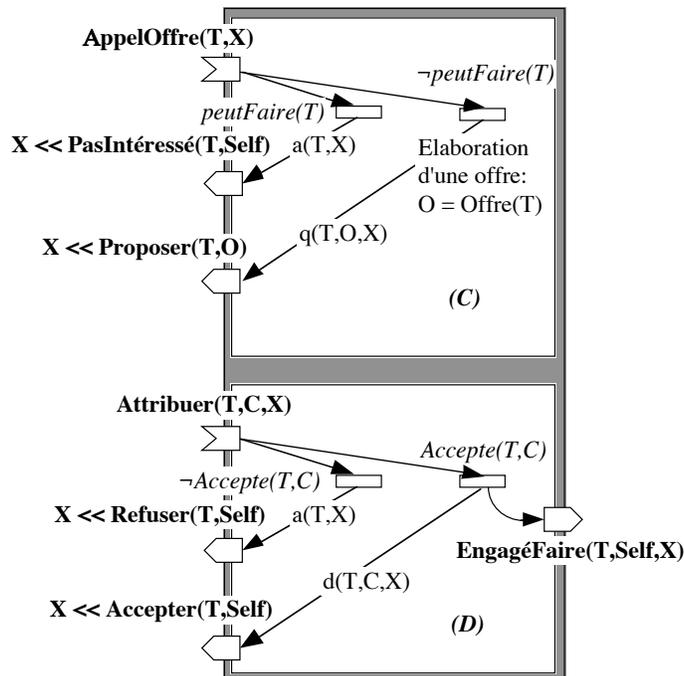


Figure 7.16: Diagramme du comportement d'un offrant. Le module (C) sert à l'évaluation des propositions et l'élaboration d'une offre. Le module (D) se charge de l'acceptation ou non d'un contrat tel qu'il a été élaboré par l'administrateur.

```

TypeDeTâche: Forer
QualitésRequises:
  DoitAvoir : SystemeForage
  DoitPouvoir : ForerSolMeubles
FormeDeProposition:
  Position : <Lat, Long>
  QualitésForage : {TypeDeSol : Performance}
DateExpiration:
  20 Jul 2193 13:06:46
Fin AppelD'offre

```

Les robots explorateurs répondront en donnant dans leurs propositions les caractéristiques demandées par l'administrateur:

```

Message: Proposer
A: Explorateur-25
De: Foreur-18
Contrat: Forer-Explorateur-25-234
DescriptionProposition:
  Position : <47N,17W>
  QualitésForage :
    {Sable: 0,5 Argile: 0,3 Schiste: 0,8}
Fin Proposition

```

et l'explorateur attribuera le "contrat de forage" à ce qu'il estime être la meilleure proposition. La simplicité apparente de l'algorithme présenté ci-dessus ne doit pas masquer un certain nombre de difficultés. Ces points obscurs portent essentiellement sur l'évaluation de la proposition, l'accès aux offrants, la prise en compte d'administrateurs multiples et des engagements personnels, la décomposition des tâches en sous-tâches et la qualité de l'allocation obtenue.

Evaluation de la proposition

Un offrant intéressé par une tâche envoie une proposition contenant une évaluation de sa capacité à réaliser cette tâche. Cette évaluation peut s'exprimer sous la forme d'une simple mesure numérique, ou d'une description comprenant plusieurs critères évalués séparément. Dans l'application de ce mécanisme à la répartition de charges dans un système multiprocesseur, il est possible de mesurer l'intérêt d'un processeur pour une charge sous la forme d'une simple valeur en prenant en compte la charge du processeur. Plus celle-ci est forte, plus l'offre est réduite, et inversement. Dans d'autres situations, il apparaît plus difficile de caractériser l'intérêt d'un agent pour une tâche et de mesurer ses capacités à offrir un "bon" service. Dans ce cas, il appartient au concepteur du système de bien choisir sa fonction d'évaluation et les critères qui seront reconnus comme pertinents par le système, car la performance du réseau contractuel y est fortement liée.

Adressage des offrants

D'une manière générale, le réseau contractuel considère que l'ensemble des agents situés dans le réseau sont accessibles depuis tous les agents du réseau et qu'obtenir leur liste ne constitue pas un problème. En particulier, dans l'algorithme donné ci-dessus, nous avons intentionnellement indiqué que l'administrateur s'adressait à tous les agents. Il s'agit néanmoins d'une conception pour le moins idyllique qui ne traite pas des problèmes de mises en œuvre dans des contextes réellement distribués et qui ne répond pas à la question: comment accéder aux agents susceptibles d'envoyer une proposition?

Des algorithmes de parcours de réseau en parallèle peuvent être utilisés pour accéder à l'ensemble des agents du réseau. Ils ressemblent essentiellement à l'algorithme utilisé pour l'allocation par délégation dans un réseau d'acointances, mais au lieu de demander aux agents s'ils acceptent d'exécuter une tâche on les prie de faire une proposition et d'envoyer cette proposition à l'administrateur. Lorsque le processus est terminé et que l'administrateur a reçu tous les acquittements, qu'ils soient porteurs d'une proposition ou d'une indication de non-intérêt, l'administrateur sait qu'il détient toutes les propositions et qu'il peut commencer son évaluation.

Evidemment ce processus est très coûteux en temps et en quantité de messages, puisqu'il suppose que tous les agents du système doivent être adressés individuellement. Il pose de plus le problème de la constitution d'un réseau d'acointances, même si celui-ci est plus simple que dans le cas d'allocation par délégation, puisqu'on ne demande pas aux agents de connaître les compétences de leurs voisins, mais seulement de savoir qu'ils disposent de voisins. Une autre solution consiste à ce que tous

les agents soient mémorisés dans une structure de donnée globale conservée par un agent unique, mais on retombe alors sur les problèmes de goulets d'étranglement et de sensibilité aux pannes caractéristiques des systèmes centralisés.

La solution la plus généralement adaptée par les concepteurs de réseau contractuel consiste à placer les agents sur un "bus" (fig. 7.17), c'est-à-dire sur une organisation en anneau dont le fonctionnement est très similaire à celui de l'*anneau à jeton* (token ring).

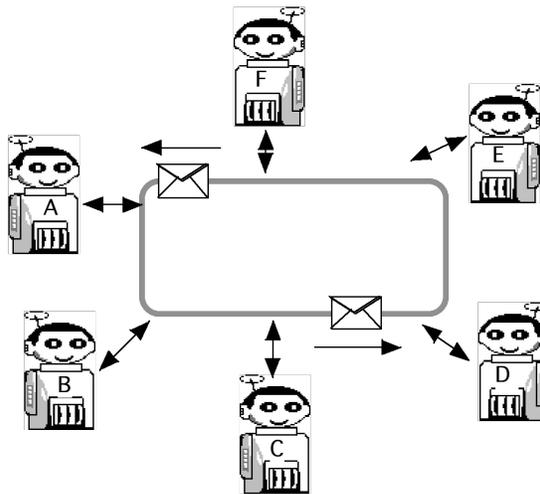


Figure 7.17: Une mise en œuvre du réseau contractuel par implémentation des agents sur un anneau

Les messages sont alors toujours véhiculés dans le même sens et circulent de proche en proche, qu'il s'agisse d'appels d'offre ou de propositions. Une solution consiste alors à considérer que les appels d'offre sont placés sur un plateau. Les offrants lisent l'appel d'offre à tour de rôle, l'évaluent et placent leur proposition sur le plateau. Lorsque le plateau retourne à l'administrateur responsable de cet appel, il sait que tout le monde a répondu et il peut choisir parmi l'ensemble des propositions. Cette mise en œuvre n'est pas non plus très rapide, puisque l'évaluation est faite séquentiellement et que l'on ne tient pas compte du parallélisme potentiel du réseau contractuel pour accélérer le processus. Néanmoins, nous verrons plus loin qu'elle présente l'avantage de pouvoir définir des allocations plus optimales que celles que nous avons données ci-dessus. Afin d'augmenter la vitesse de calcul et limiter le nombre de messages, plusieurs améliorations partielles tenant compte de la nature de l'application ont été apportées. En particulier, lorsque le réseau est peu "volatile", c'est-à-dire que l'identité des agents, leur nombre et leurs compétences varient peu, ce qui est généralement le cas, il est possible d'appliquer des techniques de mémoire tampon qui reviennent à gérer localement une table d'acointance par les administrateurs, et d'indexer les agents par les tâches auxquels ils répondent. Cette technique a été mise au point par Van Parunak pour gérer l'allocation automatique de tâches dans des ateliers flexibles (Parunak 1990).

A chaque tâche est associée un type, et l'on suppose que tous les agents qui savent effectuer les tâches d'un certain type répondent nécessairement à l'offre en

émettant une proposition. Lorsqu'un administrateur doit effectuer un appel d'offre pour un nouveau type de tâche, il diffuse cet appel dans tout le réseau et mémorise dans sa table d'accointance tous les agents qui ont répondu à ce type de tâche. En présence d'une tâche du même type, il leur suffit alors de ne parcourir que cette liste réduite pour obtenir la même efficacité. La table d'accointance joue alors le rôle d'une mémoire tampon qui optimise les accès. Parfois la liste se résume à un seul individu, et l'administrateur peut alors dresser des contrats directement sans passer par un mécanisme d'appel d'offre. Si les agents viennent à entrer ou à quitter le réseau, ou encore à voir leurs compétences modifiées, on suppose qu'ils en informeront le réseau en diffusant l'information à tous les administrateurs. Une telle technique améliore les temps de réponse et le nombre de messages envoyés, mais ses avantages ne sont évidemment perceptibles que si les modifications sont faibles devant le nombre d'appel d'offre.

Date limite

Nous avons jusqu'à maintenant considéré que l'administrateur attend toutes les réponses à son appel d'offre avant d'évaluer les propositions. Cela permet évidemment de mieux tenir compte de l'ensemble des potentialités du système en n'oubliant aucune offre, mais au prix de deux inconvénients majeurs:

1. Tous les agents doivent répondre, même ceux qui ne sont pas a priori intéressés par l'appel d'offre. De ce fait, le réseau risque d'être saturé par une trop grande quantité de messages ne contenant que très peu d'information et portant uniquement la mention "pas intéressé".
2. Un agent en panne ou en mauvais fonctionnement peut totalement bloquer le système. Il suffit qu'il ne puisse pas envoyer une réponse pour que l'administrateur l'attende indéfiniment.

Il suffit pour cela de ne pas demander aux agents de retourner de messages s'ils ne sont pas intéressés, ou évidemment s'ils sont dans l'incapacité de le faire. Chaque appel d'offre comprend en revanche une date limite pour la réception des propositions. Passé cette date, l'administrateur évalue les propositions qu'il a reçu et toutes celles qui arrivent après cette date sont directement rejetées.

Evidemment, cette solution pose d'autres problèmes: il faut bien calibrer ce temps de réaction de manière à ce que la plus grande partie des propositions aient le temps de retourner à l'administrateur. Une date trop proche et de nombreuses propositions ne seront pas prises en compte, une date trop lointaine et l'administrateur atteindra inutilement avant de commencer son évaluation. Il est néanmoins possible d'élaborer un système adaptatif: si trop de demandes arrivent après la date limite, on augmente le temps de réaction et si on attend trop après la réception des dernières réponses on diminue ce temps. De ce fait, si les agents répondent tous avec le même temps de réponse, le système convergera alors vers une valeur bien adaptée.

Administrateurs multiples et optimalité

Nous avons surtout pris en compte le cas d'un administrateur unique, mais il s'agit en fait d'un cas dégénéré de réseau contractuel, car plusieurs administrateurs peuvent travailler en même temps et donc émettent leurs appels d'offre simultanément. Il est donc nécessaire de gérer le fait que plusieurs administrateurs peuvent envoyer au même agent en même temps deux appels d'offre susceptibles d'interférer et d'avoir une influence pernicieuse sur le comportement global du système. Dans le réseau contractuel, les attributions de tâches ne sont généralement prises qu'à partir de connaissances locales des administrateurs sans prendre en compte ni ce que les autres administrateurs demandent, ni les autres propositions qui peuvent arriver par la suite. Van Parunak (Parunak 1987) considère en effet qu'à cause de la localité du processus, les agents subissent trois types d'ignorance: ignorance portant sur les aspects temporels, spatiaux et sur les charges des offrants.

1. *L'ignorance temporelle* provient du fait que les offrants ne connaissent que les appels d'offres déjà reçus et non ceux qui sont sur le point d'arriver. Un agent peut s'engager à effectuer une tâche pour laquelle il n'est en fait pas très bien adapté et ainsi rater des contrats qui correspondraient plus à ses véritables qualités.
2. *L'ignorance spatiale* provient de l'incapacité pour les administrateurs de savoir quels sont les autres appels d'offre en cours et pour les offrants de connaître les autres propositions. Par exemple, supposons que deux administrateurs, A et B envoient à deux offrants potentiels X et Y deux appels d'offre pour deux tâches TA et TB, et que les propositions qui en résultent aient la forme:

	TA	TB
X	90	80
Y	80	20

A la réception de ces propositions, l'administrateur A choisira normalement X, car sa proposition semble plus intéressante que celle de Y et de même B choisira X puisque sa proposition dans ce cas est aussi bien meilleure que celle de Y. De ce fait X va être énormément chargé pendant que Y restera inactif, alors que l'on préférerait attribuer la tâche TB à X et laisser la tâche TA à Y. Si X connaissait la proposition de Y, il pourrait diminuer sa proposition envers TA, et si Y connaissait la proposition de X, il pourrait augmenter sa proposition envers TB. Il existe ainsi des algorithmes dans lesquels tous les offrants envoient leurs propositions à tous les autres agents, ce qui permet d'obtenir des propositions plus nuancées qui tiennent compte de l'ensemble des propositions des agents. Cependant ces techniques sont très coûteuses en temps et en quantités de messages, leur complexité croissant de manière exponentielle avec le nombre d'agents dans le système.

3. *L'ignorance sur les charges* est due à ce que les offrants ne connaissent pas les charges de travail des autres agents. De ce fait, un agent qui travaille peut

être amené à gagner devant un autre dont la charge est presque nulle. Pour remédier à ce problème, Van Parunak propose que les agents disposent des informations portant sur les charges des autres, de manière à modifier leur proposition et, si besoin est, d'être plus "agressif" pour remporter le contrat.

Engagements, réservations et rejets des contrats

Le parallélisme pose un autre problème: les offrants peuvent recevoir de nouveaux appels d'offre entre le moment où ils envoient une proposition et celui où ils reçoivent l'attribution ou le rejet de l'administrateur. L'offrant se trouve alors devant un choix difficile. Il peut soit réserver de la place pour toutes les tâches pour lesquelles il émet une proposition et risquer alors de se trouver sans rien à faire, soit au contraire envoyer de nouvelles propositions sans tenir compte des précédentes, quitte à se retrouver trop chargé et dans l'incapacité de tenir ses engagements. Chacune de ces possibilités présente des avantages et des inconvénients qu'il convient d'analyser au regard du type d'application considéré.

1. En réservant les tâches, un agent considère que la probabilité de se voir attribuer une tâche est grande. De ce fait, il réserve les ressources (en temps et espace) nécessaires à l'exécution de cette tâche. Lorsqu'un nouvel appel d'offre lui parvient, il prend en compte les tâches réservées, comme s'il s'était effectivement engagé à les faire, et envisage le cas échéant d'envoyer des propositions peu intéressantes si les tâches réservées deviennent trop importantes. Il risque cependant de voir ses propositions rejetées et donc de perdre des contrats, s'il réserve trop de tâches et ne devient plus suffisamment "compétitif" dans ses propositions pour acquérir de nouveaux marchés. Le risque dans ce cas de figure est donc de se retrouver sans un nombre suffisant de tâches à accomplir. Cependant, cette méthode n'est pas sans avantage, notamment par la simplicité de sa mise en œuvre. Dès qu'une proposition est émise, l'agent se contente de réserver les ressources nécessaires à la réalisation de cette tâche et calcule ses propositions en tenant compte des tâches réservées comme si elles étaient effectivement engagées. C'est d'ailleurs cet algorithme que nous avons choisi pour la présentation générale du réseau contractuel. On peut qualifier de *prudent* un agent fonctionnant selon un tel principe, car il minimise les risques de surchauffe, mais augmente les risques de chômage technique. Cette technique est particulièrement intéressante si le nombre d'agents émettant une proposition est peu important, et si les risques liés à des surchauffes locales sont peu importantes.

2. Au contraire, en ne tenant pas compte des offres déjà émises, un agent ne réserve aucune ressource pour l'exécution des tâches pour lesquelles il émet une proposition. Il risque alors de se trouver avec de nombreux contrats à acquitter et donc d'avoir trop de travail par rapport à ses capacités. L'intérêt local d'un tel agent est donc d'être toujours relativement plus compétitif puisqu'il ne tient pas compte des ressources qu'il aura effectivement à engager sur d'autres contrats. Cette méthode est évidemment plus avantageuse que la précédente lorsque le nombre d'agents susceptibles d'envoyer une proposition est grand, car la probabilité d'attribution est plus réduite. Afin de résoudre certains conflits portant sur l'attribution des tâches, il est possible de faire en sorte que l'offrant qui se voit attribuer trop de contrats

refuse des tâches dont il a reçu l'attribution. Dans ce cas, l'administrateur doit relancer un appel d'offre ou tout du moins réévaluer les propositions qui lui ont été émises et choisir de signer un contrat avec un autre offrant. Notre algorithme tient d'ailleurs compte de cette éventualité.

3. Il est possible, enfin, de trouver une voie médiane en utilisant les principes de l'aide à la décision en déterminant l'espérance de chaque tâche. Pour cela, il faut associer deux valeurs aux tâches, la probabilité d'attribution et le gain estimé. La probabilité d'attribution correspond à la probabilité de recevoir l'attribution de la réalisation de la tâche. Par exemple, si un agent X estime qu'une proposition est susceptible d'être prise à 20%, il lui associe la probabilité 0,2. De plus, son évaluation lui permet de lui associer une valeur correspondant aux avantages qu'ils peut envisager. A partir de considérations de cet ordre, tout un pan de la recherche dans les systèmes multi-agents porte sur l'emploi des théories micro-économiques fondées sur la notion de marché et d'utilité ainsi que sur la théorie des jeux (Rosenschein & Zlotkin 1994).

L'influence des sous-contractants

Un problème qui n'a jamais été abordé par les auteurs du réseau contractuel, et d'une manière générale par très peu d'auteurs, est celui des sous-contractants, ou plus généralement de l'enrôlement des agents¹. Lorsqu'un offrant X exécute une tâche T sous contrat avec un administrateur A, il peut être amené à décomposer cette tâche en sous-tâches et ne pas être capable de réaliser certaines de ces sous-tâches lui-même. Il doit alors se comporter comme un administrateur et relancer un appel d'offre pour l'ensemble des sous-tâches $\{T_1, \dots, T_n\}$ issues de la tâche T. Un offrant confronté à une telle situation peut n'effectuer cet appel d'offre qu'après avoir reçu le contrat, ou bien demander d'abord aux sous-contractants de s'engager à effectuer ces tâches avant qu'il ne fasse lui-même des propositions à l'administrateur A. Evidemment, il s'agit d'un problème récursif, et les sous-contractants peuvent être amenés à faire appel à des sous-sous-contractants et ainsi de suite.

Prenons l'exemple des déménageurs, dans lequel coexistent deux types d'agents. Les robots "porteurs", qui savent porter des meubles, et les robots "contremaîtres", qui composent les équipes de porteurs, passent les contrats avec les administrateurs, coordonnent les actions entre les porteurs et s'assurent de la qualité du travail. Les contremaîtres jouent ici le rôle des offrants par rapport à la tâche de déménagement et les porteurs celui des sous-contractants dont les tâches sont limitées à celles du port de meubles. Si un administrateur envoie un appel d'offre pour déménager une pièce, les contremaîtres peuvent soit répondre directement à la proposition et ensuite à l'appel d'offre, soit constituer d'abord leurs équipes et répondre ensuite.

1. Dans la première solution que nous appellerons *engagement rapide* (early commitment), il se peut que l'offrant (le contremaître de notre exemple) soit coincé s'il ne trouve aucun sous-contractant libre (ouvrier) capable de l'aider. La figure 7.18 montre le problème auquel peuvent se trouver confrontés deux offrants qui demandent au même sous-contractant. L'offrant B qui vient de recevoir l'attribution

¹A notre connaissance, le seul auteur à avoir parlé explicitement de ces problèmes d'enrôlement dans le cadre du réseau contractuel est T. Bouron (Bouron 1992).

d'un contrat par l'administrateur se trouve incapable de l'exécuter car il n'a aucun sous-contractant pour l'aider, alors que A, qui possède des sous-contractants n'a pas reçu l'attribution.

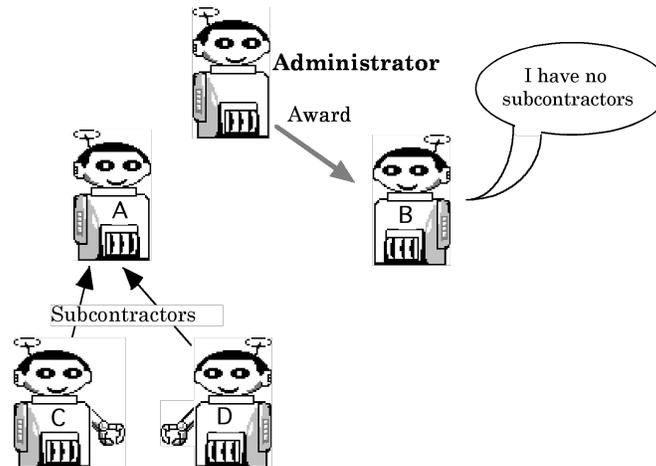


Figure 7.18: Problème de sous-contractance. B n'a pas de sous-contractant alors qu'il vient de recevoir l'attribution du contrat de l'administrateur.

Le problème vient ici de ce que B a émis sa proposition sans savoir s'il disposait de ressources suffisantes pour pouvoir exécuter le contrat. Il se trouve donc bloqué et doit rompre le contrat avec l'administrateur, ce qui entraîne évidemment une forte réorganisation. Ce type d'allocation fonctionne donc bien si les ressources sont abondantes et que le risque de se trouver sans sous-contractant est peu important.

2. La seconde solution semble évidemment être une réponse correcte au problème précédent. Pour éviter de se retrouver sans sous-contractant, il suffit de faire un sous-appel d'offre (c'est-à-dire un appel d'offre pour trouver des sous-contractants) avant de renvoyer une proposition. Si l'on ne trouve pas de sous-contractants, on envoie une proposition peu intéressante. Dans le cas contraire, on peut émettre une proposition en toute quiétude. On dit alors que l'offrant effectue un *engagement tardif* (late commitment) puisqu'il demande d'abord à ses sous-contractants de s'engager envers lui-même avant qu'il ne s'engage à son tour envers l'administrateur. Mais hélas, cette solution pose encore plus de problèmes que la première car il faut d'abord prévoir un mécanisme pour la gestion des engagements des sous-contractants. En effet, si la proposition n'est pas retenue par l'administrateur, il faut dire aux sous-contractants qu'ils sont libérés de leurs engagements. On est alors confronté à des problèmes semblables à ceux que nous avons vus dans la section précédente sur les réserves de tâches, et les solutions pour les résoudre sont analogues. De plus, même en supposant ces problèmes résolus, on se trouve néanmoins devant un problème classique en gestion de ressources des systèmes distribués, l'apparition d'étreintes fatales (dead lock) entre plusieurs processus qui doivent accéder à plusieurs ressources en même temps. Par exemple, s'il n'existe que 6 robots porteurs pour 2 contremaîtres, Berthold et Chilpéric, et qu'ayant reçu l'appel d'offre, la tâche nécessite 4 porteurs. Les deux contremaîtres peuvent se

retrouver avec des propositions de 3 porteurs chacun. Dans ce cas, ni Berthold ni Chilpéric ne peuvent répondre positivement à l'appel d'offre, alors que le nombre d'ouvriers est supérieur aux besoins de la tâche. La solution d'un tel problème passe par l'identification et le dénouement des conflits produisant ces étreintes fatales. Nous traiterons de ce problème à la section 7.3.4 qui traite du problème général de l'engagement des agents.

3. Il existe une troisième solution, qui est loin de fournir un optima, mais qui permet d'assurer une certaine stabilité au système et surtout d'éviter aussi bien les étreintes fatales que les non-exécutions de contrats. Elle consiste simplement dans la constitution d'équipes relativement permanentes de sous-contractants, ce qui revient pour un agent à "embaucher" les agents dont il a généralement besoin et de les lier à sa propre personne par un contrat d'exclusivité. Les sous-contractants ne peuvent répondre qu'aux demandes d'un seul agent, leur "patron", et refuser tout travail qui ne proviendrait pas de cet agent. En échange, le patron s'engage à leur fournir du travail. La constitution de ces équipes est alors plus permanente que dans les situations analysées ci-dessus, les agents faisant partie d'une équipe pour la réalisation de tout un ensemble de tâches. La gestion d'équipe est à la fois plus lourde et plus simple qu'un simple enrôlement pour une tâche précise. Elle est d'abord plus lourde car il faut gérer l'embauche et le départ des agents en essayant de répondre aux demandes avec des équipes déjà constituées. Mais elle est aussi plus légère, car toutes les modifications sont locales à une équipe et n'ont donc pas de conséquences sur les autres équipes. Dans l'exemple des déménageurs, si le nombre d'ouvriers pour la réalisation d'une tâche est relativement constant, il sera plus facile de constituer une équipe de déménageurs plutôt que de gérer à chaque contrat les conflits relatifs à la définition de groupes de sous-contractants. De plus, il est relativement facile d'adapter les embauches aux tâches requises, si le travail est relativement semblable. Il suffit d'embaucher des agents supplémentaires si les besoins s'en font sentir et de débaucher des agents lorsque les travaux nécessitent moins de sous-contractants, en utilisant un système de régulation assez simple.

Ce qu'il faut retenir du réseau contractuel

Le réseau contractuel offre un mécanisme d'allocation de tâche qui apporte une grande souplesse dans la manière de gérer l'accord entre les clients et les fournisseurs. Il ne s'agit cependant pas d'un modèle de résolution de problèmes, mais seulement d'un protocole d'organisation et de répartition dynamique du travail. Comme tous les protocoles, il ne dit rien sur le contenu de ce qui est échangé. En particulier, il suppose qu'il existe un langage commun pour la description des tâches et des informations à transmettre. Si R. Smith a pu nous donner quelques indications sur la manière de constituer un tel langage en s'appuyant sur la notion de description d'attributs (cf. section 7.3.2), d'autres possibilités sont ouvertes, mais restent de toute manière à la charge du concepteur d'un système multi-agent.

Avantages

Les avantages du réseau contractuel sont évidents:

1. Il s'agit d'un mécanisme d'allocation simple à mettre en œuvre, qui peut s'implémenter sur une architecture distribuée, à condition de prendre les précautions que nous avons mentionnées quant à l'accès aux agents, l'influence des administrateurs multiples et la gestion des engagements.
2. Il offre une grande dynamique puisqu'il ne fait absolument pas appel à une quelconque gestion d'acointances. Il suffit qu'un agent soit capable de s'inscrire au réseau (check-in) et de le quitter (check-out) pour qu'il soit possible de prendre facilement en compte l'apparition de nouveaux agents et la disparition d'anciens.
3. La correspondance entre tâche et agent est élaborée à partir d'un accord bilatéral entre client et fournisseur, ce qui permet de tenir compte d'un grand nombre de paramètres tels que les compétences des agents, la charge de travail, le type de la tâche à effectuer, la description des opérations, le type des données à fournir, la durée d'attente maximale autorisée, le coût, l'urgence, etc.

Inconvénients

Malheureusement, le réseau contractuel n'est pas exempt d'inconvénients:

1. Si la mise en œuvre est simple, l'exécution est lourde et produit un grand nombre de messages. En effet le nombre M de messages d'un réseau tel que celui que nous avons donné en BRIC est de l'ordre de:

$$M = N + 2\alpha N = (1 + 2\alpha)N$$

où α représente le rapport entre le nombre d'offrants intéressés par un appel d'offre et le nombre total d'agents du système. On voit que, dès que ce rapport est proche de 0,5, ce qui signifie que pour chaque offre la moitié des agents sont intéressés, le nombre total de messages est de l'ordre de N^2 . De ce fait, le réseau contractuel n'est applicable qu'à de petites sociétés d'agents dans lesquelles seules des tâches de haut niveau sont traitées par appel d'offre, de telle manière que le temps des transactions ne soit pas trop prohibitif devant celui de leur exécution. En effet, qui serait intéressé par un système dont tout le temps de calcul serait pris par des procédures de répartition et qui ne ferait donc rien de productif?

2. Nous avons vu de plus que la simplicité de mise en œuvre n'est qu'apparente et que, si l'on veut bénéficier d'une allocation relativement performance, il faut soit que le nombre d'appels d'offre soit faible, soit que l'on implémente une structure complexe de gestion des problèmes liés au parallélisme du processus. Il est d'ailleurs dommage que très peu d'études aient été faites sur des évaluations de performances de ce type d'allocation et sur les problèmes théoriques et pratiques rencontrés lors d'une mise en œuvre distribuée.
3. Enfin, lorsque les tâches sont redécomposables, il est nécessaire que les agents aient une stratégie de décision qui leur permette de choisir entre un engagement rapide, tardif ou par embauche.

Le réseau contractuel ne peut donc s'appliquer qu'à des réseaux de faible importance exécutant des tâches de forte granularité, et dans ce cas il apporte des avantages considérables en termes de simplicité et de dynamique, comme en témoigne leur application à un contexte industriel. On pourra en particulier se référer à (Parunak 1990) pour une présentation du système YAMS de gestion d'atelier flexible et du Flavor Paint Shop décrit brièvement au chapitre 1.

7.3.3 Variantes et allocations hybrides

Les mécanismes que nous avons développés auparavant correspondent à des approches "pures", qu'il s'agisse de l'approche par système d'accointances ou par réseau contractuel. Il est évidemment possible soit de modifier l'une de ces approches afin d'en définir des variantes, soit de les fusionner pour essayer de cumuler leurs avantages.

Réseau contractuel dirigé par les offres

La forme classique du réseau contractuel est dirigée par les demandes. Ce sont les administrateurs qui demandent aux offrants potentiels de faire des propositions concernant les tâches que les administrateurs voudraient voir exécuter. Comme nous l'avons vu, cette forme d'allocation est plus particulièrement adaptée à des situations de faible charge, de manière à diminuer les conflits ainsi que le grand nombre de messages nécessaires aux transactions.

Il est possible d'inverser ce mécanisme d'appel d'offre en faisant en sorte que ce soient les fournisseurs qui envoient aux clients potentiels leur capacité. Dans ce cas, le système est dirigé par les offres, c'est-à-dire par la disponibilité des offrants. Cette approche, bien qu'étant très utilisée dans le cas de répartition de tâches dans les systèmes d'exploitation distribuée, n'a été que très peu étudiée dans le contexte des systèmes multi-agents. Néanmoins, en faisant en sorte que ce soient les fournisseurs qui informent les clients de leurs compétences et de leur disponibilité, cette technique s'apparente ensuite à des mécanismes de gestion d'accointances directes, puisque l'administrateur connaît alors l'ensemble des agents susceptibles de l'intéresser.

Synthèse du réseau contractuel et des accointances

La gestion des tâches par les accointances ou par un réseau contractuel procèdent de la même problématique: comment faire effectuer une tâche par un agent compétent et faire en sorte que des transformations de l'organisation, soit par ajout ou par suppression d'agent, soit par modification de leurs capacités, puissent être prises en compte sans que le concepteur du système n'ait à dire explicitement qui doit faire quoi.

Dans un réseau d'accointances, les informations caractérisant les compétences ou l'état d'un agent sont directement accessibles par les clients, qui peuvent ainsi se décider sans avoir à diffuser des annonces générales. Au contraire, dans le réseau contractuel pur, on suppose que les agents n'ont aucune connaissance *a priori* sur

les autres agents. Il est donc nécessaire de demander à tous ceux qui sont intéressés de bien vouloir indiquer ce qu'ils savent faire au travers d'une proposition.

Nous avons vu que le réseau d'acointance est nettement plus rapide (surtout dans l'allocation directe) que le réseau contractuel, puisqu'il suffit de demander aux agents que l'on connaît d'effectuer le travail, sans avoir besoin de lancer un appel d'offre. Cependant, le réseau d'acointance manque de dynamisme puisque toute modification passe par une phase de réorganisation du réseau. Il est possible d'essayer de fusionner ces deux approches de plusieurs manières afin d'en tirer le meilleur parti. Il existe de nombreuses manières de combiner acointances et réseau contractuel.

1. La première consiste à utiliser le système d'acointances pour les "petites" tâches et le réseau contractuel pour les "grosses" tâches. Nous avons vu que le problème numéro 1 du réseau contractuel porte sur le nombre de messages ainsi que les temps d'attente éventuels avant que l'administrateur ne puisse évaluer les propositions et attribuer le contrat. Lorsqu'il s'agit de tâches importantes ces temps d'attente sont supportables. Mais il n'en est pas de même pour des tâches élémentaires dont l'exécution ne prend qu'un temps limité. Dans ce cas, on peut utiliser un réseau d'acointances pour toutes les demandes ne nécessitant pas trop de travail (et un défaut de répartition n'est pas trop grave puisque les tâches prennent peu de temps) et réserver les appels d'offres aux tâches plus importantes.
2. La seconde consiste à faire en sorte que l'apprentissage d'un réseau d'acointances passe par des appels d'offre chaque fois que cela est nécessaire. Par exemple, cela peut se produire s'il y a un problème dans le réseau d'acointances et qu'un client ne trouve pas de fournisseur correspondant à ce qu'il désire, ou encore que de nouveaux agents veuillent entrer dans le réseau. On a alors recours à un appel d'offre, les relations d'acointances étant construites et modifiées lors de cet échange.

Enfin, on peut voir aussi un système d'acointances comme une sorte de mémoire tampon du réseau contractuel ou encore comme une structure compilée de ce dernier. Les acointances agissent en effet comme une mémoire cache: leur accès est plus rapide, puisqu'il n'est plus nécessaire de poser la question aux intéressés, mais en contrepartie, il faut pouvoir gérer convenablement les incohérences entre les acointances et les agents, qui peuvent survenir notamment lors des réorganisations du réseau. C'est aussi une version compilée du réseau contractuel qui permet d'éviter l'appel d'offre et les coûts d'envois de messages associés. Dans ce cas, la réorganisation d'un réseau d'acointances peut être comparée à une recompilation d'une structure déjà compilée.

7.3.4 Contrats et engagements

Notion de contrat

Comme l'indique le Code civil, "un contrat est une convention par laquelle une ou plusieurs personnes s'engagent à donner, à faire ou à ne pas faire quelque chose". Cette définition s'applique aussi bien aux contrats entre agents. L'élaboration d'un contrat vise à la définition d'un double engagement. Le fournisseur s'engage auprès d'un client à tout mettre en œuvre pour exécuter la tâche dont il a accepté l'attribution et le client s'engage à ne pas proposer la même tâche à un autre agent (même si parfois cet engagement n'est pas toujours appliqué). Un ensemble d'engagements, comme nous l'avons vu au chapitre 5, forme des structures de dépendances. Le fait que A puisse accomplir son contrat dépend lui-même d'un autre contrat que B a accepté. Cet ensemble de dépendance constitue un réseau qui structure l'action collective et permet à des agents cognitifs d'accomplir des tâches en considérant comme acquis le fait que leurs fournisseurs accompliront bien les tâches pour lesquelles ils se sont engagés.

Fin de contrat

Un contrat passe par une série d'états dont voici les principales:

1. Il se trouve d'abord *en cours d'allocation*. Qu'il s'agisse d'un réseau contractuel ou d'un système d'allocation par accointances, la tâche n'est pas encore attribuée et le contrat n'est donc pas encore conclu.
2. Le fait que le fournisseur accepte d'effectuer une tâche engage celui-ci à l'accomplir, et le contrat se trouve donc *conclu*.
3. Si tout se passe bien, la tâche est exécutée normalement et le fournisseur retourne le fait que la tâche s'est bien terminée, et le contrat se trouve alors dans l'état *exécuté*.
4. Si au contraire, il se passe un événement non prévu au cours de l'exécution du contrat, cela signifie que le fournisseur ne peut faire face à ses engagements, et le contrat se trouve ainsi *rompu*. Dans ce cas, l'agent envoie un message `ImpossibleFaire` à son client.

Que peut faire le client lorsqu'un contrat est rompu? Plusieurs alternatives se présentent à lui, et choisir l'une plutôt que l'autre dépend de ses capacités et de la nature de la tâche T.

- S'il s'agit d'une tâche peu importante, le client de T peut décider de l'abandonner purement et simplement.
- Si la tâche est importante et qu'elle entre dans la réalisation d'une tâche T plus importante, le client de T peut (a) essayer de trouver un autre fournisseur, ou bien (b) dire au client de T qu'il ne pourra pas non plus exécuter T, ce qui risque d'entraîner toute une cascade de ruptures de contrat.

La rupture de contrat peut aussi provoquer des pénalités. Cela signifie en fait que dans le cas d'un réseau d'accointances, le client de T puisse rayer ce fournisseur de ses accointances pour les compétences nécessaires à la réalisation de T. Dans un réseau contractuel, ces pénalités forceront le fournisseur pour que ses propositions futures concernant l'exécution d'une tâche du type de T soit minorées.

7.4 Intégrer les tâches et les états mentaux

7.4.1 Le système SAM

Comment intégrer les états mentaux, tels que les intentions et les engagements, et les algorithmes d'allocation tels que le réseau contractuel ou la gestion des accointances? Afin de répondre à cette question, nous étudierons le cas d'un système SAM (Social Agent Model) développé par Thierry Bouron (Bouron 1992). L'intérêt de son travail est d'avoir montré l'importance des mécanismes de collaboration pour la performance du groupe. Il a appliqué son modèle à l'exemple très classique de la poursuite entre des prédateurs et des proies, dans lequel des prédateurs doivent entourer des proies qui fuient en se déplaçant au hasard, comme nous l'avons présenté au chapitre 1. Toute la difficulté porte sur la conception des prédateurs et, en particulier, sur leur manière de coopérer pour attraper des proies. Il s'agit donc de concevoir différentes stratégies de coopération et de les évaluer par rapport aux performances de capture. En particulier ses stratégies consistent à définir des *associations* éphémères, qui ne durent que le temps de réalisation de la tâche globale.

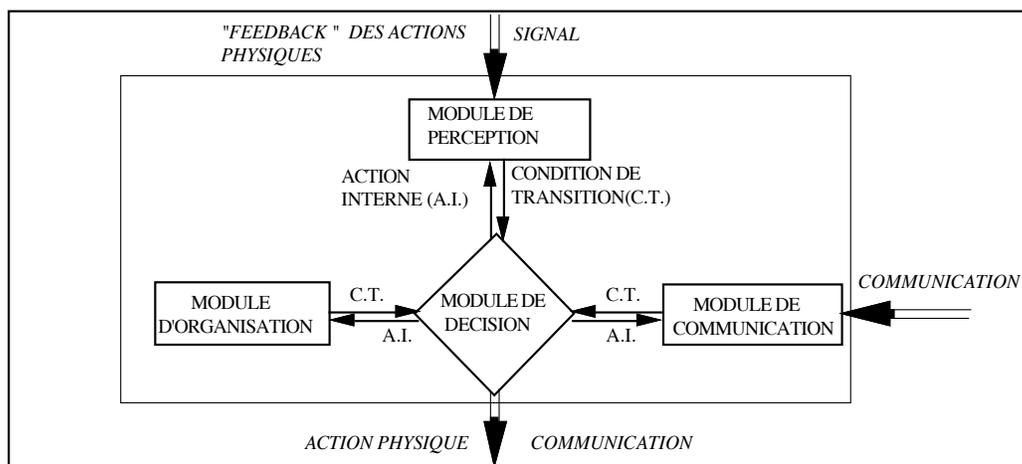


Figure 7.19: L'organisation d'un agent SAM

T. Bouron développe un modèle complet de coopération entre agents fondé sur la notion d'actes de langage et d'engagement et qui intègre des mécanismes de coopération tels que le réseau contractuel. Pour cela les agents s'organisent de manière à exécuter des tâches globales, appelées tâches sociales par T. Bouron, qui ont comme objectif la capture d'une proie. Les agents sont définis à l'aide

d'une architecture modulaire (cf. chap. 3) comprenant 4 modules – perception, organisation, communication et décision – comme le montre la figure 7.19.

Les deux premiers modules traitent les perceptions et les communications. Le troisième, le module d'organisation, gère les engagements tenus par l'agent. Il contient une représentation de la tâche globale à l'origine de ses engagements, une description de la tâche locale attribuée à l'agent dans le contexte de l'association et au moins le nom ou l'adresse d'un autre agent impliqué dans cette association.

Le module de décision est le plus important des quatre, car il détermine le comportement de l'agent, c'est-à-dire les actions physiques et les messages envoyés à d'autres agents. Ce module est représenté sous la forme d'un automate à états finis, dans lequel chaque décision est représentée sous la forme d'une transition qui ne peut être franchie que si l'agent est dans l'état désiré et que toutes les conditions d'application de la décision sont remplies. L'état d'un agent est déterminé par une fonction booléenne portant sur un ensemble de variables d'états appelées *statuts*. Ces variables, comme les conditions et les actions de transitions, peuvent être classées en plusieurs catégories comme le montre la figure 7.20.

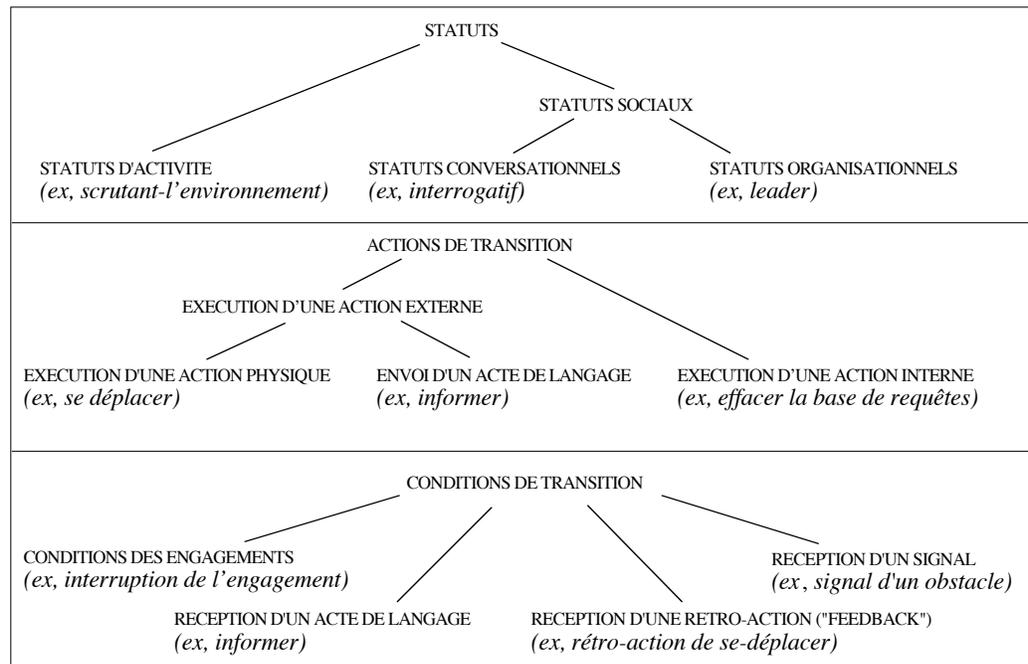


Figure 7.20: Les statuts, actions et conditions de transition d'un agent SAM

7.4.2 La hiérarchie des architectures

Afin d'analyser plus finement l'efficacité de la coopération des agents, T. Bouron a défini une hiérarchie qui distingue entre plusieurs niveaux d'agents (fig. 7.21).

Au premier niveau, les agents sont uniquement mobiles. Ils ne possèdent qu'une capacité de perception et ne peuvent pas communiquer avec les autres. Au niveau suivant, les agents sont capables de se communiquer des informations sur l'état

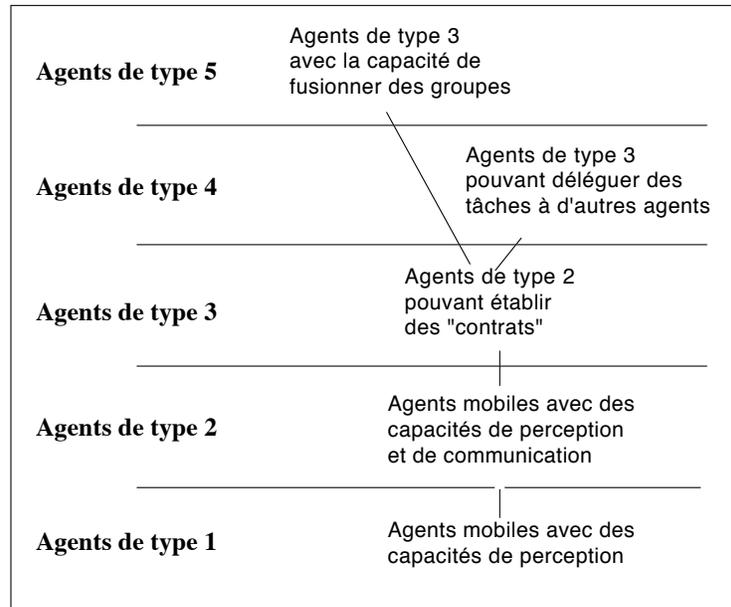


Figure 7.21: Les différents niveaux de comportements

du monde extérieur et donc d'augmenter les capacités de perception des agents. Le niveau 3 introduit la notion de contrat. Un *leader* est capable d'engager des collaborateurs pour former des groupes de prédateurs. Par le système de contrat, il est possible de ne définir que des groupes qui aient la "bonne taille", c'est-à-dire qu'il y ait ni trop ni pas assez d'agents qui s'engagent à chasser une proie particulière. La technique d'élaboration de contrat est fondée sur celle du réseau contractuel, que nous avons présentée précédemment. Au niveau suivant est défini un processus de délégation qui autorise les agents non leader à déléguer leur contrat à un autre agent s'ils découvrent une proie qui n'est pas encore poursuivie. Dans ce cas, l'agent se désengage de son groupe et devient leader d'un nouveau groupe lorsque la délégation de sa tâche courante est achevée. Ce mécanisme permet d'augmenter le nombre de tâches prises en compte en utilisant les agents engagés comme des leaders potentiels de nouveaux groupes. Enfin, au dernier niveau, un processus de regroupement permet à plusieurs petits groupes ne disposant pas d'assez d'agents pour accomplir leur tâche de s'associer et d'unir leurs forces. Ce processus est une réponse au problème de la sous-contractance, décrit à la section 7.3.2, qui laisse des groupes incapables de réaliser la tâche pour laquelle ils se sont engagés.

7.4.3 Les résultats

L'intérêt de cette étude réside non seulement dans la définition de cet ensemble d'architectures emboîtées, mais aussi dans l'ensemble des expérimentations ayant pour but d'analyser la performance de ces systèmes.

La figure 7.22 présente les résultats obtenus avec ces différentes architectures pour une densité d'agents (nombre de prédateurs/nombre de proies) de 2. On peut constater que, pour cette densité, les agents les plus élaborés sont plus performants

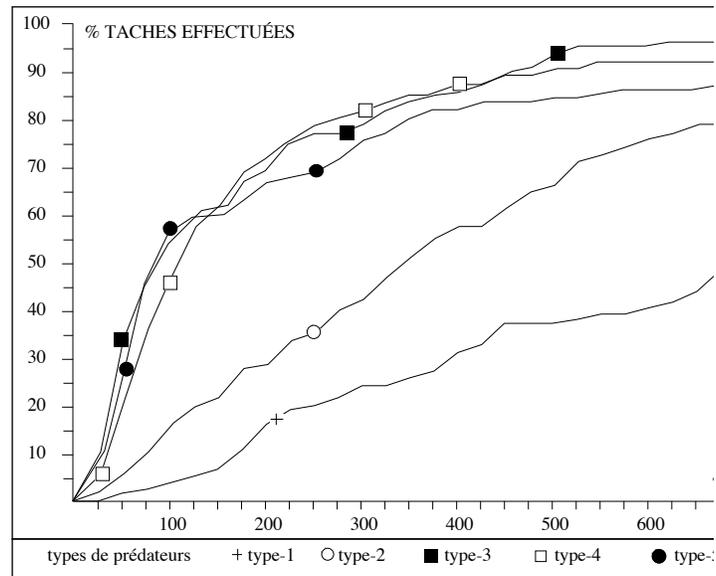


Figure 7.22: Les performances des groupes de prédateurs en fonction des types d'agents SAM

que les agents de type 1 et 2 puisqu'ils peuvent attraper un plus grand nombre de proies en un temps moindre. Il faut néanmoins remarquer que les agents de type 3, qui ne comprennent que des capacités contractuelles sans mécanismes de délégation ni de réorganisation des groupes, sont plus performants ici, car la densité des prédateurs est relativement importante. Pour des densité très faible, les agents de type supérieur s'avèrent plus performants, le fait de pouvoir déléguer des tâches ou de se regrouper constituant alors un intérêt appréciable par rapport à un simple réseau contractuel. On peut noter cependant que le phénomène de regroupement est surtout intéressant si le temps imparti n'est pas trop important (en dessous de 500 unités de temps dans les simulations). En revanche, les phénomènes de délégations, qui ralentissent l'accomplissement des tâches, deviennent plus favorables au-delà par leur capacité à recouvrir une plus grande portion de territoire. Les graphes

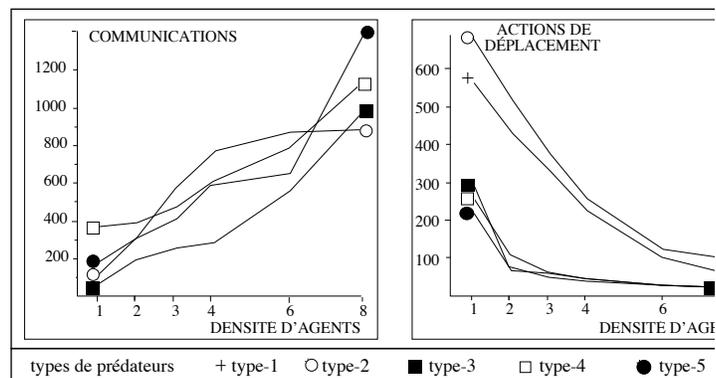


Figure 7.23: Le coût des communications et des actions physiques dans SAM

de la figure 7.23 donnent le nombre moyen d'actions de déplacements exécutées et le nombre de communications envoyées en fonction de la densité des agents. On peut constater que les techniques de collaborations évoluées sont plus coûteuses en communication, mais qu'elles améliorent le nombre de déplacements effectués par les agents, ce qui est une parfaite illustration de l'intérêt de la coopération pour améliorer les performances et diminuer les coûts, si l'on suppose bien sûr qu'il est plus coûteux de se déplacer que de communiquer.

7.4.4 L'implémentation des architectures

Dans le système SAM de T. Bouron, les comportements de ces différents agents sont définis sous la forme d'automates à états finis, les états étant donnés par les fonctions booléennes des statuts. Malheureusement, comme nous l'avions signalé au chapitre 4, les automates à états finis ne sont pas facilement composables et ils ne permettent pas de définir de manière modulable les comportements des agents. De ce fait, la lecture des comportements définis par T. Bouron est assez difficile. Afin d'en simplifier la compréhension, nous en donnerons ici une représentation en termes de BRIC, ce qui permettra de les intégrer aux mécanismes d'allocation vus précédemment. Nous ne présenterons ici que les trois premiers niveaux de SAM.

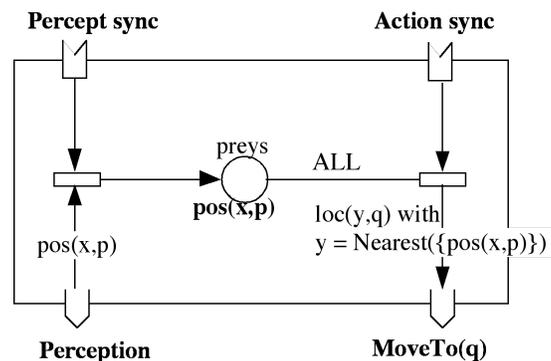


Figure 7.24: Le comportement d'un agent de niveau 1 se contente d'aller vers la proie la plus proche.

7.4.5 Le niveau 1

L'architecture de niveau 1 est extrêmement simple, puisque l'agent se borne à sélectionner la proie qui lui semble la plus proche de toutes celles qu'il perçoit et à aller dans sa direction (fig. 7.24). Les entrées de synchronisation servent à la modélisation des actions. Nous en avons parlé au chapitre 4 concernant la modélisation des architectures d'agents en BRIC.

7.4.6 Le niveau 2

Au niveau 2, les agents communiquent leurs informations à tous les autres agents. Dans la version que nous présentons ici, nous supposons que les agents peuvent s'abonner s'ils veulent recevoir des informations d'autres agents.

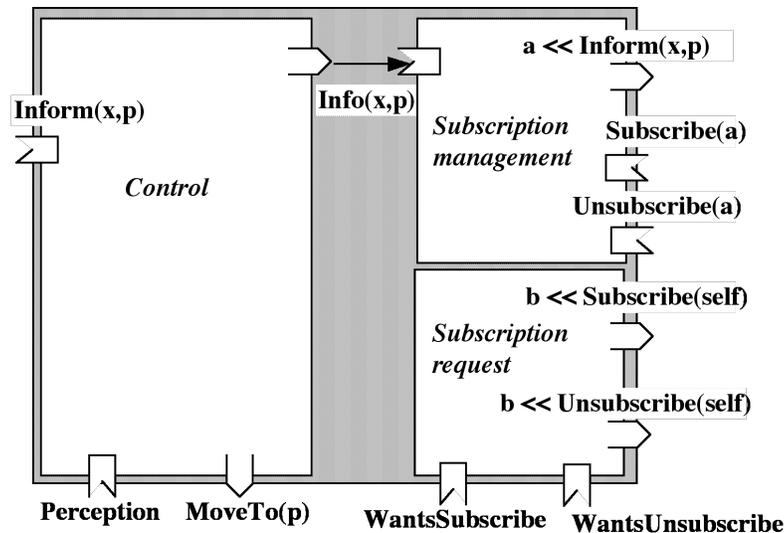


Figure 7.25: Le comportement d'un agent de niveau 2 peut être décomposé en un module de contrôle (qui implémente la fonction conative) et deux modules de gestion des abonnements.

La figure 7.25 présente une architecture de niveau 2 dans laquelle on a omis de faire mention des bornes de synchronisation. Le système organisationnel est décomposé en deux modules: un pour la gestion des abonnements (fig. 7.26.A) et un autre pour la demande d'abonnement (fig. 7.26.B).

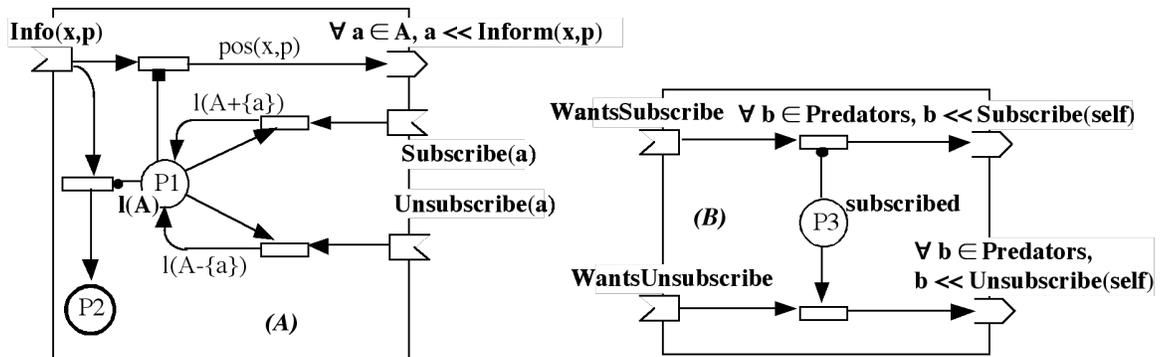


Figure 7.26: Les deux modules de gestion (A) et de demande d'abonnement (B)

Le module de contrôle implémente la fonction conative et se contente de recevoir les informations des agents auprès desquels il est abonné et les percepts (fig. 7.27) et de déclencher les actions correspondantes. L'agent demande un abonnement s'il ne perçoit aucune proie et se désabonne lorsqu'il en perçoit une.

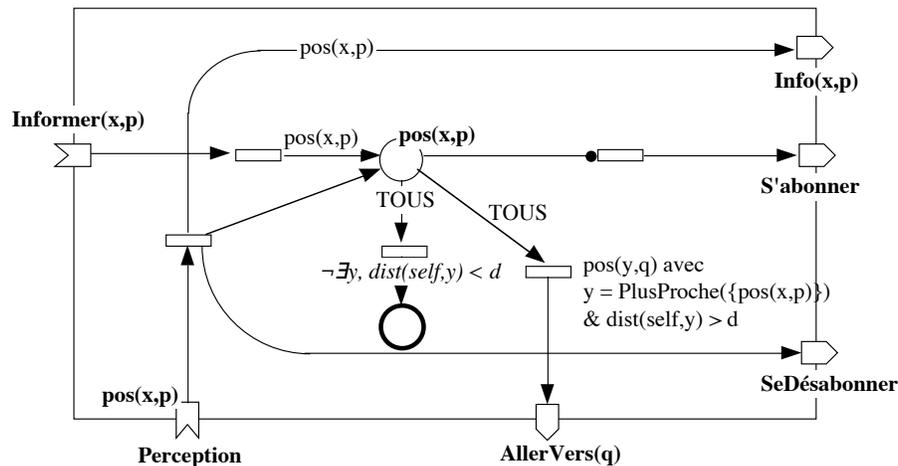


Figure 7.27: Le module de contrôle (système conatif) d'un agent de niveau 2

7.4.7 Le niveau 3

Le niveau 3 fait intervenir la notion d'appel d'offre en plus de la gestion des abonnements aux informations du niveau 2. Si un agent aperçoit une proie, il se dirige vers elle, devient leader et envoie un appel d'offre pour constituer une équipe. Les agents intéressés et qui n'ont pas déjà été recrutés s'engagent dans une telle équipe. C'est le leader qui se charge de donner en permanence la position de la proie et, lorsqu'elle est attrapée, il prévient tous les agents de l'équipe que la tâche est terminée, ce qui les délie de leur engagement. L'architecture d'un agent de niveau 3 est donnée figure 7.28. Il se compose de cinq modules: un module de contrôle qui implémente la fonction conative de l'agent, deux modules pour traiter l'appel d'offre (modules d'administrateur et d'offrant) et deux modules pour gérer respectivement l'équipe et l'engagement auprès d'une équipe.

Ces deux derniers modules sont particulièrement simples (fig. 7.29). Le gestionnaire de l'engagement se contente de mémoriser s'il est engagé ou non, les modules de contrôle et d'offrant pouvant y avoir accès pour savoir si l'agent est engagé ou non.

Le module de gestion de l'équipe transmet la position de la proie en cours de capture à tous les membres de l'équipe ainsi que l'indication de fin de tâche. Dans ce cas, l'équipe est démantelée. La définition de l'équipe provient des modules d'appel d'offre. Ces derniers, qui ne sont pas représentés ici, ne diffèrent des modules que nous avons examinés précédemment que par les points suivants: la sélection des candidats ne conduit pas à recruter un seul prédateur, mais trois pour constituer une équipe; le résultat est donné sous la forme d'un ensemble d'agents qui s'engagent à poursuivre une proie; enfin le test `peutFaire` qui indique si un offrant est intéressé par l'appel d'offre est remplacé ici par un test sur la distance de l'offrant à la proie d'une part et par une requête au module d'engagement pour savoir si l'agent est engagé ou non, car seul un agent "libre" peut s'engager dans une équipe.

Le module de contrôle (fig. 7.30) reçoit les perceptions des proies ainsi que des informations provenant de son leader s'il s'est engagé dans une équipe. Il prend ses

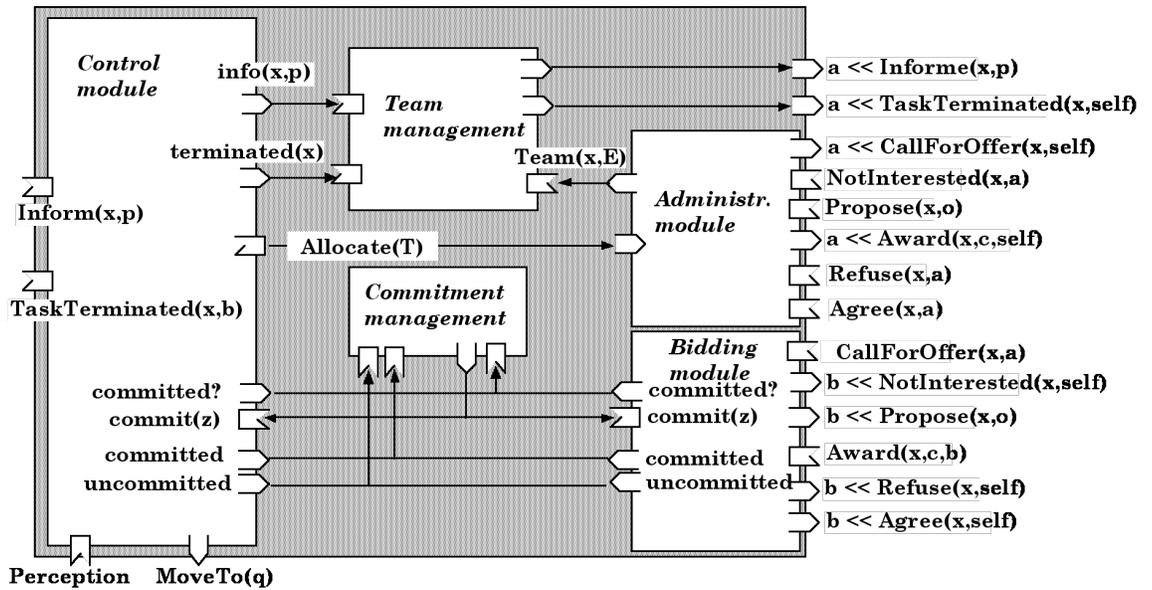


Figure 7.28: Diagramme général d'une architecture d'agent de niveau 3 qui comprend un système d'appel d'offre et une gestion d'engagement dans une équipe

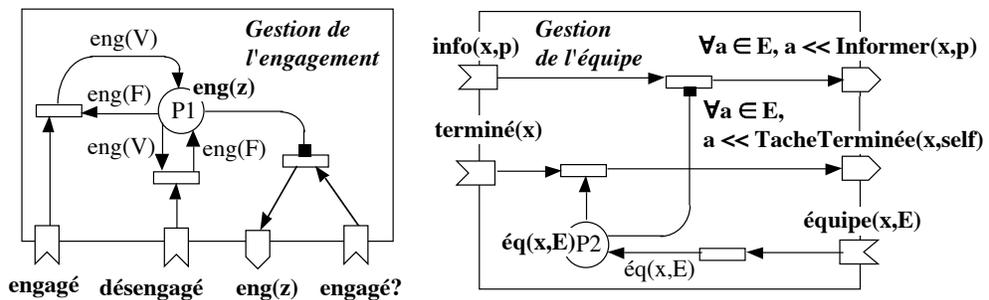


Figure 7.29: Les modules de gestion de l'engagement auprès d'un leader et de gestion d'une équipe

décisions à partir des règles suivantes:

1. Si l'agent est leader (et engagé, on suppose que les leaders s'engagent auprès de leur propre équipe) et s'il perçoit une proie libre, alors il se dirige vers elle et envoie aux membres de son équipe l'information de sa position.
2. Si l'agent n'est pas leader, mais qu'il est engagé et reçoit des informations concernant la position de la proie, alors il se dirige vers cette position, négligeant les perceptions concernant d'autres proies qu'il pourrait apercevoir en chemin.
3. Si l'agent n'est pas engagé et s'il perçoit une proie libre, alors il devient leader et cherche à constituer une équipe en envoyant un appel d'offre.
4. Si l'agent est leader et s'il perçoit une proie capturée, alors il envoie un message indiquant que la tâche est terminée et il se désengage.

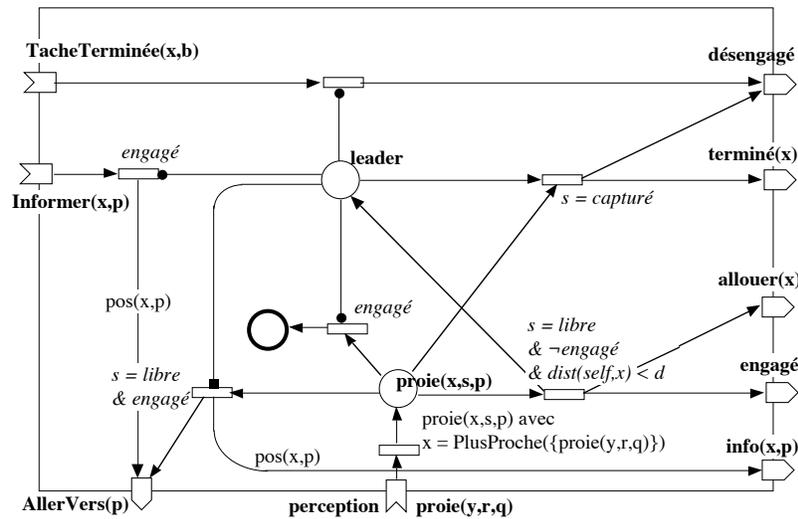


Figure 7.30: Le module de contrôle d'un agent de niveau 3

Ces règles, implémentées dans le module de contrôle BRIC, sont suffisantes pour gérer le comportement des agents prédateurs de niveau 3. Néanmoins, un certain nombre de simplifications ont été glissées dans la définition de ce comportement:

1. On n'a pas pris en compte les désengagements éventuels d'un membre d'une équipe de son propre chef. La seule manière pour une équipe de se dissoudre est de parvenir à capturer la proie afin que le leader s'en aperçoive et dissolve l'équipe ainsi constituée. De ce fait, lorsqu'un agent s'est engagé auprès d'un leader, il ne peut pas revenir sur ses engagements.
2. Un agent offrant (c'est-à-dire non-leader) devient "aveugle" une fois engagé, et les informations qu'il pourrait percevoir ne sont pas disponibles pour les autres prédateurs. C'est pourquoi les agents de niveau supérieur délèguent leurs informations à tous les agents qui ne sont pas engagés pour qu'ils disposent d'un plus grand nombre d'informations et améliorent leur performance de capture.
3. Les leaders, s'ils ne peuvent recruter le bon nombre d'agents, demeurent bloqués jusqu'à ce qu'une équipe soit constituée. Pour supprimer ce blocage, les agents SAM de niveau supérieur peuvent soit dissoudre une équipe insuffisante, soit négocier avec d'autres leaders ne disposant pas d'une équipe complète pour fusionner leurs équipes.

Ce travail permet cependant de comprendre comment des techniques d'appel d'offre peuvent s'incorporer à des mécanismes d'engagement et définir ainsi des agents capables de répartir leur travail et de demander à d'autres de les aider. De plus, on peut constater que la modularité est relativement bien assurée: il serait très facile de modifier l'architecture d'agents de niveau 3 pour qu'ils gèrent un réseau d'accointances. Il suffit de changer les modules d'administrateur et d'offrant pour les remplacer par des modules de client et fournisseur d'un système d'accointances

pour que le comportement global de l'agent soit remanié sans devoir remettre l'architecture générale et les autres modules.

7.5 Allocation émergente

Jusqu'à maintenant nous ne nous sommes intéressés qu'à l'allocation de tâches par agents cognitifs, capables de communiquer intentionnellement entre eux. Cette activité prend la forme de dialogues que s'échangent les agents pour parvenir à ce que le travail soit effectué par un agent compétent.

On peut être surpris de penser qu'il est aussi possible de parler d'allocation de tâches dans le cas d'agents réactifs. En effet, on pourrait supposer que leurs capacités limitées ne leur permettraient pas de mettre en œuvre des mécanismes aussi complexes que les algorithmes que nous venons d'analyser. Et pourtant, il est possible de faire en sorte, d'une part, que les tâches soient distribuées aux agents et, d'autre part, que les agents eux-mêmes se spécialisent dans l'accomplissement de telle ou telle tâche.

L'allocation réactive fait appel à la notion de signal et non à celle de message. Les signaux, comme nous l'avons vu au chapitre 6, sont des formes non intentionnelles de communication, transmises par diffusion et propagation dans l'environnement, le signal décroissant généralement d'intensité avec la distance. Ces signaux sont donc des formes de communication élémentaires pour lesquelles il n'existe pas une sémantique unique: un même signal pourra induire deux comportements différents dans deux agents différents. L'exemple le plus caractéristique est le cri d'un jeune animal: pour ses congénères il pourra déclencher (en fonction de l'espèce) un comportement de protection, alors que pour un prédateur il donnera lieu au contraire à un comportement de poursuite et de chasse.

L'un des intérêts majeurs de ces signaux pour l'allocation de tâches vient de ce que leur intensité décroît avec la distance à la source. En effet l'intensité d'un signal qui se propage instantanément est généralement donnée par l'équation:

$$I(x) = \frac{k}{dist(x, x_0)^2} \quad \text{ou} \quad I(x) = \frac{k}{dist(x, x_0)}$$

où k est une constante qui représente l'intensité maximale du signal à la source, et $dist$ est la fonction de calcul de la distance entre deux points. Bien que la deuxième équation, qui stipule que l'intensité d'un signal décroît linéairement avec la distance, n'ait pas de répondant physique immédiat, elle est souvent utilisée en pratique du fait de sa simplicité de mise en œuvre. La dégradation de l'intensité des signaux en fonction de la distance introduit une différence dans les agents qui perçoivent ce signal. Il y a ceux qui se trouvent proches de la source et pour lesquels le signal sera fort, et ceux qui se trouveront plus éloignés et qui donc ne recevront qu'un signal atténué. Le principe de l'allocation de tâche repose alors sur le déclenchement différencié d'un comportement en fonction de l'intensité du signal qu'un agent reçoit. Par exemple si un agent peut effectuer l'une des deux tâches T_1 ou T_2 et qu'à ces tâches soient associées respectivement les signaux S_1 et S_2 , il exécutera la tâche

pour laquelle le stimulus (calculé comme le produit de l'intensité du signal par un coefficient interne correspondant à la tendance qu'un agent a de faire une tâche) est le plus fort:

$$\begin{aligned} \text{exécuter}(T_1) & \text{ si } S_1 \geq S_2 \\ \text{exécuter}(T_2) & \text{ si } S_2 < S_1 \end{aligned}$$

Lorsque les intensités sont identiques, et pour éviter le célèbre blocage de l'âne de Buridan qui ne pouvait choisir entre deux tas de foin s'il se trouve à égale distance des deux tas, on tire aléatoirement entre les deux actions. On peut aussi déclencher les tâches de manière aléatoire, l'intensité des stimuli S_i définissant la probabilité de déclenchement des tâches T_i . Ce déclenchement différentiel introduit dans les systèmes multi-agents deux formes de compétition.

1. La première correspond à une *compétition intra-agent* qui porte sur l'ensemble des tâches qu'un agent peut accomplir. La prise de décision d'un agent réactif dépend de trois facteurs: l'intensité initiale du signal, la tendance à accomplir telle ou telle tâche, la distance à la source et la capacité de perception de l'agent. Deux signaux qui ont même intensité pourront être différenciés par un même agent en fonction de ses propres tendances. Par exemple, un prédateur ne portera pas le même intérêt à un signal lui indiquant la position d'une proie s'il a le ventre creux ou s'il vient de manger. Ces quatre facteurs se combinent pour donner un stimulus dont l'intensité décide du choix de la tâche à déclencher, la tâche ayant le plus fort stimulus remportant la compétition (cf. section 3.6.5).
2. La seconde concerne la relation respective des agents vis-à-vis de la source du signal. Comme l'intensité relative est plus forte lorsqu'on se trouve près de la source, les agents les plus proches auront plus tendance à accomplir les tâches liées à ce stimulus que les plus éloignés. Cette compétition n'est cependant pas sans poser des problèmes: que se passe-t-il en effet lorsque deux agents décident de répondre à ce stimulus et qu'un seul agent suffit pour répondre à la demande? Dans l'exemple des robots explorateurs, supposons qu'un robot foreur ait besoin de faire transporter son minerai jusqu'à la base. Pour ce faire, il envoie un signal **DemandeTransport** qui se propage et diminue d'intensité avec la distance. Si plusieurs transporteurs reçoivent le signal et décident de se déplacer jusqu'à la source, lequel aura effectivement la possibilité de transporter le minerai? La solution évidente consiste à servir le premier arrivé. Si les transporteurs ont la même capacité de locomotion, le plus proche de la source arrivera normalement le premier et aura donc le bénéfice de l'exécution de la tâche.

Mais tous les problèmes ne sont pas résolus pour autant. Les techniques de compétition n'optimisent pas toujours le comportement des agents. Par exemple, supposons qu'il existe deux zones de forage relativement éloignées l'une de l'autre, et que les deux robots foreurs F_1 et F_2 placés dans ces zones aient besoin d'un transporteur en même temps (fig. 7.31). Si les transporteurs A_1 et A_2 sont situés plus près de F_1 que

de F_2 , ils répondront tous deux favorablement à la demande de F_1 et se dirigeront vers F_1 (1). Mais comme A_1 est plus près de F_1 que A_2 c'est lui qui transportera le minerai de F_1 ; A_2 se sera déplacé pour rien (2). Mais si F_2 a toujours besoin d'un transporteur, A_2 sera maintenant sensible à son appel et se dirigera vers lui (3 et 4).

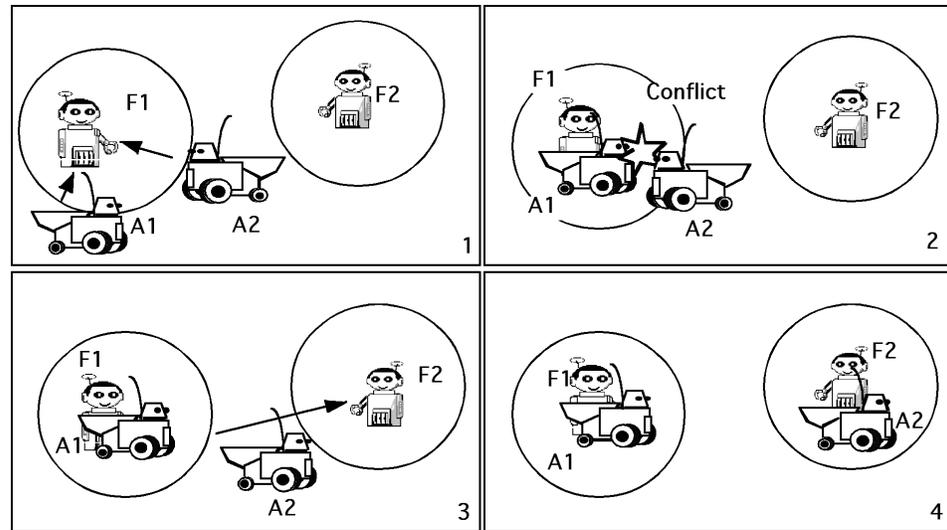


Figure 7.31: Les différentes phases d'un système d'allocation réactive de tâches

On voit que dans cette situation, le transporteur A_2 a d'abord été vers F_1 pour se rendre compte qu'il était déjà aidé par A_1 et qu'il n'avait pas besoin de lui. Il a donc effectué le chemin $A_2 \rightarrow F_1$ puis le parcours $F_1 \rightarrow F_2$, au lieu du simple trajet $A_2 \rightarrow F_2$, ce qui peut s'avérer coûteux si les agents A_1 et A_2 sont suffisamment loin.

Le fait que le trajet soit non optimisé fait malheureusement parti des techniques réactives qui ne tendent généralement pas vers l'optimum, mais qui compensent cette perte par une grande capacité d'adaptabilité. Il est clair en effet que cette technique est totalement indépendante du nombre d'agents mis en œuvre et, mieux encore, plus le nombre d'agent est important plus on a de chance de répondre aux demandes de réalisation de tâches. Que l'on supprime un agent et le système continue à fonctionner comme s'il ne s'était rien passé. Nul besoin de protocoles complexes, ni de vérification de terminaison, ni d'une manière générale de tout l'attirail algorithmique que nous avons examiné précédemment.

Il existe un deuxième problème: les agents réactifs peuvent être piégés dans des zones d'ombre pour des signaux provenant d'autres demandes. Dans ce cas, l'agent devient sourd aux appels d'autres agents. On peut résoudre cette difficulté de plusieurs manières: la première porte sur le nombre d'agents. Plus celui-ci est grand plus la probabilité que certains agents puissent répondre à ces besoins devient importante. On peut aussi éliminer cette difficulté en augmentant continuellement les intensités des sources des signaux tant que les sources n'ont pas reçu de satisfaction. Si l'on peut faire croître l'intensité indéfiniment, toutes les zones d'ombre seront un jour recouvertes et les agents qui s'y trouvent pourront recevoir ces stimuli.

7.6 Un exemple: le système Manta

De manière générale, les techniques réactives sont surtout utilisées pour l'avantage qu'elles confèrent en termes de simplicité et d'adaptabilité à des conditions qui évoluent souvent. En effet, elles assurent une bonne répartition des tâches en fonction des besoins au prix néanmoins d'une certaine redondance de moyens. Le système MANTA de modélisation d'un nid de fourmis en est un bon exemple.

7.6.1 Description générale

Le système MANTA est caractéristique d'une approche réactive dans laquelle l'informatique et les sciences de la vie s'apportent mutuellement de leurs apports. En effet ce projet, qui porte sur la modélisation par systèmes multi-agents d'une colonie de fourmis *Ectatomma ruidum* au cours de son évolution, est né de la rencontre de deux domaines de recherches: l'éthologie et les systèmes multi-agents. Les expérimentations ont porté notamment sur sa sociogenèse, c'est-à-dire sur la constitution d'une colonie mature à partir d'une (ou plusieurs) reine(s), sur l'adaptation d'une colonie à son environnement et, en particulier, sur la division des tâches (polyéthisme) et la spécialisation des ouvrières, et enfin sur la polygynie (Drogoul et al. 1992). Il s'agissait en particulier de tester une hypothèse éthologique concernant l'aspect distribué de la prise de décision dans une colonie de fourmis. Le comportement des individus suffit-il à expliquer la génération et la stabilité des formes sociales observées (division du travail, polyéthisme d'âge, dynamique de la fondation de la société, etc.)? Peut-on montrer qu'une société d'agents peut survivre sans faire appel à un système de contrôle centralisé ni à une quelconque organisation hiérarchique? C'est pour répondre à ce type de questions que le système MANTA fut réalisé. Il mit en évidence que l'organisation du travail dans une colonie de fourmis peut résulter d'un ensemble d'interactions et de contrôles locaux, sans qu'il soit nécessaire de faire intervenir une quelconque entité centrale de régulation, les performances adaptatives de la société étant le résultat des comportements, forcément élémentaires, de chacun de ses membres. Dans un système de simulation multi-agent tel que MANTA, il s'agit de modéliser les différentes entités qui interviennent dans un nid de fourmis sous formes d'agents réactifs sans faire appel à une organisation prédéfinie ni à des spécialisation préétablies. On ne présentera ici que quelques-uns des traits saillants de cette étude. Le lecteur intéressé pourra se référer à (Drogoul 1993) qui présente la problématique, le système et les expériences avec toute la précision nécessaire.

7.6.2 L'architecture du système

Les agents

Une colonie de fourmis peut être divisée en trois groupes d'agents: le groupe des "assistés" qui comprend les œufs, les larves et les cocons, celui des "assistants" qui inclut la reine, les ouvrières et les mâles, et celui des "divers" qui regroupe toutes les autres entités "actives" du nid, avec en particulier la nourriture, les cadavres de fourmis ainsi que les sources de lumière et d'humidité. L'environnement de MANTA

est directement accessible à l'écran et représente un nid de fourmis semblable à ceux qui sont quotidiennement employés dans des laboratoires. La figure 7.32 montre l'écran servant à la définition du nid et à l'observation directe du comportement de la fourmilière. Les boutons de contrôle sur la droite permettent de modifier la topologie du nid, de rajouter des fourmis ou d'autres agents, et de modifier ou d'inspecter l'état de l'environnement et des agents qui y circulent à tout instant. Chaque agent est

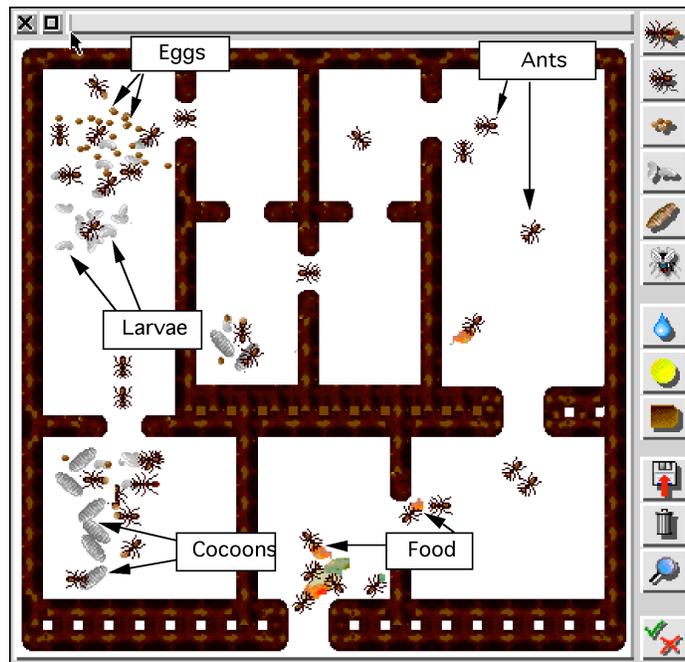


Figure 7.32: Un nid de fourmis tel qu'il est visualisé par MANTA (d'après (Drogoul 1993))

représenté sous la forme d'un petit automate qui possède la capacité de choisir parmi un ensemble de tâches en compétition, chaque tâche s'exprimant sous la forme d'une séquence de *primitives de comportement*, c'est-à-dire d'une suite d'actions atomiques exécutables par l'agent.

A chaque stimulus de l'environnement est associé une tâche et une seule, et un agent ne perçoit que les stimuli associés à ses propres tâches. Les stimuli sont caractérisés par leur *intensité*, c'est-à-dire par le niveau du signal correspondant au stimulus à la position où se trouve l'agent. D'un point de vue éthologique, les tâches représentent des chaînes d'action figées (fixed action patterns). Chaque tâche est caractérisée par le stimulus permettant de la sélectionner, par son *poids* qui précise l'importance relative de la tâche par rapport aux autres, par son *seuil de déclenchement* et son *niveau d'activité*. Le niveau d'activité est donné par la formule:

$$a_i(t) = \frac{w_i(t)}{\sum_{j=1}^n w_j(t)} x_i(t)$$

où w_i est le poids de la tâche, x_i est l'intensité du stimulus déclencheur. Pour qu'une nouvelle tâche soit sélectionnée, il faut que son niveau d'activité soit supérieur au

seuil de déclenchement de la tâche et au niveau d'activité de la tâche courante, c'est-à-dire que $a_i(t) > \sigma_i$ et $a_i(t) > a_c(t)$ où σ_i est le seuil de déclenchement et $a_c(t)$ le niveau d'activité de la tâche courante.

Les tâches sont donc en compétition: si plusieurs tâches sont sélectionnables, le contrôleur choisit celle dont le niveau d'activation est le plus grand. La figure 7.33 montre l'architecture générale d'un tel agent. C'est le mécanisme que nous avons décrit au chapitre 3 sur les architectures à base de tâches compétitives.

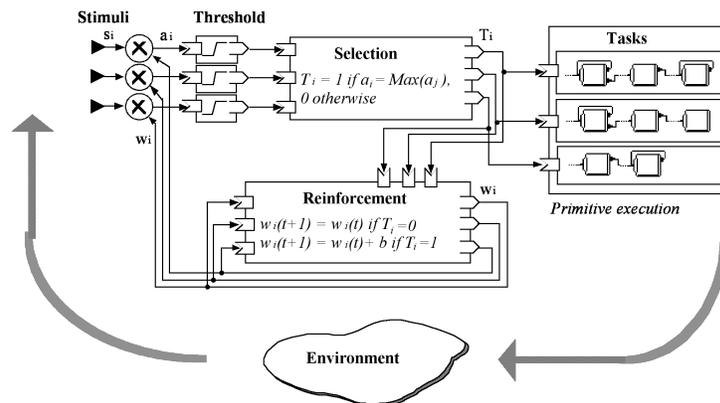


Figure 7.33: Architecture générale d'un agent MANTA

Lorsqu'une tâche est sélectionnée, elle devient la tâche courante, elle exécute son *comportement d'activation* et son niveau d'activité prend la valeur de son niveau d'activation. Au contraire lorsqu'une tâche est désactivée, elle exécute son *comportement de désactivation*. Chaque agent dispose d'une tâche spéciale qui est choisie lorsqu'aucune tâche n'est sélectionnable et lorsque le niveau d'activité de la tâche courante devient nul. Cette tâche, appelée *défaut*, spécifie le comportement par défaut d'un agent lorsque son environnement est peu attractif, c'est-à-dire pauvre en stimuli.

Les processus de rétroaction positive

Chaque agent voit son comportement renforcé. Le poids de chaque tâche est en effet augmenté d'une valeur chaque fois qu'une tâche w_i est sélectionnée:

$$\begin{aligned} w_i(t+1) &= w_i(t) + \delta \text{ si } w_i \text{ est sélectionnée} \\ w_i(t+1) &= w_i(t) \text{ sinon} \end{aligned}$$

Ce mécanisme de renforcement constitue une véritable rétroaction positive exprimée au niveau de chaque agent, en rendant l'agent plus sensible aux stimuli de l'environnement correspondant aux tâches qu'il a déjà exécutées. De ce fait, plus un agent accomplit une tâche, plus il aura tendance à l'accomplir de nouveau.

L'environnement et les communications

Les communications sont effectuées par diffusion de signaux dans l'environnement. Chaque agent dispose d'un ensemble de signaux qu'il peut émettre en fonction

de son état interne. Par exemple, une larve qui a faim va propager un signal de `LarveAffamée` qui pourra être perçu par une ouvrière et éventuellement déclencher sa tâche `nourrirLarve`. Les signaux se propagent dans l'environnement en perdant de l'intensité avec la distance, comme nous l'avons vu ci-dessus.

Les processus de rétroaction négative

Les actions des ouvrières peuvent être considérées comme des actions de régulations visant à maintenir l'homéostasie du nid. En effet, la plupart d'entre elles servent à satisfaire les besoins exprimés par des agents tels que les œufs, les larves ou les cocons qui nécessitent des soins répétés et de la nourriture en quantité (surtout pour les larves). Chaque fois qu'une ouvrière satisfait une demande, la demande diminue et le signal correspondant à cette demande est lui aussi diminué. Prenons un exemple, en produisant le signal `LarveAffamée` une larve exprime son besoin de nourriture. Mais si une ouvrière parvient effectivement à lui fournir de la nourriture (en exécutant sa tâche `nourrirLarve`), alors son besoin de nourriture diminuera et l'intensité du signal `LarveAffamée` sera plus faible. De ce fait les actions tendent à réduire l'intensité des demandes et donc à réguler leur niveau.

7.6.3 Les expérimentations

L'allocation des tâches et la division du travail

Dans les premières versions de MANTA (Drogoul & Ferber 1994a; Drogoul & Ferber 1994b; Drogoul et al. 1992), l'accent fut mis sur les problèmes de différenciation sociale en étudiant l'apparition d'une organisation sociale caractérisée par une division du travail au sein d'ouvrières initialement identiques. Ces expériences portèrent sur une population de 30 fourmis, 50 larves, 50 œufs et 50 morceaux de nourriture aléatoirement réparties dans le nid. Les fourmis n'ont que 3 tâches possibles: s'occuper des œufs, des larves et de la nourriture, en regroupant en tas de la même espèce ces différentes entités. Voici le code correspondant à la tâche `soinsAuxOeufs` qui est déclenchée lorsqu'une fourmi perçoit le signal `Oeuf` avec un niveau suffisant:

```
soinsAuxOeufs
  suivreGradient(Oeuf)
  si il y a un tas d'oeuf alors
    prendre ce tas
    faire suivreGradient(Oeuf) et fuirGradient(humidité)
    déposer le tas
  sinon arrêter soinsAuxOeufs
```

Dans ce programme, la primitive `suivreGradient` fait suivre à la fourmi le chemin de plus grande pente pour la conduire à la source du signal correspondant (ici `œuf`) alors que la primitive `fuirGradient` travaille dans le sens contraire en faisant en sorte que la fourmi s'éloigne le plus possible de la source du signal. Les primitives `prendre` et `déposer` sont utilisées pour prendre et déposer quelque chose là où se trouve l'agent. La simulation s'arrête lorsque tous ces éléments sont totalement triés

et qu'il ne reste plus que trois tas. Bien que cet exemple n'ait pas pour objectif de simuler un véritable nid, deux leçons peuvent en être tirées:

1. La distribution moyenne du temps de travail global entre les trois tâches correspond bien à la répartition initiale de la quantité d'œufs, de larves et de nourriture, ce qui indique que la société globale adapte ses moyens à la demande et agit comme un système de régulation distribué.
2. La division du travail qui apparaît au sein du nid est caractérisée par plusieurs groupes fonctionnels (fig. 7.34):
 - a) Les "nounous des œufs" (groupe 1, 8 fourmis) caractérisées par un haut niveau de soins aux œufs et un faible niveau d'inactivité.
 - b) Les fourmis "non spécialisées" (groupe 2, 8 fourmis) caractérisées par un fort degré d'inactivité, bien qu'elles contribuent globalement aux autres activités du nid.
 - c) Les fourmis "nourricières" (groupe 3, 7 membres) caractérisées par un haut niveau de traitement de la nourriture. Les membres de ce groupes montrent aussi un fort niveau d'inactivité.
 - d) Les "spécialistes des larves/inactifs" (groupe 4, 3 fourmis) caractérisés par un haut niveau de soin aux larves et aussi un niveau élevé d'inactivité et un très faible niveau de soins aux œufs.
 - e) Enfin, les "nounous des larves" (groupe 5, 4 fourmis) qui présentent une grande spécialisation dans le soin aux larves et un très faible niveau d'activité dans les autres domaines.

Cette division du travail est évidemment plus simple que celle que l'on observe dans un nid réel, du fait de la simplification du comportement des fourmis. Néanmoins, la structure est très stable au travers de l'ensemble des expériences réalisées, ce qui confirme le fonctionnement de la rétroaction négative dont nous avons parlé plus haut: les demandes provenant des œufs et des larves en particulier sont diminuées par le travail accompli par les ouvrières dans ce petit "marché du travail" que représente un nid de fourmis.

Sociogenèse

La sociogenèse est le processus fondateur par lequel un simple individu (une reine) évolue vers une société complète en produisant de nouveaux individus (œufs, larves, cocons et ouvrières), en allant chercher de la nourriture et en s'occupant du couvain. Drogoul, Corbara et Fresneau (Corbara et al. 1993; Drogoul et al. 1993) ont montré qu'il était possible de reproduire artificiellement les conditions de réelles sociogenèses telles qu'elles ont pu être étudiées en laboratoire avec des fourmis *Ectatomma ruidum*. Une quantité variable de nourriture, qui dépend de la taille de la colonie, est fournie chaque jour à l'entrée du nid. Chaque fourmi (dont la reine) peut accomplir une vingtaine de tâches environ.

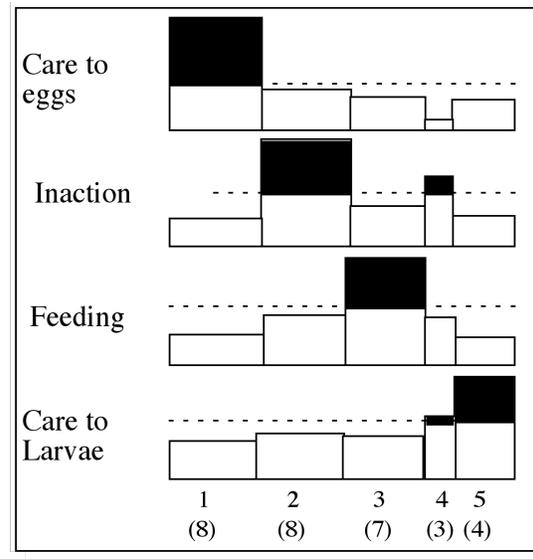


Figure 7.34: Sociogramme d'un nid de fourmis, montrant la différenciation comportementale des fourmis. Les lignes donnent les tâches effectuées et les colonnes indiquent le temps (en pourcentage) passé par les différents groupes de fourmis à accomplir ces tâches (d'après (Drogoul et al. 1992)).

Une expérimentation commence en plaçant une reine seule dans un nid vide et en laissant le système évoluer par lui-même. Elle s'arrête lorsque l'une des conditions suivantes est vérifiée: la reine meurt de faim, ce qui signifie la fin de la colonie, ou la population du nid dépasse 20 ouvrières ou plus, ce qui indique que le processus de sociogenèse a été mené à bien. Environ 78% des sociogenèses échouent, ce qui est assez proche des résultats réels où environ 86% de colonies n'atteignent jamais l'étape des 10 travailleuses. De plus, la dynamique d'une société artificielle est très semblable à celle que l'on observe dans la réalité. La figure 7.35 montre le processus de sociogenèse d'une véritable colonie caractérisée par un fort décalage entre la population des œufs et celle des larves. Bien que les courbes d'une colonie simulée soit plus régulières (fig. 7.36), elles montrent une grande similarité avec la précédente.

Ces expérimentations ont pu montrer en premier lieu que la dynamique d'une société simulée correspond assez bien à celle que l'on observe dans la réalité et montre qu'il est possible de faire évoluer et de stabiliser l'organisation sociale d'une société d'agents réactifs à partir de comportements simples ne comprenant qu'un petit nombre de primitives. Il est ainsi possible de se passer d'un contrôle centralisé pour réguler l'activité et l'évolution d'une société d'agents autonomes. Ce type de régulation des naissances émerge uniquement des interactions locales entre les agents et est conditionné uniquement par les contraintes environnementales auxquelles sont soumis les agents, telles que la restriction de nourriture. En second lieu, cela a permis de corroborer l'analyse théorique correspondant au mécanisme de la double rétroaction: une rétroaction positive qui agit au niveau des agents et provoque une différenciation comportementale et une rétroaction négative qui agit globalement par un mécanisme d'offre et de demande semblable à celui d'un

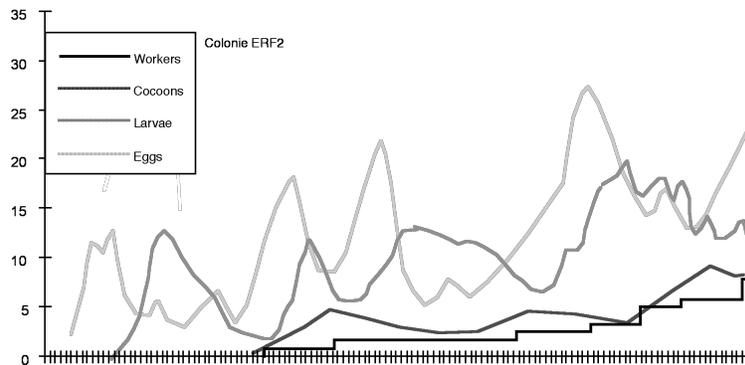


Figure 7.35: L'évolution du couvain dans une colonie naturelle (d'après (Corbara 1991))

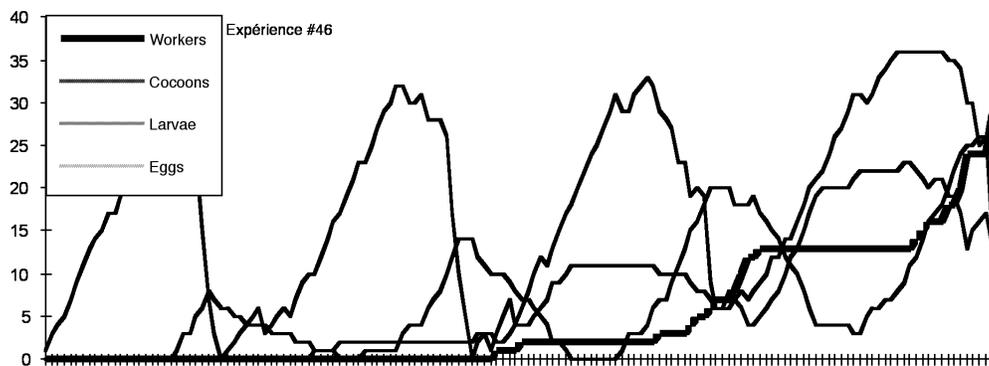


Figure 7.36: L'évolution d'une colonie simulée (d'après (Drogoul 1993))

marché. Enfin, la composante spatiale ne doit pas être négligée. Elle est en grande partie responsable du succès et de la stabilisation de la société et agit comme un coefficient d'amortissement dans la dynamique globale de la société.

7.6.4 Des fourmis aux robots fourmis

Des robots réactifs peuvent faire preuve d'une grande souplesse dans l'élaboration de tâches complexes et montrer des comportements collectifs émergents relativement spectaculaires à partir de quelques principes élémentaires. Mais peut-on les utiliser effectivement dans la réalisation de tâches utiles pour l'être humain? C'est à cette tâche que s'est consacrée Lynne Parker (Parker 1994). A partir des travaux de R. Brooks sur l'architecture de subsomption, elle a développé un système de robots coopératifs, ALLIANCE, dont elle a vérifié l'intérêt à la fois sur une simulation et sur de véritables robots. Dans l'un de ses exemples, plusieurs robots doivent nettoyer un terrain vague en le débarrassant de ses déchets, sachant qu'un ensemble de boîtes de conserves ont été déversées à un endroit du terrain. Dans un autre, ils doivent enlever la poussière de meubles (simulés) et nettoyer le sol.

Chaque agent possède une architecture qui semble issue d'une combinaison de subsomption et de tâches en compétitions similaire à celle du système MANTA. Comme dans ce dernier, les tâches de haut niveau sont en compétition et mutuellement incompatibles. Les robots peuvent s'envoyer des signaux qui indiquent leur état et la tâche qu'ils sont en train d'accomplir, afin éventuellement de ne pas accomplir les tâches effectuées par les autres. Les tâches sont ensuite définies sous la forme d'une structure de subsomption composée d'un ensemble de modules connectés par des dispositifs d'inhibition. Les tâches sont sélectionnées par ce que L. Parker appelle un système "motivationnel", qui se résume en fait à un système de déclenchement par stimuli comme nous l'avons vu avec MANTA. Elle a d'ailleurs développé des mécanismes d'adaptation consistant essentiellement à faire varier le seuil de déclenchement des tâches en fonction des agents, c'est-à-dire à reproduire une division du travail semblable à celle des fourmis.

Même si ce travail ne semble pas très performant en termes organisationnels par rapport à l'adaptation sociale obtenue avec MANTA, il faut reconnaître que l'utilisation de véritables robots entraîne une dégradation générale des phénomènes observés à cause essentiellement des informations erronées et incomplètes obtenues par leurs capteurs. Ce travail permet de valider dans un environnement réel les hypothèses que nous avons vérifiées sous la forme de simulations avec MANTA, à savoir la possibilité pour un ensemble d'agents réactifs de s'organiser globalement et de se répartir le travail sans requérir de coordinateur centralisé.

Chapter 8

Coordination d'actions

Lorsque Sigebert, robot réparateur, reçoit de la part du robot foreur Theodoric une demande de réparation de sa tête de foreuse, il sait que cela lui prendra du temps et lui demandera beaucoup d'efforts. En effet, une foreuse, cela ne se change pas comme ça. Il va falloir qu'il demande à un robot de l'aider à porter une foreuse neuve de la soute jusqu'à l'atelier de réparation. Mais les robots transporteurs lourds sont très occupés à cette heure. Il faudra demander à Frédégonde, qui organise les tournées, de le mettre en liste d'attente pour la demande de robots lourds et, en attendant, aller chercher Theodoric pour l'amener à l'atelier.

Quelques robots sont en train de déménager une pièce qui servait de remise pour un ensemble de petits matériels. Mais la porte et le couloir qui permet d'atteindre cette remise ne sont pas très larges et deux robots ne peuvent se croiser de front. Il faut donc que l'équipe s'organise. Pourquoi ne pas faire la chaîne et se passer les objets de robots en robots?

Dans ces deux cas, issus du même contexte, la gestion des tâches est problématique. Dans le premier cas, il faut gérer les tournées et les listes d'attente des demandes de robots transporteurs lourds. Dans le second, les actions des robots doivent être coordonnées pour que les objets soient passés de main en main sans qu'ils se cassent et sans provoquer d'embouteillage. Mais comment? C'est à la résolution de ces problèmes que ce chapitre est consacré.

8.1 Qu'est-ce que la coordination d'actions?

Lorsque plusieurs agents travaillent ensemble, il faut gérer un certain nombre de tâches supplémentaires qui ne sont pas directement productives mais servent à améliorer l'accomplissement des activités qui le sont. Ces tâches supplémentaires font partie du système organisationnel, et on les appelle *tâches de coordination*. Celles-ci sont indispensables dès que l'on se trouve en présence d'un ensemble d'agents autonomes qui poursuivent leurs propres buts, la réalisation de tâches productives entraînant avec elles tout un cortège de tâches de coordination sans lesquelles les premières ne peuvent être accomplies.

8.1.1 Définitions

La coordination d'actions a ainsi été décrite par Thomas W. Malone (Malone 1988) comme *l'ensemble des activités supplémentaires qu'il est nécessaire d'accomplir dans un environnement multi-agents et qu'un seul agent poursuivant les même buts n'accomplirait pas*¹.

Par exemple, dans un aéro-club où seuls quelques avions atterrissent par jour, le contrôle de trafic est presque inexistant. Au contraire, l'aéroport Charles de Gaulle, où un avion décolle ou se pose toutes les minutes, le contrôle aérien est une activité fondamentale assurée par des techniciens spécialement formés pour éviter que les avions n'entrent en collision. L'activité de coordination devient fondamentale et contraint même le problème initial, le transport de passagers, en limitant le nombre d'appareils en circulation. Que le trafic diminue et toute cette activité devient inutile. Qu'il augmente et les problèmes de coordination deviennent encore plus cruciaux.

La coordination des actions, dans le cadre de la coopération, peut donc être définie comme l'articulation des actions individuelles accomplies par chacun des agents de manière à ce que l'ensemble aboutisse à un tout cohérent et performant. Il s'agit ainsi de disposer dans l'espace et le temps les comportements des agents de telle manière que l'action du groupe soit améliorée soit par une augmentation des performances, soit par une diminution des conflits. La coordination des actions est donc l'une des principales méthodes pour assurer la coopération entre agents autonomes (cf. chap. 2).

Les exemples de coordination d'actions sont multiples et se rencontrent dans tous les domaines d'activités. Deux véhicules qui se croisent à un carrefour doivent adapter leurs actions les uns par rapport aux autres de manière à éviter les collisions. Un trapéziste qui doit agripper les poignets de son partenaire à la fin de son envol doit calculer très précisément le moment où il se lance en l'air. Les avions de la patrouille de France lorsqu'ils effectuent des voltiges aériennes doivent aussi coordonner leurs actions avec une très grande justesse. Des déménageurs qui déplacent un piano ou des ouvriers du bâtiment qui construisent une maison doivent aussi coordonner leurs actions de manière à venir à bout de leur tâche. Il en est de même des musiciens dans un orchestre de jazz ou de musique de chambre qui doivent faire en sorte que leur différentes parties se coordonnent les unes aux autres, dans un ordre précis et rigoureux afin que le résultat soit agréable à l'oreille². Enfin, des processus informatiques doivent eux aussi synchroniser leurs actions lorsqu'ils accèdent à des ressources communes de manière à ce que le système reste dans un état cohérent. La coordination d'actions est nécessaire pour quatre raisons principales:

1. Les agents ont *besoin d'informations et de résultats* que seuls d'autres agents peuvent fournir. Par exemple, un agent qui construit des murs aura besoin d'être approvisionné en briques, un agent qui surveille l'activité d'un procédé

¹T. Malone place dans les tâches de coordination toutes les tâches non productives et donc aussi celles que nous avons liées à la notion de *collaboration*, en particulier à l'allocation de tâches (cf. chap. 7)

²L'interprétation d'un morceau de musique, en particulier quand il s'agit d'improviser sur un thème, met d'ailleurs en œuvre les quatre formes de coordination qui sont examinées plus loin.

industriel en un point donné aura besoin d'informations sur l'état de ce procédé en d'autres endroits. Un agent logiciel qui apporte de la valeur ajoutée à un réseau de communication, tel qu'un service d'agence de voyage par exemple, nécessite les services d'autres agents, comme ceux de réservation de chambres et de places d'avions.

2. Les *ressources sont limitées*. On ne fait parfois attention aux actions des autres que parce que les ressources dont on dispose sont réduites et qu'on se retrouve à plusieurs à les utiliser. Qu'il s'agisse de temps, d'espace, d'énergie, d'argent ou d'outils, les actions pour être accomplies ont besoin de ressources qui n'existent pas en quantité infinie (c'est parce que l'espace est limité que des véhicules sont obligés de se coordonner pour s'éviter). Et la coordination est d'autant plus importante que les ressources sont faibles (pour des déménageurs, plus les objets à déplacer sont lourds et encombrants, plus ils doivent faire attention en accordant leur conduite). Il faut donc partager ces ressources de manière à optimiser les actions à effectuer (éliminer les actions inutiles, améliorer le temps de réponse, diminuer les coûts, etc.) tout en essayant d'éviter les conflits éventuels (conflits d'accès, collisions, actions contradictoires, etc.).
3. On cherche à *optimiser les coûts*. Coordonner des actions permet aussi de diminuer les coûts en éliminant les actions inutiles et en évitant les redondances d'action. Par exemple, si Sigebert reçoit deux demandes différentes pour aller chercher des pièces détachées, il sera plus rentable de combiner ces demandes et de ne faire qu'un voyage pour aller chercher ces pièces. De même, si deux personnes doivent se rendre au même endroit, elles pourront n'utiliser qu'une voiture et économiser ainsi l'essence correspondant à un trajet supplémentaire.
4. On veut permettre à des agents ayant des *objectifs distincts mais dépendants* les uns des autres de satisfaire ces objectifs et d'accomplir leur travail en tirant éventuellement parti de cette dépendance. La figure 8.1 illustre cette situation dans le monde des cubes. Deux agents doivent transformer un empilement de cubes en un autre en ne déplaçant à chaque fois qu'un seul cube. Plusieurs cas peuvent se présenter:
 - a) Si les objectifs sont compatibles et si les empilements initiaux et finaux sont totalement indépendant (fig. 8.1.a), on se trouve dans une situation neutre ou d'indépendance (cf. chap. 2), les deux agents ayant la possibilité d'accomplir leurs actions de manière totalement autonome.
 - a) Si les objectifs sont compatibles mais les cubes placés de telle façon que les agents aient à tenir compte de leurs actions réciproques. Dans ce cas, les agents doivent éviter de se gêner et si possible s'aider mutuellement pour que chacun satisfasse ses buts (fig. 8.1.b). Il s'agit alors d'une situation de coopération, que nous avons appelée collaboration coordonnée, nécessitant une coordination des actions.
 - a) Si, enfin, les objectifs sont incompatibles, c'est-à-dire que les buts de l'un viennent en contradiction avec ceux de l'autre (fig. 8.1.c), alors soit on

estime que le système est incohérent, soit on se situe dans le cadre d'une situation de compétition (cf. chap. 2).

Dans ce chapitre, nous ne nous intéresserons qu'aux situations de coopération nécessitant une coordination d'action, laissant de côté les situations neutres, du fait qu'elles ne posent aucun problème, et les situations de compétition qui nécessiteraient une étude d'un autre ordre.

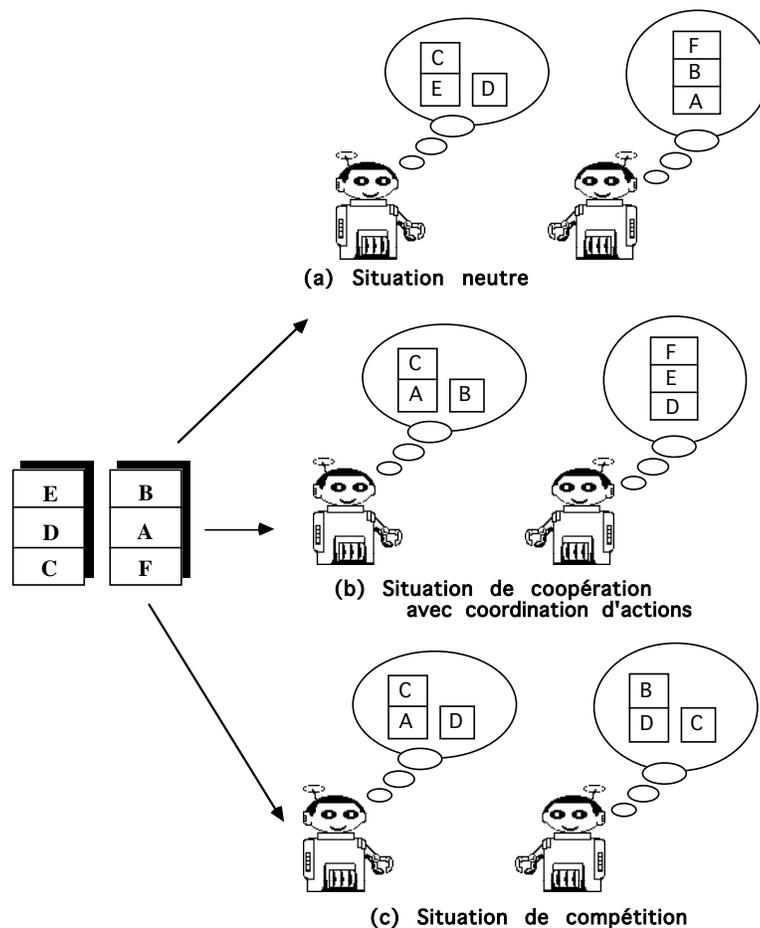


Figure 8.1: Quelques situations d'interaction. Seule la situation (b) est à la fois coopérative et nécessite une coordination des actions des deux agents.

8.1.2 La coordination comme résolution de problème

Le problème de la coordination d'action soulève plusieurs questions. Elles portent sur les agents avec lesquels un agent doit coordonner ses actions, sur le moment et le lieu d'accomplissement de ces actions coordinatrices, sur la détection des conflits ou synergies éventuelles et enfin sur la manière d'effectuer cette coordination.

Avec qui coordonner ses actions?

La première difficulté consiste à déterminer avec qui un agent doit coordonner ses actions. Dans certains cas la réponse est simple. Par exemple, dans le contrôle aérien, un appareil qui se rend d'un point à un autre doit tenir compte des actions de tous les avions qui trouvent trop près de lui et avec lesquels il risque d'entrer en collision. Identifier ces agents revient alors à déterminer tous ceux qui se trouvent à proximité, c'est-à-dire à percevoir les appareils qui se situent au voisinage de l'avion et à établir leur position et leur vitesse avec précision. Le cas du trafic aérien n'est pas unique. Au contraire il s'agit d'une situation typique que l'on retrouve avec tous les robots mobiles en général et dans les systèmes anti-collision en particulier. La coordination repose alors sur la relation que les agents entretiennent avec l'espace et notamment sur la perception de leurs distances et vitesses relatives.

Néanmoins, la définition des agents avec lesquels on doit se coordonner est souvent donnée par le problème lui-même. Par exemple, deux déménageurs, qui doivent synchroniser leurs gestes pour éviter de faire tomber les meubles qu'ils portent, n'ont pas à se poser la question de savoir avec qui ils doivent coordonner leurs action, car c'est de toute évidence avec l'autre porteur.

Les dépendances mutuelles entre actions

Percevoir et déterminer les agents avec lesquels on doit coordonner ses actions ne suffit pas. Il faut pouvoir aussi gérer les interdépendances qui existent entre les actions d'un ensemble d'agents. Trois agents doivent tenir compte de contraintes et de dépendances mutuelles que deux agents peuvent ignorer. Par exemple, si trois agents A_1 , A_2 et A_3 doivent coordonner leurs actions, cela signifie non seulement que A_1 doit organiser ses mouvements avec A_2 mais qu'il doit aussi les coordonner avec A_3 , qui lui-même doit tenir compte du comportement de A_2 . Par exemple, supposons que trois bateaux doivent se croiser en mer, comme le montre la figure 8.2. Si le bateau A_1 ne prend en compte que le bateau A_2 , il peut aller sur sa droite, mais se faisant il risque d'entrer en collision avec le bateau A_3 . Or A_3 ne peut pas changer sa route pour éviter A_1 sous peine de s'échouer sur la petite île. De ce fait, il lui faut tenir son cap, et c'est à A_2 et A_1 de l'éviter.

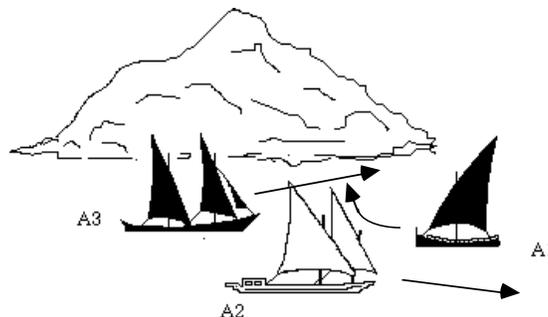


Figure 8.2: Evitement de collision entre bateaux

D'une manière générale, s'il existe une action coordonnée, toute action d'un agent

A_i dépend des actions de tous les autres agents $\{A_1, \dots, A_n\}$, sachant que leurs actions dépendent elles aussi des actions de tous les autres agents. Il est alors possible de considérer que coordonner les actions d'un ensemble d'agents revient à résoudre le système à n -équations et n -inconnues suivant:

$$\begin{aligned} \sigma_1(t+1) &= \theta_1(\text{React}(\text{Exec}(o_1(t)||\dots||o_n(t)))) \\ &\dots \\ \sigma_n(t+1) &= \theta_n(\text{React}(\text{Exec}(o_1(t)||\dots||o_n(t)))) \end{aligned}$$

où $o_i(t)$ représente l'opération que l'agent A_i exécute au temps t , et les $\theta_1, \dots, \theta_n$ les dépendances entre les actions des agents. Ce système peut encore s'exprimer par la simple équation:

$$O(t+1) = \Theta(\text{React}(\text{Exec}(O(t))))$$

où $O(t)$ représente le vecteur $\langle o_1(t), \dots, o_n(t) \rangle$ des opérations choisies par les agents $\langle A_1, \dots, A_n \rangle$ à l'instant t , et Θ représente le vecteur des fonctions de dépendances. Résoudre cette équation, revient donc à déterminer l'ensemble des fonctions comportementales (la suite des choix des opérations à appliquer) qui sont des solutions de cette équation.

La difficulté réside dans le fait que l'on ne sait pas résoudre ces équations dans le cas général et donc que, sauf dans des cas particuliers (et notamment ceux mettant en jeu des agents mobiles, et dans lesquelles les actions sont des déplacements dans un espace euclidien), il faut s'en remettre à des techniques heuristiques. Une théorie de la coordination reste donc à faire³. Celle-ci pourra se faire sur l'idée que les comportements sont contraints et, l'heuristique générale pourrait être de ne modifier que les comportements les moins contraints. C'est d'ailleurs ce que nous avons expliqué avec l'exemple de la figure 8.2. Lorsqu'il y a un conflit dû à une limitation des ressources, on modifie le moins possible le comportement des agents les plus contraints et au contraire on demande aux agents les moins contraints de tenir compte des actions des autres agents et d'agir en conséquence.

Relations entre actions

Lorsque les agents accomplissent leurs actions, certaines actions sont particulièrement sensibles, soit parce que leur exécution simultanée entraîne des conflits, soit, contraire, parce qu'elles conduisent à une amélioration des performances.

F. von Martial (von Martial 1992) a classé en deux grandes catégories les relations pouvant exister entre les actions lorsqu'elles sont accomplies simultanément par plusieurs agents, c'est-à-dire ce que nous avons appelé relations opératoires au chapitre 3. Les *relations négatives* ou conflictuelles sont celles qui gênent ou empêchent plusieurs actions de s'accomplir simultanément. Cet aspect négatif peut évidemment être dû soit à l'incompatibilité des objectifs, soit à la limitation des ressources, comme nous l'avons vu au chapitre 2. Au contraire, les *relations positives*, ou synergiques, sont celles qui favorisent les actions en les faisant bénéficier les

³Lorsque les opérations peuvent se traduire sous la forme de vecteurs (vitesses, accélération) dans un espace géométrique, K. Zeghal propose des prémisses de cette théorie dans (Zeghal 1993).

unes des autres et donc tendent à une plus grande efficacité que si les actions étaient réalisées de manière indépendante. La figure 8.3 montre cette classification des relations. La *relation d'égalité* signifie que certaines actions ne sont pas liées à un agent particulier et peuvent être réalisées par un autre agent. La *relation de subsomption*⁴ indique que l'action *a* d'un agent A fait partie des actions *b* d'un agent B, et donc qu'en accomplissant *b*, B réalise du même coup *a*. La relation de *faveur* précise qu'une action, en s'accomplissant, favorise la possibilité d'accomplir une autre action.

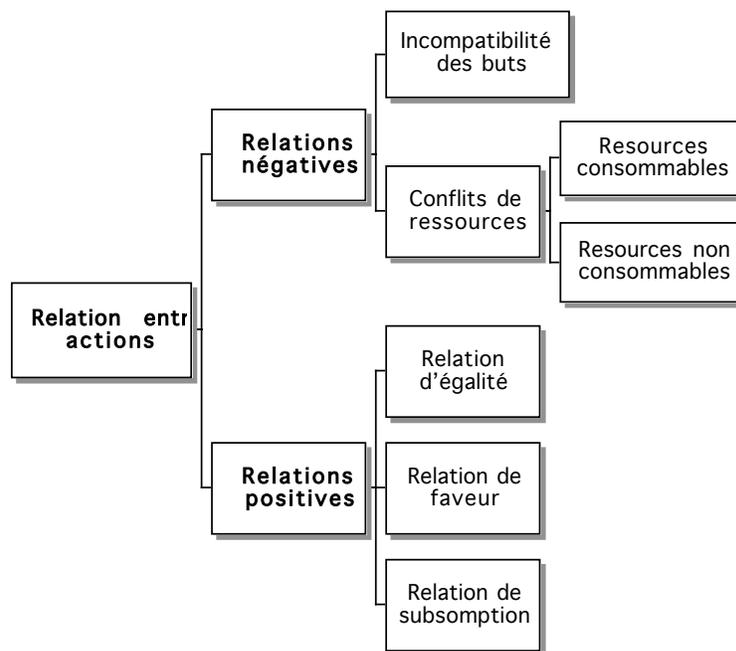


Figure 8.3: Les types de relations entre actions

Détecter qu'une relation existe entre deux actions constitue un enjeu majeur de l'organisation d'un ensemble d'actions, mais aussi un problème crucial pour l'organisation et l'amélioration de la coordination des actions. Parfois, surtout si la coordination d'actions est effectuée de manière centralisée et s'il s'agit de formes conflictuelles, les relations apparaissent facilement. En revanche, si la coordination est totalement distribuée, la plupart des relations deviennent difficile à reconnaître avant que l'action n'ait eu lieu, et la difficulté de mettre en œuvre de véritables méthodes de planification distribuée provient en grande partie de cette peine.

8.1.3 Caractéristiques des systèmes de coordination

Il existe plusieurs paramètres pour caractériser les différentes formes de coordination d'action. Alors que Durfee, Lesser et Corkill (Durfee et al. 1989) n'en retiennent que trois (la réactivité, la prédictivité et le nombre de messages communiqués),

⁴Ce terme est malheureux, puisqu'on parle aussi d'architecture de subsomption (cf. chap. 3). Je l'ai reproduit néanmoins tel quel puisqu'il fait partie du bagage des concepts de von Martial.

j'en propose ici onze, qui sont classées en quatre catégories: les caractéristiques temporelles, organisationnelles, de réalisation et de généralisation. Leur étude assure une meilleure compréhension de l'intérêt de tel ou tel mode de coordination.

A. Caractéristiques temporelles

La rapidité, l'adaptabilité et la prédictivité caractérisent le rapport du système de coordination avec le temps.

- A.1. La *rapidité* porte sur la capacité d'un système à agir plus ou moins promptement à un événement, que cet événement soit prévu ou imprévu. Moins un système raisonne, plus il est "câblé" et plus il est rapide. Le temps de réaction d'un système dépend évidemment de la complexité du calcul, mais aussi d'autres paramètres tels que le nombre d'agents, le nombre de tâches, etc.
- A.2. L'*adaptabilité* concerne la capacité du système à prendre en compte des événements inattendus. Plus un système est adaptable, plus il peut réagir à des événements ou des situations non prévues, et donc plus il pourra être mis en œuvre dans des contextes évolutifs. A l'inverse les systèmes moins adaptables sont souvent plus efficaces dans des contextes dans lesquels toutes les données sont connues à l'avance.
- A.3. La *prédictivité* se rapporte à la faculté d'anticiper sur l'avenir, c'est-à-dire à déterminer avec plus ou moins de précision l'état du monde et des autres agents dans le futur. Plus un système est prédictif, moins il a tendance à être rapide et adaptable. Les systèmes prédictifs sont particulièrement efficaces dans des contextes où l'on a beaucoup d'informations sur les événements futurs et sur la manière dont il faudra les prendre en charge. Inversement, les systèmes peu prédictifs s'avèrent plus intéressants dans des contextes qui évoluent très vite ou lorsqu'on dispose de peu d'informations sur ce qui va se passer. Essentiellement, on distingue les modes *anticipatifs*, qui cherchent à déterminer la manière dont les actions doivent être coordonnées avant de les exécuter (planification), et les modes *réactifs* qui déterminent l'ordre et la manière d'agir au moment même où les actions sont exécutées (coordination réactive, synchronisation). Evidemment, ces modes ne sont que des extrêmes et toutes les hybridations intermédiaires sont possibles.

B. Caractéristiques organisationnelles

Ces caractéristiques portent sur la manière dont est organisée la coordination d'actions et sur sa centralisation.

- B.1. La structure organisationnelle peut être décrite le long d'un axe de *centralisation/distribution*, les organisations les plus centralisées étant généralement les plus simples à mettre en œuvre et les plus cohérentes, les organisations distribuées pouvant plus facilement s'adapter aux modifications imprévues de l'environnement et en particulier aux dysfonctionnements éventuels de certains

agents. On retrouve ici les caractéristiques organisationnelles que nous avons développées au chapitre 3.

- B.2. Le *mode de communication* définit la manière dont les agents prennent connaissance des actions des autres agents. Si les agents sont situés dans un environnement, ils pourront percevoir les actions des autres agents, soit directement en identifiant leur comportement, soit indirectement par l'intermédiaire de traces et de propagation de signaux. Puisque toute coordination passe nécessairement par des communications, il est certain que les modes de communication jouent un rôle déterminant dans la manière de faire interagir des agents entre eux et, comme d'habitude, nous retrouverons l'importance de la dichotomie cognitif/réactif dans les communications et donc dans la manière de coordonner les actions.
- B.3. La *liberté de manœuvre* laissée à un agent caractérise le degré d'indépendance du comportement d'un agent vis-à-vis des consignes qu'il peut recevoir d'un coordinateur ou à l'égard du protocole de coordination. Plus les agents sont libres d'agir comme bon leur semble, plus ils peuvent s'adapter à des circonstances particulières mais aussi plus ils sont à même de créer des "pièges", c'est-à-dire des situations bloquées localement, et dont il s'avère souvent difficile de sortir (cf. les caractéristiques de qualité).

C. Caractéristiques de qualité et d'efficacité

La qualité et l'efficacité des résultats obtenus sont certainement les deux principales propriétés que l'on attend d'une méthode de coordination.

- C.1. La *qualité* de la coordination est évidemment l'enjeu principal d'un système de coordination. Les comportements sont-ils optimaux ou simplement corrects? Utilisent-ils au mieux les ressources disponibles? Sont-ils capables de trouver des synergies ou bien s'épuisent-ils dans des actions inutiles et redondantes? On peut aussi se demander si les techniques de coordination dégradent ou améliorent les performances du système. Par exemple, en quoi les feux de signalisation, nécessaires pour régler le trafic et éviter les collisions entre voitures, modifient-ils la circulation? Améliorent-ils le trafic ou, au contraire, sont-ils perturbateurs et créateurs de bouchons?
- C.2. *Evitement de conflits*. La qualité d'une méthode de coordination porte aussi sur sa capacité à éviter et à sortir des conflits éventuels. Certaines techniques ont malheureusement l'inconvénient de tomber parfois dans des "pièges", c'est-à-dire des situations dans lesquels les agents ont bien du mal à se sortir par eux-mêmes. Par exemple, un bouchon de voitures à un carrefour peut devenir si dense qu'il devient quasiment impossible de l'éliminer. Il s'agit d'un piège correspondant à une "étrointe fatale" (dead-lock), chaque véhicule ayant besoin que l'autre bouge pour pouvoir se dégager. Il est donc préférable d'éviter autant que possible ces situations en utilisant des techniques appropriées. Par exemple, la priorité à droite lorsque l'on circule à droite favorise l'apparition

de ces étreintes fatales, alors que la priorité à gauche permettrait d'éliminer un grand nombre de ces pièges néfastes.

- C.3. *Nombre d'agents.* La qualité d'une technique de coordination se mesure aussi au nombre d'agents que l'on est capable de coordonner ensemble. Certaines ne permettent pas de dépasser quelques unités alors que d'autres peuvent aisément ordonner les actions de plusieurs centaines d'agents.

D. Caractéristiques de réalisation

Ces caractéristiques se rapportent aux moyens nécessaires à la réalisation d'un système de coordination.

- D.1. *Degré et quantité d'informations.* Cette caractéristique concerne la quantité d'informations que les agents doivent échanger pour coordonner leurs actions. Ces échanges peuvent se faire à deux moments différents, soit dans l'élaboration préalable des plans d'action, si les agents sont capables d'employer des techniques anticipatrices, soit au cours de l'exécution. On privilégie généralement les modes de coordination permettant de diminuer le nombre de communications nécessaires pour éviter leur possible engorgement.
- D.2. *Le degré de représentation mutuelle requis* est lui aussi un critère de performances. En effet, plus il est nécessaire d'avoir des informations sur le monde et sur autrui, plus il en coûte pour mettre à jour cette connaissance et plus il est donc difficile qu'elle soit en adéquation avec l'état de l'environnement et celui des autres agents. Inversement, plus on dispose d'informations sur les autres plus on est à même de prévoir l'évolution du système et donc de répondre de manière adaptée aux actions des différents agents.
- D.3. *La difficulté de mise en œuvre.* Certaines techniques supposent l'existence de mécanismes complexes et sont difficiles à programmer et à mettre au point. D'autres au contraire s'avèrent très simples. Toutes choses égales par ailleurs, il est toujours préférable de s'en tenir aux méthodes les plus simples, ne serait-ce que pour des raisons de facilité de conception, de réalisation et de maintenance.

E. Caractéristiques de généralisation

Celles-ci indiquent dans quelle mesure une méthode de coordination est générale, en autorisant une certaine hétérogénéité des agents ou en s'appliquant à différents domaines.

- E.1. *L'hétérogénéité* de l'approche, c'est-à-dire l'aptitude à définir des systèmes de coordination qui sachent prendre en compte des agents qui diffèrent par leurs capacités de perception, de raisonnement et de communication, et donc de pouvoir interconnecter des agents qui ne disposent pas des mêmes mécanismes de coordination.

- E.2. La *généralité* de la méthode. Certaines techniques sont souvent plus adaptées à certains domaines (par exemple la coordination de déplacements de véhicules autonomes) que d'autres. S'il est préférable de disposer de méthodes générales qui peuvent s'adapter à n'importe quel domaine, il faut savoir que généralement plus une méthode est générale moins elle a tendance à être efficace.

Evidemment, ces critères ne sont pas totalement indépendants. Un système très rapide et capable de faire face à n'importe quelle situation ne permettra généralement pas de donner de bonnes prévisions à long ou moyen terme. De même, des méthodes réclamant une forte communication entre les agents et des capacités de représentation mutuelle auront du mal à prendre en compte des agents fortement hétérogènes.

8.1.4 Formes de coordination d'action

A partir des critères précédents, il est possible de dégager quatre formes principales de coordination d'actions, chacune pouvant donner lieu à plusieurs variantes (cf. tableau 8.1):

1. *Coordination par synchronisation.* Il s'agit là de la forme la plus élémentaire et "bas niveau" de la coordination. Toute coordination d'action doit décrire précisément l'enchaînement des actions, ce qui induit une nécessaire synchronisation de leur exécution. On parle généralement de synchronisation lorsqu'il s'agit de gérer la simultanéité de plusieurs actions et de vérifier que le résultat des opérations soit cohérent. Les principaux problèmes et algorithmes de synchronisation sont issus des systèmes distribués et sont utilisés tels quels dans le contexte des systèmes multi-agents. Nous n'en parlerons donc que très peu.
2. *Coordination par planification.* Cette technique, qui est la plus traditionnelle en IA, repose sur un découpage de l'action en deux phases: Dans la première, on réfléchit sur l'ensemble des actions à effectuer pour atteindre un but en produisant un ensemble de plans. Dans la seconde, on choisit l'un de ces plans que l'on exécute. Du fait que l'environnement peut évoluer, les plans peuvent être amenés à être révisés au cours de leur exécution, ce qui nécessite alors des possibilités de replanification dynamique. De plus, dans les systèmes multi-agents, les différents plans élaborés par les agents peuvent entraîner des conflits d'objectifs ou d'accès à des ressources. Il faut alors coordonner les plans de manière à résoudre ces conflits et satisfaire ainsi les buts des différents agents. D'une manière générale les techniques de planification assurent une bonne qualité de coordination, mais se révèlent incapables de prendre en compte des situations imprévues ou trop complexes.
3. *Coordination réactive.* Cette technique, qui est plus récente, considère qu'il est souvent plus facile de mettre en œuvre des mécanismes de coordination fondés sur des agents réactifs que de planifier l'ensemble des actions et de leurs interactions avant d'agir. D'une manière générale ces techniques font surtout appel

	Synchro.	Planification	Réactive	Réglement.
Rapidité	très bonne	faible	très bonne	bonne
Adaptabilité	très faible	faible	très bonne	bonne
Prédictivité	faible	très bonne	faible	moyenne
Centralisation/ distribution	indifférent	indifférent	indifférent	centralisé
Mode de communication	messages	messages	stimuli/ marques	indifférent
Liberté de manoeuvre	très faible	faible	grande	assez faible
Qualité de la coordination	assez bonne	très bonne	assez bonne	assez bonne
Evitement de conflits	bon	bon	faible	bon
Nombre d'agents	grand	faible	très grand	grand
Quantité d'échanges	moyenne	élevé	faible	faible
Représentations mutuelles	faible	élevé	faible	faible
Difficulté de mise en oeuvre	moyenne	élevée	faible	moyenne
Hétérogénéité	faible	très faible	élevée	moyenne
Généralisation	faible	moyenne	moyenne	moyenne

Table 8.1: Valeurs des paramètres en fonction des formes de coordination

à la liaison perception-action d'un agent, c'est-à-dire qu'elles s'accomplissent in situ, et non a priori comme les précédentes. On peut différencier plusieurs méthodes de coordination réactive qui diffèrent par leur usage de champs ou de routines (réflexes), par leur usage ou non de représentations limitées des autres et par l'emploi ou non de contraintes et de dépendances. Les approches très réactives n'assurent pas nécessairement une très grande optimisation de l'ensemble des résultats mais sont capables d'intervenir dans des contextes évolutifs et d'une manière générale dans des situations où il est difficile d'anticiper ce qui va se passer.

4. *Coordination par réglementation.* Il s'agit d'une méthode rarement décrite dans la littérature, mais souvent mise en pratique dans les systèmes nécessitant une coordination limitée. Le principe est de donner des règles de comportement qui visent à éliminer les conflits possibles. Par exemple, on attribuera des règles de priorités à des véhicules à un croisement de manière à éviter le conflit. Cette technique est évidemment inspirée du code de la route et d'une manière

générale de toutes les réglementations utilisées pour définir ce qu'est un "bon" comportement et éviter les conflits autant que possible.

8.2 Synchronisation d'actions

Synchroniser plusieurs actions, c'est définir la façon dont s'enchaînent les actions de manière à ce que leur déroulement dans le temps concorde et qu'elles puissent être accomplies "juste au bon moment". On parle aussi de synchronisation si l'objectif est d'assurer une certaine cohérence dans le système et d'empêcher que le résultat des actions des uns ne viennent perturber celles des autres.

Le problème de la synchronisation d'actions a surtout été étudiée dans le domaine des automatismes industriels ou celui des systèmes d'exploitation répartis. La synchronisation constitue le "bas niveau" de la coordination d'action, l'endroit où sont implémentés les mécanismes de base permettant aux différentes actions de s'articuler correctement.

8.2.1 Synchronisation de mouvements

Dès que plusieurs éléments doivent se mouvoir ensemble, il est nécessaire de synchroniser leur déplacement. Par exemple, un cycliste doit faire attention que lorsque sa jambe droite appuie sur la pédale de droite, la gauche se relâche et vice-versa. La coordination est alors affaire de rythme et de positionnement dans le temps des actions en fonction des événements. Si l'on considère chacune des jambes du cycliste comme un agent, il faut donc synchroniser la poussée de chaque jambe de manière à ce qu'elle ne puisse s'effectuer que si la pédale correspondante se trouve en position haute.

De nombreux formalismes ont été développés pour décrire et résoudre les problèmes de synchronisation, mais les plus connus et les plus employés sont issus des réseaux de Petri (cf. chap. 4). Par exemple, la synchronisation des jambes du cycliste s'expriment très simplement sous la forme d'un tel réseau (fig. 8.4). Chaque jambe est représentée sous la forme d'une place et d'une transition, la place représentant la jambe dans l'état "haut", et la transition correspondant à l'action d'appuyer sur la pédale. A l'état initial on suppose qu'une seule des places est marquée.

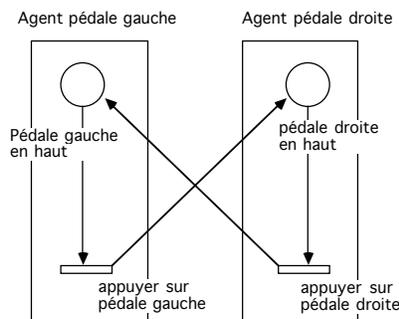


Figure 8.4: Synchronisation des actions des deux jambes d'un cycliste

8.2.2 Synchronisation d'accès à une ressource

Lorsque plusieurs agents doivent partager une ressource, ils doivent synchroniser leurs actions et éviter ainsi que leurs actions deviennent incohérentes. Supposons, comme dans (Georgeff 1983) que deux robots, Clotaire et Chilpéric, aient à utiliser la même machine pour construire le premier des écrous, le second des boulons (fig. 8.5). Le comportement de Clotaire consiste à répéter indéfiniment la suite des actions

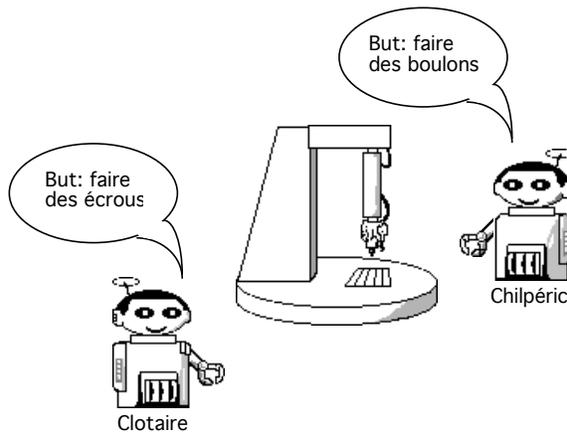


Figure 8.5: Clotaire et Chilpéric doivent partager la machine pour faire des écrous et des boulons.

suivantes:

Comportement Clotaire

- Aller jusqu'à la machine
- Placer un morceau de métal sur la machine
- Faire un écrou
- Porter l'écrou dans le stock des écrous

Celui de Chilpéric ne diffère pas beaucoup du précédent:

Comportement Chilpéric

- Aller jusqu'à la machine
- Placer un morceau de métal sur la machine
- Faire un boulon
- Porter le boulon dans le stock des boulons

Les actions de l'un peuvent évidemment interférer avec celles de l'autre. Lorsque Clotaire utilise la machine, il ne faut pas que Chilpéric vienne toucher ses boutons et dérégler le travail de l'autre robot. De ce fait, il est nécessaire d'introduire des mécanismes de synchronisation qui auront pour but de placer l'utilisateur de la machine en section critique et d'obliger l'autre à attendre son tour. Le concept de *section critique*, et les techniques associées, est utilisé par les spécialistes de systèmes d'exploitation pour indiquer qu'un processus utilise des ressources et qu'il ne doit pas être dérangé pendant son travail. Pour résoudre ce problème, il faut faire en

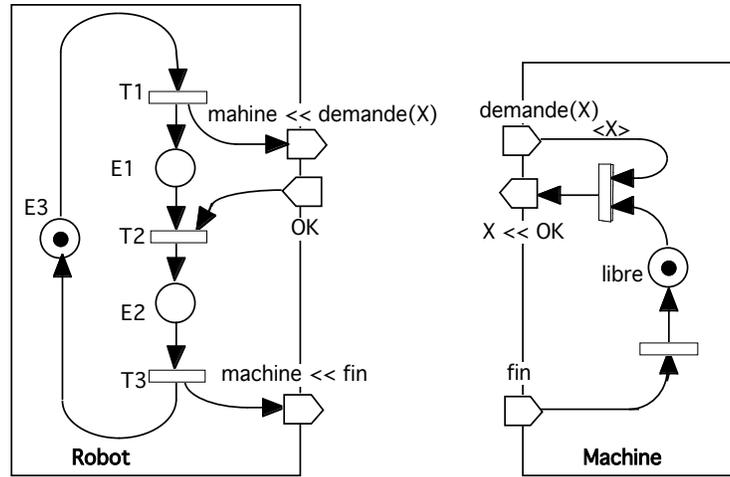


Figure 8.7: Description du comportement des agents de type Robot et Machine permettant de synchroniser l'accès à la machine

8.3 Coordination d'actions par planification

De loin le plus classique en IAD, la coordination d'actions par planification utilise les résultats de la planification mono-agent, élaborée dans le cadre de l'intelligence artificielle, et les étend aux situations dans lesquelles interviennent plusieurs agents. Avant de décrire la problématique de la planification multi-agent, nous ferons un petit rappel des techniques classiques de planification mono-agent.

8.3.1 Planification multi-agent

Les systèmes multi-agents n'ont pas simplifié la problématique de la planification, au contraire. S'il est déjà difficile pour un agent unique de déterminer la suite des actions qu'il doit accomplir, l'introduction d'agents supplémentaires n'a fait qu'aggraver les choses. En premier lieu, la multitude ne peut qu'aggraver le risque de modifications non prévues par le système de planification, ce qui réclame une plus grande réactivité de l'organe d'exécution et peut nécessiter de nombreuses re-planifications. Pour les mêmes raisons, l'interdépendance entre actions ne peut aller qu'en s'amplifiant, ce qui augmente les difficultés d'obtenir un ordre convenable pour l'application des opérations, difficultés que nous avons déjà décrites au chapitre 4. Enfin, l'introduction de plusieurs agents qui accomplissent leurs actions en parallèle contredit l'hypothèse de séquentialité d'application des opérations que l'on rencontre avec STRIPS. Cependant, être en présence d'un ensemble d'agents favorise l'exécution des actions puisqu'il n'est plus nécessaire de linéariser des plans parallèles, et que les problèmes de synchronisation trouvent une solution naturelle.

Limite des modèles à la STRIPS

Le choix d'une "bonne" représentation d'actions qui assure la prise en compte du parallélisme dans l'exécution des actions devient donc encore plus crucial que pour la planification pour agents uniques. Les modèles d'action "à la STRIPS", déjà limités en planification mono-agent, présentent de nombreuses difficultés dans le cadre de systèmes multi-agents, du fait de l'exécution nécessairement simultanée des actions des agents et des interactions qui en découlent. De ce fait, les hypothèses de STRIPS, et en particulier celles liées à l'intemporalité des actions et à la séquentialité de l'application des opérateurs, sont des contraintes excessives lorsqu'il s'agit de faire interagir des agents entre eux. Cependant, du fait de l'hégémonie extrême des principes d'action à la STRIPS en IA et de la difficulté d'offrir d'autres modèles d'actions puissants avec lesquels il soit possible de construire des plans, nombre d'auteurs en intelligence artificielle distribuée continuent d'utiliser des opérateurs à la STRIPS pour décrire les actions de leurs agents. Très conscient des limites de ce modèle, Georgeff (Georgeff 1984; Georgeff 1986) est l'un des auteurs qui a le plus contribué à essayer d'introduire de nouvelles manières de représenter les actions et de les appliquer à des univers multi-agents. Cependant, comme nous le verrons, ces extensions ne sont pas elles-mêmes sans présenter quelques problèmes.

L'action comme processus

Nous avons vu au chapitre 4 que l'on peut représenter les actions sous la forme de processus. C'est ce qu'a fait Georgeff en 1984 en proposant une autre approche de la représentation d'actions dans un univers multi-agent en partant de cette notion (Georgeff 1984). Un processus est une abstraction qui représente une action et qui peut elle-même se décomposer en sous-processus jusqu'à aboutir aux processus atomiques indécomposables. Par exemple, `préparerSpaghettiBolognaise` est un processus qui représente l'ensemble des manières particulières pour préparer des spaghetti bolognaise. Pour s'accomplir, ce processus nécessite l'exécution de sous-processus tels que `préparerSauce` et `cuireSpaghetti` qui peuvent s'exécuter simultanément. Ce dernier peut s'exprimer sous la forme `prendreCasserole`, `remplirCasseroleEau`, `chaufferCasserole`, `mettreSpaghettiDansCasserole`, `retirerSpaghetti`. L'action est alors une instantiation particulière du processus qui décrit un ordre d'exécution des différents sous-processus qui le composent. Nous avons vu au même chapitre qu'un processus peut être décrit sous la forme d'un automate dont les états représentent les états d'exécution du processus à un instant donné, et les transitions correspondent au passage d'un état du processus à un autre. Chaque nœud est étiqueté par l'ensemble des états admissibles pour cet état d'exécution. Les arcs sont étiquetés par les transitions atomiques.

Dans ce contexte, un plan est un mot reconnu par l'automate, et l'ensemble des plans possibles pour "exécuter le processus" correspond à l'ensemble des mots acceptés par l'automate. Par exemple, imaginons le monde de la figure 8.8 où Clotaire peut être localisé en cinq endroits différents. Supposons qu'il veuille aller en 5 pour récupérer l'outil. Il peut y aller soit en passant par les pièces 2 et 3 ou par 2 et 4. Le modèle de processus associé à `allerPièce5` s'exprime sous la forme de la figure

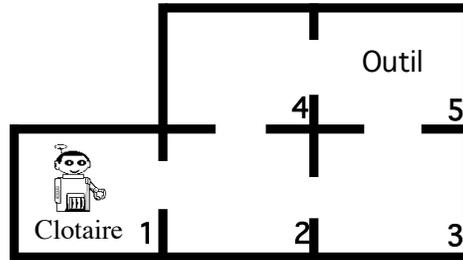


Figure 8.8: Le robot Clotaire veut aller en 5.

8.9.

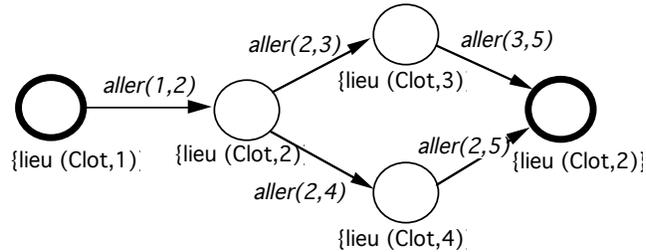


Figure 8.9: Un plan de Clotaire pour parvenir à son but

Chaque état de l'automate représente un état du monde défini comme un ensemble de formules atomiques, et les transitions correspondent aux actions possibles que peuvent effectuer les agents. L'intérêt de cette conception est de pouvoir composer les processus entre eux de manière à former des séquences, des itérations et des compositions parallèles. Par exemple, deux processus P_1 et P_2 peuvent être composés en séquence, ce que l'on écrit $P_1; P_2$. Il suffit pour cela de calculer l'intersection de l'état du monde obtenu en sortie du processus P_1 , avec l'état du monde obtenu en entrée du processus P_2 . L'itération consiste à boucler un processus P sur lui-même pour construire un nouveau processus P^* . Il suffit ici encore de fusionner l'état initial et l'état final en calculant l'intersection des états du monde correspondant.

La composition parallèle est particulièrement intéressante pour les systèmes multi-agents: pour composer deux processus P_1 et P_2 de manière concurrente, il suffit de créer un automate P_3 obtenu de telle manière que tout état de P_3 puisse s'exprimer comme une combinaison des états de P_1 et P_2 . Dans l'exemple du robot ci-dessus, supposons que l'on veuille ajouter un autre robot, Chilpéric, qui soit initialement dans la pièce 3 et qui ait comme but d'aller dans la pièce 5 et de fermer la porte entre les pièces 3 et 5. Le fait que Chilpéric puisse fermer la porte réduit l'ensemble des possibilités de Clotaire. En effet, si Chilpéric ferme la porte alors que Clotaire est en 3, ce dernier ne pourra plus passer directement de la pièce 3 à la pièce 5. Le processus définissant Chilpéric est défini à la figure 8.10.

Malheureusement, si l'idée de composer des automates peut paraître a priori intéressante, elle s'avère pratiquement inutilisable, du fait que le nombre d'états

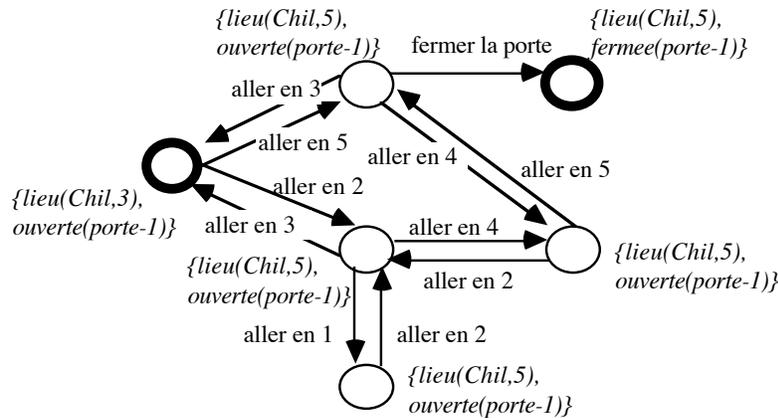


Figure 8.10: Le processus définissant Chilpéric

d'un automate P_3 résultant de la composition de deux sous-automates P_1 et P_2 est égal au produit des états de ses sous-automates. Ainsi, la composition des modèles des deux processus P_{Clotaire} et $P_{\text{Chilpéric}}$ qui comportent respectivement 6 et 5 états est un automate qui en comprend 30. De ce fait, la taille des automates résultant croît exponentiellement avec le nombre d'agents, ce qui impose de sérieuses limites quant à l'applicabilité d'un tel système.

Mais l'idée de considérer l'action comme des processus ne doit pas être nécessairement abandonnée pour autant. En utilisant des réseaux de Petri comme modèle de processus, il est possible de les composer sans tomber dans le travers de la croissance exponentielle du nombre d'états. Dans ce cas, le nombre de places d'un réseau composé est égal à la somme des places de chaque sous-réseau, ce qui nous donne des réseaux dont la taille est linéaire en nombre d'agents.

Pour reprendre notre exemple de robots, on modélisera non seulement Clotaire et Chilpéric comme des processus, mais aussi la porte qui relie les pièces 3 et 5. Les actions des agents sont modélisées sous la forme de transitions du réseau de Petri et les conditions par des places. Ces conditions sont vérifiées s'il existe (au moins) une marque dans la place correspondante. Certaines marques sont totalement consommées par les actions alors que d'autres sont au contraire conservées. Les marques consommées (et produites) représentent les transformations d'état, alors que les marques conservées représentent des caractéristiques nécessaires aux actions mais qui n'entrent pas dans les transformations. Par exemple, le fait que la porte soit ouverte est évidemment une condition nécessaire pour que le robot puisse la franchir, et cette condition n'est pas modifiée par ce déplacement. Les situations initiales et finales sont données par des marquages des différentes places, et planifier revient donc à déterminer un ordre de déclenchement des transitions qui permette de passer d'un marquage initial à un marquage final. Bizarrement, il n'existe aucune étude à notre connaissance qui ait considéré la représentation par réseau de Petri comme système de base d'analyse de l'action. Néanmoins, nous en reparlerons dans le cadre de la planification multi-agent, car elle permet de présenter aisément les problèmes qui se posent lors de la coordination d'action multi-agents.

Modes de planification multi-agent

Planifier des actions en univers multi-agents peut se décomposer en trois étapes distinctes. La construction de plans, la synchronisation/coordination des plans et l'exécution de ces derniers. Comme il est possible d'utiliser un ou plusieurs agents à chacune de ces étapes, il est possible d'obtenir un grand nombre d'organisations différentes, dont le principe général est donné à la figure 8.11. Un système de planification est alors composé d'un ensemble d'agents pouvant planifier, synchroniser ou exécuter des plans, un même agent pouvant accomplir une seule ou plusieurs de ces tâches.

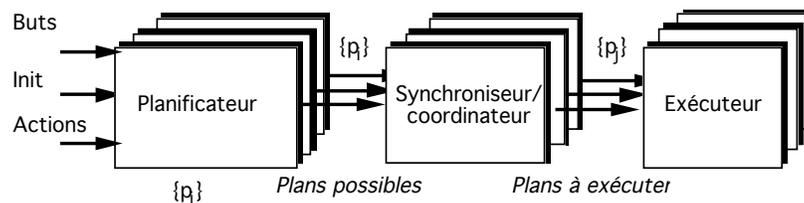


Figure 8.11: La planification multi-agent peut faire appel à plusieurs planificateurs, plusieurs synchroniseurs/coordonneurs et, évidemment, à plusieurs exécuteurs.

En premier lieu, il s'agit de construire des plans. Pour cela, on peut faire appel à un agent unique, et on parle alors de *planification centralisée* ou au contraire distribuer la tâche de planification au sein de chacun des agents, et obtenir alors un système de *planification distribuée*. Dans ce cas, chaque agent construit un *sous-plan* qui satisfait ses buts locaux, sans nécessairement tenir compte des autres agents. Dans ce cas, ces plans ont de fortes chances de ne pas pouvoir s'intégrer facilement les uns avec les autres et il devient alors nécessaire de les coordonner entre eux. Cette tâche peut être soit déléguée à un agent unique qui centralisera tous ces sous-plans et essaiera de les synthétiser afin de déterminer un plan global, soit laissée à l'initiative de chacun des agents, en leur demandant de coordonner leurs actions entre eux et de résoudre leurs conflits potentiels par des techniques de négociation.

De ce fait, les trois grands modes d'organisation classiques de la planification multi-agent sont:

1. la planification centralisée pour agents multiples,
2. la coordination centralisée pour plans partiels,
3. la planification distribuée.

8.3.2 Planification centralisée pour agents multiples

La planification centralisée pour agents multiples suppose qu'il n'existe qu'un seul planificateur, c'est-à-dire qu'un seul agent capable de planifier et d'organiser les actions pour l'ensemble des agents, comme le montre la figure 8.12. Cet agent traite

aussi généralement de la synchronisation des plans ainsi que de l'allocation des tâches aux autres agents dont le rôle se cantonne alors à n'être que de simples exécutants.

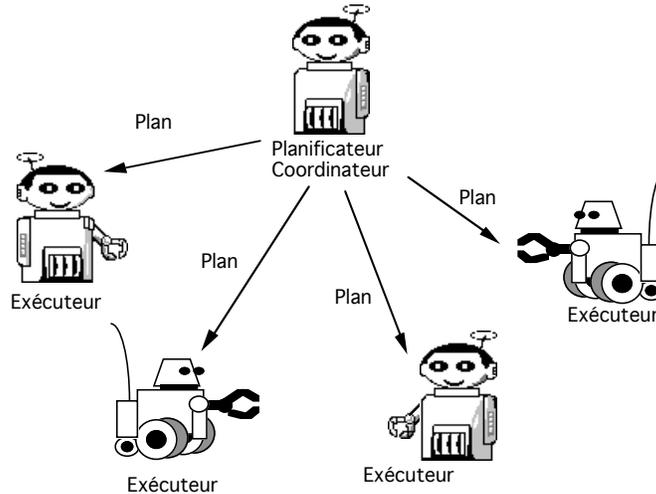


Figure 8.12: Dans la planification centralisée pour agents multiples, un agent unique, le planificateur est chargé de produire un plan global dont les actions sont ensuite allouées aux exécuteurs.

Les techniques de construction centralisée de plans pour agents multiples reprennent celles de la planification pour agents uniques et s'effectuent généralement en trois temps:

1. On cherche d'abord un plan général partiel, qui puisse s'exprimer sous la forme d'un graphe acyclique.
2. On détermine ensuite les branches qui peuvent être exécutées en parallèle et on introduit des points de synchronisation chaque fois que deux branches de calcul se rejoignent.
3. Enfin, on alloue l'exécution des tâches aux différents agents concernés.

Reprenons le cas des cubes traité au chapitre 4 dans le cas d'agents uniques et essayons de l'étendre au cas où deux robots peuvent manipuler ces cubes (fig. 8.13). Dans le cas d'une planification centralisée, on construit un plan comme s'il n'y avait qu'un seul agent, et le résultat de l'étape 1 donne un graphe partiel acyclique qui présente deux branches d'exécution parallèles pour la réalisation des opérations `poserTable(D)` et `poser(A,D)` d'une part et `poser(B,C)` d'autre part.

Le plan une fois réalisé, il faut placer des points d'exécution pour empêcher les agents de se trouver en conflit et faire en sorte que les actions effectuées le long d'une branche de calcul ne soient exécutées que si les précédentes sont bien terminées. Par exemple, on peut faire exécuter simultanément `poserTable(D)` et `poser(B,C)` puisqu'il n'existe aucune interférence possible dans l'exécution de ces actions. Il est alors possible de construire un graphe d'exécution sous la forme d'un réseau de Petri

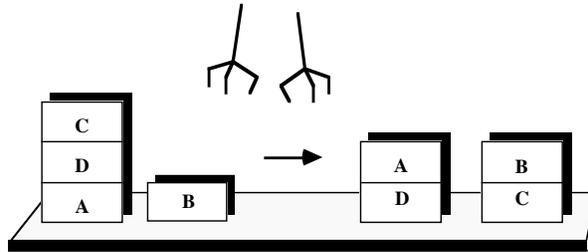


Figure 8.13: Un problème de déplacement de cubes, mais on cherche ici à l'exécuter par deux agents simultanément.

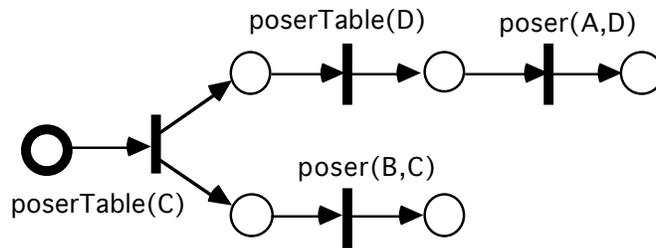


Figure 8.14: Plan de résolution d'un problème de cube

qui montre les points de parallélisation ainsi que ceux de synchronisation nécessaire (fig. 8.14).

Le problème de la coordination se résume alors à une simple combinaison de synchronisation et d'allocation de tâches. Deux cas limites peuvent se présenter. Soit l'allocation est gérée a priori par le planificateur (ou tout autre agent centralisateur) qui précise quel agent doit faire quoi, soit l'allocation est gérée de manière dynamique à chaque nouvelle action. Evidemment, entre ces deux extrêmes il est possible d'envisager toute une gamme de solutions intermédiaires.

1. *Gestion dynamique de l'allocation.* Une gestion dynamique de l'allocation peut être effectuée par un mécanisme d'appel d'offre comme nous l'avons vu au chapitre précédent. Il est aussi possible de gérer à la fois la synchronisation et l'allocation de tâches de manière dynamique en considérant les exécuteurs comme des ressources qui sont a priori d'accord pour effectuer le travail. Par exemple, le problème de la gestion des cubes peut être modélisé par un réseau de Petri comme le montre la figure 8.15. Par rapport au graphe d'exécution, on ajoute une place supplémentaire qui représente les ressources (c'est-à-dire les pinces en attente) et cette place est reliée à l'entrée et à la sortie de chaque action. Cela signifie qu'une fois l'action accomplie, la ressource est de nouveau disponible pour une autre action, ce qui résout élégamment le problème de l'allocation de tâches.

2. *Gestion statique de l'allocation.* Dans une gestion statique de l'allocation, la répartition des agents est effectuée au préalable, chacun sachant ce qu'il doit faire et quand. La coordination se résume alors à une simple synchronisation. La modélisation par réseau de Petri est ici aussi encore très simple: il suffit de créer autant de places qu'il existe d'agents et de relier ces places aux actions associées à

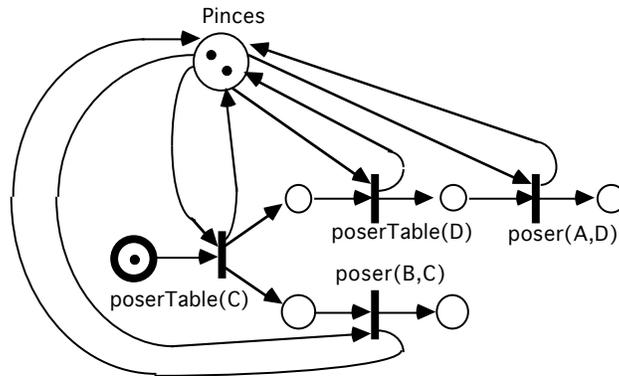


Figure 8.15: Exécution d'un plan avec allocation dynamique

chaque agent.

8.3.3 Coordination centralisée pour plans partiels

Il est possible aussi de ne centraliser que la coordination. Dans ce cas, chaque agent construit indépendamment son propre plan partiel qu'il envoie au coordinateur. Ce dernier tente alors de synthétiser tous ces plans partiel en un seul plan global cohérent (fig. 8.16). Il s'agit alors de dégager les conflits potentiels et de les supprimer, soit en ordonnant les actions, soit en déterminant les points de synchronisation nécessaires.

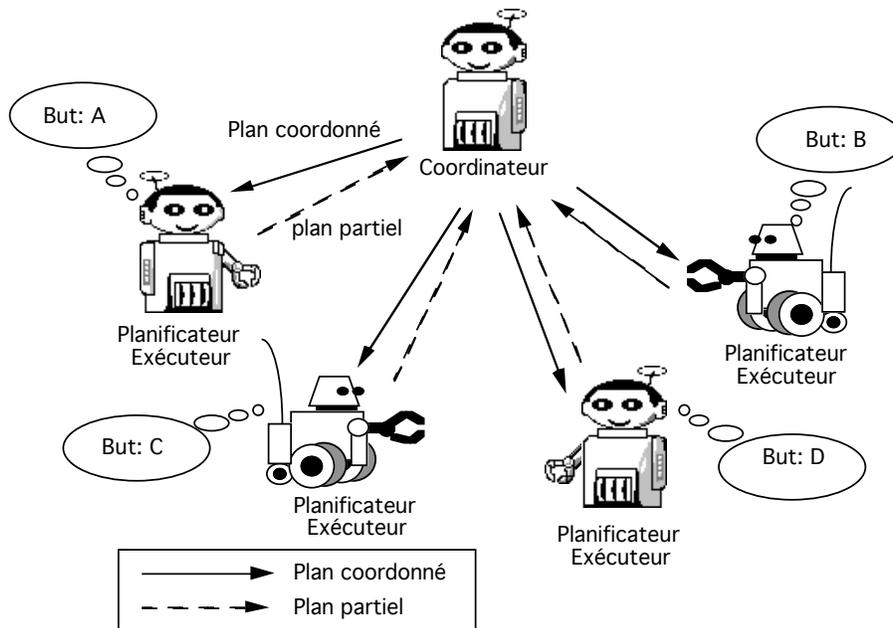


Figure 8.16: Dans la coordination centralisée pour plans partiels, un agent est chargé de la fusion et de la synchronisation des plans partiels construits indépendamment par les agents.

Coordination par fusion de plans partiels

La planification multi-agent repose sur une fusion réussie des plans partiels provenant des divers agents concernés de manière à n'en faire qu'un seul qui soit cohérent. Si l'on représente les plans sous la forme de réseaux de Petri, fusionner deux plans consiste à coller les deux graphes représentant les réseaux en identifiant les places semblables. Dans ce cas, plusieurs possibilités sont offertes selon que les actions des plans sont indépendantes, positives ou négatives.

Indépendance des actions

Il n'y a aucune place commune entre les agents et il est donc impossible de fusionner les deux réseaux en un seul réseau connexe. Les plans des agents sont donc indépendants, et l'exécution des actions des uns et des autres ne sont absolument pas modifiées.

Relations positives entre actions

Les deux plans comportent des actions qui sont en relation positive (cf. section 8.1.2). Il est alors possible de fusionner les plans pour obtenir un plan unique au moins aussi performant que les deux plans exécutés séparément. Le problème revient alors simplement à synchroniser l'exécution des plans partiels de manière à gérer les ressources en conflits sans modifier les plans eux-mêmes. Cela consiste à entrecroiser les actions des différents plans en établissant l'ordre d'exécution des actions et en déterminant les sections où il est possible d'exécuter plusieurs actions en parallèles, c'est-à-dire à élaborer un graphe acyclique représentant les différentes actions à accomplir.

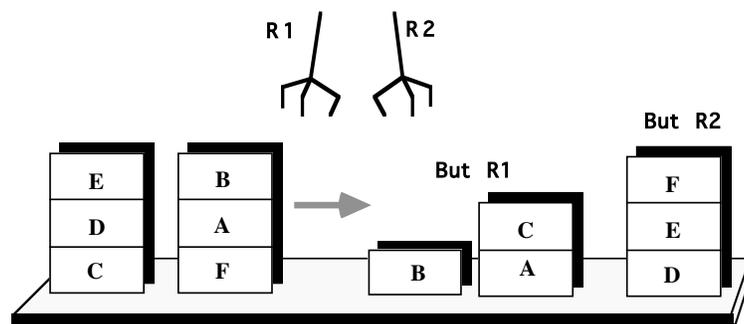


Figure 8.17: Un problème de cubes avec deux agents

Supposons par exemple que deux robots aient à réaliser chacun un empilement de cubes à partir d'un ensemble de cubes communs, comme le montre la figure 8.17, et que chaque agent n'ait aucune connaissance des buts de l'autre. Les buts des robots sont les suivants:

Buts R1 = sur(B,Table), sur(A,Table), sur(C,A)
 Buts R2 = sur(D,Table), sur(E,D), sur(F,E)

Si chaque robot dispose d'un planificateur qui lui est propre, il peut construire son propre plan sans tenir compte de l'autre. De ce fait, les plans obtenus par R_1 et R_2 peuvent prendre les formes des plans (partiellement ordonnés) P_1 et P_2 de la figure 8.18.

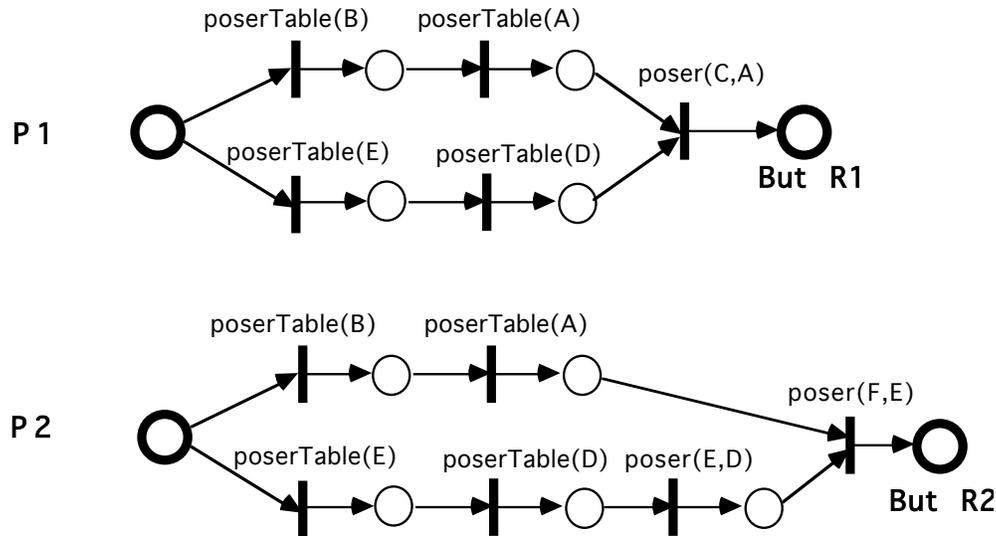


Figure 8.18: Les plans partiels des deux agents

La fusion des plans P_1 et P_2 est relativement simple, puisqu'il n'y a pas de point de conflits. Il suffit donc à un superviseur général de superposer les deux plans et donc de définir un plan global P_G qui soit la synthèse de ces deux plans partiels, comme le montre la figure 8.19.

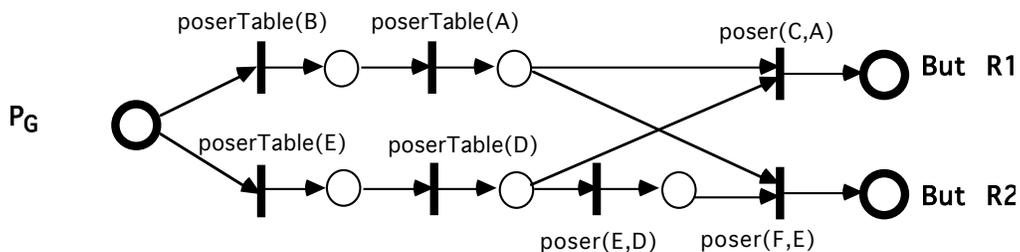


Figure 8.19: Synthèse des plans partiels de R_1 et R_2

Il suffit ensuite de lui appliquer une technique statique ou dynamique d'allocation et de synchronisation de tâches telle que celle que nous avons examinée à la section précédente.

Relations négatives et conflits de ressources

La résolution du problème précédent est particulièrement simple, puisqu'il n'y a ni conflit ni synergie, mais seulement une intersection non vide de la structure des plans, chaque agent pouvant satisfaire ses buts de manière indépendante. Mais

évidemment cette situation n'est qu'un cas particulier de la coordination d'agents multiples. Supposons par exemple que nous ayons à faire face à une situation comme celle de la figure 8.20 où plusieurs agents doivent s'aider mutuellement pour que les actions soient accomplies. Il s'agit de placer une petite caisse au fond d'un placard et un piano au fond d'un réduit, le réduit donnant accès au placard.

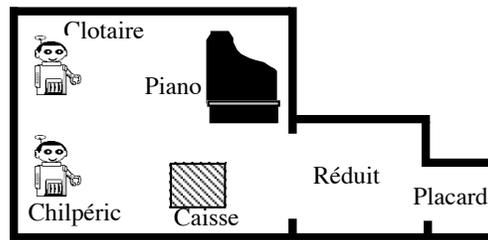


Figure 8.20: Comment placer une caisse et un piano, sachant qu'il faut deux agents pour transporter le piano?

Les deux opérateurs servant à définir ces actions peuvent s'écrire ainsi:

```
Opérateur: placerPiano
  nbre agents : 2
  pré: lieu(Piano,la),vide(Reducit)
  suppr: lieu(Piano,la),
         vide(Reducit)
  ajouts: lieu(Piano,reduit),
         plein(Reducit)
fin

Opérateur: placerCaisse
  nbre agents : 1
  pré: lieu(Piano,lb),vide(Reducit),
       vide(Placard)
  suppr: lieu(Caisse,lb),vide(Reducit)
  ajouts: lieu(Caisse,Placard),plein(Placard)
fin
```

Supposons que Clotaire décide de déplacer le piano et Chilpéric la caisse. Chacun n'a donc qu'un seul opérateur et s'il n'a pas connaissance des buts de l'autre, Clotaire demandera de l'aide à Chilpéric pour l'aider à ranger le piano. Mais s'il accomplit cette tâche, alors il ne pourra plus ranger la caisse dans le placard.

Le problème consiste donc à ordonner les actions des deux agents de manière à ce qu'elles soient accomplies dans le bon ordre. Ici encore, dès qu'il existe un agent centralisateur, le problème revient à fusionner les plans et à déterminer un parcours dans l'exécution des plans qui satisfasse tous les buts. La figure 8.21 montre les plans des deux agents avant la fusion, et la figure 8.22 montre le plan obtenu après fusion

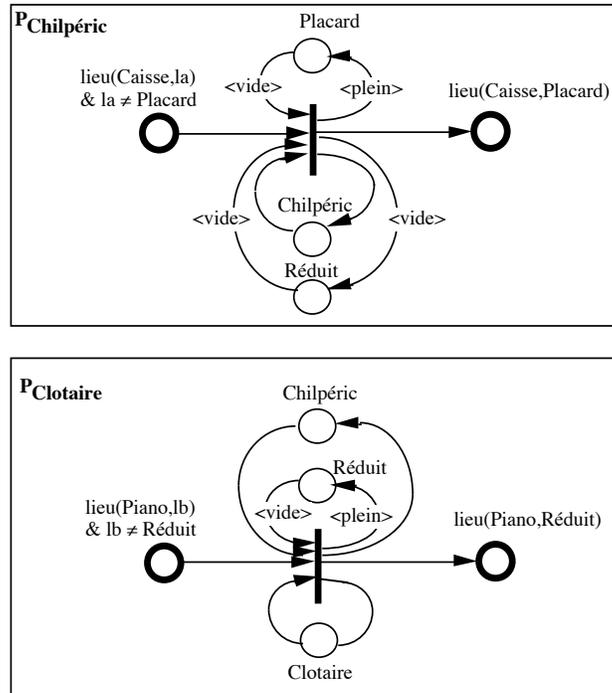


Figure 8.21: Les plans partiels de Clotaire et Chilpéric sous la forme de réseaux de Petri

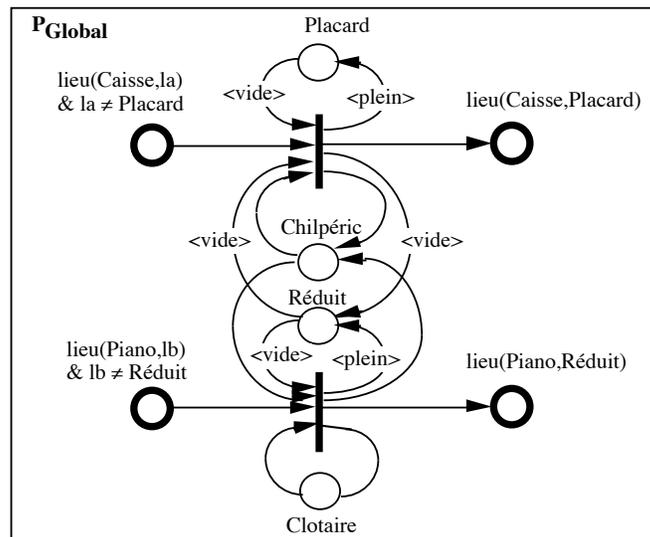


Figure 8.22: Le plan global obtenu après fusion des plans partiels de Chilpéric et Clotaire

des plans de Chilpéric et de Clotaire, en identifiant la place de Chilpéric et celle du réduit communes aux deux plans.

Résoudre le problème revient alors à trouver une suite de transitions telle que les deux états terminaux soient accessibles, c'est-à-dire à dire que les deux places terminales comprennent chacune un jeton.

8.3.4 Coordination distribuée pour plans partiels

Principe général

La planification distribuée introduit encore plus de répartition des travaux en supposant qu'il n'existe aucun agent centralisateur, ni pour planifier des plans globaux, ni pour coordonner des plans partiels. Dans la planification distribuée, chaque agent planifie individuellement les actions qu'il compte accomplir en fonction de ses propres buts. La difficulté porte alors non seulement sur la résolution des conflits potentiels qui peuvent intervenir lors de l'exécution des plans, mais aussi, comme nous l'avons précisé plus haut (cf. section 8.1.2), sur la reconnaissance des situations synergiques qui peuvent se présenter lorsque les actions des uns peuvent être utiles à la réalisation des buts des autres, les agents se rendant alors mutuellement service.

La figure 8.23 montre une organisation de planification distribuée. Chaque agent dispose de buts et de plans lui permettant de résoudre ses buts. Le problème consiste pour ces agents à échanger des informations portant sur leurs plans et leurs buts, afin que chacun puisse satisfaire ses objectifs.

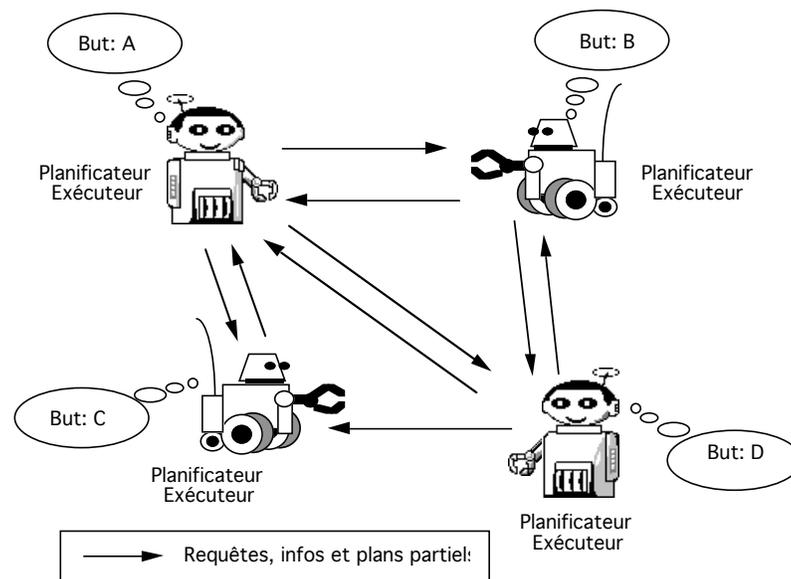


Figure 8.23: Le modèle de planification distribuée. Chaque agent dispose de buts et de plans partiels qu'il peut communiquer à d'autres agents.

Peu de chercheurs ont abordé ce problème dans toute sa généralité, hormis V.

Lesser et E. Durfee qui, avec le modèle PGP (Partial Global Planning ou planification globale partielle), ont tenté de donner un cadre général pour la planification distribuée pour agents multiples (Durfee et Lesser 1991).

Dans le modèle de planification globale partielle, chaque agent dispose d'une base de connaissance disposée sur trois niveaux. Au premier niveau se trouvent les plans locaux qui organisent les activités futures de l'agent avec tous les détails requis: buts à court ou moyen terme, coûts, durée, etc. Au deuxième niveau on trouve les *nœuds-plans* (dans la terminologie de Lesser et Durfee, un nœud est un agent), qui sont des résumés des plans locaux, débarrassés des détails inutiles. Enfin au troisième niveau se trouvent les PGP (Plans Globaux Partiels) qui contiennent une information générale sur le déroulement d'une partie des activités globales. Un PGP est une structure d'information que les agents utilisent pour échanger des informations sur leurs objectifs et leurs plans. Un PGP contient des informations portant sur l'ensemble des plans dont il assure la fusion, les objectifs poursuivis, ce que les agents sont en train de faire, les coûts et résultats espérés ainsi que la manière dont les agents devraient interagir. Un agent échange alors des PGP avec d'autres et, à partir d'un modèle de soi et des autres, il tente d'identifier les agents dont les buts appartiennent à un objectif de groupe, appelé PGG (pour Partial Global Goal) et de combiner les PGP associés en un PGP plus général pour satisfaire ce but.

Conflits d'accès aux ressources et construction des plans

Nous allons illustrer la difficulté qu'il y a à fusionner des plans locaux, en donnant un exemple qui n'appartient pas au domaine des SMA mais à la vie courante. Il a l'avantage de bien montrer les problèmes qui se posent lorsqu'il faut non seulement réorganiser l'ordre des actions, mais aussi, pour chaque agent, reprendre localement son plan en fonction du plan des autres.

Jules décide d'emmener son fils, Hector, au cinéma. Pour cela il envisage de prendre la voiture et donc de partir juste avant le début de la séance, ce qui lui permet de finir un travail en retard avant et de coucher Hector ensuite. Mais sa femme, Hélène, doit impérativement se rendre à un rendez-vous de travail pour lequel elle a absolument besoin de la voiture.

Ici les deux plans partiels sont en conflit du fait que l'accès à une ressource, la voiture, est limitée et que deux actions sont impératives: le rendez-vous d'Hélène et le coucher d'Hector, cette dernière pouvant être effectuée indifféremment par Jules ou Hélène, comme le montre la figure 8.24.

Comment résoudre ce problème? Voici une solution "humaine":

Jules accepte de prendre les transports en commun et donc de partir plus tôt. Il devra finir son travail en retard le soir et ne pourra plus s'occuper d'Hector. Hélène lui propose alors de s'en occuper en revenant de son rendez-vous.

En considérant qu'il existe non pas deux mais trois agents (Hector est aussi un agent) et que la voiture et les transports en commun agissent comme des ressources pour accomplir des actions, le problème et sa solution sont présentés à la figure 8.25.

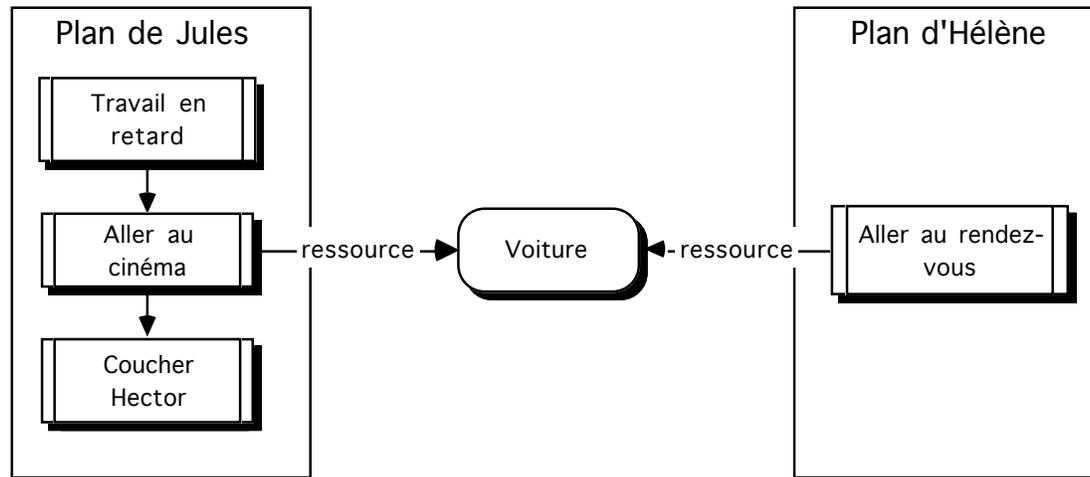


Figure 8.24: Les plans originaux de Jules et Hélène, avant coordination

A partir de ce réseau, il est facile de voir que l'on ne peut atteindre les place finales (cerclées en gras) que si effectivement Jules emprunte les transports en commun et qu'Hélène se charge donc de coucher Hector. C'est donc simplement à partir d'une technique de recherche de marquage dans un réseau de Petri que l'on peut définir un "bon plan" qui tiendra compte des interactions entre les agents et des ressources disponibles.

Mais en fait, le problème consiste généralement à trouver une bonne représentation du plan. En effet, dans le cas précédent, ce n'est que parce qu'il y avait un conflit d'accès aux ressources, que Jules a imaginé une alternative à son premier plan, en décidant de prendre les transports en commun et de ne faire son travail en retard qu'ensuite. C'est cette identification des conflits, mais aussi des possibilités annexes tels que l'accès à une autre ressources non envisagée au préalable qui constitue le travail le plus difficile.

Difficultés et quelques solutions envisageables

Ces approches générales fondées sur des plans partiels subissent encore les limites suivantes:

1. *Hypothèse d'homogénéité.* Les agents sont supposés homogènes et disposer des mêmes mécanismes de raisonnement. On suppose que tous les agents, placés dans les mêmes conditions, arriveront au même but, ce qui est évidemment loin d'être le cas dans la réalité.
2. *Limitation du nombre d'agents.* la puissance de ces techniques de fusion de plans sont limitées à des situations ne faisant intervenir qu'un très petit nombre d'agents, du fait de la difficulté à faire converger l'ensemble des plans locaux vers un plan global cohérent, en présence d'un ensemble de contraintes. D'une manière générale, toutes ces techniques ont une complexité qui croît

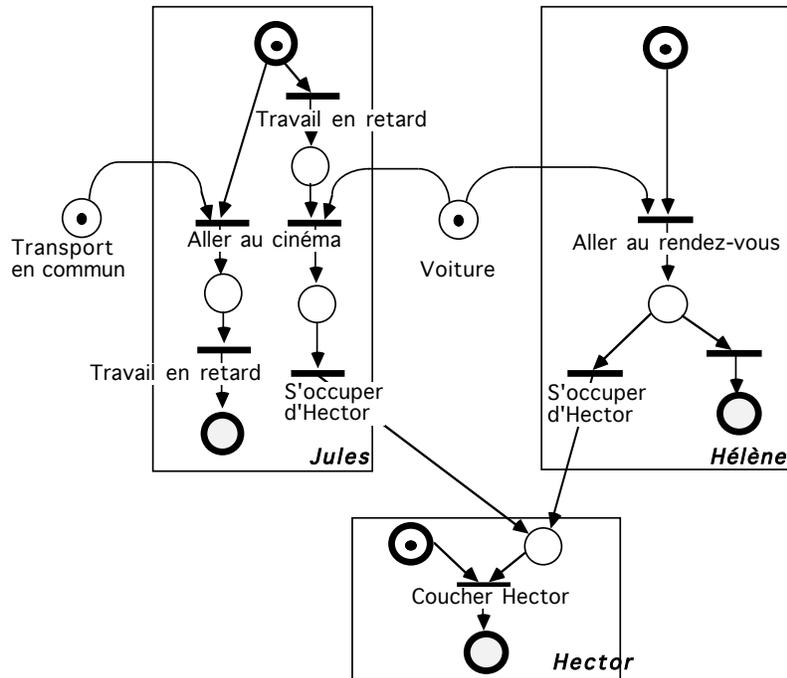


Figure 8.25: L'ensemble des plans de Jules et d'Hélène représentés sous la forme d'un réseau de Petri

exponentiellement avec le nombre d'agents et le nombre de choix possibles pour chaque agent.

3. *Hypothèse de régularité du monde.* Elles supposent que l'environnement ne se modifie pas trop vite par rapport à l'accomplissement des actions et que les agents ne changent pas de tâches facilement. En d'autres termes, on suppose que les agents s'engagent à accomplir les actions qu'ils ont planifiées. De ce fait ces techniques ne savent pas très bien prendre en compte des situations réclamant une réaction rapide, non planifiée sous la forme d'un acte réflexe.
4. *Problèmes temporels.* Enfin, ces techniques doivent être améliorées si l'on veut qu'elles prennent en compte la durée effective des actions et donc qu'elles gèrent les problèmes de simultanéité et de ressources uniquement disponibles pendant une plage de temps réduite.

Par rapport à tous ces problèmes et face à l'ensemble des difficultés qui se dressent devant la planification distribuée, on peut envisager des solutions qui permettent d'obtenir des plans cohérents et applicables? Pour répondre à cette question, nous nous bornerons à donner quelques conseils permettant d'aboutir à des solutions pratiques.

Hierarchisation des plans

On peut hiérarchiser les plans et supposer que leur affinage ne conduira pas à des conflits, en reprenant les travaux portant sur les plans hiérarchiques (lire par

exemple (Nilsson 1980) pour une présentation de la planification hiérarchique). Si des conflits apparaissent, on devra alors les traiter en utilisant au mieux les heuristiques présentées ci-dessous.

Engagement des agents

La notion d'engagement est essentielle pour la planification multi-agent. Si le monde, et donc les autres agents, était totalement imprévisible, s'il ne présentait pas quelques régularités, il serait impossible d'exécuter le moindre plan, puisqu'il deviendrait caduc sur le champ. Pour comprendre ce point de vue, il faut le mettre en correspondance avec un monde chaotique. Par exemple, supposons, comme l'a décrit L. Suchman (Suchman 1987) que nous devons franchir un rapide en canoë. On peut bien entendu s'asseoir un moment et essayer de trouver un plan qui nous permette de le passer sans encombre. Mais dès que l'on s'élançait dans le rapide, les courants nous font rapidement sortir de notre plan et l'on doit alors s'en remettre à ses réflexes, son expérience et son habileté. Le plan ne sert plus à rien. Cette conception permet ainsi de donner des arguments aux partisans de la planification réactive que nous étudierons à la section suivante. Mais heureusement le monde ne se comporte pas toujours comme un rapide. On peut planifier sa journée de travail car l'environnement ne change pas trop. Les rendez-vous sont des points de synchronisation sur lesquels on peut compter si les agents sont sûrs et s'ils ne remettent pas leur engagement (ou seulement très rarement). Dans ce cas, la planification est possible car le monde est relativement prévisible, les engagements des agents jouant le rôle des lois de la physique pour déterminer (partiellement) un état futur du monde. Par exemple, Clotaire, qui se trouve à Paris doit aller à Toulouse. Il apprend que Chilpéric doit aussi se rendre à Toulouse à la même époque et qu'il peut l'emmener. Clotaire peut donc tenir compte de ce que vient de lui dire Chilpéric, si celui-ci tient évidemment ses promesses, et planifier son voyage en considérant que le problème du moyen de transport est résolu.

Stratégies de fonctionnement

On peut aussi utiliser des stratégies de fonctionnement qui déterminent un mode d'approche du monde, en sachant que certaines ne conduiront certainement pas à une solution optimale, mais se révéleront faciles et rapides à mettre en œuvre, alors que d'autres, plus précises, réclameront des moyens plus importants. Par exemple, on peut supposer que tout se passera bien tant qu'il n'y aura pas de conflits et on ne replanifiera qu'à la détection de ces conflits. Dans ce cas, on exécute toujours ses propres plans localement sans tenir compte des autres et c'est uniquement lorsqu'on détecte des problèmes que l'on intègre les actions des autres dans ses plans. Cette manière de faire est évidemment mauvaise puisqu'elle ne permet pas de tenir compte des relations synergiques entre actions. Elle présente aussi l'inconvénient de ne pas assez tenir compte des autres et donc de devoir beaucoup s'en remettre à des techniques de résolution de conflits chaque fois qu'un problème se présente. Néanmoins, dans des situations où l'indépendance est de règle, où les situations de conflit sont mineures et où l'environnement est modifié rapidement, il s'agit d'une stratégie à envisager, présentant l'avantage d'être particulièrement simple à mettre en œuvre.

On peut au contraire préférer avoir une stratégie de général de corps d'armée et essayer d'utiliser toutes les informations disponibles sur les autres pour tenir compte de tous les conflits ou opportunités qui se présentent et replanifier au fur et à mesure de l'obtention de ces informations. Evidemment, cette technique correspond à une recherche d'optimalité, mais elle nécessite des moyens de calcul importants pour modéliser l'environnement et les autres agents et tenter de prévoir leurs plans d'actions.

Entre ces deux extrêmes, il existe de nombreuses approches, mais il faut savoir que, avec une vision distribuée, il n'est jamais certain (sauf dans des cas élémentaires) d'obtenir une solution optimale, et il faudra donc utiliser des heuristiques tout en sachant qu'elles peuvent conduire à des échecs alors qu'il existait une solution.

8.4 Coordination réactive

A l'opposé des approches précédentes, la coordination réactive considère qu'il est souvent plus simple d'agir directement sans planifier au préalable ce que l'on doit faire. Comme nous l'avons déjà signalé aux chapitres précédents, les agents réactifs sont très simples et ne possèdent pas de représentations de leur environnement. De ce fait, toutes les informations relatives à leur comportement se trouvent dans l'environnement et leurs réactions dépendent uniquement de la perception qu'ils peuvent en avoir. Les agents réactifs sont donc situés (cf. chap. 1) et opèrent dans "l'ici et le maintenant", c'est-à-dire qu'ils ne mémorisent pas les événements passés et ne peuvent pas non plus anticiper sur le futur. Bien que cette simplicité les rende souvent sans défense lorsqu'ils sont utilisés de manière isolée, les agents réactifs prennent leur force du travail en groupe, ce qui leur donne la possibilité de réaliser des actions dont ils auraient été bien incapables individuellement. Comme nous l'avons vu au chapitre précédent, les agents réactifs sont très sensibles aux relations spatiales qui déterminent tout à la fois des contraintes, des capacités d'actions et définissent des relations de coopération privilégiées.

8.4.1 Coordination par actions situées

Nous avons vu au chapitre 4 comment il était possible de définir le comportement d'un agent isolé en termes d'actions situées. Mais qu'en est-il pour des systèmes composés de plusieurs agents? Par exemple que se passe-t-il si l'on demande à plusieurs robots d'effectuer certaines tâches et comment peuvent-ils accomplir leurs objectifs simultanément. Nous nous trouvons évidemment dans une situation de coopération et plusieurs cas peuvent se présenter (cf. chap. 2). Si les agents ont des buts indépendants, les seuls problèmes qu'ils peuvent rencontrer viennent des conflits éventuels d'accès à des ressources communes. On peut alors distinguer deux grandes catégories de situations:

1. Les premières ont trait à l'accès d'une même portion d'espace par un ensemble d'agents. Il s'agit alors pour eux d'accomplir leurs buts tout en s'évitant, comme dans le cas du contrôle aérien. Les techniques les plus utilisées dans

ce cadre sont celles qui se rapportent à l'utilisation de champs de potentiels définissant des comportements attractifs et répulsifs.

2. Les secondes portent sur l'utilisation d'une ressource ponctuelle, telle qu'un outil commun. Il est alors possible d'utiliser des mécanismes de synchronisation semblables à ceux que nous avons étudiés précédemment section 8.2

Un problème particulier se pose lorsque les agents ont des buts dépendants et que les actions des uns peuvent améliorer celles des autres et donc augmenter les performances du groupe tout entier. Le principe général consiste à utiliser les capacités des agents réactifs à réagir aux modifications de l'environnement et souvent à marquer cet environnement pour coordonner les actions des agents entre eux. De ce fait, presque toutes les techniques de coordination réactives se résument à l'utilisation de quelques techniques essentielles:

1. Utilisation de *champs de potentiels* ou plus généralement de *champs de vecteurs* pour la détermination du déplacement des agents mobiles.
2. Utilisation de *marques* pour coordonner l'action de plusieurs agents, ces marques permettant d'utiliser l'environnement comme un système de communication souple, robuste et simple.

8.4.2 Du comportement de meute aux systèmes anti-collisions

L'un des premiers comportements que l'on peut demander à des agents réactifs situés est de savoir évoluer en groupe tout en s'évitant mutuellement.

Les techniques élémentaires

Le premier à s'être intéressé à la définition de comportements collectifs de meutes (flocking) est Craig Reynolds. En 1987, il produisit un modèle de meute fondé sur l'observation par chacun d'un comportement extrêmement simple (Reynolds 1987). Le comportement agrégatif de ses créatures qu'il appelle "boids" (prononcer "boïd") est un phénomène émergent qui résulte de leur interaction, chacun d'eux se contentant de suivre les règles comportementales suivantes:

1. Maintenir une distance minimale par rapport aux autres objets dans l'environnement et en particulier les autres boids.
2. Adapter sa vitesse à la moyenne de celle de ses voisins.
3. Aller vers le centre de gravité des boids voisins.

Ces règles sont suffisantes pour que les boids adoptent un comportement semblable à ceux d'un vol d'oiseaux migrateurs. A l'encontre des escadrilles aériennes, qui possèdent un chef, les boids évoluent sans leader et sans contrôle global. Leur comportement est particulièrement fluide et semble très naturel lorsqu'ils évitent tous ensemble des obstacles en restant naturellement groupés. Parfois la meute se divise et produit deux groupes qui se meuvent indépendamment l'un de l'autre. Parfois aussi deux sous-groupes se réunissent pour produire un nouveau groupe.

Des robots collectifs

Les agents réactifs peuvent faire preuve d'une grande souplesse dans la réalisation d'une tâche et montrer des comportements collectifs émergents relativement spectaculaires à partir de principes très élémentaires. En reprenant les travaux de C. Reynolds et en les adaptant à des robots réels, M. Mataric, du Mobot Lab de R. Brooks au M.I.T., a constitué de véritables meutes robotiques (Mataric 1992; Mataric 1994). Chaque robot dispose d'une batterie de comportements de haut niveau que ce dernier sélectionne en fonction de la situation dans laquelle il se trouve: évitement de collision, filature d'un autre robot, dispersion et répartition sur un territoire, agrégation, guidage sur un but. Ces comportements font appel à des primitives de bas niveau telles que la perception par infrarouge d'un obstacle ou le calcul d'une distance par sonar. Voici, par exemple, les comportements de filature et d'agrégation.

Comportement filature

```
Si il y a un objet sur la droite seulement
    alors aller à droite.
Si il y a un objet sur la gauche seulement
    alors aller à gauche.
Si il y a un objet sur la gauche et la droite
    alors continuer tout droit et compter le temps.
Si il y a un objet sur la gauche et la droite qui
    sont là depuis un moment, alors stopper.
```

Comportement Agrégation

```
Si aucun robot n'est trouvé dans une certaine zone,
    alors aller vers un centre de gravité local.
```

Le comportement de meute est donné comme la somme pondérée des comportements d'évitement, de filature, d'agrégation et de dispersion:

Comportement Meute

```
Pondérer les sorties des actions d'évitement,
    de filature, d'agrégation et de dispersion
    et calculer un vecteur de déplacement.
Si il se trouve au début de la meute
    alors ralentir.
Si il se trouve à la fin de la meute,
    alors accélérer.
```

D'après M. Mataric, le comportement de meute est très difficile à implémenter lorsqu'il s'agit de véritables robots, à cause de la dynamique propre de leur mécanique ainsi qu'aux limitations des capteurs. Il faut alors jouer sur les paramètres de pondération et faire attention que les robots ne cherchent pas trop à s'éloigner ou, au contraire, en viennent à former des groupes trop compacts qui gêneraient leurs évolutions. Néanmoins, et à condition de bien ajuster ces paramètres, il est possible

de faire émerger de tels comportements, que les agents soient simulés ou réels. On peut cependant se poser la question de savoir si les problèmes viennent effectivement des capteurs ou bien s'ils proviennent d'une analyse encore trop élémentaire des comportements.

Les bancs de poissons

Car en effet, si ces approches s'inspirent de la nature, elles n'ont parfois qu'un lointain rapport avec la manière dont les animaux se comportent effectivement. Qui n'a jamais été impressionné par ces comportements collectifs que sont les vols d'oiseaux migrateurs, les bancs de poissons, les meutes de loups ou les hordes de gazelles? Comment ces animaux font-ils pour évoluer ainsi en groupe et accomplir parfois des figures fort complexes? Comment synchronisent-ils leurs actions pour donner l'impression qu'un chef dirige leur chorégraphie?

C'est pour répondre à ces questions que R. Meslé — sous ma direction et en collaboration avec J.-P. Treuil et P. Fréon de l'Orstom — a développé un programme de simulation de bancs de poissons, *ICHTYUS* (fig. 8.26), montrant comment il est possible de comprendre leur constitution et leur évolution à partir de considérations entièrement distribuées (Mesle 1994).

Un banc de poissons est une agrégation spatiale d'un certain nombre d'individus fondée sur une approche égalitaire, tous les membres ayant la même influence. Ces bancs ont des comportements collectifs particulièrement intéressants, car leurs mouvements semblent parfaitement synchronisés comme s'ils répondaient tous au même signal ou s'il existait un organe de commande. De plus, tout en agissant de manière parfois brusque, les poissons ne se heurtent jamais, tous les mouvements semblant toujours formidablement coordonnés.

L'approche distribuée permet de rendre compte de ce phénomène. Il suffit de considérer que chaque poisson est un agent dont la direction est fonction de l'influence de l'ensemble de ses voisins, c'est-à-dire des poissons se trouvant à proximité, et que le comportement collectif émerge de ces interactions. Chaque poisson calcule la direction qu'il doit prendre en fonction de sa propre direction antérieure et des directions et des positions des différents poissons qui l'entourent, en effectuant un calcul de somme vectorielle pondérée par la distance. Le résultat est impressionnant: à partir d'une situation initiale, dans laquelle les poissons sont placés de manière aléatoire, et en faisant tourner une simulation portant sur plusieurs cycles (de 40 à 100 cycles environ), on constate que les poissons tendent en effet à construire des bancs dans lesquels les agents coordonnent leurs mouvements pour aller dans la même direction.

De plus, la présence d'un obstacle ne les gêne nullement: ils l'évitent de concert comme s'ils s'étaient donné le mot. Tous les poissons évitent l'obstacle tout en restant groupés sans que l'on puisse dégager un leader, ni même un déclencheur du mouvement.

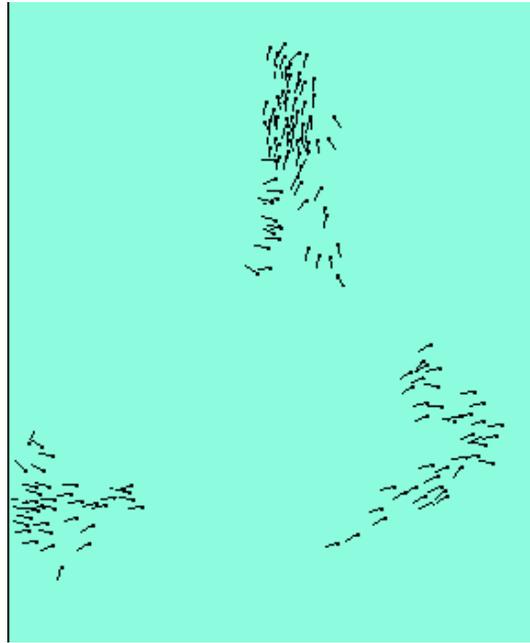


Figure 8.26: Les poissons se rassemblent en bancs dans ICHTYUS.

Les champs de forces symétriques

Si le comportement de meute peut s'expliquer par une simple combinaison vectorielle en est-il de même lorsque plusieurs agents mobiles, des avions par exemple, doivent coordonner leurs actions pour s'éviter mutuellement à des distances importantes tout en essayant de suivre leurs plans de vols? Nous avons vu au chapitre 4 qu'il était possible de modéliser le déplacement d'un agent à partir des forces attractives et répulsives résultant de la combinaison de potentiels correspondant à des buts ou à des obstacles. Peut-on généraliser cette méthode à des agents mobiles? Dans ce cas, les agents sont des obstacles qui modifient le paysage des potentiels des autres agents. Chacun d'eux construit sa propre vision du monde, son propre champ de potentiels, à partir des informations dont il dispose et donc de ce qu'il perçoit. Les potentiels résultants sont donc définis pour chaque agent A_i comme la somme du potentiel attractif correspondant au but avec la somme de tous les potentiels répulsifs définis pour les autres agents à éviter.

$$U_i = U_i^{attr} + \sum_{j=1}^n U_j^{rep}$$

La force résultante s'exprime alors comme le gradient (c'est-à-dire la dérivée) de ce potentiel au point p :

$$\vec{F}_i(p) = -\vec{\nabla} U_i(p)$$

Mais cette technique, si elle s'avère très simple à mettre en œuvre, présente néanmoins un inconvénient majeur: du fait que les agents se déplacent, les champs de potentiels sont continuellement modifiés, et les trajectoires que suivent les agents deviennent parfois aberrantes. Par exemple, la figure 8.27 montre une situation caractéristique

introduite par les champs de potentiels. Deux agents qui suivent des routes qui se croisent cherchent à s'éviter et donc à fuir. Pour éviter l'agent A_2 , l'agent A_1 s'écarte vers la gauche, mais l'agent A_2 en cherchant lui aussi à éviter A_1 incurve sa trajectoire vers la droite. Bien que ces réactions soient logiques du point de vue de chaque agent, elles conduisent à des modifications de trajectoires inadéquates. Cela est dû au fait que, en utilisant simplement des techniques fondées sur le gradient de potentiels, les agents n'arrivent pas à tenir compte du déplacement des autres agents, la prise de décision locale ne permettant pas de résoudre le problème de manière globale.

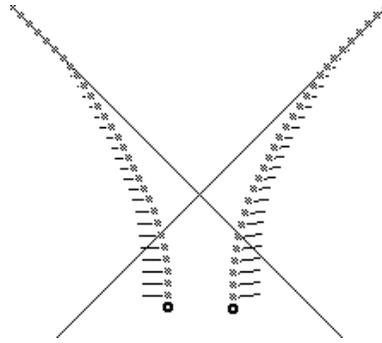


Figure 8.27: Une conséquence malheureuse de l'utilisation de champs de potentiels lorsque le comportement d'évitement est réduit à celui d'une fuite.

Il est cependant possible de résoudre ce problème en introduisant une nouvelle force, une *force de glissement*, comme l'a fait K. Zeghal⁵ (Zeghal 1993), qui permet d'embrasser les problèmes de coordinations spatiales tout en restant dans la même logique de champs de force. L'idée est très simple: elle consiste à dire que pour contourner un obstacle il suffit de rester sur une ligne qui conserve la même valeur du potentiel et donc à tenter de rester à même distance d'un obstacle. Mathématiquement, cela revient à dire que cette force \vec{F}_{ij} qui agit sur l'agent A_i à partir d'un potentiel provenant de l'agent A_j est perpendiculaire au gradient du potentiel, ce qui est donné par l'équation suivante:

$$\vec{F}_{ij}(p) \cdot \vec{\nabla} U_{ij}(p) = 0$$

De plus l'intensité de cette force en un point p est égale à celle du gradient généré par A_j :

$$\|\vec{F}_{ij}(p)\| = \|\vec{\nabla} U_{ij}(p)\|$$

Il est facile de donner une image de cette théorie: le meilleur moyen pour contourner un obstacle n'est pas de le fuir, mais d'essayer de se déplacer en restant toujours à une distance supérieure à une distance limite, ce que les alpinistes font naturellement lorsqu'ils tentent de rester sur une ligne de même altitude, une ligne "iso" dans

⁵De manière indépendante, De Medio et Oriolo d'une part (De Medio et Oriolo 1991) et Masoud et Bayouni d'autre part (Masoud et Bayouni 1993) ont proposé des techniques voisines pour traiter le problème de l'évitement entre agents mobiles autonomes dans le cadre de la robotique.

leur jargon. L'intensité de cette force est donnée comme étant égale à l'intensité du gradient au même point, ce qui signifie que plus on est loin, moins on cherche à contourner un obstacle, et plus on est prêt, plus on doit effectivement avoir un déplacement qui se rapproche de la perpendiculaire du gradient. Enfin, il s'agit de savoir de quel côté on doit chercher à contourner l'obstacle. Si cela est (presque) indifférent lorsqu'il s'agit d'un obstacle fixe ou d'un agent ne réagissant pas à la présence d'autrui, il n'en est pas de même lorsque deux agents réagissent de la même manière, comme nous l'avons vu à la figure 8.27. Pour passer d'un agent cherchant simplement à éviter un obstacle à une interaction d'évitement entre agents, il suffit de donner à la force de glissement le sens défini par la différence des vecteurs vitesse, comme le montre la figure 8.28. Lorsqu'il existe la moindre petite différence entre les vecteurs vitesse des deux agents, les vecteurs vitesse vont dans des sens opposés, ce qui permet de choisir la bonne direction d'évitement.

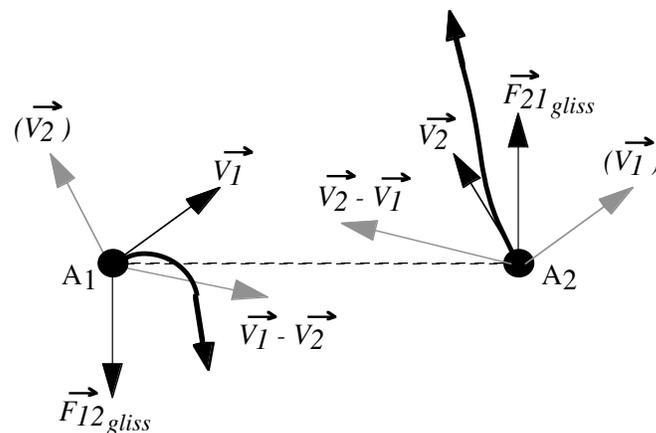


Figure 8.28: Contournement symétrique de deux mobiles. Les formes de glissement ont une direction perpendiculaire au gradient du potentiel de répulsion, une intensité égale à celle du gradient et leur sens est donné par la différence des vecteurs vitesse des agents.

Des situations intéressantes émergent de ce type de coordination d'actions. Lorsque plusieurs agents doivent s'éviter simultanément, des structures dynamiques de tourbillon apparaissent spontanément, donnant l'impression que les agents tournent autour d'un rond-point (fig. 8.29).

K. Zeghal a implémenté cette technique dans le système CRAASH, qui simule l'évolution d'un ensemble d'avions disposant ou non de son système d'anti-collision (Zeghal et Ferber 1993). Il a ainsi pu montrer que plus d'une cinquantaine d'appareils civils (Airbus, Boeing, etc.) étaient capables d'évoluer pour tous venir atterrir de manière automatique sur le même aéroport, sans entrer en collision.

8.4.3 Marquage de l'environnement

Nous avons vu l'importance des techniques utilisant des champs de vecteurs pour définir les positions respectives des buts, des obstacles et des agents, et permettre

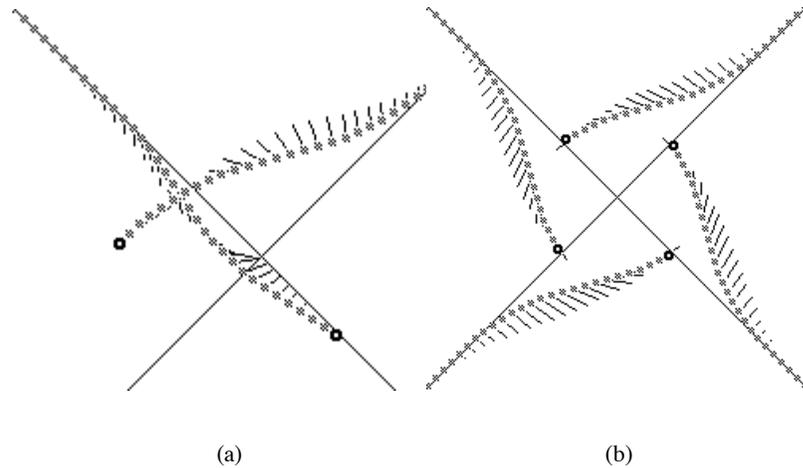


Figure 8.29: L'introduction d'un comportement d'évitement par l'usage de forces glissantes permet non seulement de résoudre le problème posé par les champs de potentiels (a), mais aussi de provoquer des structures qui émergent des interactions tels que celle du "rond point" (b). Cette dernière apparaît alors qu'aucune règle de priorité n'est donnée (d'après (Zeghal 1993)).

ainsi à ces derniers de coordonner leur déplacement dans l'espace. Mais elles ne se bornent pas à la gestion des mouvements. Elles se généralisent facilement à des problèmes de coordination, le marquage de l'environnement constituant la clé de la coordination coopérative d'agents réactifs.

Qu'est-ce qu'une marque? Une marque est un signe matériel ou une empreinte qui sert à indiquer artificiellement l'environnement afin de coordonner les actions des agents. Les marques peuvent être déposées, lues et retirées par les agents, mais leur effet peut aussi disparaître lentement, par "évaporation naturelle" dans l'environnement.

Marques et synchronisation d'actions

Connah et Wavish (Wavish et Connah 1990) ont montré la difficulté qu'il pouvait y avoir de concevoir un ensemble d'agents réactifs coopérants et, en particulier, qu'il ne suffisait pas d'ajouter "bêtement" des agents réactifs pour qu'automatiquement les performances du système soient améliorées.

Par exemple, supposons que l'on cherche à assembler dynamiquement des composants de manière à réaliser un appareil électronique tel qu'un téléviseur. Dans une pièce comprenant un tapis roulant et cinq casiers contenant cinq types de composants pour téléviseurs (carcasses, tubes, déflecteurs, circuits électroniques et vis), des robots doivent placer les composants dans un ordre précis au point d'assemblage (fig. 8.30). On considère que, lorsque ces cinq composants ont été placés sur le tapis roulant dans cet ordre, la télévision est achevée et la tâche finie (Hickman et Shiels 1991).

Le comportement de chaque agent est défini sous la forme d'un ensemble de

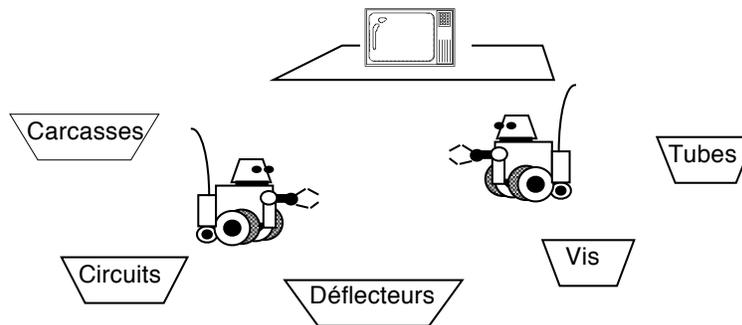


Figure 8.30: Des robots doivent assembler des composants de téléviseurs dans un ordre précis, les pièces détachées se trouvant dans des casiers séparés.

règles de production qui implémentent des actions situées (cf. chap. 4). Si un agent distingue une certaine situation, et notamment ce qui se trouve au point d'assemblage, il déclenche une action associée à la situation perçue. Par exemple, si un agent s'aperçoit qu'une pièce particulière se trouve déjà au point d'assemblage, il peut déclencher la recherche de l'élément suivant.

Règle prendre

```

si je ne porte rien
et je suis devant un casier
alors je prend ce qu'il y a dans le casier
et je me dirige vers le point d'assemblage

```

Règle déposer

```

si je porte quelque chose
et je suis au point d'assemblage
alors je dépose ce que je tiens

```

Règle chercherDéflecteur

```

si je ne porte rien
et je suis au point d'assemblage
et je perçois un tube au point d'assemblage
alors je me dirige vers le casier des déflecteurs

```

Règle chercherCircuitElectronique

```

si je ne porte rien
et je suis au point d'assemblage
et je perçois un déflecteur au point d'assemblage
alors je me dirige vers le casier des
circuits électroniques

```

Les deux premières règles sont générales, mais les suivantes doivent être hiérarchisées. En effet, la règle `chercherCircuitElectronique` doit être prioritaire devant `chercherDéflecteur`, puisque, s'il existe un déflecteur, il existe aussi un tube, mais

la perception de ce dernier ne doit pas pour autant déclencher la recherche d'un autre déflecteur.

Avec un seul agent, tout se passe bien: il suffit qu'il se présente au point d'assemblage et constate qu'il n'y a rien pour se diriger vers le casier des carcasses ou qu'il aille directement devant ce casier pour que les composants soient placés dans le bon ordre. Mais que se passe-t-il avec seulement deux agents? Avec le même comportement, ils risquent de déposer certains composants en double. En effet, lorsque l'agent A voit qu'il y a un tube au point d'assemblage, il va naturellement chercher un déflecteur; mais si l'agent B arrive au même endroit avant qu'A ne revienne, il ira lui aussi chercher un déflecteur, ce qui aura pour conséquence de doubler le nombre de déflecteurs. Le comportement de B est donc redondant et mal synchronisé avec celui de A. On peut résoudre ce problème de plusieurs manières.

La solution présentée dans (Hickman et Shiels 1991) consiste à donner aux agents la possibilité de voir ce que les autres agents transportent: si l'agent B voit que A transporte un déflecteur, il se dirigera vers le casier des circuits électroniques, évitant ainsi de ramener un autre déflecteur. Mais cette solution n'est absolument pas satisfaisante et pose d'autres problèmes. D'abord B doit non seulement voir que A transporte un déflecteur, mais aussi qu'il *a l'intention* d'en apporter un lorsque A se dirige vers les casiers, ce qui est en contradiction avec l'hypothèse de réactivité associée aux actions situées. De plus, cette technique n'est pas très généralisable. S'il y a un nombre n d'agents, chacun doit voir ce que les $n - 1$ autres portent et donc être à même de décider à partir d'une perception globale de la configuration des autres agents, hypothèse peu applicable dans un contexte réel, notamment à cause de l'accroissement exponentiel du nombre de règles.

Heureusement, des solutions plus simples existent, qui consistent à marquer ce que les agents sont en train de faire au point d'assemblage. Par exemple, si l'agent A décide d'aller chercher un déflecteur, il placera une marque de la forme RechercheDéflecteur. De ce fait, un agent B arrivant après et voyant cette marque pourra se diriger directement vers le casier des circuits électroniques. Il suffit pour cela de légèrement modifier les règles de recherche de manière à prendre en compte la perception des marques. Voici, par exemple, la nouvelle règle de recherche d'un circuit électronique:

```
Règle chercherCircuitElectronique
  si je ne porte rien
  et je suis au point d'assemblage
  et je perçois un déflecteur au point d'assemblage
  ou une marque RechercheDéflecteur
  alors je me dirige vers le casier des circuits électroniques
  et je dépose une marque RechercheCircuitElectronique
```

Dans ce cas, les marques jouent le rôle d'indices pour la coordination des actions entre les agents et empêchent les actions inefficaces éventuelles. Il est aussi possible d'utiliser les marques pour amplifier les performances des agents en servant à la fois d'indice de collaboration et de coordination d'actions.

Marques et amélioration des performances

L'utilisation des marques peut aussi servir à améliorer les performances d'un groupe et à former des structures autocatalytiques, c'est-à-dire des organisations qui émergent des interactions. Les premiers travaux ayant porté sur l'usage de telles marques pour résoudre le problème de la récupération de minerai sont dus à J.-L. Deneubourg (Deneubourg et Goss 1989) et à L. Steels (Steels 1989). Une discussion des approches possibles ainsi que de leur efficacité peut être trouvée dans (Drogoul et Ferber 1992).

L'exemple des robots explorateurs (cf. chap. 1) permet en effet de comprendre comment l'utilisation des marques peut servir à réduire l'espace de recherche des robots et donc à améliorer leur productivité. Rappelons que le problème de l'exploration consiste pour les robots à trouver des échantillons de minerai et à les rapporter à la base. Le comportement des robots peut donc être décrit simplement par la succession de phases d'exploration et de transport des échantillons jusqu'à la base. En termes d'actions situées, on pourrait décrire ce comportement de la manière suivante:

Règle explorer

```
si je ne porte rien
et je ne perçois aucun minerai
alors j'explore de manière aléatoire
```

Règle trouver

```
si je ne porte rien
et que je perçois du minerai
alors je prend un échantillon du minerai
```

Règle rapporter

```
si je porte du minerai
et je ne suis pas à la base
alors retourner à la base
```

Règle déposer

```
si je porte du minerai
et je suis à la base
alors déposer le minerai
```

Nous considérerons que retourner à sa base n'est pas un problème pour un robot et qu'il lui est toujours possible de revenir à son point d'attache. Par exemple, on supposera que la base émet un signal inversement proportionnel à sa distance (ou au carré de sa distance) qui permet à chaque robot de trouver son chemin. Le problème demeure donc celui de la découverte des échantillons. Si ces derniers sont disséminés de manière aléatoire et que les robots n'ont aucun indice de leur présence, les robots suivront un déplacement aléatoire jusqu'à ce qu'ils perçoivent un tas de minerai pour en rapporter un échantillon à la base.

L'inconvénient de cette technique est d'être relativement coûteuse, puisque, si les agents se déplacent de manière aléatoire, le temps mis pour trouver un tas de minerai est proportionnel à la surface d'exploration. Dans ce cas, les actions des robots ne sont absolument pas coordonnées, et tout le travail effectué par un robot pour trouver un tas de minerai est totalement perdu pour les autres. Une méthode qui peut être employée pour améliorer l'efficacité de la recherche consiste évidemment à ce que les agents se communiquent la position des tas de minerai. Plusieurs solutions utilisant des communications réactives (cf. chap. 6 et 7) peuvent être envisagées. En s'inspirant de la façon dont certaines fourmis se transmettent la position des sources de nourriture en déposant de petites quantités de phéromones, Deneubourg et Steels ont imaginé que des robots explorateurs pourraient déposer des marques dans l'environnement lorsqu'ils ont trouvé du minerai et qu'ils retournent à la base. Ces marques améliorent le processus de recherche du minerai car, lorsqu'un robot en phase d'exploration perçoit une marque il se dirige naturellement vers cette marque et remonte ainsi jusqu'à la source du minerai. Le comportement de ces robots est alors modifié de la manière suivante (on modifie les règles `explorer` et `rapporter` et on ajoute la règle `suivreMarque`):

Règle `explorer`

```
si je ne porte rien
et je ne perçois aucun minerai
et je ne perçois aucune marque
alors j'explore de manière aléatoire
```

Règle `rapporter`

```
si je porte du minerai
et je ne suis pas à la base
alors retourner à la base
et déposer une marque
```

Règle `suivreMarque`

```
si je ne porte rien
et je ne perçois aucun minerai
et je perçois une marque
alors je me dirige vers cette marque
```

Un robot ayant trouvé du minerai et qui retourne vers la base va créer un "chemin d'odeur", c'est-à-dire une suite de marques déposées sur le sol. Les robots en train d'explorer vont se trouver captés par le chemin et auront donc tendance à suivre les traces, ce qui les amènera normalement vers le tas de minerai. De ce fait, avec le temps, de plus en plus de robots seront amenés à trouver le tas de minerai et donc à marquer eux-mêmes le chemin et à déposer des marques, ce qui aura pour effet de renforcer la tendance des robots à suivre ce chemin, dans un phénomène autocatalytique caractéristique des rétroactions positives.

La figure 8.31 montre la situation d'un ensemble de robots cherchant des échantillons de minerai et utilisant des marques pour améliorer le processus. Cela

donne l'impression que les agents suivent un chemin d'odeurs pour retrouver l'échantillon de minerai.

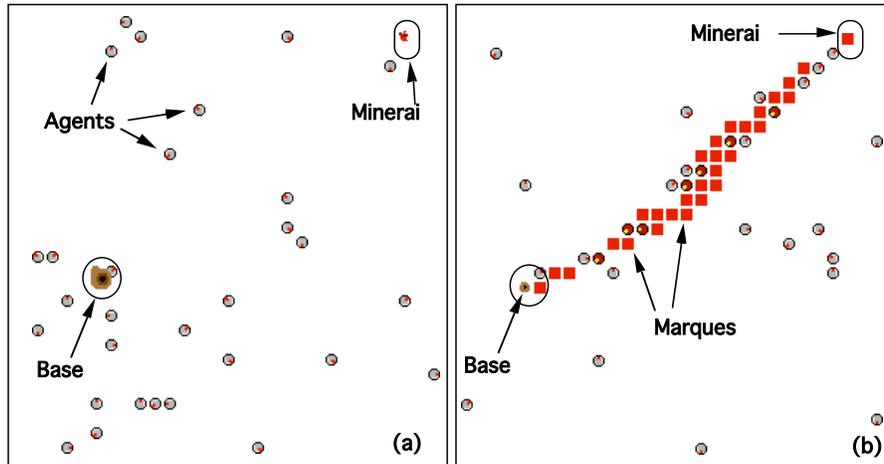


Figure 8.31: Utilisation de marques par des robots pour rapporter du minerai à la base et améliorer leurs performances collectives

Cependant, l'utilisation de marques pose parfois un problème: leur présence peut induire des agents en erreur. Par exemple, si la source de minerai est éteinte et s'il reste des marques, alors les agents suivront les marques pour aboutir à un endroit maintenant vide. Il faut donc que les marques soient elles-mêmes des éléments dynamiques capables de disparaître lorsqu'il n'y a plus d'échantillon à aller chercher, c'est-à-dire que le chemin soit volatil et disparaisse avec le temps. Deux solutions sont envisageables:

1. La première consiste à utiliser des marques volatiles. Si le chemin n'est pas constamment renforcé par le dépôt de nouvelles marques, les marques disparaîtront. Cette technique, qui s'apparente à ce qui se passe dans la nature avec des fourmis, nécessite seulement une bonne paramétrisation de la durée de rémanence des marques. Si celle-ci est trop grande, on se retrouve dans la situation précédente qui "piège" des robots alors que cela n'est plus nécessaire, et si elle est trop faible, le mécanisme de marquage n'aura pas le temps de fonctionner pour de nouvelles fourmis.
2. On peut aussi faire en sorte que les agents enlèvent une partie des marques. Au retour, au lieu de ne déposer qu'une seule marque, les robots en déposent deux et, à l'aller, lorsqu'ils suivent le chemin des marques, ils en reprennent une. De ce fait s'il n'existe plus de minerai, les robots piégés par le chemin des marques élimineront les marques déposées préalablement, ce qui aura pour effet de "nettoyer" le chemin de toutes les marques inutiles.

8.4.4 Actions de coordination

Nous avons envisagé dans les sections précédentes l'utilisation de marques déposées dans l'environnement pour coordonner les actions d'agents réactifs. Mais il est possible de coordonner les actions de tels agents même en l'absence de marques. Il faut alors ajouter des *actions de coordination* qui offrent aux agents, au-delà des mécanismes de synchronisation que nous avons vus plus haut (cf. section 8.2), la possibilité de s'entraider (ou simplement de ne pas se gêner) dans leurs tâches.

Un problème particulièrement intéressant est celui de la formation d'une chaîne de transport d'objets. Le déplacement d'objets d'un point à un autre est une des activités fondamentales des organisations humaines. La production des biens ne se trouve généralement pas au même endroit que leur consommation et il faut donc les transporter. Mais comment transporter ces objets d'un point X à un point Y? Il existe essentiellement trois techniques de transport de bases. La première consiste à porter directement un objet de X à Y sur toute la longueur du trajet. Mais, dès que la distance devient longue et que la quantité d'affaires à transporter devient importante, il s'avère plus rentable d'utiliser des moyens de transport mécaniques qui permettent de transférer de grandes masses d'objets sur de longues distances. Ces moyens vont du char à bœuf à l'avion cargo, en passant par les véhicules routiers, le chemin de fer, les bateaux et la brouette. Le problème consiste alors à *charger* ces systèmes de transport, c'est-à-dire à poser un problème semblable mais sur des distances beaucoup plus petites, celles qui séparent le point d'origine du lieu où se trouve le véhicule. Soit on porte tout à la force de ses bras, soit on utilise des techniques intermédiaires permettant de transporter sur de plus courtes distances. Mais il existe une autre technique de transport qui ne peut s'effectuer qu'à plusieurs et qui consiste à *faire la chaîne*, c'est-à-dire à se passer un objet de mains en mains depuis son lieu d'origine jusqu'à sa destination. A la différence des deux premières techniques, l'objet n'est pas porté par la même personne (ou par le même système de transport) pendant tout son trajet. Il est au contraire continuellement passé à un tiers, et c'est son transfert d'agent à agent qui assure son déplacement.

Cette technique suppose deux choses: qu'il existe un grand nombre d'agents, d'une part, et que le transfert de l'objet d'un agent à un autre puisse s'effectuer de manière aisée, d'autre part. Mais l'organisation d'une chaîne, les gestes de transfert des objets nécessitent une grande coordination des actions pour que les objets arrivent à bon port. Cette coordination peut être décrite simplement à partir du comportement des agents récupérateurs de minerai vus précédemment. Lorsqu'ils vont du lieu d'origine (X) vers le lieu de destination (Y) et qu'ils portent quelque chose, les agents transmettent leur charge à celui qui revient de Y les mains vides (Drogoul et Ferber 1992):

```
Règle explorer
  si je ne porte rien
  et je ne perçois aucun minerai
  et je ne perçois aucun agent avec du minerai
  alors explorer de manière aléatoire
```

```
Règle donnerUnCoupDeMain
  si je ne porte rien
  et je ne perçois aucun minerai
  et je perçois un agent avec du minerai
  alors proposer à l'agent mes services
  et retourner à la base

Règle rapporter
  si je porte du minerai
  et je ne suis pas à la base
  et aucun agent ne me propose ses services
  alors retourner à la base

Règle échanger
  si je porte du minerai
  et je ne suis pas à la base
  et un agent me propose ses services
  alors donner mon chargement à l'agent
  et retourner d'où je viens
```

Il est alors possible de constituer des chaînes dynamiques qui se forment dès qu'il existe un tas de minerai à rapporter. Les agents qui reviennent donnent leur chargement à ceux qui vont vers la source et repartent charger. Mais en repartant, ils peuvent être amenés à rencontrer des agents qui reviennent de la source et donc à prendre leur charge et à retourner vers la base. De ce fait, très vite, les agents ne se déplacent plus le long de l'ensemble du chemin, mais ne font qu'aller et venir sur de courtes distances. Un agent libre s'empare de la charge d'un agent porteur de minerai, la transporte jusqu'à trouver un agent libre et retourne vers la source de manière continue. Si le nombre des agents devient plus important, les agents ne bougent plus de place et se contentent alors de transférer leur charge d'un voisin à l'autre, créant ainsi une véritable chaîne d'agents. La figure 8.32 montre deux exemples de chaînes. Le premier (a) ne comprend qu'une seule chaîne puisqu'il n'y a qu'une seule source de minerai, alors que le second (b) forme une fourche du fait de la présence de deux sources. On pourra noter que ces chaînes comprennent parfois des "trous", des agents se déplaçant alors entre les segments.

L'intérêt de cette approche est de montrer qu'avec une toute petite modification du comportement des agents, il est possible de construire des structures de coordination possédant des propriétés très intéressantes, telles que celle de produire des structures stationnaires "ouvertes", qui restent stables tant qu'elles sont approvisionnées en éléments "extérieurs". Elles se comportent alors comme des structures autocatalytiques, c'est-à-dire des systèmes ouverts qui restent stables en dehors de leur point d'équilibre.

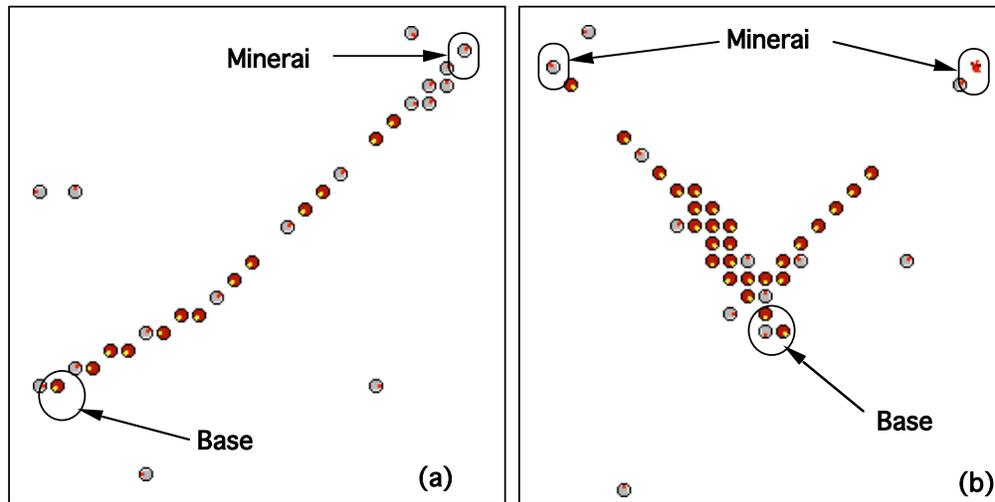


Figure 8.32: Un exemple de chaînes. La première (a) ne comprend qu'un seul bras, puisqu'il n'y a qu'une seule source, alors que la seconde (b) forme une fourche vers chacune des sources.

8.5 Résolution par coordination : l'éco-résolution

Alors que toutes les techniques que nous venons de voir se chargent d'aider à la coordination des actions d'agents définis "au préalable", nous allons voir qu'il est aussi possible d'utiliser de telles approches pour résoudre des problèmes classiques en reformulant le problème. L'éco-résolution, en particulier, est une méthode de résolution de problème qui prend le contre-pied des techniques classiques. Plutôt que de formaliser un problème de manière globale et de définir ensuite des méthodes de résolution s'appliquant directement à sa définition, on considère qu'il est préférable de reformuler le problème de manière qu'il soit conçu comme un ensemble d'agents en interaction qui tentent de satisfaire individuellement leur propre but.

8.5.1 Principes de l'éco-résolution

Dans le cadre de l'éco-résolution, appréhender un problème, c'est définir une population d'agents dont l'ensemble des comportements tend à aboutir à un état stable, que l'on appelle la solution du problème. Chaque agent répond aux principes d'autonomie et de localité, c'est-à-dire que les actions qu'il entreprend sont la conséquence à la fois de perceptions locales et des relations qu'il a établies avec d'autres agents.

Au cours de leur comportement, ces agents effectuent parfois des actions particulières qui, si elles sont mémorisées, correspondent directement aux actions qui devraient être effectuées par un système centralisé (un programme de planification ou un individu) pour aboutir au résultat. Le plan est alors obtenu comme un effet secondaire (un effet de bord disent les informaticiens) du comportement des agents. Les particularités essentielles de cette approche sont les suivantes:

- A. Il n'y a pas d'exploration globale de l'espace des états possibles du monde. Les états du monde ne sont pas directement utilisés par l'éco-résolution. Seuls les états internes des agents et leur relation perceptive avec l'environnement sont pris en compte.
- B. Ces systèmes résistent très bien au bruit. Une perturbation ne modifie pas le mécanisme de résolution. En fait, une perturbation est presque une donnée normale dans l'évolution du système et se trouve donc prise en compte sans aucune difficulté. C'est d'ailleurs cette caractéristique qui rend très intéressants ces systèmes dans des univers évolutifs où de nouvelles informations arrivent en permanence.
- C. De ce fait, cette méthode ne présente que très peu d'explosion combinatoire, ce qui la rend capable de résoudre des problèmes de taille relativement plus importante.

ECO est un modèle général de système de résolution de problèmes fondé sur l'éco-résolution, qui se compose de deux parties:

1. Un noyau qui définit le protocole suivi par l'ensemble des agents. Ce noyau est absolument indépendant du domaine du problème à résoudre. Il consiste en une définition abstraite des agents du système, appelés *éco-agents*, et surtout des interactions entre les différents comportements élémentaires des agents. Ces comportements très simples peuvent être définis à l'aide d'un automate d'état comme nous le verrons ci-dessous.
2. Une partie dépendante du domaine d'application, où les comportements des agents spécifiques au domaine sont codés.

8.5.2 Les éco-agents

Chaque agent dispose d'un ensemble de comportements élémentaires prédéfinis qui l'entraînent dans une quête perpétuelle d'un état de satisfaction. Dans cette recherche de satisfaction, ces agents peuvent être gênés par d'autres agents. Dans ce cas, ils agressent ces gêneurs, lesquels sont dans l'obligation de fuir. Au cours de cette fuite, ils peuvent être amenés à agresser d'autres gêneurs qui les empêchent de fuir, cette opération se poursuivant jusqu'à ce que les gêneurs se déplacent effectivement.

Structure des éco-agents

Chaque éco-agent peut être caractérisé par:

- Un *but*, c'est-à-dire un autre agent avec lequel il doit être dans une relation particulière, appelée *relation de satisfaction*, pour qu'il soit satisfait.
- Un *état interne*. Un éco-agent peut être dans l'un des trois états internes suivants: *satisfait*, *en recherche de satisfaction* ou *en recherche de fuite*. Le passage d'un état à un autre est donné par une table de transition caractéristique d'un automate à états finis (voir ci-dessous).

- Des *actions élémentaires*, dépendantes du domaine, qui correspondent aux comportements de satisfaction et de fuite des agents.
- Une fonction de *perception des gêneurs*, c'est-à-dire de l'ensemble des agents qui empêchent l'agent courant d'être satisfait ou de fuir.

A cela, il faut ajouter le concept des dépendances très souvent utilisé en ECO:

- Les *dépendances* sont les agents dont l'agent courant est le but, et ces dépendances ne pourront être dans l'état *satisfait* que si l'agent courant se trouve lui-même satisfait. Ces dépendances sont produites par la définition des buts, et donc par l'établissement des relations de satisfactions. On dit que A est directement dépendant de B si A est dans les dépendances de B (ou B est un but pour A). "Être dépendant de" est une relation d'ordre partielle qui peut s'énoncer ainsi: on dit que A *dépend de* B si B est directement dépendant de A ou s'il existe un autre agent C tel que A soit directement dépendant de C et tel que C dépende de B.

Comportement des éco-agents

Les comportement élémentaires d'un éco-agent sont indépendants du domaine d'application et leur interaction est définie dans le noyau d'ECO. Ces comportements peuvent être décrits comme des procédures⁶. Voici la définition simplifiée de ces fonctions :

1. *La volonté d'être satisfait*: Les éco-agents cherchent à se trouver dans un état de satisfaction. S'il ne peuvent y parvenir, parce qu'empêchés par quelques gêneurs, ils agressent ces derniers:

```
Procédure essayerSatisfaire(x)
  si le but de x est satisfait alors
    pour tous les agents y qui gênent x
      fuir(y, x, but(x))
  dès qu'il n'y a plus de gêneurs,
    alors faireSatisfaction(x)
```

La fonction `faireSatisfaction` est dépendante du domaine d'application et doit être adaptée pour chaque type d'éco-agent. Elle se charge d'exécuter l'opération dont le résultat aura pour conséquence que l'agent vérifie sa condition de satisfaction. Cette action est rendue possible par la fuite de l'ensemble des gêneurs.

2. *L'obligation de fuir*: Lorsqu'un éco-agent est attaqué, celui-ci doit fuir. Il doit alors chercher une situation qui satisfasse la contrainte `c` qui est passée en argument de la fonction `fuir`. Voici la définition de cette fonction:

```
Procédure fuir(x, y, c) // x fuis y avec la contrainte c
  si x était satisfait, x devient insatisfait
```

⁶En fait, ces fonctions étant dépendantes d'un type d'agent, il est plus facile de les représenter comme des méthodes d'un langage objet.

```

soit p := trouverPlacePourFuir(x,y,c)
  si P est NIL, alors "pas de solution",
  sinon
    pour tous les agents z qui gênent x dans sa
    fuite vers p,
      fuir(z, x, p)
    dès qu'il n'y a plus de gêneurs pour fuir,
      alors faireFuite(x,p)

```

Les fonctions `trouverPlacePourFuir` et `faireFuite` sont elles aussi dépendantes de l'application. La première cherche une place dans l'environnement où l'agent peut fuir, et la seconde réalise effectivement l'action de fuite.

Les éco-agents comme automates à états finis

Chaque éco-agent peut être vu comme un automate à états finis comprenant 4 états, 2 entrées et 3 sorties. Les états de l'automate correspondent aux états de satisfactions des éco-agents. Ces quatre états sont la *satisfaction (S)*, la *recherche de satisfaction (RS)*, la *recherche de fuite (RF)* et la *fuite (F)*⁷. L'état de départ est RS. Les deux entrées correspondent aux deux situations qui peuvent être observées par un éco-agent: l'agression par un autre agent et le fait d'avoir des gêneurs qui l'empêchent d'accomplir ses actions. Les commandes de sorties correspondent d'une part aux actions élémentaires (`faireFuite` et `FAIRESATISFACTION`) et d'autre part à un message d'agression envoyé à un autre agent. La figure 8.33 représente cet automate. Le graphe de l'automate est donné 8.34. Chaque changement d'état est

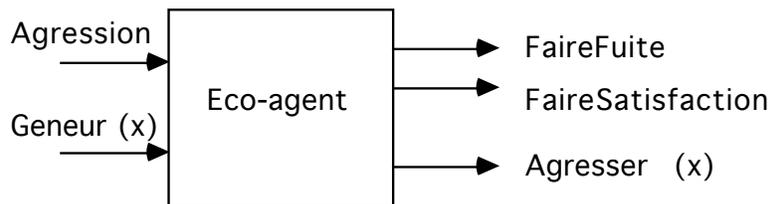


Figure 8.33: Représentation d'un éco-agent sous la forme d'un automate

caractérisé par un couple de variables d'entrée $\langle \text{Agression}, \text{Gêneurs} \rangle$, où a et g dénotent des variables d'entrée à 1, et $\neg a$, $\neg g$ des variables d'entrée à 0, la valeur x indiquant une indifférence. Une marque '-' indique qu'aucune action n'est effectuée en sortie.

8.5.3 Exemples simples d'éco-problèmes

Un éco-problème est défini par:

⁷Il existe parfois des agents qui n'ont pas de buts. Dans ce cas, l'automate est légèrement différent, car les agents ne cherchent pas à se satisfaire et ils ne possèdent qu'un comportement de fuite.

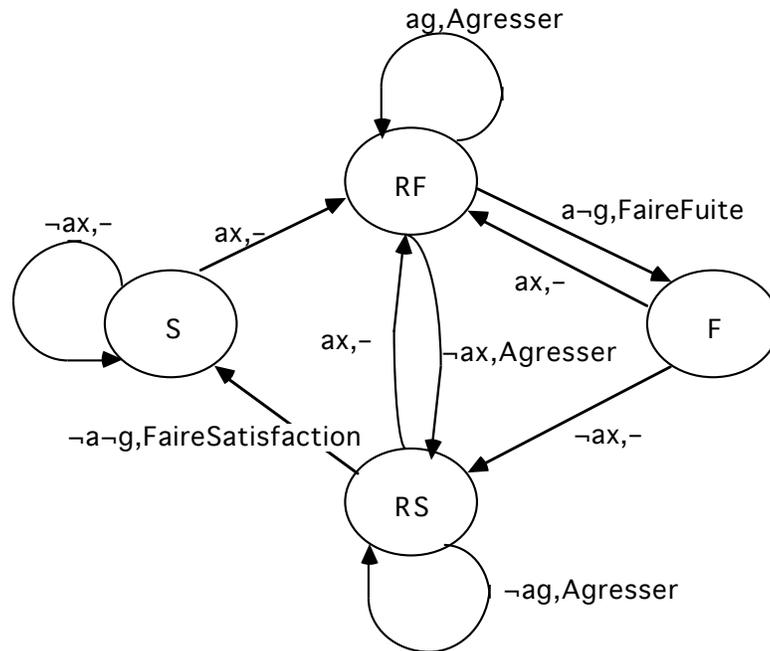


Figure 8.34: Le graphe des états d'un éco-agent

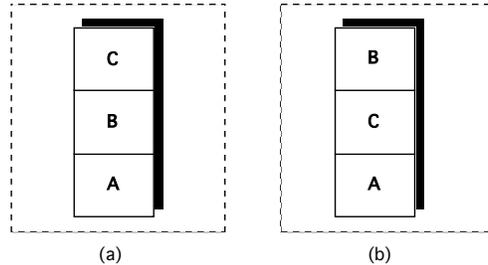
1. Un ensemble d'agents, chaque agent étant caractérisé par un but et un comportement (composé de l'automate et des actions élémentaires dépendantes de l'application).
2. Une configuration initiale, décrite par un ensemble d'agents placés dans leur état de départ (RS).
3. Un critère de terminaison défini comme une fonction des états de satisfaction des éco-agents. Dans la plupart des exemples suivants, ce critère correspond à la satisfaction de tous les agents présents.

De ce fait, l'étape la plus complexe pour résoudre un problème en ECO est de déterminer l'ensemble des agents et la manière de les utiliser. Elle nécessite une méthode d'analyse radicalement différente de la résolution de problème classique. Alors que cette dernière utilise généralement des opérateurs de transition d'état, l'éco-résolution suppose que *ce sont les entités du monde et non les opérateurs* qui interagissent. *Ce type de démarche est en fait à la résolution de problèmes et à la planification ce que la programmation par objet est à la programmation procédurale.* Ce ne sont plus les opérateurs qui sont le centre d'intérêt du système mais, au contraire, les objets sur lesquels portent ces actions.

Les cubes

L'exemple le plus simple et en même temps peut-être le plus parlant est donné par le problème, bien connu en planification, du monde des cubes. Il s'agit d'inverser deux cubes dans une pile, ce qui revient à échanger la valeur de 2 registres. Un cube ne

peut supporter qu'un seul cube et on ne prend qu'un seul cube à la fois. Supposons que l'on désire obtenir un état (b) à partir d'une configuration (a):



La solution classique consiste à explorer un ensemble d'états engendré par l'application de l'ensemble des opérations disponibles sur la configuration initiale, comme nous l'avions décrit au chapitre 4. Si l'on explore cet espace avec des opérateurs de type STRIPS, on est parfois conduit à des situations d'échec, en particulier si l'on a placé le cube B sur C avant de s'être assuré que C soit sur A. Cet échec réclame un retour en arrière de la part du planificateur, qui va essayer une autre issue. Pour pallier cet inconvénient, certains planificateurs modernes utilisent ou trouvent des heuristiques pour essayer d'effectuer le bon choix et donc limiter les retours en arrière.

En revanche, dans le cadre de l'éco-résolution, la solution consiste à donner aux agents des conditions de satisfaction relatives à l'état final qui doit être engendré: le cube A sera satisfait s'il est sur la table, le cube B s'il est sur C, le cube C s'il se trouve sur A. Fuir consiste pour un cube à se poser sur la table. L'ensemble des relations de satisfaction est donc:

$$\{\text{sur}(B,C), \text{sur}(A,\text{Table}), \text{sur}(C,B)\}$$

D'où l'on déduit les relations de dépendance entre agents:

$$\{B \text{ dépend de } C, C \text{ dépend de } A, A \text{ dépend de Table}\}$$

Sachant que la table est toujours satisfaite, on peut en déduire que le cube A est satisfait. Restent les cubes B et C. Commençons par C, puisque B dépend de C. B est un gêneur de C puisqu'il est un obstacle à la satisfaction directe de C. De ce fait, C demande à B de fuir. Mais C est alors un obstacle à la fuite de B, donc B demande à C de fuir. Ce dernier se pose sur la table, ce qui permet à B de fuir en se posant sur la table. C peut alors se poser sur A et devenir satisfait et B peut se poser alors sur C, terminant ainsi la solution du problème. Il suffit alors de mémoriser la trace de l'exécution des actions pour en déduire un plan.

Pour écrire cet exemple et le résoudre grâce à ECO, la première chose à faire est d'identifier les différents types d'éco-agents, puis de leur associer une structure et enfin de définir les comportements dépendants de l'application tels que `faireFuite` et `faireSatisfaction`. Dans le monde des cubes, il existe deux types d'agents: les cubes et les surfaces dont la table est l'unique exemplaire. On suppose qu'un cube ne peut soutenir qu'un seul cube et que les surfaces ont une place infinie. Voici la définition de ces types d'agents:

```
Type Cube
  Sous type de EcoAgent
```

Champs

```

auDessus : Cube
auDessous : EcoAgent (Cube ou Surface)

```

Procédure faireSatisfaction(x:Cube)

```

x.but.auDessus(x)
x.auDessous.auDessus(nil)
x.auDessous.but

```

Procédure faireFuite(x:Cube, p:EcoAgent)

```

x.auDessous.auDessus(nil)
x.auDessous := p

```

Procédure satisfait?(x:Cube)

```

    si x.auDessous = x.but
alors retourner(Vrai)
sinon retourner(Faux)

```

Procédure trouverPlacePourFuir(x:Cube)

```

retourner(Table)

```

Voici maintenant les procédures définissant les gêneurs de fuite et de satisfaction. Un cube est gêné pour fuir s'il existe un cube au-dessus de lui et il est gêné pour se satisfaire s'il existe un cube sur lui ou sur son but:

Procédure gêneursFuite(x:Cube)

```

si x.auDessus <> nil alors
// on retourne la liste des gêneurs
retourner(x.auDessus)
sinon
retourner(nil) // sinon rien

```

Procédure gêneursSatisfaction(x:Cube)

```

local r
r := x.auDessus ou x.but.auDessus
si r <> nil alors
retourner(r)
sinon
retourner(nil)

```

La définition des surfaces est beaucoup plus simple: elles sont toujours satisfaites et comme elles sont toujours libres (on suppose que la table est infinie), elles ne possèdent aucun gêneur.

Type Surface

Sous type de EcoAgent

```
Procédure satisfait?(x:Surface)
  retourner(Vrai)
```

On notera que toutes ces procédures correspondent à des actions simples: elles n'appellent aucune méthode de résolution (et, en particulier, elles n'appellent pas le noyau d'ECO) et reflètent directement les données du problème. Les procédures `générateursFuite` et `générateursSatisfaction`, notamment, décrivent directement les contraintes définies dans le monde des cubes, à savoir qu'on ne peut poser un cube sur la table que s'il est libre (il n'a aucun cube au dessus), et on ne peut poser un cube sur un autre cube que si les deux sont libres. Une procédure en particulier mérite notre attention. Il s'agit de `trouverPlacePourFuir`. Cette procédure, qui est ici très élémentaire puisqu'elle retourne toujours la table, est généralement la plus complexe à écrire et celle où il est possible d'ajouter des heuristiques permettant des résolutions plus rapides. Nous en verrons quelques-unes dans les exemples suivants.

Les cubes avec contraintes

Le problème des cubes tel qu'il a été décrit ici peut sembler trop facile. Il suffit en effet de poser tous les cubes sur la table puis de les reposer un par un pour résoudre le problème. En revanche, si l'on considère que la taille de la table est limitée, c'est-à-dire si le nombre de cubes qu'elle peut contenir est fini, le problème n'est plus aussi simple. Cette limitation entraîne des contraintes qui obligent à "défaire" des situations, en supposant qu'un cube, pour fuir, peut être amené à se poser sur un autre. Evidemment cette fuite peut provoquer une gêne supplémentaire pour les autres cubes. La situation suivante en est un exemple. Pour atteindre la configuration finale, les cubes ne disposent que de trois places sur la table (fig. 8.35). En (1), A cherche à se satisfaire (en allant sur E) et demande à B de fuir en lui donnant comme contrainte de ne pas se poser sur E. B peut fuir sur D et agresse C en lui interdisant d'aller sur D. Ainsi, en (2), A est bien libre, mais C lui bloque l'accès à E et il doit de nouveau l'agresser pour parvenir à son but. De la même façon, en (3), B chasse C en lui interdisant A comme destination.

La procédure `trouverPlacePourFuir` renvoie pour ce problème un des éco-agents en choisissant préférentiellement un emplacement libre sur la table, puis un cube libre différent de la contrainte passée lors de l'agression. *C'est la seule méthode qu'il est nécessaire de modifier dans la définition du problème précédent.* Le code en est le suivant:

```
Procédure trouverPlacePourFuir(x: Cube, y:Cube)
  local ensemble : liste de EcoAgent
  ensemble := Table.emplacements
  pour tout z de ensemble
    si (z.sur) = nil alors
      retourner z
  // on parcourt l'ensemble des cubes
  pour tout z de EnsembleDesCubes
    si (z.sur) = nil et (z <> y) et
```

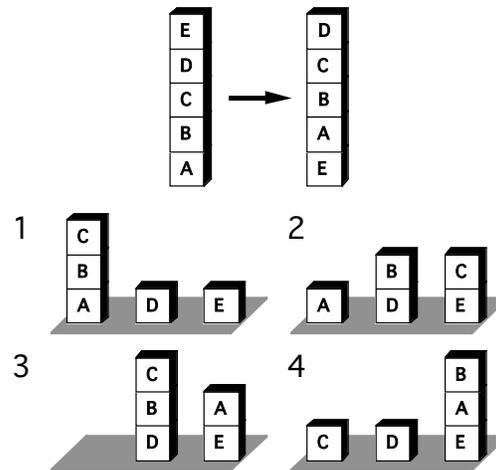


Figure 8.35: Un problème de cubes contraint. La table ne peut recevoir que trois piles de cubes.

```

(z <> contrainte)
// contrainte provenant de l'agression
alors
  retourner(z)
retourner(nil) // pas de place pour fuir

```

Les tours de Hanoï

Le problème des tours de Hanoï, dans lequel une tour de disques doit être déplacée d'un piquet à un autre sans qu'un disque ne soit jamais posé sur un autre disque de taille inférieure, hérite directement du problème des cubes avec contraintes. En effet, il s'agit là d'un cas particulier du problème dans lequel une contrainte supplémentaire est ajoutée lors de la recherche d'une place pour fuir. On voit ainsi émerger la notion de *famille de problèmes* dont les membres sont des éco-agents ayant des comportements similaires.

On traite ce problème de manière semblable à la technique des cubes avec contraintes. Les deux classes d'agents sont (naturellement) les disques et les tours, ces derniers étant satisfaits en permanence. Les disques sont satisfaits s'ils reposent sur un disque de taille immédiatement supérieure, ou sur une tour libre. Leur comportement se borne aux deux réactions classiques:

1. *Comportement de satisfaction*: aller sur un disque de taille immédiatement supérieure.
2. *Comportement de fuite*: aller sur une autre tour. Mais cette réaction a comme contrainte le fait de ne pas aller sur la tour sur laquelle l'agresseur (le disque qui provoque la fuite) veut aller. Cette contrainte permet ainsi d'éliminer les problèmes de conflits qui pourraient survenir si le disque supérieur allait à l'endroit où le disque inférieur voulait aller.

De plus, dès que l'acte d'agression est terminé, et donc que l'objet de la fuite disparaît, la réaction de satisfaction se remet en place. Le disque A a comme but (condition de satisfaction) d'aller sur le disque 3. Pour cela il agresse le disque B en lui interdisant d'aller en 3. De ce fait, B veut s'en aller en 2. Comme il est gêné, il agresse le disque C en lui interdisant d'aller en 1. De ce fait, le disque C ne peut aller qu'en 3. Une fois placé, il indique à B qu'il peut se déplacer et B se place en 2. Il indique alors qu'il a fini son agression envers C et C vient se replacer sur B, de plus il donne comme continuation le fait d'indiquer à A qu'il peut maintenant se déplacer. De ce fait, A se place en 3 et précise qu'il a terminé son agression envers B. De ce fait, B veut se placer sur A, mais comme il est gêné, il est obligé d'agresser C en lui interdisant d'aller sur la tour 3. C va en 1, ce qui libère B qui va sur A. B enlève son agression envers C, et C peut se placer sur B, ce qui résout entièrement le problème, les trois disques étant satisfaits.

A l'analyse de la trace ci-dessus on constate que la solution obtenue correspond directement à l'algorithme classique des tours de Hanoï. Cette remarque a été observée directement à partir de l'analyse du comportement des tours, et n'avait pas été imaginée auparavant: c'est en regardant le fonctionnement du système que cette correspondance a été mise à jour, et non à partir d'une volonté a priori de réaliser la même opération. Devant une telle constatation, on ne peut s'empêcher de formuler un certain nombre de remarques et questions et, en particulier, celles-ci: Quelle est la nature du lien entre le comportement défini et l'algorithme? Est-ce que cet algorithme est un état stable de comportement, ou bien n'a-t-on pas programmé l'algorithme de manière déguisée, ou enfin s'agit-il simplement d'une coïncidence pour cet exemple-là, la correspondance ne se généralisant pas?

En fait, le problème des tours de Hanoï est très contraint, et toute solution passe soit par une exploration d'espaces d'états avec retour en arrière, soit par un comportement qui est nécessairement celui de l'algorithme: il n'existe pas d'alternative. Si, par contre, on diminue les contraintes associées à ce jeu (en augmentant le nombre de piquets par exemple) le comportement s'éloigne alors de l'algorithme parfait, et des variantes commencent à apparaître. Par exemple, si l'on utilise 4 piquets avec seulement 3 disques, l'algorithme le plus parfait consiste à placer le disque C sur la tour 2, le disque B sur la tour 3, puis le disque A sur la tour 4 et enfin à empiler tous les disques les uns après les autres dans le bon ordre (fig. 8.36).

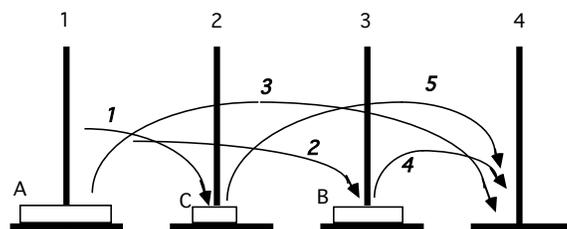


Figure 8.36: Dans un problème de tour de Hanoï à plus de trois piquets, ECO utilise automatiquement toute la place qui lui est disponible.

L'étude de systèmes aussi simples que celui des cubes ou des tours de Hanoï montre l'intérêt de ce type de technique, mais aussi ses limites, puisqu'on n'est

pas certain a priori d'obtenir un plan optimal. Cependant, il serait certainement possible d'y intégrer des techniques d'apprentissage afin de modifier localement et incrémentalement les comportements des agents et, en particulier, la coordination des différentes réactions.

Le taquin

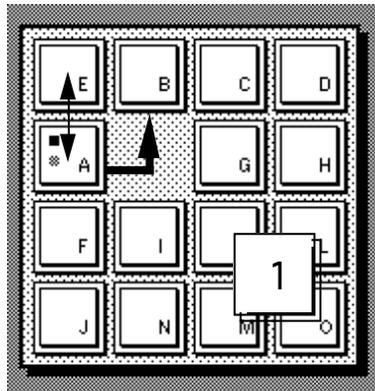
Bien plus que les tours de Hanoï, le taquin a inspiré nombre d'études en planification. Son univers parfaitement délimité, générant des contraintes évidentes, est en effet particulièrement propice à l'exploration de domaines d'états ou d'arborescences. Les travaux les plus aboutis à ce jour, non seulement en terme de réduction de la complexité des recherches, mais aussi en terme de résultats empiriques, sont ceux de R.E. Korf, commencés en 1985 avec l'utilisation de l'algorithme IDA* (Iterative-Deepening A*) (Korf 1985), et poursuivis avec les algorithmes de recherche RTA* (Real-Time A*) et ses successeurs, dont notamment LRTA* (Learning Real-Time A*) (Korf 1988). Pour réussis et intéressants qu'ils soient, ces travaux n'en sont pas moins fidèles à la ligne directrice qui, depuis le début, sous-tend toute tentative de résolution du problème du taquin: l'exploration plus ou moins profonde de domaines d'états guidés par une technique heuristique.

La réalisation du taquin a montré que, même si la complexité des fonctions relatives au problème augmente considérablement, il n'est pas nécessaire de modifier les fonctions du noyau d'ECO. Les algorithmes utilisés sont linéaires en nombre de cases du taquin, aussi la complexité croît-elle linéairement avec la taille du taquin. De plus cette implémentation en ECO ouvre des possibilités supplémentaires (telles le taquin à cases vides multiples, le passage immédiat d'une taille de taquin à une autre, la résolution de taquins non carrés, etc.), ce qui n'est pas du tout le cas des approches exploratoires précitées.

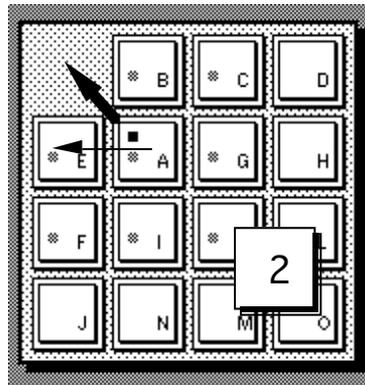
Pour résoudre ce problème en ECO, on considère que les cases comme les palets sont des éco-agents. Alors que les premiers sont toujours satisfaits (les cases sont immobiles), les seconds ne le sont que s'ils se trouvent sur la case correspondant à la configuration désirée. Par exemple, le palet marqué de la lettre "A" ne sera satisfait que s'il se trouve sur la case de coordonnée (1,1). La fonction `trouverPlacePourFuir` est ici un peu plus compliquée, car elle doit estimer quel palet il faut agresser pour fuir. On trouvera une définition précise de l'algorithme, ainsi qu'une preuve de la complétude et de la décidabilité de la solution quelle que soit la taille du taquin dans (Drogoul et Dubreuil 1992). Comme exemple de la souplesse du modèle, voici la résolution du problème des coins : le palet A veut prendre la place de E pour compléter la première ligne (fig. 8.37).

Ce mécanisme, qui utilise les propriétés générales du problème, permet de résoudre des taquins de très grande taille, dont certains n'avaient jamais été résolus à cause du coût du calcul. Le tableau 8.38 montre les performances moyennes du système en fonction de la taille du taquin, chaque exemple étant résolu 100 fois. Le chronométrage a été effectué sur une version d'ECO écrite en Smalltalk-80 4.1, tournant sur une station de travail Sun Sparc 10 sans affichage graphique.

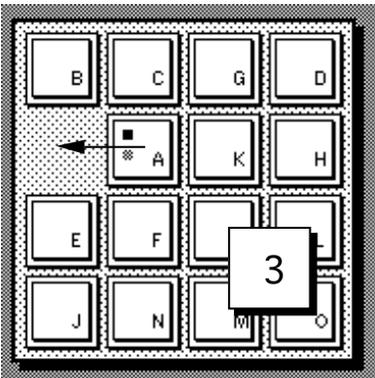
- Palet bloqué
- Palet actif



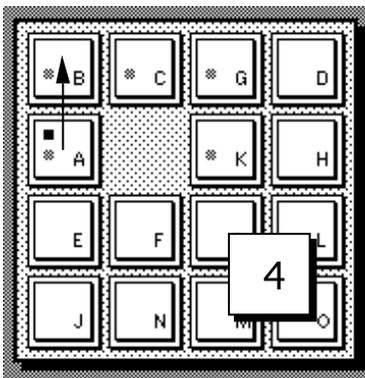
A agresse E en lui passant comme contrainte de ne pas toucher à B (B est le palet dont "dépend" A). E n'a plus comme solution que d'agresser A.



A réagresse E en lui passant comme contrainte, cette fois, de ne pas fuir sur le blanc (son but). E agresse donc F. Sont successivement agressés: I, K, G, C et B



B, C, G, K, I, F et E ayant bougé, A agresse le blanc en vue de se satisfaire. Le blanc fuit en échangeant sa place avec le palet qui l'agresse.



A agresse maintenant B qui occupe son but sans lui passer aucune contrainte B agresse donc C, qui agresse G, etc...

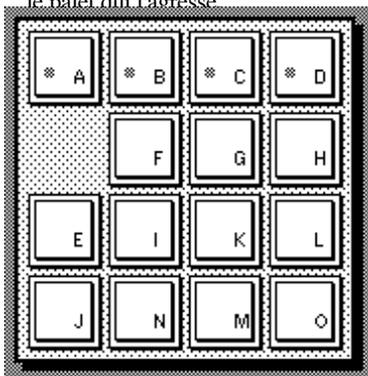


Figure 8.37: Le problème des coins dans le taquin, tel qu'il est géré dans ECO

Configuration		Solution		
Largeur	Palets	Temps moyen	Longueur Moyenne	Longueur par Palet
n	N	Secondes	Nombre de Mouvements	
3	8	0,2218	48,90	6,11
4	15	0,7547	142,10	9,47
5	24	2,4812	283,00	11,79
6	35	4,5146	482,00	13,77
7	48	9,6559	746,30	15,55
8	63	17,027	1 109,20	17,61
9	80	23,1532	1 455,20	18,19
10	99	35,4682	1 947,70	19,67
11	120	52,7616	2 483,10	20,69
12	143	65,7846	2 889,10	20,20
13	168	86,7198	3 361,90	20,01
14	195	106,9209	4 199,70	21,54
15	224	116,4785	4 881,50	21,79
16	255	165,8702	5 581,90	21,89
17	288	204,4582	6 267,10	21,76
18	323	230,687	6 947,60	21,51
19	360	271,5735	7 797,20	21,66
20	399	292,0156	9 004,70	22,57
21	440	340,8768	9 882,00	22,46
22	483	415,1151	11 093,70	22,97
23	528	477,3549	12 404,00	23,49
24	575	544,9122	13 340,10	23,20
25	624	625,5805	15 181,20	24,33
26	675	655,0309	16 514,00	24,47
27	728	782,0069	18 496,80	25,41
28	783	897,6368	19 999,40	25,54
29	840	1036,3034	21 229,20	25,27
30	899	1179,58	23 009,00	25,59

Figure 8.38: Résultats expérimentaux de la résolution du Taquin. Les valeurs sont des moyennes sur 100 expérimentations pour chaque exemple (d'après (Drogoul 1993)).

8.5.4 Univers évolutifs

Les possibilités d'ECO ne se limitent pas à la résolution de problèmes. Il s'est révélé être un outil particulièrement bien adapté à la simulation d'univers évolutifs.

Comme nous l'avons signalé précédemment, les méthodes combinatoires sont très sensibles aux perturbations. Au contraire, dans un système d'éco-agents, une perturbation extérieure est traitée exactement comme n'importe quelle autre donnée du problème: tout agent qui perçoit la perturbation y réagit localement, selon ce que lui dicte son comportement. C'est dans cette optique qu'a été implémenté le problème du "Misachieving baby" (Schoppers 1987): Lors de la résolution d'un problème du monde des cubes, un bébé malfaisant déplace des cubes, des parties de piles, voire des piles entières à sa guise (fig. 8.39). Par exemple, dans la situation (1), le cube B n'a qu'à se poser sur C pour achever la pile désirée. C'est alors que le bébé intervient et déplace la pile composée des cubes A et C et la pose sur B (2). Le cube A n'est plus satisfait et B est maintenant obligé d'agresser les cubes au-dessus de lui pour pouvoir atteindre son but.

Du point de vue de l'utilisateur, la définition du problème n'est pas différente de celui des cubes examiné précédemment. C'est le noyau d'ECO qui prend en charge la dynamique du système en gérant le changement des conditions: il suffit simplement

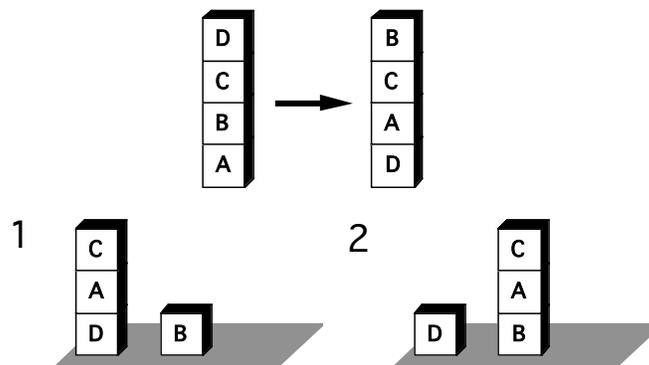


Figure 8.39: Dans un problème de cubes, un bébé malicieux vient perturber une pile de cubes.

de mettre à jour l'état de satisfaction (le cube A n'est plus satisfait) et surtout de réinitialiser les gêneurs et les fuyards, en considérant que la situation résultante de l'action du bébé malicieux est une nouvelle situation initiale.

Le graphique de la figure 8.40 montre la moyenne du nombre de déplacements des cubes (sans compter ceux qui sont effectués par le bébé) en fonction de la période (P) des perturbations. A chaque activation d'un cube, le bébé avait une chance sur P de provoquer une perturbation. Le problème à résoudre, décrit ci-dessus, demande 6 déplacements s'il n'y a aucune perturbation. L'opération a été répétée 100 fois pour chaque valeur de P.

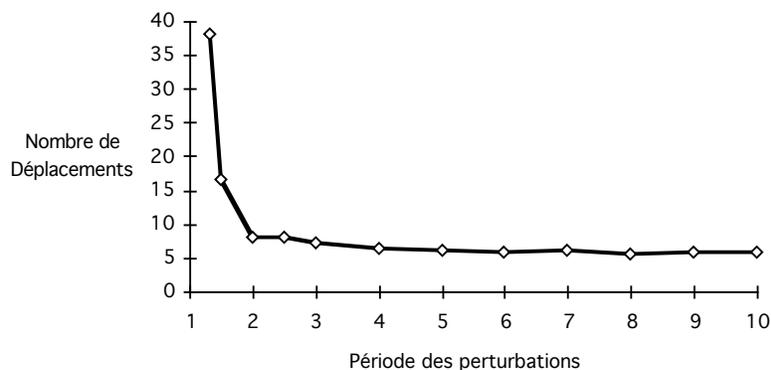


Figure 8.40: Le nombre de déplacements de cubes nécessaires en fonction de la période des perturbations P

On constate que même si la période des perturbations diminue, la solution est rapidement atteinte. Quand le rythme des perturbations est plus élevé que celui des actions des cubes ($P < 2$), le système parvient tout de même à son état final lorsqu'à un moment (statistiquement d'autant plus éloigné dans le temps que P est proche de 1), les perturbations n'affectent plus que des agents non satisfaits. D'autre part, on peut expliquer la grande réussite d'ECO par le fait que les perturbations sont parfois positives: le bébé peut placer des cubes dans leur état de satisfaction de manière aléatoire. Si le bébé continue ses interventions une fois la configuration

finale atteinte, le système va naturellement retrouver son état d'équilibre, ce qui tend à faire osciller le système autour de cet état. ECO induit ainsi naturellement un comportement de système homéostatique.

8.5.5 Formalisation

Il est possible de donner une formalisation d'ECO à partir d'un critère permettant de savoir si une suite d'opérations a bien abouti au but désiré.

La problématique de la planification a été tout particulièrement étudiée par Chapman (Chapman 1987), lequel présente tout à la fois un modèle formel de la planification, un critère permettant de déterminer la validité des plans et un planificateur conçu comme une réalisation calculatoire de ce critère.

L'idée générale est de décrire une condition permettant de décider si une proposition, considérée comme une description de l'état final, peut être établie par une suite d'opérateurs de type STRIPS. Il est possible d'adapter ce critère à l'éco-résolution, en considérant les opérateurs de satisfaction et de fuite comme mécanisme principal de construction de plan.

Soit $A = \{a_1, \dots, a_n\}$ un ensemble d'agents qui doivent être satisfaits pour que le problème soit considéré comme résolu, Σ un ensemble d'état du monde, et si l'on nomme $Sat(a_i, \sigma)$ l'action de satisfaction de l'agent a_i conduisant à l'état σ , et $Fuite(a_i, \sigma)$ l'action de fuite, alors ce critère peut être défini ainsi (Bura et al. 1991):

Critère de satisfaction: La satisfaction d'un agent a_i peut être affirmée dans l'état du monde final σ_f , s'il est affirmé directement dans l'état final, ou s'il a été affirmé dans un état précédent σ_n et qu'il a été conduit à fuir dans aucun état σ_b entre l'état σ_p et σ_f . Pour chaque état σ_b où l'agent a été conduit à fuir, il doit y avoir un état σ_a entre l'état σ_b et l'état σ_f qui resatisfasse cet agent. Pour chaque état où un agent doit fuir et peut être satisfait simultanément, l'agent effectue alors l'opération qui le conduit à la satisfaction.

Voici une traduction plus formelle de ce critère:

$$\forall a_i \in A, \exists \Sigma, Sat(a_i, \sigma_p) \wedge (\forall \sigma_b \in \Sigma, (\sigma_p \leq \sigma_b) \wedge Fuite(a_i, \sigma_b)) \Rightarrow ((\exists \sigma_a \in \Sigma, (\sigma_p \leq \sigma_b) \wedge Sat(a_i, \sigma_a)) \vee Sat(a_i, \sigma_b))$$

Ce critère permet de comprendre comment fonctionne Eco dans les cas précédents: par exemple, dans le monde des cubes (cubes contraints, tour de Hanoi...), les dépendances entre agents forment un ordre partiel tel que pour satisfaire un agent, il suffit de déplacer tous les gêneurs qui se trouvent entre l'agent qui cherche à se satisfaire et son but, c'est-à-dire un agent déjà satisfait. Comme le déplacement de ces gêneurs ne modifie pas le statut des agents déjà satisfaits (il suffit de déplacer tous les cubes qui gênent sur d'autres empilements), il s'ensuit que le critère de vérité est vérifié. Cela signifie que ECO sait définir un plan d'action pour tous les problèmes dont la situation finale peut être donnée sous la forme d'une structure arborescente. C'est en particulier vrai pour l'assemblage des structures données "en kit", qui supposent que tous les éléments s'emboîtent les uns sur les autres. La généralisation à des structures de dépendance plus complexes est en cours.

8.5.6 Résolution de contraintes par éco-résolution

Les techniques réactives ne se limitent pas à la planification d'action. Il est possible de les utiliser pour résoudre d'autres problèmes tels que ceux que l'on rencontre dans le domaine de la satisfaction de contraintes. Comment, par exemple, placer au mieux un ensemble de tâches sur un ensemble de machines? Comment constituer un emploi du temps pour un collègue, lorsqu'il n'existe pas nécessairement une solution satisfaisant à la fois toutes les contraintes pédagogiques et les préférences des professeurs? Résoudre ces types de problèmes est du ressort de la satisfaction de contraintes. Les problèmes de contraintes mettent en présence deux types d'entités: les variables et les contraintes. Résoudre un problème de satisfaction de contraintes, c'est trouver une assignation de valeurs pour les variables qui satisfasse les contraintes. Par exemple, résoudre le système d'équations suivant:

$$3 * X + Y = Z, \quad Z > 2 * X, \quad 4 * X < 2 * Y$$

dans lequel X , Y et Z sont des variables entières définies dans le domaine $[1, 10]$, c'est trouver l'ensemble des valeurs qui satisfont ces contraintes. Un système de satisfaction de contraintes donnerait ainsi les solutions:

$$\begin{aligned} X=1, Y=3, Z=6 \\ X=1, Y=4, Z=7 \\ X=1, Y=5, Z=8 \\ X=1, Y=6, Z=9 \\ X=1, Y=7, Z=10 \end{aligned}$$

Pour une bonne introduction aux problèmes de satisfaction de contraintes, et en particulier à la programmation par contraintes, on pourra se référer à (Fron 1994). La figure 8.41 montre un réseau de contraintes pour le problème précédent.

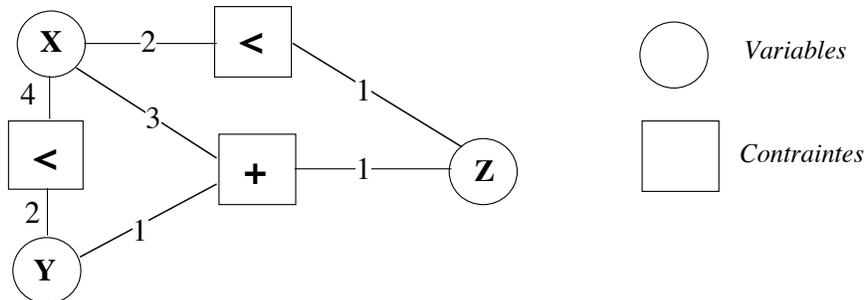


Figure 8.41: Un petit réseau de contraintes résolvant le système d'équation à trois inconnues défini plus haut

En partant d'un point de vue initialement très différent de celui des techniques classiques fondées sur la propagation de contraintes et l'énumération implicite de solutions, K. Ghedira a développé une méthode originale de résolution de contraintes qui associe l'éco-résolution et le recuit simulé (Ghedira 1993; Ghedira 1994). L'idée générale consiste à considérer les variables et les contraintes comme des éco-agents qui cherchent à trouver un état de satisfaction.

Les contraintes

Une contrainte est satisfaite si sa relation associée est vérifiée. Autrement, une contrainte insatisfaite demande à l'une de ses variables de changer de valeur. Le graphe d'état d'une contrainte est donné figure 8.42.a et comprend 4 états principaux: **non-instanciée**, **instanciée-satisfaite**, **instanciée-insatisfaite** et **en-attente**. La première correspond à un état dans lequel des variables associées à la contrainte ne sont pas encore instanciées, la seconde à une situation où la contrainte est à la fois instanciée et satisfaite, la troisième au cas où elle est à la fois instanciée mais non satisfaite et la quatrième à une situation intermédiaire où la contrainte attend le verdict d'une variable.

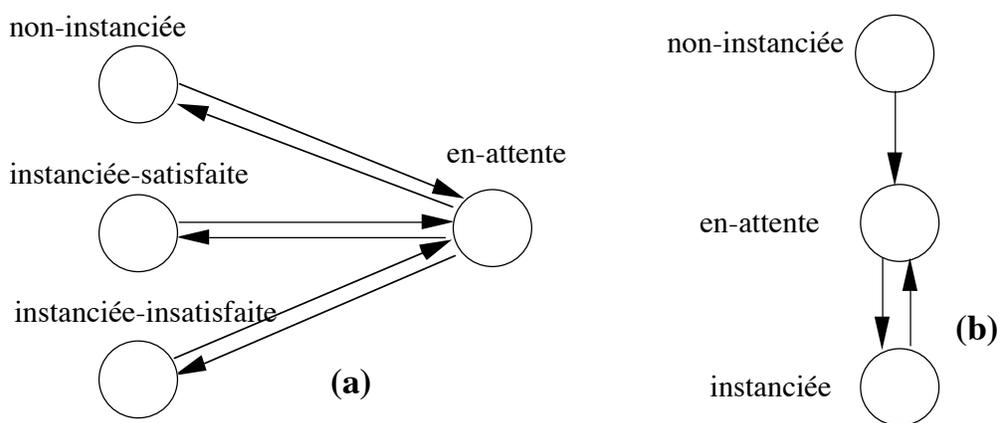


Figure 8.42: Le diagramme des états d'une contrainte (a) et d'une variable (b)

Les variables

Les variables tentent de trouver des valeurs appartenant à leur domaine de définition. Lorsqu'on lui demande de changer de valeur, une variable choisit aléatoirement une valeur de son domaine de définition et demande à ses contraintes leur opinion, c'est-à-dire si ces valeurs les satisfont ou non. A partir de toutes ces réponses, elle décide d'accepter ou non cette valeur en fonction d'un critère de choix dépendant d'un mécanisme fondé sur la notion de recuit simulé. La figure 8.42.b montre le graphe de transition d'une variable. La fonction de décision est fondée sur le processus suivant: soit i , la valeur courante d'une variable v et j la nouvelle valeur, soit $S(i)$ le nombre de ses contraintes satisfaites, la variable choisira la valeur j si:

$$S(j) \geq S(i)$$

$$S(j) < S(i) \text{ et } \text{Rand}(0, 1) < e^{-\frac{S(j)-S(i)}{T(k,v)}}$$

et décidera de garder la valeur i autrement. La fonction $\text{Rand}(0, 1)$ choisit une valeur aléatoire entre 0 et 1 et la fonction $T(k, v)$ détermine la tolérance d'acceptation de la variable v au temps k . Afin de restreindre au fur et à mesure les choix d'une variable, sa tolérance décroît régulièrement avec le temps: $T(k+1, v) = T(k, v) - 1$,

la valeur initiale de la fonction T dépendant du nombre de contraintes associées à la variable v . De ce fait, plus une variable a déjà effectué des choix, moins elle aura tendance à accepter une valeur qui satisfait un moins grand nombre de contraintes. Elle restera de plus en plus “figée” sur ses choix antérieurs, comme si elle se “gelait” avec le temps.

L'intérêt de cette méthode est de pouvoir facilement se généraliser à des situations de satisfactions partielles, c'est-à-dire de problèmes pour lesquels il n'existe aucune solution globale, et de trouver rapidement une solution. De plus, des travaux théoriques montrent qu'elle est capable de donner la solution optimale. Enfin, des modifications légères de l'ensemble des contraintes ne remettent pas en cause l'ensemble des situations déjà trouvées. Les insatisfactions se propageront de proche en proche aux variables et aux autres contraintes, et le système privilégiera donc les solutions proches des solutions déjà trouvées. Une autre approche fondée sur une conception distribuée de la satisfaction de problèmes a été proposée dans (Liu et Sycara 1993), mais elle ne semble pas offrir, pour l'instant, d'aussi bons résultats que celle de K. Ghedira. Quoi qu'il en soit, l'utilisation de la coordination d'agents comme mécanisme de résolution ouvre de nouvelles perspectives dans les techniques de résolution de problèmes.

Chapter 9

Conclusion

La kénétique propose une vision nouvelle à la fois de la construction des logiciels informatiques, mais aussi, et c'est ce que j'ai voulu montrer tout au long de cet ouvrage, de la résolution de problèmes qui ne s'appuie plus sur une vision centralisée du monde et de l'individu mais qui pense tout comme interaction et comme émergence de fonctionnalités liées à ces interactions. Les systèmes multi-agents proposent une révolution tranquille, un simple changement d'attitude et de perspective, mais dont les conséquences sont radicales. Le fait d'ignorer volontairement les solutions globales au profit de visions purement locales suppose encore une certaine ouverture d'esprit, car tout notre enseignement, tous les paradigmes dominants posent encore le global comme le seul moyen de résoudre les problèmes. Mais la complexité du monde, et en particulier des systèmes informatiques, est maintenant telle que seules les perspectives qui mettent en avant des conceptions locales et qui prônent avant tout la capacité d'émergence et d'auto-organisation de ces systèmes seront capables d'apporter des réponses viables et efficaces.

J'ai déjà participé, en tant que chercheur et enseignant, à la révolution apportée par la programmation par objets. Mais il faut aller encore un peu plus loin et donner aux composants logiciels l'autonomie comportementale dont ils ont besoin. Et ce n'est qu'à ce prix que l'on pourra effectivement réaliser ces systèmes ouverts, communicants et interactifs.

Un peu d'histoire anticipée...

Quel est le futur des logiciels, et quelle sera la structure des programmes de demain? Pour le savoir, projetons-nous un peu dans le futur, en 2045 par exemple. Voilà ce que nous lisons dans un magazine d'informatique (penser à *Byte* ou au *Dr Dobb's Journal*) à cette époque¹.

La structure des programmes du vingtième siècle nous semble maintenant bien étrange. Seuls quelques individus de mœurs désuètes programment encore comme nos grands-parents. Comment ne pas sourire à la vision de ces étranges systèmes

¹J'avais publié ce texte en 1989 (Ferber 1989) dans ma thèse d'Etat. Je le présentais alors comme de la science-fiction. Je pense maintenant qu'il est le prolongement logique des travaux décrits ici et qu'il s'agit d'une perspective prévisible du futur génie logiciel.

qui séparaient arbitrairement les données du code, qui s'exprimaient de manière séquentielle, tordant l'esprit humain pour qu'il s'adapte à la machine. On aboutissait à des logiciels quasiment incapables de s'adapter à un contexte en constante mutation. L'évolution était impossible, si ce n'est au prix de coûteuses manipulations de code, qui obligeaient parfois les programmeurs à tout recommencer.

Dans ces temps-là, un logiciel était une chose inerte, manipulée de l'extérieur, sans comportement propre. Certains cherchaient à fixer ces programmes dans une rigidité voisine du cristal. On appelait cela "spécifier" et "prouver des programmes". Evidemment, les logiciels réalisés avec des techniques de spécification et de synthèse de programmes s'exécutaient de manière correcte, mais les problèmes étaient reportés ailleurs: dans l'élaboration des spécifications qui exigeaient des programmeurs une gymnastique mentale au moins aussi complexe que celle de la programmation et, surtout, dans la très grande difficulté de faire évoluer le logiciel, toute modification nécessitant un travail considérable pour veiller à l'indépendance des transformations apportées. Programmer était alors un art, qui réclamait du doigté et du savoir-faire, afin d'éviter ce que l'on appelait à l'époque un "bug", c'est-à-dire une erreur de programme. Les "bugs", qui ont disparu aujourd'hui — le Silicium Museum en conserve quelques intéressants et rares exemplaires — étaient l'angoisse des programmeurs, car leur présence pouvait totalement arrêter le programme. Ils ne savaient pas comment engendrer un logiciel par auto-organisation! Ils ne connaissaient pas les techniques liées à l'émergence! Ils ne savaient pas utiliser l'interaction avec l'environnement comme facteur de complexification et d'adaptation! Leur technique était rudimentaire à l'extrême, et ils nous apparaissent aujourd'hui comme relevant d'un âge presque antédiluvien.

Aujourd'hui, heureusement, il n'en est plus de même. Les programmes sont devenus des entités quasi vivantes, formées d'un nombre gigantesque de petites unités indépendantes et douées d'une certaine autonomie, qui communiquent entre elles, s'organisent, se réorganisent, se complexifient et s'adaptent à leur environnement, afin d'accomplir les tâches qui leur sont imparties. Cette architecture est complexe et sans cesse remise en question par l'ensemble des unités organisées en colonies, lesquelles présentent de nombreuses analogies tant avec la structure neuronique du cerveau qu'avec celle d'une société animale ou humaine. On fait ainsi "pousser" des programmes, comme on fait pousser des plantes ou comme on élève un animal. Pour disposer d'une application, il suffit de placer une "souche" dans une architecture informatique et de la mettre en interaction avec d'autres agents (programmes et/ou êtres humains) dans un environnement adéquat, pour qu'ils s'adaptent et réalisent les tâches pour lesquelles ils sont spécialisés. Il est possible ensuite de "greffer" des programmes, c'est-à-dire de les agréger afin qu'ils coopèrent et que le résultat ne soit pas seulement la somme des deux programmes, mais le produit d'une alchimie particulière liée aux interactions qui ont lieu entre les deux morceaux initiaux. On obtient alors un logiciel original, qui n'a plus que de lointaines ressemblances avec ceux dont il est issu, un peu comme un enfant par rapport à ses parents.

Dans le même journal, on peut lire la publicité suivante:

Nouveau: souche de logiciel à prise rapide pour l'organisation et le man-

agement des PME

Cette souche permet le développement d'un multi-organisateur intégré qui prend en charge la prise de décision, la gestion et la communication dans l'entreprise. Elle s'adapte à toute architecture "Hyper Compatible" disposant d'au moins 3 millions de Giga EO virtuels². Elle s'installe très simplement dans l'appareil. Comme pour les souches ordinaires, il suffit de lui donner de plus en plus de travail à effectuer, afin qu'elle apprenne sa tâche et s'adapte à vos besoins. Sa nouvelle organisation "Multi World", qui lui procure des simulations d'interactions hypothétiques internes, lui permet d'être totalement opérationnelle au bout de seulement quinze jours au lieu de deux mois avec les souches ordinaires. Cette souche révolutionnaire s'intègre naturellement à toutes celles disposant d'un mécanisme d'éco-hétéro-adaptation.

En l'an 2045, tout ce qui constitue notre culture informatique actuelle sera dépassé. L'intelligence artificielle n'existera peut-être plus en tant que discipline à part, si ce n'est comme étude de la "psycho-socio-biologie des artefacts". Elle sera totalement fusionnée et amalgamée avec tous les secteurs de l'informatique et en collaboration avec les sciences physiques, biologiques, sociales et psychologiques. La structure des ordinateurs aura changé. La notion même de processeur central aura totalement disparu. Chaque ordinateur sera constitué de plusieurs milliers de petits processeurs travaillant ensemble, constituant ainsi le "milieu" dans lequel les petites entités des programmes évolueront.

Cette vision du futur suppose que les logiciels puissent coopérer harmonieusement avec nous en s'insérant directement dans l'ensemble de nos activités. Ils apprendront en nous observant (mais pas en nous contrôlant, il ne s'agit pas de créer Big Brother) et s'intégreront de plus en plus dans les processus de traitement, dans les tâches de routines, mais aussi pour nous aider à prendre rapidement et efficacement des décisions. A l'instar des humains, qui apprennent par "osmose", c'est-à-dire par confrontation avec la réalité, en imitant, en essayant et en modifiant des comportements "pour voir ce que ça fait", les logiciels de l'avenir devront être en contact avec nous, en regardant et en apprenant à partir des flux d'informations qui circulent normalement entre les humains. Simples outils aujourd'hui, collaborateurs de demain, les agents informatiques nous seront aussi indispensables qu'une calculatrice ou, pour certains, qu'un ordinateur portable. Ces agents, en voyant circuler des informations, en s'adaptant au fonctionnement des êtres humains (qui en retour s'accommoderont eux aussi des comportements des agents informatiques), se réorganiseront et se restructureront dynamiquement pour s'améliorer et s'adapter aux conditions de leur environnement. Ce type de système verra alors l'émergence d'un écosystème de nature complètement nouvelle, formé d'une population hybride d'agents humains et informatiques, qui seront amenés à cohabiter pour, tout simplement, accomplir leurs destins et vivre.

²les EO sont des "Equivalents Octets", une mesure utilisée pour décrire des organisations mémoires.

Appendix A

BRIC

BRIC (Block-like Representation of Interactive Components) est un langage de haut niveau permettant de concevoir et de réaliser des systèmes multi-agents à partir d'une approche modulaire. Un système BRIC comprend un ensemble de composants reliés entre eux par des liens de communication. Nous n'aborderons ici que les éléments nécessaires à la compréhension des composants décrits dans cet ouvrage. D'autres extensions de BRIC sont possibles mais nous ne les décrivons pas ici.

A.1 Les composants

Un composant BRIC est une structure logicielle caractérisée extérieurement par un certain nombre de bornes d'entrées et de sorties et de manière interne par un ensemble de composants. Les bornes d'entrées et de sorties jouent le rôle d'interface en recevant ou en transmettant des messages vers l'extérieur (fig. A.1). Tout composant est instance d'une classe qui décrit sa structure interne. Les classes servent donc, à l'image des langages objets, à réunir toutes les informations communes à un ensemble de composants semblables.

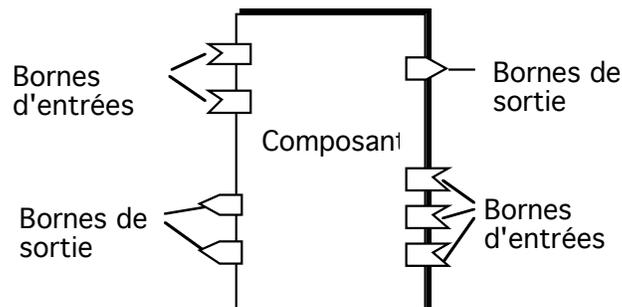


Figure A.1: Structure d'un composant et de ses bornes

Le comportement de ces composants peut être défini soit par l'assemblage d'autres composants dont le comportement est déjà spécifié, et on parle alors de *composants structurés*, soit directement par l'intermédiaire d'un réseau de Petri associé. Dans ce dernier cas, on dira qu'il s'agit de *composants élémentaires*. Enfin, il

est aussi possible d'introduire des *composants primitifs* dont le comportement n'est pas décrit par un réseau de Petri, mais dont l'intérêt est de prendre en charge des fonctionnalités externes à l'agent ou au système.

A.2 Composants composites

Un composant structuré est défini par l'ensemble des sous-composants qui le compose. Les bornes d'entrées du composant structuré sont liées aux bornes d'entrées des sous-composants. On peut aussi fusionner les bornes d'entrées (et de sorties) des composants composites avec ses sous-composants, comme le montre la figure A.2.

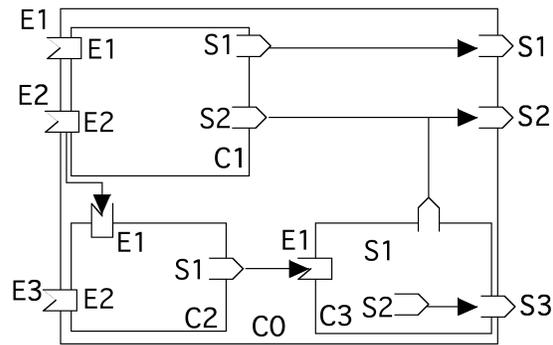


Figure A.2: Un composant composite

A.3 Constitution des composants élémentaires

Les composants élémentaires ont leur comportement décrit en termes de réseaux de Petri. Le formalisme de réseau normalement utilisé en BRIC est celui des réseaux de Petri colorés avec arcs inhibiteurs. Mais d'autres formalismes, tels que celui des réseaux synchrones ou des réseaux temporisés, peuvent être employés.

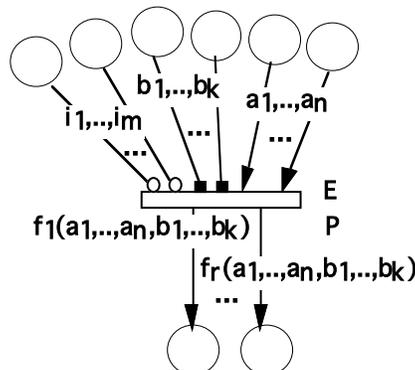


Figure A.3: Forme générale d'une transition dans le formalisme BRIC

La forme générale d'une transition est représentée figure A.3. Une transition est définie par des arcs entrants, des arcs sortants et une précondition d'activation.

- a) Les arcs entrants associent une place à une transition. Ils sont porteurs d'une condition sous la forme d'une description de marque, cette description comprenant des variables. Lorsque la place contient une marque correspondant à cette description, l'arc est validé. Il existe trois catégories d'arcs entrants:
1. Les *arcs standard*, notés a_1, \dots, a_n , ne déclenchent la transition que s'ils sont tous validés. Ces arcs "consomment" les marques qui servent au déclenchement, lesquelles sont donc supprimées des places en entrées.
 2. Les *arcs inhibiteurs*, notés i_1, \dots, i_m , inhibent le déclenchement de la transition s'il est validé. Ces arcs ne sont pas consommateurs de marques.
 3. Les *arcs non consommateurs*, notés b_1, \dots, b_k , fonctionnent comme les arcs standard, à la seule différence qu'ils ne consomment pas les marques en entrée. Les arcs non consommateurs peuvent toujours s'exprimer sous la forme d'arcs standard, leur usage servant uniquement à rendre plus lisible les figures (fig. A.4).

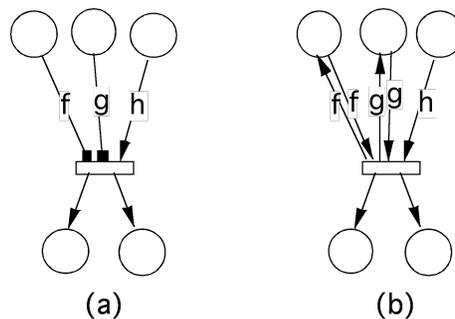


Figure A.4: Les arcs non consommateurs (a) peuvent être transformés en réseaux ne comprenant que des arcs consommateurs (b).

- b) Les arcs sortants, qui associent une transition à une place de sortie, produisent de nouvelles marques qui viennent se placer dans ces places. Ces marques produites sont fonction des marques utilisées pour le déclenchement de la transition.
- c) La précondition associée à une transition porte sur des conditions externes. Il est nécessaire que cette précondition soit vérifiée pour que la transition soit déclenchée.

A.4 Liens de communication

Les composants communiquent en s'échangeant des informations, que l'on appelle parfois "message"¹, le long de liens de communication qui relient des bornes de sorties à des bornes d'entrées. Le passage des informations est régi par les lois de franchissement des réseaux de Petri, introduisant de fait une exécution de type "événementiel asynchrone". Le déroulement des opérations est cadencé par les informations émises et reçues par les composants, l'ensemble constituant un réseau de propagation d'informations. Les informations sont véhiculées dans le réseau sous la forme de marques. Une marque est soit un jeton, c'est-à-dire une information élémentaire qui ne vaut que par sa présence ou son absence, soit un prédicat de la forme $p(\mathbf{a}_1, \dots, \mathbf{a}_n)$, où les \mathbf{a}_i représentent des symboles dans un alphabet fini ou des nombres.

A.5 Conventions de notations et équivalences

Afin de faciliter la lecture des composants BRIC, on fera un certain nombre d'hypothèses concernant les notations:

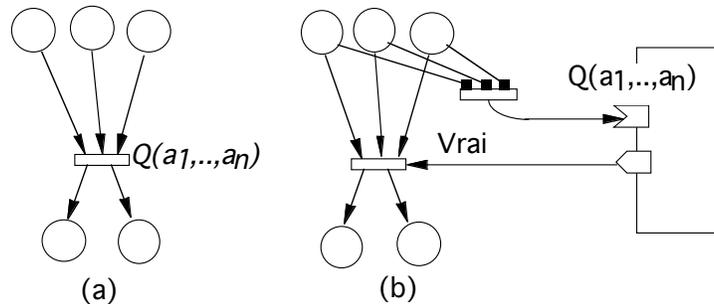


Figure A.5: Les conditions externes peuvent être traduites sous la forme d'une demande d'information à un autre composant.

1. On suppose que tout arc en sortie d'une place ayant un libellé de la forme $P(\mathbf{a}_1, \dots, \mathbf{a}_n)$ aura pour condition de validité une description identique à la description de la place, sauf mention du contraire. De ce fait, il n'est en général pas nécessaire de reproduire la description sur un arc entrant sur une transition, sauf si l'on cherche à donner plus de précision à une condition. Les conditions externes associées à des transitions (fig. A.5.a) peuvent être considérées comme des demandes d'information. La figure A.5.b montre ainsi un réseau équivalent, dans lequel la transition externe a été traduite sous la forme d'un accès à un composant.

2. Les noms des bornes d'entrée seront pris comme des libellés de place, puisqu'elles sont considérées comme des places.

¹Afin d'éviter d'éventuels conflits sur le terme "message" lorsqu'il s'applique aux communications entre les composants ou entre les agents, on appelle les premiers "messages internes" et les seconds "messages externes" chaque fois que la confusion s'avère possible.

3. Tout lien direct entre une borne d'entrée d'un composant englobant et une borne d'entrée d'un composant englobé est supposé comprendre une transition, d'après les règles de construction des réseaux de Petri. Néanmoins, et afin de simplifier les figures, on pourra écrire des liens directs entre ces bornes en supposant qu'il existe une transition "virtuelle" entre ces bornes. La figure A.2 doit donc se réécrire sous la forme de la figure A.6.

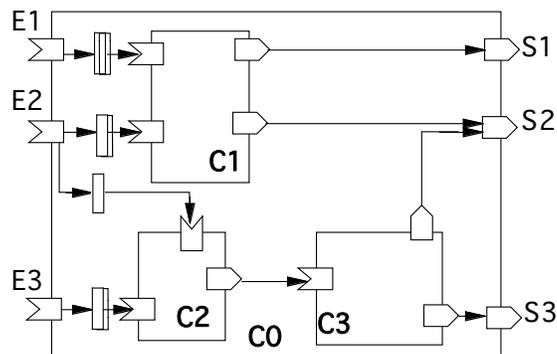


Figure A.6: Représentation des transitions intermédiaires entre bornes d'entrées

A.6 Traduction sous forme de réseaux de Petri

Il est toujours possible de transformer un ensemble de composants en un réseau de Petri équivalent qui représente une sorte "d'éclaté" de la structure BRIC, en considérant que les bornes d'entrée des composants correspondent à des places, alors que les bornes de sortie ne correspondent ni à des places ni à des transitions, mais servent uniquement de "fils conducteurs" vers les bornes d'entrée d'autres composants.

Par exemple, l'ensemble de composants BRIC de la figure A.7.a peut être représenté par le réseau de Petri équivalent de la figure A.7.b, les bornes de sortie ayant disparu, et les bornes d'entrée étant remplacées par des places.

A.7 Exemple

La figure A.8 montre un composant reprenant l'exemple célèbre en programmation par objet du compte en banque (Ferber 90), sur lequel on peut déposer (borne **déposer**) ou retirer (borne **retirer**) des sommes. Il est aussi possible de consulter le solde des dépôts et retraits. Le solde est représenté par une place $S(v)$.

On constate qu'on ne peut retirer que des sommes laissant le compte couvert. Toute tentative pour retirer une somme supérieure au montant du solde entraînera la production du message `impossibleTirer(x)` en sortie où x est la valeur que l'on cherche à retirer.

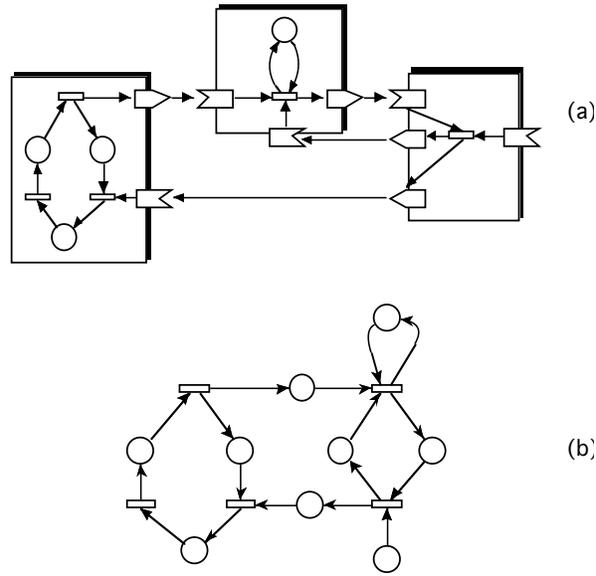


Figure A.7: La définition d'un ensemble de composants BRIC (a) se transforme en un réseau de Petri équivalent (b).

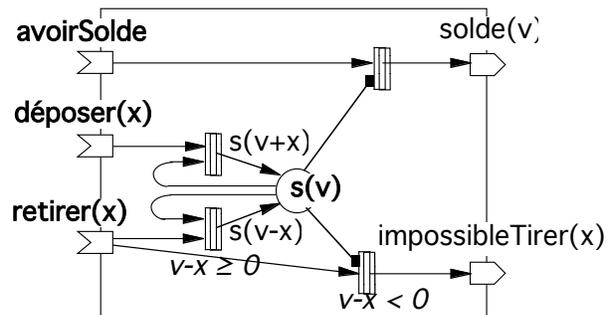


Figure A.8: Un composant BRIC représentant un compte en banque

Références bibliographiques

- Agha G. (1986) *Actors: A Model of Concurrent Computation for Distributed Systems*. MIT Press.
- Agre P. E. et Chapman D. (1987) "Pengi: an Implementation of a Theory of Activity." In *AAAI-87*, p. 268-272, Morgan Kaufmann.
- Allen J. F. et Perrault C. R. (1980) "Analyzing Intention in Utterances". *Artificial Intelligence*. **15**, p. 143-178.
- Andler D. (Ed.) (1992) *Introduction aux sciences cognitives*. Gallimard.
- Atlan H. (1979) *Entre le cristal et la fumée. Essai sur l'organisation du vivant*. Le Seuil.
- Austin J. L. (1962) *How to Do Things With Words*. Clarendon Press. Trad. fr. *Quand dire c'est faire*, Le Seuil, 1970.
- Axelrod R. (1992) *Donnant, Donnant. Théorie du comportement coopératif*. Editions Odile Jacob.
- Bachatène H. et Estrailier P. (1992) *A Modular Design on Stepwise Refinements of Coloured Petri Nets*. Research Report, MASI, Université Paris 6.
- Bachimont B. (1992) *Le Contrôle dans les systèmes à base de connaissances*. Hermès.
- Barraquand J. et Latombe J.-C. (1989) *Robot Motion Planning: A Distributed Representation Approach*. Research Report, STAN-CS-89-1257, Department of Computer Science, Stanford University.
- Barwise J. et Perry J. (1983) *Situations and Attitudes*. MIT Press.
- Bateson G. (1979) *La Nature et la pensée*. Le Seuil.
- Bateson G., Birdwhistell R., Goffman E. et al. (1981) *La Nouvelle communication*. Le Seuil.
- Baylon C. et Mignot X. (1991) *La Communication*. Nathan.
- Beckers R., Holland O. E. et Deneubourg J.-L. (1994) "From Local Actions to Global Tasks: Stigmergy and Collective Robotics." In *Proc. of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, Artificial Life IV*, R. Brooks et P. Maes (Ed.), MIT Press.
- Beer R. D. et Chiel H. J. (1992) "The Neural Basis of Behavioral Choice in an Artificial Insect." In *From Animals to Animats: Proc. Of the First International Conference on Simulation of Adaptive Behavior (SAB'90)*, Paris, J.-A. Meyer et S. W. Wilson (Ed.), p. 247-254, MIT Press.
- Benda M., Jaganathan V. et Dodhiawala R. (1986) "On Optimal Cooperation of Knowledge Sources." In *Sixth Workshop on Distributed Artificial Intelligence*, Boing AI Center, Seattle, WA.

- Bernard-Weil E. (1988) *Précis de systémique ago-antagoniste. Introduction aux stratégies bilatérales*. Limonest, L'Interdisciplinaire.
- Bertalanffy L. v. (1968) *General System Theory*. New York, Braziller.
- Bond A. et Gasser L. (1988) *Readings in Distributed Artificial Intelligence*. Morgan Kaufman.
- Bond A. H. (1989) "Commitment, Some DAI Insights from Symbolic Interactionist Sociology." In *Ninth Workshop on Distributed Artificial Intelligence*, Seattle.
- Bond A. H. et Gasser L. (1988) "An Analysis of Problems and Research in Distributed Artificial Intelligence." In *Readings in Distributed Artificial Intelligence*, A. H. Bond et L. Gasser (Ed.), Morgan Kaufman.
- Bonnet C., Ghiglione R. et Richard J.-F. (1990) *Traité de psychologie cognitive, tome 1*. Dunod.
- Booch G. (1991) *Object Oriented Design with Applications*. Benjamin Cummings.
- Bouron T. (1991) "What Architecture for Multi-Agent Systems." In *AAAI'91 Workshop on Cooperation among heterogeneous intelligent systems*.
- Bouron T. (1992) *Structures de communication et d'organisation pour la coopération dans un univers multi-agents*. Thèse de l'université Paris 6.
- Bouron T., Ferber J. et Samuel F. (1990) "MAGES: a Multi-Agent Testbed for Heterogeneous Agents." In *Decentralized Artificial Intelligence (Vol II)*, Y. Demazeau et J.-P. Muller (Ed.), North-Holland.
- Bousquet F., Cambier C., Mullon C. et al. (1993) "Simulating the Interaction between a Society and a Renewable Resource". *Journal of Biolog. Systems*. **1**(2), p. 199-214.
- Boutot A. (1993) *L'invention des formes*. Editions Odile Jacob.
- Braitenberg V. (1984) *Vehicles: Experiments in Synthetic Psychology*. MIT Press.
- Brassac C. (1992) "Analyse de conversation et théorie des actes de langage". *Cahiers de linguistique française*. **13**.
- Bratman M. (1987) *Intention, Plans, and Practical Reason*. Harvard University Press.
- Bratman M. (1990) "What Is Intention?" In *Intentions in Communications*, P. R. Cohen, J. Morgan et M. E. Pollack (Ed.), Cambridge, Bradford Book/MIT Press.
- Briot J.-P. (1989) "Actalk: a Testbed for Classifying and Designing Actor Languages in the Smalltalk-80 Environment." In *Proc. of ECOOP '89*, Nottingham, UK, p. 109-129.
- Brooks R. et Connell J. H. (1986) "Asynchronous Distributed Control System for a Mobile Robot". *SPIE*. **727**, Mobile Robots.
- Brooks R. (1990) "Elephants Don't Play Chess". *Robotics and Autonomous Systems*. **6**.
- Bura S., Drogoul A., Ferber J. et Jacopin E. (1991) "Eco-résolution: un modèle de résolution par interactions." In *Huitième congrès sur la reconnaissance des formes et l'intelligence artificielle (RFIA'91)*, Lyon, AFCET.
- Bura S., Guérin-Pace F., Mathian H., Pumain D. et Sanders L. (1993) "Multi-Agents Systems and the Dynamics of a Settlement System." In *Simulating Societies Symposium*, Siena, C. Castelfranchi (Ed.).

- Burg B. et Arlabosse F. (1994) "ARCHON: une plateforme industrielle pour l'intelligence artificielle distribuée." In *Actes des deuxièmes journées francophones sur l'intelligence artificielle distribuée et les systèmes multi-agents (JFI-ADSMA '94)*, Voiron, Y. Demazeau et S. Pesty (Ed.), IMAG.
- Burloud A. (1936) *Principes d'une psychologie des tendances*. Paris.
- Cambier C. (1994) *SIMDELTA: un système multi-agent pour simuler la pêche sur le Delta Central du Niger*. Thèse de l'université Paris 6.
- Cambier C., Bousquet F. et Dansoko D. (1992) "Un Univers multi-agents pour la modélisation du système de la pêche du Delta Central du Niger." In *Premier colloque africain sur la recherche en informatique (CARI'92)*, Yaoundé.
- Cammarata S., McArthur D. et Steeb R. (1983) "Strategies of Cooperation in Distributed Problem Solving." In *Proc. of the 1983 IJCAI Conference*.
- Carle P. (1992) *Un Langage d'acteur pour l'intelligence artificielle distribuée intégrant objets et agents par réflexivité compilatoire*. Thèse de l'université Paris 6.
- Castañeda H.-N. (1975) *Thinking and Doing*. Dordrecht, Holland, D. Riedel.
- Castelfranchi C. et Conte R. (1992) "Mind is not Enough: Precognitive Bases of Social Interaction." In *Simulating Societies Symposium*, Guildford, N. Gilbert (Ed.).
- Castelfranchi C. et Werner E. (Ed.) (1994) *Artificial Social Systems, Proc. of Maamaw'92*. Springer-Verlag.
- Chaib-Draa B. (1989) *Contribution à la résolution distribuée de problème: approche basée sur les comportements intentionnels des agents*. Thèse de l'université de Valenciennes.
- Chaib-Draa B., Moulin M., Mandiau R. et Millot P. (1992) "Trends in Distributed Artificial Intelligence". *Artificial Intelligence Review*, **6**, p. 35-66.
- Chapman D. (1987) "Planning for Conjunctive Goals". *Artificial Intelligence Journal*, **32**, p. 333-367.
- Charest J. (1980) *La Conception des systèmes: une théorie, une méthode*. Québec, Gaëton Morin.
- Cherian S. et Troxell W. O. (1993) "Neural Network Based Behavior Hierarchy for Locomotion Control." In *From Animal to Animats 2, Proceedings of the Second International Conference on Simulation of Adaptive Behavior (SAB'92)*, Hawaii, J.-A. Meyer, H. L. Roitblat et S. W. Wilson (Ed.), p. 61-70, MIT Press.
- Chevrier V. (1993) *Etude et mise en œuvre du paradigme multi-agent: de Atome à Gtmas*. Thèse de l'université de Nancy I.
- Cliff D., Harvey I. et Husbands P. (1993) "Exploration in Evolutionary Robotics". *Adaptive Behavior*, **2**(1), p. 73-110.
- Cohen P. R. et Levesque H. J. (1990a) "Intention is Choice with Commitment". *Artificial Intelligence*, **42**, p. 213-261.
- Cohen P. R. et Levesque H. J. (1990b) "Persistence, Intention and Commitment." In *Intentions in Communication*, P. R. Cohen, J. Morgan et M. E. Pollack (Ed.), p. 33-69, MIT Press.
- Cohen P. R. et Levesque H. J. (1990c) "Rational Interaction as the Basis for Communication." In *Intentions in Communications*, P. R. Cohen, J. Morgan et

- M. E. Pollack (Ed.), p. 508, MIT Press.
- Cohen P. R. et Levesque H. J. (1995) "Communicative Actions for Artificial Intelligence." In *First International Conference on Multi-Agent Systemes (ICMAS'95)*, San Francisco, V. Lesser (Ed.), MIT Press.
- Cohen P. R. et Perrault C. R. (1979) "Elements of a Plan Based Theory of Speech Acts". *Cognitive Science*. **3**(3).
- Collins R. J. et Jefferson D. R. (1991) "Representation for Artificial Organisms." In *From Animal to Animats*, Paris, J.-A. Meyer et S. W. Wilson (Ed.), p. 382-390, MIT Press.
- Conry S., Meyer R. A. et Lesser V. (1988) "Multistage Negotiation in Distributed Artificial Intelligence." In *Readings in Distributed Artificial Intelligence*, A. Bond et L. Gasser (Ed.), Morgan Kaufman.
- Conte R., Miceli M. et Castelfranchi C. (1991) "Limits and Levels of Cooperation: Disentangling Various Types of Prosocial Interaction." In *Distributed A.I. 2*, Y. Demazeau et J.-P. Müller (Ed.), North-Holland.
- Corbara B. (1991) *L'Organisation sociale et sa genèse chez la fourmi *Ectatomma ruidum* Roger*. Thèse de l'université Paris XIII.
- Corbara B., Drogoul A., Fresneau D. et Lalande S. (1993) "Simulating the Sociogenesis Process in Ant Colonies with MANTA." In *Towards a Practice of Autonomous Systems II*, P. Bourguin et F. Varela (Ed.), MIT Press.
- Coria M. (1993) "Stepwise Development of Correct Agents: a Behavioural Approach Based of Coloured Petri Nets." In *Proc. of the Canadian Conf. on Electrical and Computer Engineering*, Vancouver, Canada.
- Corrêa M. et Coelho H. (1993) "Around the Architectural Agent Approach to Model Conversations." In *Proc. of the 5th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Neuchâtel.
- Couffignal L. (1963) *La Cybernétique*. Vol. 638, PUF.
- Crozier M. et Friedberg E. (1977) *L'Acteur et le système*. Le Seuil.
- Cyrułnik B. (1989) *Sous le signe du lien*. Hachette.
- David R. et Alla H. (1989) *Du Grafset aux réseaux de Petri*. Hermès.
- Davis R. et Smith R. J. (1983) "Negotiation as a Metaphor for Distributed Problem Solving". *Artificial Intelligence*. **20**(1), p. 63-109.
- De Medio C. et Oriolo G. (1991) "Robot Obstacle Avoidance Using Vortex Fields." In *Advances in Robot Kinematics*, S. Stifter et Lenarcic (Ed.), p. 227-235, Springer Verlag.
- Delattre P. (1968) "Structure et fonction (Biologie)." In *Encyclopaedia Universalis*, vol. 15, (Ed.), p. 442-444.
- Delattre P. (1971) *Système, structure, fonction, évolution*. Maloine.
- Delaye C. (1993) *Structures et organisations des systèmes multi-agents autonomes et adaptatifs*. Thèse de l'université Paris 6.
- Demazeau Y., Boissier O. et Koning J.-L. (1994) "Using Interaction Protocols to Control Vision Systems." In *IEEE International Conference on Systems, Man and Cybernetics*, San Antonio.
- Demazeau Y. et Müller J.-P. (Ed.) (1990) *Decentralized Artificial Intelligence*. Elsevier North-Holland.

- Demazeau Y. et Müller J.-P. (Ed.) (1991) *Decentralized AI 2*. Elsevier North-Holland.
- Deneubourg J.-L. et al. e. (1993) "Self-Organisation and life: from Simple Rules to Global Complexity." In *Second European Conference on Artificial Life.*, Bruxelles.
- Deneubourg J.-L., Goss S., Sendova-Franks A., Detrain C. et Chretien L. (1991) "The Dynamics of Collective Sorting Robot-like Ants and Ant-like Robots." In *From Animals to Animats*, Paris, J.-A. Meyer et S. W. Wilson (Ed.), p. 356-363, MIT Press.
- Deneubourg J. L., Aron S., Goss S., Pasteels J. M. et Duerinck G. (1986) "Random Behaviour, Amplification Processes and Number of Participants: How they Contribute to the Foraging Properties of Ants". *Physica*. **22**(D), p. 176-186.
- Deneubourg J. L. et Goss S. (1989) "Collective Patterns and Decision-Making". *Ecology, Ethology and Evolution*. **1**, p. 295-311.
- Dennett D. (1991) *La Conscience expliquée*. Editions Odile Jacob.
- Dennett D. C. (1983) "Intentional Systems in Cognitive Ethology: the Panglossian Paradigm Defended". *The Behavioral and Brain Sciences*. **6**, p. 343-390.
- Dennett D. C. (1987) *The Intentional Stance*. M.I.T. Press. Trad. fr. *La Stratégie de l'interprète*, Essais, Gallimard, 1990.
- Devlin K. (1991) *Logic and Information*. Cambridge University Press.
- Doyle J. (1979) "A Truth Maintenance System". *AI Journal*. **12**, p. 231-272.
- Dreyfus H. (1979) *What Computers can't do: a Critique of Artificial Reason*. Harper.
- Dreyfus H. L. (1992) "La Portée philosophique du connexionisme." In *Introduction aux sciences cognitives*, D. Andler (Ed.), p. 352-373, Gallimard.
- Drogoul A. (1993) *De la simulation multi-agent à la résolution collective de problèmes. Une étude de l'émergence de structures d'organisation dans les systèmes multi-agents..* Thèse de l'université Paris 6.
- Drogoul A., Corbara B. et Fresneau D. (1993) "MANTA: New Experimental Results on the Emergence of (Artificial) Ant Societies." In *Simulating Societies Symposium*, Siena, C. Castelfranchi (Ed.).
- Drogoul A. et Dubreuil C. (1992) "Eco-Problem Solving: Results of the N-Puzzle." In *Decentralized Artificial Intelligence III*, Y. Demazeau et E. Werner (Ed.), North Holland.
- Drogoul A. et Ferber J. (1992) "From Tom Thumb to the Dockers: Some Experiments with Foraging Robots." In *From Animals To Animats: Second Conference on Simulation of Adaptive Behavior (SAB 92)*, Hawaii, J.-A. Meyer, H. Roitblat et S. Wilson (Ed.), MIT Press.
- Drogoul A. et Ferber J. (1994a) "Multi-Agent Simulation as a Tool for Modeling Societies: Application to Social Differentiation in Ant colonies." In *Artificial Social Systems*, vol. 830, C. Castelfranchi et E. Werner (Ed.), p. 3-23, Berlin, Springer-Verlag.
- Drogoul A. et Ferber J. (1994b) "Multi-agent Simulation as a Tool for Studying Emergent Processes in Societies." In *Simulating Societies: the Computer Simulation of Social Phenomena*, N. Gilbert et J. Doran (Ed.), p. 127-142, UCL Press.
- Drogoul A., Ferber J., Corbara B. et Fresneau D. (1992) "A Behavioral Simulation Model for the Study of Emergent Social Structures." In *Towards a Practice*

- of *Autonomous Systems*, Paris, P. Bourguine et F. Varela (Ed.), p. pp161-170, MIT Press.
- Drogoul A., Ferber J. et Jacopin E. (1991) "Pengi: Applying Eco-Problem Solving for Behavior Modeling in an Abstract Eco-System." In *European Simulation Multiconference, ESM'91*, E. Mosekilde (Ed.).
- Dupuy J.-P. (1994) *Aux origines des sciences cognitives*. La Découverte.
- Durfee E. et Montgomery T. (1989) "MICE: A Flexible Testbed for Intelligent Coordination Experiments." In *Proc. of Ninth Workshop on Distributed AI*, Orcas Islands, Seattle, M. Benda (Ed.), Boing Computer Services.
- Durfee E. H. et Lesser V. R. (1991) "Global Partial Planning: A Coordination Framework for Distributed Hypothesis". *IEEE Transactions on Systems, Man and Cybernetics*. **21**(6).
- Durfee E. H., Lesser V. R. et Corkill D. D. (1987a) "Coherent Cooperation Among Communicating Problem Solvers". *IEEE Transactions on Computers*. **C**(36), p. 1275-1291.
- Durfee E. H., Lesser V. R. et Corkill D. D. (1987b) "Cooperation through Communication in a Distributed Problem Solving Network." In *Distributed Artificial Intelligence*, M. Huhns (Ed.), Pitman.
- Durfee E. H., Lesser V. R. et Corkill D. D. (1989) "Cooperative Distributed Problem Solving." In *The Handbook of Artificial Intelligence*, vol. IV, A. Barr, P. R. Cohen et E. A. Feigenbaum (Ed.), p. 83-148, Addison-Wesley.
- Durkheim E. (1897) *Le Suicide*. PUF.
- Eco U. (1988) *Sémiotique et philosophie du langage*. PUF.
- El Fallah-Seghrouchni A. et Haddad S. (1994) "Représentation et manipulation de plans à l'aide de réseaux de Petri." In *Actes des deuxièmes journées francophones sur l'intelligence artificielle distribuée et les systèmes multi-agents (JFI-ADSDMA'94)*, Voiron, Y. Demazeau et S. Pesty (Ed.), IMAG.
- Engelmore R. et Morgan T. (1988) *Blackboard Systems*. Addison-Wesley.
- Enjalbert P. (1989) "Notes préliminaires à une théorie opérationnelle du sens". *Intellectica*. **8**, p. 109-159.
- Enjalbert P. (1993) *Théories du signe selon Umberto Eco*. Rapport interne, 93-10, Laboratoire d'algorithmique et d'intelligence artificielle de Caen, Université de Caen.
- Erceau J. et Ferber J. (1991) "L'intelligence artificielle distribuée." In *La Recherche*, Juin.
- Erman L., Hayes-Roth F., Lesser V. et Reddy D. (1980) "The HEARSAY-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty". *ACM Computing Surveys*. **12**.
- Estraillier P. et Girault C. (1992) "Applying Petri Net Theory to the Modelling Analysis and Prototyping of Distributed Systems." In *Proceedings of the IEEE International Workshop on Emerging Technologies and Factory Automation*, Cairns, Australia.
- Ferber F. (1906) *Pas à pas, saut à saut, vol à vol*. Berger-Levrault.
- Ferber J. (1987) "Des objets aux agents: une architecture stratifiée." In *Sixième congrès RFIA*, Antibes, AFCET.

- Ferber J. (1989) *Objets et agents: une étude des structures de représentation et de communications en Intelligence Artificielle*. Thèse d'Etat, Université Paris 6.
- Ferber J. (1990) *Conception et programmation par objets*. Hermès.
- Ferber J. (1994) "La Kénétique: des systèmes multi-agents à une science de l'interaction". *Revue internationale de systémique*. **8**(1), p. 13-27.
- Ferber J. (1995a) "Basis of Cooperation in Multi-Agent Systems." In *Proc. of the 95 European Conference on Cognitive Science*, Saint-Malo, INRIA.
- Ferber J. (1995b) "Reactive Distributed Artificial Intelligence: Principles and Applications." In *Foundations of Distributed Artificial Intelligence*, N. Jennings (Ed.), Wiley.
- Ferber J. et Carle P. (1991) "Actors and Agents as Reflective Concurrent Objects: a Mering IV Perspective". *IEEE Trans on Systems, Man and Cybernetics*. **21**(6).
- Ferber J., Carle P., Mousseau D. et Furet T. (1993) "Synchronisation de messages par utilisation de la réflexivité du langage d'acteurs Mering IV." In *Actes des journées françaises sur les langages applicatifs, JFLA'93*.
- Ferber J. et Drogoul A. (1992) "Using Reactive Multi-Agent Systems in Simulation and Problem Solving." In *Distributed Artificial Intelligence: Theory and Practice*, L. Gasser et N. Avouris (Ed.), Kluwer Academic Publishers.
- Ferber J. et Erceau J. (1994) "L'intelligence artificielle distribuée et la conception de systèmes multi-agents." In *Intelligence Collective*, E. Bonabeau et P. Bourguin (Ed.), Hermès.
- Ferber J. et Magnin L. (1994) "Conception de systèmes multi-agents par composants modulaires et réseaux de Petri." In *Actes des journées du PRC-IA*, Montpellier.
- Ferraris C. et Haton M. C. (1992) "Un guide méthodologique pour l'acquisition des connaissances dans le cadre d'un système multi-experts." In *Actes des journées d'acquisition des connaissances*, Dourdan.
- Fikes R. E. et Nilsson N. J. (1971) "STRIPS: a New Approach to the Application of Theorem Proving to Problem Solving". *Artificial Intelligence*. **2**(3-4), p. 189-208.
- Finin T., Fritzson R., McKay D. et McEntire R. (1994) "KQML as an Agent Communication Language." In *3rd International Conference on Information and Knowledge Management (CIKM'94)*, ACM Press.
- Fodor J. (1983) *The Modularity of the Mind*. Cambridge, Mass, MIT Press. Trad. fr. *La modularité de l'esprit*, Editions de Minuit, 1983.
- Forrester J. W. (1980) *Principes des Systèmes*. Presses Universitaires de Lyon.
- Fox M. (1981) "An Organizational View of Distributed Systems". *IEEE Trans. on Man, Systems and Cybernetics*. **11**(1), p. 70-79.
- Fresneau D. (1994) *Biologie et comportement social d'une fourmi ponerine néotropicalique*. Thèse d'Etat, Université Paris XIII.
- Friedberg E. (1993) *Le Pouvoir et la règle*. Le Seuil.
- Fron A. (1994) *Programmation par contraintes*. Addison-Wesley France.
- Galliers J. R. (1991) "Modelling Autonomous Belief Revision in Dialogue." In *Decentralized Artificial Intelligence 2: Proc. of the Second European Workshop on Autonomous Agents in a Multi-Agents World (MAAMAW'90)*, Y. Demazeau et P. Müller (Ed.), Elsevier North Holland.

- Gasser L. (1991) "Social Conceptions of Knowledge and Action: DAI Foundations and Open Systems Semantics". *Artificial Intelligence*. **47**(1-3), p. 107-138.
- Gasser L. (1992) "An Overview of DAI." In *Distributed Artificial Intelligence: Theory and Praxis*, L. Gasser et N. M. Avouris (Ed.), Kluwer Academic Publishers.
- Gasser L. (1995) "Computational Organization Research." In *1st International Conference on Multi-Agent Systems*, San Francisco, V. Lesser (Ed.), MIT Press.
- Gasser L., Braganza C. et Herman N. (1987a) "Implementing Distributed Artificial Intelligence Systems using MACE." In *IEEE Conference on Artificial Intelligence Applications*, p. 315-320.
- Gasser L., Braganza C. et Herman N. (1987b) "MACE: a Flexible Testbed for Distributed AI Research." In *Distributed Artificial Intelligence*, M. N. Huhns (Ed.), p. 119-152, Pitman.
- Gasser L. et Hill R. W. (1990) "Coordinated Problem Solvers". *Annual Reviews of Computer Science*. **4**, p. 203-253.
- Gasser L. et Huhns M. (Ed.) (1989) *Distributed Artificial Intelligence*. Vol. II, Pitman/Morgan Kaufman.
- Gasser L., Rouquette N., Hill R. et Lieb J. (1989) "Representing and Using Organizational Knowledge in Distributed AI Systems." In *Distributed Artificial Intelligence*, vol. II, L. Gasser et M. Huhns (Ed.), Pitman.
- Genesereth M. R. et Nilsson N. J. (1987) *Logical Foundations of Artificial Intelligence*. Morgan Kaufman.
- Georgeff M. (1984) "A Theory of Action for MultiAgent Planning." In *AAAI-84*, p. 121-125.
- Georgeff M. (1986) "The Representation of Events in Multi-Agent Domains." In *AAAI-86*, p. 70-75.
- Georgeff M. P. (1983) "Communication and Interaction in Multi-Agent Planning." In *AAAI-83*, p. p125-129.
- Ghallab M., Grandjean P., Lacroix S. et Thibault J.-P. (1992) "Représentation et raisonnement pour une machine multi-sensorielle." In *Actes des quatrième journées nationales du PRC-GDR intelligence artificielle*, Marseille, p. 121-167, Teknea.
- Ghedira K. (1993) *MASC: une approche multi-agents des problèmes de satisfaction de contraintes*. Thèse de l'ENSAE.
- Ghedira K. (1994) "Partial Constraint Satisfaction by a Multi-Agent Simulated Annealing approach." In *Int. Conf. of AI, KBS, ES and NL*, Paris.
- Ghiglione R. (1986) *L'Homme communicant*. Armand Colin.
- Ghiglione R. (1989) "Le "qui" et le "comment"." In *Perception, Action, Langage - Traité de Psychologie Cognitive*, vol. 3, C. Bonnet, R. Ghiglione et J.-F. Richard (Ed.), p. 175-226, Dunod.
- Giroux S. (1993) *Agents et systèmes, une nécessaire unité*. Thèse de Doctorat, Université de Montréal.
- Giroux S. et Senteni A. (1992) *Steps to an Ecology of Concurrent Computation*. Rapport de recherche, Université de Montréal.
- Gleick J. (1989) *La Théorie du chaos*. Flammarion.

- Gleizes M.-P., Glize P. et Trouilhet S. (1994) "Etude des lois de la conversation entre agents autonomes". *Revue internationale de systématique*. **8**(1), p. 39-50.
- Goldberg A. et Robson D. (1980) *Smalltalk-80: The language and its Implementation*. Addison-Wesley.
- Goldberg D. (1989) *Genetic Algorithms*. Addison-Wesley.
- Grice H. P. (1975) "Logic and Conversation." In *Syntax and Semantics*, vol. 3, Speech Acts, P. Cole et J. L. Morgan (Ed.), p. 41-58, Academic Press.
- Guha R. V. et Lenat D. (1990) "CYC: A Midterm Report." In *AI Magazine*,
- Gurvitch G. (1963) *La vocation actuelle de la sociologie*. PUF.
- Harel D. (1979) *First Order Dynamic Logic*. Vol. 68, Springer-Verlag.
- Haton J.-P., Bouzid N., Charpillat F. et al. (1991) *Le Raisonnement en intelligence artificielle*. InterEditions.
- Hayes-Roth B. (1985) "A Blackboard Architecture for Control". *Artificial Intelligence*. **26**(3), p. 251-321.
- Hayes-Roth B. et Collinot A. (1993) "A Satisficing Cycle for Real-Time Reasoning in Intelligent Agents". *Expert Systems with Applications*. **7**, p. 31-42.
- Hayes-Roth B., Washington R., Ash D. et al. (1992) "Guardian, a Prototype Intelligent Agent for Intensive-Care Monitoring". *Artificial Intelligence in Medicine*. **4**(2).
- Hayes-Roth F., Erman L., Fouse S., Lark J. S. et Davidson J. (1988) "ABE: a Cooperative Operating System and Development Environment." In *Readings in Distributed Artificial Intelligence*, A. Bond et L. Gasser (Ed.), Morgan Kaufman.
- Heudin J.-C. (1994) *La Vie artificielle*. Paris, Hermès.
- Hewitt C. (1977) "Viewing Control Structures as Patterns of Message Passing". *Artificial Intelligence*. **8**(3), p. 323-374.
- Hewitt C. (1985) "The Challenge of Open Systems." In *Byte*,
- Hewitt C. (1991) "Open Information Systems Semantics for Distributed Artificial Intelligence". *Artificial Intelligence (special issue on foundations of AI)*. **47**(1-3), p. 79-106.
- Hewitt C., Bishop P. et Steiger R. (1973) "A Universal Modular Actor Formalism for Artificial Intelligence." In *Third International Joint Conference on Artificial Intelligence*.
- Hickman S. et Shiels M. (1991) "Situated Action as a Basis for Cooperation." In *Decentralized A.I. 2*, E. Werner et Y. Demazeau (Ed.), Elsevier North Holland.
- Holland J. H. (1968) "Processing and Processors for Schemata." In *Associative Information Processing*, E. L. Jacks (Ed.), p. 127-146, American Elsevier.
- Holland J. H. (1978) "Cognitive Systems Based on Adaptive Algorithms." In *Pattern directed inference systems*, D. A. Waterman et F. Hayes-Roth (Ed.), p. 313-329, Academic Press.
- Holland J. H., Holyoak K. J., Nisbett R. E. et Thaghard P. R. (1986) *Induction: Processes of Inference, Learning and Discovery*. MIT Press.
- Huhns M. N. (Ed.) (1987) *Distributed Artificial Intelligence*. Pitman.
- Husserl E. (1950) *Idées directrices pour une phénoménologie*. Gallimard.
- Iffenecker C. (1992) *Un Système multi-agents pour le support des activités de conception de produits*. Thèse d'université, Université Paris 6.

- Iffenecker C. et Ferber J. (1992) "Using Multi-Agent Architecture for Designing Electomechanical Products." In *Actes de la conférence Avignon'92 sur les Systèmes Experts et leurs Applications*, Avignon, EC2.
- Ishida T. (1989) "CoCo: A Multi-Agent System for Concurrent and Cooperative Operation Tasks." In *9th Distributed Artificial Intelligence Workshop*, Orcas Island, M. Benda (Ed.).
- Ishikawa Y. et Tokoro M. (1986) "A Concurrent Object-Oriented Knowledge Representation Language Orient84/K: its features and implementation." In *OOP-SLA '86*, ACM Sigplan Notices.
- Jacopin E. (1993) *Algorithmique de l'interaction: le cas de la planification*. Thèse de l'université Paris 6.
- Jagannathan V., Dodhiawala R. et Baum L. S. (1989) *Blackboard Architectures and Applications*. Academic Press.
- Jakobson R. (1963) *Essai de linguistique générale*. Editions de Minuit.
- Jennings N. (1994) *Cooperation in Industrial Multi-Agent Systems*. Vol. 43, World Scientific Press.
- Jennings N., Corera J. M. et Laresgoiti I. (1995) "Developing Industrial Multi-Agent Systems." In *First International Conference on Multi-Agent Systems*, San Francisco, V. Lesser (Ed.), MIT Press.
- Jennings N. et Wittig T. (1992) "ARCHON: Theory and Practice." In *Distributed Artificial Intelligence: Theory and Practice*, L. Gasser et N. Avouris (Ed.), Kluwer Academic Publishers.
- Jensen K. (1992) *Coloured Petri Nets. Basic Concepts, Analysis Method and Practical Use*. Vol. 1, Springer-Verlag.
- Konolige K. (1986) *A Deduction Model of Belief*. Pitman.
- Korf R. E. (1985) "Depth-first Iterative-Deepening: an Optimal Admissible Tree Search". *Artificial Intelligence*. **27**.
- Korf R. E. (1988) "Real Time Heuristic Search". *Artificial Intelligence*. **42**.
- Kornfeld W. (1979) "ETHER: a Parallel Problem Solving System." In *Proc. of 6th IJCAI*.
- Kornfeld W. et Hewitt C. (1980) *The Scientific Community Metaphor*. AI Lab Memo, 641, MIT.
- Kripke S. A. (1991) "Semantical Considerations on Modal Logic." In *Reference and Modality*, Linsky (Ed.), p. 63-72, Oxford University Press.
- Kuhn T. (1962) *The Structure of Scientific Revolutions*. University of Chicago Press.
- Lâasri H. et Maître B. (1989) *Coopération dans un univers multi-agent basée sur le modèle du blackboard: études et réalisations*. Thèse de l'université de Nancy I.
- Lâasri H., Maître B. et Haton J.-P. (1987) "ATOME: outil d'aide au développement de systèmes multi-experts." In *Actes des sixièmes journées sur la reconnaissance des formes et l'intelligence artificielle (RFIA '87)*, Antibes, p. 749-759.
- Lakatos I. (1976) *Proofs and Refutations*. Cambridge University Press.
- Lalanda P., Charpillat F. et Haton J. P. (1992) "A Real Time Blackboard Based Architecture." In *10th European Conference on Artificial Intelligence*, Vienne, Autriche.
- Langton C. (Ed.) (1988) *Artificial Life*. Addison-Wesley.

- Langton C., Taylor C., Farmer J. D. et Rasmussen S. (Ed.) (1990) *Artificial Life II*. Addison-Wesley.
- Langton C. G. (Ed.) (1994) *Artificial Life III*. Addison-Wesley.
- Lapierre J.-W. (1992) *L'Analyse de systèmes*. Syros.
- Latombe J.-C. (1991) *Robot Motion Planning*. Kluwer Academic Publishers.
- Latour B. (1989) *La Science en action*. La Découverte.
- Le Gallou F. (1992) "Activités des systèmes, décomposition des systèmes." In *Systémique. Théorie et applications*, F. Le Gallou et B. Bouchon-Meunier (Ed.), p. 71-100, Lavoisier.
- Le Moigne J.-L. (1977) *La Théorie du système général*. PUF.
- Le Strugeon E. (1995) *Une méthodologie d'auto-adaptation d'un système multi-agents cognitifs*. Thèse de l'université de Valenciennes.
- Lefèvre C. (1995) *Agents logiques communicants*. Thèse de l'université de Caen.
- Lehmann F. (Ed.) (1992) *Semantic Networks in Artificial Intelligence*. Pergamon Press.
- Lenat D. (1975) "BEINGS: Knowledge as Interacting Experts." In *Proc. of the 1975 IJCAI Conference*, p. 126-133.
- Lenat D. et Brown J. S. (1984) "Why AM and Eurisko Appear to Work". *Artificial Intelligence*. **23**, p. 269-294.
- Lenat D. et Guha R. V. (1990) *Building Large Knowledge-Based Systems*. Addison-Wesley.
- Lespérance Y., Levesque H., Lin F. et al. (1995) "Fondements d'une approche logique à la programmation d'agents." In *Actes des troisièmes journées francophones sur l'intelligence artificielle distribuée et les systèmes multi-agents*, Chambéry.
- Lesser V. R. et Corkill D. D. (1983) "The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks". *AI Magazine*. **4**(3), p. 15-33.
- Lestel D. (1986) *Contribution à l'étude du raisonnement expérimental dans un domaine sémantiquement riche*. Thèse de doctorat, EHESS.
- Lestel D., Grison B. et Drogoul A. (1994) "Les agents réactifs et le vivant dans une perspective d'évolution coopérative". *Intellectica*. **19**, p. 73-90.
- Levine M. D. (1985) *Vision in Man and Machine*. Prentice-Hall.
- Lin L.-J. (1992) "Self-improving Reactive Agents: Case Studies of Reinforcement Learning Frameworks." In *From Animals to Animats: Proc. Of the First International Conference on Simulation of Adaptive Behavior (SAB'90)*, Paris, J.-A. Meyer et S. W. Wilson (Ed.), p. 297-305, MIT Press.
- Liu J. S. et Sycara K. (1993) "Emergent Constraint Satisfaction through Multi-Agent Coordinated Interaction." In *Maamaw'93*, Neuchâtel.
- Livet P. (1994) *La communauté virtuelle*. Editions de l'Eclat.
- Lorenz K. (1984) *Les Fondements de l'éthologie*. Flammarion.
- Maes P. (1990) "Situated Agents can have Goals". *Journal of Robotics and Autonomous Systems*. **6**, p. 49-70.
- Maes P. (1991) "A Bottom-Up Mechanism for Behavior Selection in an Artificial Creature." In *From Animals to Animats: Proc. Of the First International Conference on Simulation of Adaptive Behavior (SAB'90)*, Paris, J.-A. Meyer et S. W.

- Wilson (Ed.), p. 238-246, MIT Press.
- Maes P. (1992) "Learning Behavior Networks from Experience." In *Toward a Practice of Autonomous Systems, Proc. of the First European Conference on Artificial Life*, Paris, F. Varela et P. Bourguine (Ed.), p. 48-57, MIT Press.
- Maes P. et Nardi D. (Ed.) (1988) *Meta-level Architectures and Reflexion*. North-Holland.
- Malinowski B. (1944) *A Scientific Theory of Culture and Others Essays*. Chapel Hill, North Carolina Press. Trad. fr. *Une théorie scientifique de la culture*, Le Seuil, 1968.
- Malone T. W. (1988) "What is coordination theory." In *National Science Foundation Coordination Theory Workshop*, MIT.
- Marr D. (1982) *Vision: a Computational Investigation into the Human Representation and Processing of Visual Information*. Freeman.
- Maruichi T. (1989) *Organizational Computation*. Ph.D. Thesis, Keio University.
- Maruichi T., Ichikawa M. et Tokoro M. (1990) "Modeling Autonomous Agents and Their Groups." In *Decentralized A.I. 1*, Y. Demazeau et J.-P. Müller (Ed.), North-Holland.
- Masini G., Napoli A., Colnet D., Léonard D. et Tombre K. (1989) *Les Langages à objets*. InterEditions.
- Masoud A. et Bayoumi M. (1993) "Robot Navigation Using the Vector Potential Approach." In *Proc. of IEEE International Conference on Robotics and Automation*, Atlanta, p. 805-811.
- Mataric M. J. (1992) "Designing Emergent Behaviors: from Local Interactions to Collective Intelligence." In *From Animals to Animats 2, Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, Hawaii, J.-A. Meyer, H. L. Roitblat et S. W. Wilson (Ed.), p. 432-441, MIT Press.
- Mataric M. J. (1994) "Learning to Behave Socially." In *From Animals to Animats 3, Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, Brighton, D. Cliff, P. Husbands, J.-A. Meyer et S. W. Wilson (Ed.), p. 453-462, MIT Press.
- Maturana H. et Varela F. (1980) *Autopoiesis and Cognition: the Realization of the Living*. Boston, D. Reidel.
- Maturana H. et Varela F. (1987) *The Tree of Knowledge*. Boston, New Science Library. Trad. fr. *L'Arbre de la connaissance*, Addison-Wesley France, 1994.
- McCarthy J. (1980) "Circumscription: A Form of Non-Monotonic Reasoning". *Artificial Intelligence*. **13**, p. 27-39.
- McFarland D. (1990) *Dictionnaire du comportement animal*. Robert Laffont.
- McFarland D. (1994) "Towards Robot Cooperation." In *94' Conference on the Simulation of Adaptive Behavior*, Brighton, MIT Press.
- Mesarovic M. D. et Takahara Y. (1975) *General Systems Theory: Mathematical Foundations*. Academic Press.
- Mesle R. (1994) *ICHTYUS: architecture d'un système multi-agents pour l'étude de structures agrégatives*. Rapport de DEA, LAFORIA, Université Paris 6.
- Meyer J.-A. et Guillot A. (1989) "Simulation of Adaptive Behavior in Animats: Review and Prospect." In *From Animals to Animats*, Paris, J.-A. Meyer et S. W.

- Wilson (Ed.), MIT Press.
- Meyer J. A., Roitblat H. L. et Wilson S. W. (Ed.) (1993) *Simulation of Adaptive Behavior: From Animals to Animats II*. Cambridge, MIT Press.
- Meyer J. A. et Wilson S. (Ed.) (1991) *Simulation of Adaptive Behavior: From Animals to Animats*. MIT Press.
- Milner R. (1989) *Communication and Concurrency*. Prentice-Hall International.
- Minsky M. (1988) *La Société de l'Esprit*. InterEditions.
- Mintzberg H. (1982) *Structure et dynamique des organisations*. Les Editions d'Organisation.
- MIRIAD E. (1992) "Approcher la notion de collectif." In *Actes des journées multi-agents du PRC-IA*, Nancy.
- Moeschler J. (1985) *Argumentation et conversation*. Hatier.
- Moeschler J. (1989) *Modélisation du dialogue: représentation de l'inférence argumentative*. Hermès.
- Montague R. (1974) *Formal Philosophy*. Yale University Press.
- Moore R. C. (1988) "Autoepistemic Logic." In *Non-Standard Logics for Automated Reasoning*, Smets (Ed.), p. 105-136, Academic Press.
- Morin E. (1977) *La Méthode (1): la Nature de la Nature*. Le Seuil.
- Morin E. (1991) *La Méthode (4): Les Idées. Leur habitat, leur vie, leurs moeurs*. Le Seuil.
- Morley R. E. et C. S. (1993) "An Analysis fo a Plant-Specific Dynamic Scheduler." In *Proc. of the NSF Workshop on Dynamic Scheduling.*, Cocoa Beach.
- Moulin B. et Cloutier L. (1994) "Collaborative Work Based on Multiagent Architectures: a Methodological Perspective." In *Soft Computing: Fuzzy Logic, Neural Networks and Distributed Artificial Intelligence*, F. Aminzadeh et M. Jamshidi (Ed.), p. 261-296, Prentice-Hall.
- Murata T. (1989) "Petri Nets: Properties, Analysis and Applications". *Proceedings of the IEEE*. **77**(4), p. 541-580.
- Newell A. (1982) "The Knowledge Level". *Artificial Intelligence*. **18**(1), p. 87-127.
- Newell A., Shaw J. et Simon H. (1957) "Empirical Explorations of the Logic Theory Machine: a Case Study in Heuristics." In *Computers and Thought (1963)*, E. Feigenbaum et J. Feldman (Ed.), p. 109-133, McGraw-Hill.
- Newton I. (1985) *Principia Mathematica*. C. Bourgeois.
- Nilsson N. (1980) *Principles of Artificial Intelligence*. Springer-Verlag.
- Overgaard L., Petersen H. G. et Perram J. W. (1994) "Motion Planning for an Articulated Robot: a Multi-Agent Approach." In *MAAMAW'94*, Odense, Danemark, Y. Demazeau, J.-P. Müller et J. Perram (Ed.).
- Pacherie E. (1993) *Naturaliser l'intentionnalité. Essai de philosophie de la psychologie*. PUF.
- Parker L. E. (1994) *Heterogeneous Multi-Robot Cooperation*. PhD Thesis, MIT.
- Parunak H. V. D. (1987) "Manufacturing Experience with the Contract Net." In *Distributed Artificial Intelligence*, M. N. Huhns (Ed.), Pitman.
- Parunak H. V. D. (1990) "Distributed AI and Manufacturing Control: Some Issues and Insights." In *Decentralized AI*, Vol 1, Y. Demazeau et J.-P. Muller (Ed.), p. 81-104, North-Holland.

- Parunak H. V. D. (1993) *Industrial Applications of Multi-Agent Systems*. Rapport de recherche, Industrial Technology Institute.
- Pattison H. E., Corkill D. et Lesser V. (1987) "Instantiating Descriptions of Organizational Structures." In *Distributed Artificial Intelligence*, M. Huhns (Ed.), Pitman.
- Pavé A. (1994) *Modélisation en biologie et en écologie*. Lyon, Aléas.
- Pednault E. (1986) "Formulating Multiagent, Dynamic-World Problems in the Classical Planning Framework." In *Reasoning about Actions and Plans*, Timberline, M. Georgeff et A. Lansky (Ed.), p. 425, Morgan Kaufman.
- Piaget J. et Inhelder B. (1966) *La psychologie de l'enfant*. PUF.
- Pitrat J. (1990) *Metaconnaissances. Futur de l'intelligence artificielle*. Hermès.
- Popper K. (1963) *Conjectures and Refutations*. Londres, Routledge and Kegan Paul. Trad. fr. *Conjectures et réfutations*, Payot, 1985.
- Prigogine I. et Stengers I. (1979) *La nouvelle alliance*. Gallimard.
- Rao A. et Georgeff M. (1992) "Social Plans: Preliminary Report." In *Decentralized AI 3 - Proc. of MAAMAW'91*, E. Werner et C. Castelfranchi (Ed.), p. 127-146, Elsevier North Holland.
- Quinqueton J., Reitz P. et Sallantin J. (1991) "Les Schémas mentaux, un cadre conceptuel pour l'apprentissage à partir d'exemples." In *Actes du congrès sur la reconnaissance des formes et l'intelligence artificielle*, Lyon, J.-P. Laurent (Ed.).
- Raynal M. (1987) *Systèmes répartis et réseaux: concepts outils et algorithmes*. Eyrolles.
- Récanati F. (1981) *Les Énoncés performatifs*. Editions de Minuit.
- Regnier S. et Duhaut D. (1995) "Une approche multi-agent pour la résolution de problèmes robotiques." In *Actes des troisièmes journées francophones sur l'Intelligence artificielle Distribuée et les systèmes multi-agents (JFIADSMA '95)*, Chambéry.
- Reiter R. (1980) "A Logic for Default Reasoning". *AI Journal*. **13**, p. 81-131.
- Reynolds C. (1987) "Flocks, Herds and Schools: A Distributed Behavioral Model". *Computer Graphics*. **21**(4), p. 25-34.
- Rocher G. (1968) *Introduction à la sociologie générale. T1: l'action sociale*. Le Seuil.
- Rosenblatt K. J. et Payton D. W. (1989) "A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control." In *Proc. of the IEEE/INNS International Joint Conference on Neural Networks*.
- Rosenschein J. et Zlotkin G. (1994) *Rules of Encounters: Designing Conventions for Automated Negotiation among Computers*. MIT Press.
- Sabah G. (1989) *L'Intelligence artificielle et le langage. Tome 2: Compréhension*. Hermès.
- Sabah G. (1990) "CARMEL: A Computational Model of Natural Language Understanding using Parallel Implementation." In *Proc. of the Ninth European Conference on Artificial Intelligence (ECAI 90)*, Stockholm.
- Schoppers M. (1987) "Universal Plans for Reactive Robots in Unpredictable Environment." In *IJCAI-87*, Milan.
- Schwartz E. (1993) "A Coherent and Holistic Metamodel for the Functioning and

- Evolution of Viable Systems. An application to Human Societies." In *4th International Symposium on Systems Research Informatics and Cybernetics*, Baden-Baden.
- Searle J. (1991) *The Rediscovery of the Mind*. MIT Press.
- Searle J. et Vanderveken D. (1985) *Foundations of illocutionary logic*. Cambridge University Press.
- Searle J. R. (1969) *Speechs Acts*. Cambridge University Press.
- Searle J. R. (1979) *Expression and Meaning*. Cambridge University Press.
- Searle J. R. (1985) *L'intentionnalité: essai de philosophie des états mentaux*. Editions de Minuit.
- Shannon C. et Weaver W. (1948) *The Mathematical theory of Communication*. Urbana, University of Illinois Press.
- Shoham Y. (1988) *Reasoning About Change*. MIT Press.
- Shoham Y. (1993) "Agent Oriented Programming". *Artificial Intelligence*. **60**(1), p. 51-92.
- Sibertin-Blanc C. (1985) "High level Petri nets with Data Structure." In *6th European Workshop on Petri Nets and Applications*, Espoo, Finlande.
- Smith R. G. (1979) "A Framework for Distributed Problem Solving." In *Proceedings of IJCAI'79*.
- Smith R. G. (1980) "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver". *IEEE Trans. on Computers*. **29**(12), p. 1104-1113.
- Smolensky P. (1992) "IA connexioniste, IA symbolique et cerveau." In *Introduction aux sciences cognitives*, D. Andler (Ed.), Gallimard.
- Sombé L. (1988) "Raisonnement sur des informations incomplètes en intelligence artificielle." *Revue d'intelligence artificielle*. **2**(3-4), p. 9-210.
- Sowa J. (1984) *Conceptual Structures*. Addison-Wesley.
- Sperber D. et Wilson D. (1986) *Relevance: Communication and Cognition*. Blackwell. Trad. fr. *La Pertinence: communication et cognition*, 1989, Editions de Minuit.
- Steeb R., Camarata S., Hayes-Roth F., Thorndyke P. et Wesson R. (1988) "Architectures for Distributed Air Traffic Control." In *Readings in Distributed Artificial Intelligence*, A. Bond et L. Gasser (Ed.), Morgan Kaufman.
- Steels L. (1989) "Cooperation between Distributed Agents through Self-Organization." In *Decentralized A.I.*, Y. Demazeau et J.-P. Müller (Ed.), Elsevier/North-Holland.
- Steels L. (1994a) "The Artificial Life Roots of Artificial Intelligence". *Artificial Life*. **1**(1).
- Steels L. (1994b) "A Case Study in the Behavior-Oriented Design of Autonomous Agents." In *Proc. of the 94' Conference on the Simulation of Adaptive Behavior*, Brighton, MIT Press.
- Stoetzel J. (1978) *La Psychologie sociale*. Flammarion.
- Suchman L. (1987) *Plans and Situated Actions: the Problem of Human-Machine Communication*. Cambridge University Press.

- Sutton E. (1994) *Etat de l'art des algorithmes génétiques*. SGDN/STS/VST/5, Secrétariat général de la défense nationale.
- Sycara K. (1989) "Multiagent Compromise via Negotiation." In *Distributed Artificial Intelligence*, L. Gasser et M. Huhns (Ed.), Pitman.
- Tanenbaum A. (1991) *Architecture des ordinateurs*. InterEditions.
- Theraulaz G. (1991) *Morphogenèse et auto-organisation des comportements dans les colonies de guêpes Polistes dominulus (Christ)*. Thèse de troisième Cycle, Université d'Aix-Marseille I.
- Theraulaz G., Goss S., Gervet J. et Deneubourg J. L. (1991) "Task Differentiation in Polistes Wasp Colonies: a Model for Self-Organizing Groups of Robots." In *From Animals to Animats*, J. A. Meyer et S. Wilson (Ed.), p. 346-354, MIT Press.
- Thibault J.-P. (1993) *Interprétation d'environnement évolutif par une machine de perception multi-sensorielle*. Thèse de l'université Paul Sabatier, Toulouse.
- Thom R. (1988) *Esquisse d'une sémiophysique*. InterEditions.
- Tinbergen N. (1951) *The Study of Instinct*. Clarendon Press. Trad. fr. *L'Etude de l'instinct*, Payot, 1971.
- Todd P. M. et Miller G. F. (1991) "Exploring Adaptive Agency II: Simulating the Evolution of Associative Learning." In *From Animals to Animats: Proc. Of the First International Conference on Simulation of Adaptive Behavior (SAB'90)*, Paris, J.-A. Meyer et S. W. Wilson (Ed.), p. 306-315, MIT Press.
- Tokoro M. (1993) "The Society of Objects." In *Proc. of the OOPSLA '93 Conference*.
- Trognon A. et Brassac C. (1988) "Actes de langages et conversation". *Intellectica*. **6**(2), p. 211-232.
- Trognon A. et Brassac C. (1992) "L'Enchaînement conversationnel". *Cahiers de linguistique française*. **13**.
- Tyrrell T. (1993a) *Computational Mechanisms for Action Selection*. PhD Thesis, Edinburgh University.
- Tyrrell T. (1993b) "The Use of Hierarchies for Action Selection". *Adaptive Behavior*. **1**(4), p. 387-420.
- Uexküll J. v. (1956) *Mondes animaux et mondes humains*. Denoël.
- Vanderveken D. (1988) *Les actes de discours*. Pierre Mardaga.
- Vanderveken D. (1992) *Meaning and speech acts*. Cambridge University Press.
- Varela F. (1989) *Autonomie et Connaissance*. Le Seuil.
- Varela F., Thomson E. et Rosch E. (1993) *L'Inscription corporelle de l'esprit*. Le Seuil.
- Varela F. J. et Bourguin P. (Ed.) (1992) *Toward a Practice of Autonomous Systems*. MIT Press.
- Vauclair J. (1992) *L'Intelligence de l'animal*. Le Seuil.
- Vernadat F. et Azemat P. (1993) "Prototypage de systèmes d'agents communicants." In *Actes des premières journées francophones sur l'intelligence artificielle distribuée et les systèmes multi-agents*, Toulouse.
- Volterra V. (1926) "Variation and fluctuations of the number of individuals of animal species living together." In *Animal Ecology*, (Ed.), McGraw-Hill.
- von Martial F. (1992) *Coordinating Plans of Autonomous Agents*. Vol. 610, Springer-Verlag.

- Waldinger R. (1977) "Achieving Several Goals Simultaneously". *Machine Intelligence*. **8**, p. 94-136.
- Wavish P. (1992) "Exploiting Emergent Behaviour in Multi-Agent Systems." In *Decentralized A.I. 3*, E. Werner et Y. Demazeau (Ed.), North-Holland.
- Wavish P. R. et Connah D. M. (1990) *Representing Multi-Agent Worlds in ABLE*. Technical Note, TN2964, Philips Research Laboratories.
- Weihmayer R. et Brandau R. (1990) "A Distributed AI Architecture for Customer Network Control." In *GLOBECOM'90*, San Diego.
- Werner E. (1989) "Cooperating Agents: A Unified Theory of Communication and Social Structure." In *Distributed Artificial Intelligence, Vol II*, L. Gasser et M. Huhns (Ed.), Pitman.
- Werner E. et Demazeau Y. (Ed.) (1992) *Decentralized A.I. 3*. Amsterdam, Elsevier North-Holland.
- Werner G. M. (1994) "Using Second Order Neural Connections for Motivation of Behavioral Choices." In *From animals to animats 3: proc. of the third international conference on simulation of adaptive behavior.*, Brighton, D. Cliff, P. Husbands, J.-A. Meyer et S. W. Wilson (Ed.), p. 154-161, MIT Press.
- Werner G. M. et Dyer M. G. (1992) "Evolution of Communication in Artificial Organisms." In *Artificial Life II (1990)*, Santa Fe, C. Langton, C. Taylor, J. D. Farmer et S. Rasmussen (Ed.), p. 659-687, Addison Wesley.
- Wiener N. (1948) *Cybernetics*. Paris, Hermann.
- Wilson E. O. (1971) *The Insect Societies*. Belknap Press of Harvard University Press.
- Wilson S. W. (1991) "Knowledge Growth in an Artificial Animal." In *First International Conference on Genetic Algorithms and their Applications*, Pittsburgh, p. 16-23, Carnegie-Mellon University.
- Winograd T. et Flores F. (1986) *Understanding Computers and Cognition: a New Foundation for Design*. Ablex Publishing Corp.
- Wittgenstein (1951) *Le Cahier bleu et le cahier brun*. Gallimard.
- Wittig T. (Ed.) (1992) *ARCHON: An Architecture for Multi-Agent Systems*. Ellis Horwood.
- Wooldridge M. et Jennings N. (1994) "Towards a Theory of Cooperative Problem Solving." In *MAAMAW'94*, Odense, Danemark, Y. Demazeau, J.-P. Muller et J. Perram (Ed.).
- Yonezawa A. (Ed.) (1990) *ABCL: An Object-Oriented Concurrent System*. MIT Press.
- Yonezawa A. et Tokoro M. (Ed.) (1987) *Object-Oriented Concurrent Programming*. MIT Press.
- Zeghal K. (1993) "Un Modèle de Coordination d'Actions Reactive appliqué au trafic aérien." In *Journées francophones "IAD et SMA"*, Toulouse, AFCET.
- Zeghal K. et Ferber J. (1993) "CRAASH: A Coordinated Collision Avoidance System." In *European Simulation Multiconference*, Lyon.
- Zeghal K., Ferber J. et Erceau J. (1993) "Symmetrical, transitive and Recursive Force: A Representation of interactions and commitments." In *IJCAI Workshop on Coordinated Autonomous Robots*, Chambéry.

Zlotkin G. et Rosenschein J. S. (1992) "A Domain Theory for Task Oriented Negotiation." In *Proc. of the International Joint Conference on Artificial Intelligence*, Chambéry.

Index

- éco
 - résolution, 38
- émergence, 8, 9, 17–20, 41, 59, 87, 88, 118, 126, 189, 275, 343, 398, 434, 439
- état
 - d’automate, 171
- état du monde, 153
- états mentaux, 225–227, 338, 379
- évolution en groupe, 434–439
- ABCL, 145
- abduction, 310
- accointances, 14
 - réorganisation, 360, 378
- acheminement
 - direct, 313
 - par propagation, 314
 - par voie d’affiche, 315
- ACTALK, 145
- acte consommatoire, 286, 290
- actes de langage, 316, 320–340, 379
- action, 150
 - comme commande, 152, 191
 - comme déplacement, 182–187
 - comme déplacement physique, 152
 - comme modifications locales, 152, 187
 - comme processus, 151, 169
 - comme réponse à des influences, 151
 - comme transformation d’état, 152
 - comme transformation d’état, 164
 - comme transformation d’états, 151
 - de coordination, 446
 - problématique de l’, 54
 - rationnelle, 304
 - située, 195–201, 288, 433
- adéquation, 255, 259
- adaptation, 56
- administrateur, 361–388
- adoption de but, 74
- agent, 8
 - cognitif, 20, 23, 129, 388
 - définition, 13
 - et organisation, 19
 - fanatique, 305
 - hystérique, 206
 - intentionnel, 20
 - logiciel, 14, 16
 - pulsionnel, 21, 66
 - réactif, 20, 23, 129, 388, 433
 - rationnel, 20, 269
 - situé, 112
 - totipotent, 119, 127
 - tropique, 21, 66, 194–206
- Agent Oriented Programming, 308
- AGENT0, 308
- algorithmes génétiques, 139
- ALLIANCE, 398
- allocation
 - émergente, 345
 - centralisée, 345
 - centralisée, 346–350
 - de ressources, 422, 425
 - distribuée, 345
 - appel d’offre, 345, 378, 361–378
 - directe, 351–353
 - par délégation, 353–359
 - réseau d’accointances, 345, 350–361
 - optimisation, 348
 - réactive, 388
- ANA, 291
- analyse
 - fonctionnelle, 95, 100, 122, 124
 - grille, 106–110
 - structurale, 110, 122, 124

- animat, 31, 46, 140
 anneau à jeton, 368
 anti-collision, 52, 297, 439
 appel d'offre, 126, 385, 361–388, 422
 arbitrage, 68, 77, 82, 85
 architecture, 24, 111, 129
 - classifieurs, 139
 - connexioniste, 140, 289, 293
 - modulaire, 131
 - multi-agent, 144, 291
 - subsumption, 135, 181, 398
 - systèmes de production, 137
 - système dynamique, 142, 295
 - tâches compétitives, 136, 291, 293
 - tableaux noirs, 28, 37, 132, 315
- ARCHON, 53
 assertif, 321, 331, 334–337
 ATOME, 134
 augmentation
 - qualitative, 80
 - quantitative, 80, 83
 autocatalytique, 444, 447
 automate
 - à états finis, 171–174, 179, 207, 328, 380, 383, 417, 451
 - à registres, 174–176
 - cellulaire, 31, 188–191
 autonomie, 270
 autopoïèse, 58, 100, 270
 banc de poissons, 436
 BB1, 133
 BEINGS, 29
 boids, 434
 BRIC, 209–211, 332, 347, 383
 but, 303, 333
 catalyse, 81
 chaîne
 - comportement de, 447
 chaîne d'action, 276, 288
 champ
 - de potentiel, 184–186, 289, 296, 434, 437
 Chilpéric, 414, 418
 client, 344, 346, 388
 Clotaire, 153–155, 169, 180, 196, 198–204, 221, 255, 414, 417
 code (de communication), 310–311
 cogniton, 55, 227–231, 240, 262, 333, 338
 collaboration, 56, 84
 commande (cogniton), 228, 271
 communicant
 - SMA, 14
 communication, 83, 278
 - diffusion, 313, 394
 - incidente, 315, 316, 345
 - intentionnelle, 315, 316
 - point à point, 312
 - schéma général, 310
 compétece, 231, 239, 266–267, 333, 343, 346, 350
 compatibilité des buts, 66
Comport, 194, 207
 comportement, 24, 26, 111
 - appétitif, 286, 290
 - réflexe, 20
 - téléonomique, 20
 conception
 - fonctionnelle, 111
 - horizontale, 114
 - verticale, 114
 - objet, 111
 conceptualisation, 235
 CONDOR, 36
 connaissance, 233–241
 connexionisme, 240
 contournement (comportement de), 438
 contrôle
 - problème du, 271
 contrat, 361–367, 371–374, 377–379
 - cognitif, 319
 conversation, 326, 328
 coopération, 73–89, 343
 - avantages, 88
 - intentionnelle, 46, 78
 - méthodes, 82
 - réactive, 46, 78
 - système des activités, 86
 coordination
 - caractéristiques

- généralisation, 410
- organisationnelles, 408
- qualité, 409
- réalisation, 410
- temporelles, 408
- centralisée
 - pour plans partiels, 423–428
- d'actions, 56, 85
- définition, 402
- distribuée
 - pour plans partiels, 428–433
- planification, 411, 416–433
- réactive, 411, 433–447
- réglementation, 412
- synchronisation, 411, 413
- tâches de, 85
- couplage, 116
- CRAASH, 213, 439
- croissance, 229, 233, 246, 255, 303, 333
- CYC, 29, 238
- déclaratif, 321
- décomposition des problèmes, 344
- délégation, 353, 356
- délégation de tâches, 266
- délibération, 193
- dépendance des actions, 406
- Decision*, 206
- demande, 229
 - fonctionnelle, 276
- diffusion
 - par vague, 185, 191, 203
- dimension
 - d'analyse, 99
 - environnementale, 99, 264
 - personnelle, 99
 - physique, 99
 - relationnelle, 99, 265
 - sociale, 99
- directif, 321
- distributeur de bonbons, 171, 175
- DVMT, 30, 37, 132
- éco-agent, 449
- éco-problème, 451
- effet de masse, 81
- EMF, 293
- encombrement, 72
- engagement, 131, 230, 279–285, 320, 333, 371, 378–388, 432
 - rapide, 373
 - tardif, 373
- environnement, 13, 212, 434
 - centralisé, 213, 215, 217
 - distribué, 213
 - spécialisé, 216
- ETHER, 28, 315
- Exec*, 166, 193, 194, 406
- exercitif, 322, 325, 329, 331, 333–334
- expressif, 321, 331
- Flavor Paint Shop, 376
- Flavors Paint Shop, 36
- fonction
 - conative, 96, 102, 318, 385, 386
 - décisionnelle, 96
 - de transition, 173
 - exécutrice, 96
 - expressive, 318
 - interactionnelle, 96, 103
 - métalinguistique, 318
 - métaconceptuelle, 318
 - paralinguistique, 318
 - motivationnelle, 96
 - organisationnelle, 96, 102, 343
 - perceptive, 96
 - phatique, 318, 325
 - poétique, 318
 - productive, 96, 104, 124
 - référentielle, 318
 - représentationnelle, 95, 100
 - végétative, 97, 105
- fournisseur, 344, 346, 378, 388
- fuite
 - comportement de, 456
- GUARDIAN, 134
- HEARSAY II, 28, 133
- homéostasie, 394
- hypothèse, 229
- ICHTYUS, 436

- IDEAL, 37
- illocutoire
 acte, 322, 324–326, 332
 force, 323, 327
- indice, 310
 conflictuel, 82
 dans les actions situées, 196
 de coopération, 75
 de coordination, 442
 de performance, 79
 de survie, 78
- inférence, 241
- influences, 164–168, 215, 260
- information (cogniton), 228, 260
- intelligence artificielle distribuée, 28
- intention, 229, 267, 270, 299–308, 312, 315, 333, 379
 sociale, 306
 théorie de Cohen et Levesque, 308
 théorie de Cohen et Levesque, 302–333
- intentionnalité, 225–226, 316–317
- interaction, 63
- Internet, 52
- interprétation, 250
 logique, 243–244
- interrogatif, 322, 331, 334
- jeton, 176
- justification (des croyances), 258
- kénétique, 8, 12, 31, 52, 57, 60, 63, 116, 128, 237, 240
 conception, 52
- KBS-SHIP, 36
- KQML, 33, 338–340
- langage
 communication, 26, 310
 d'acteurs, 26, 29, 61, 145
 implémentation, 25
 représentation des connaissances, 26
 spécification et formalisation, 27
- leader, 381, 385–388
- locutoire (acte), 322
- logique
 épistémique, 248
 des défauts, 257
 des prédicats, 245, 248, 256
 modale, 249–253
 opérationnelle, 254
- lois de l'univers, 14, 26, 163, 165, 166, 168, 260, 262, 265
- médiateur, 313, 344–350
- méthode, 230
- MACE, 30, 146, 213
- MAGES, 146, 213
- MANTA, 41, 271, 293, 391–398
- marquage
 dans l'environnement, 202, 218, 434, 440, 444–446
 dans les réseaux de Petri, 177, 419, 430
- marque
 de réseaux de Petri, 176
- Mem*, 206
- mentalais, 241
- MERING IV, 145
- meute (comportement de), 434–436
- micro-théorie, 238, 245
- modèle
 des actions, 149
 factuel, 264
 notion de, 147
 prévisionnel, 264, 265
 psychologique, 264
- mondes synthétiques, 43
- motivation, 271–273
 contractuelle, 274
 environnementale, 273
 personnelle, 273
 contractuelle, 274
 hédoniste, 274, 277
 relationnelle, 273, 278
 sociale (déontique), 273, 276
- négociation, 7, 19, 27, 29, 30, 56, 57, 68, 77, 82, 85, 116, 332, 420
- neurones, 140
- norme, 228

- objets et agents, 61
- offrant, 361–388
- omniscience logique, 253
- organisationnel
 - système, 85, 230
- organisations, 7, 19, 57, 63, 64, 78, 81, 88, 92–110, 116, 123–129, 266, 280, 395, 408
 - émergentes, 118, 443
 - évolutives, 117, 127
 - analyse des, 94, 122
 - artificielles, 8, 18, 57, 114, 130
 - collectives, 93
 - concrètes, 92, 268
 - coopératives, 48
 - définition, 91
 - engagement des, 281
 - fixes, 116, 125, 345
 - hiérarchiques, 117, 289, 392
 - individuelles, 93, 129
 - niveaux, 17, 93, 104, 113
 - prédéfinies, 118, 345, 392
 - réactives, 24
 - sociologie des, 20
 - variables, 117, 126, 345
- PALADIN, 146
- PANDORA II, 213
- passage à l'acte
 - intentionnel, 285, 299, 308
 - réactif, 285–298
- PENGI, 197, 204
- Percept, 194
- Percept*, 193, 206
- percept, 193, 196, 206, 208, 228, 260, 275, 385
- PerceptIndice*, 196
- perception, 231–233, 260, 333
 - active, 233
 - passive, 233
- perception/délibération/action, 193
- performance, 122
- performatif, 322, 330–340
- perlocutoire (acte), 323, 324
- PGP, 429
- place
 - de réseaux de Petri, 176
- plan, 267, 302
 - définition, 155
 - exécution du, 160
 - fusion de, 424, 426
 - linéaire, 158
 - partiel, 159, 160, 421, 425, 429, 430
 - total, 159
- planification, 22, 131, 155–161, 408, 416, 420, 462
 - centralisée, 420
 - pour agents multiples, 420–423
 - critique, 184
 - de mouvements, 182
 - de trajectoires, 182, 205
 - distribuée, 420, 428
 - linéaire, 158
 - multi-agent, 416–430, 433
- potentiel d'action spécifique, 276
- poursuite (problème de), 44, 379
- prévision, 259
- pragmatique, 316
- processus, 169–171, 417
 - composition des, 418
- proies et prédateurs (problème de), 44, 379
- promissif, 321, 331, 338
- protocole, 327, 391
 - allocation de tâches, 313
- pulsion, 229, 274, 275
- référent, 242
- réification, 111
- réseau
 - contractuel, 30, 85, 345, 361–379, 381, 382
 - d'accointances, 345, 350–361, 377, 378
 - de Petri, 26, 33, 176, 329, 415, 419
 - colorés, 177
 - de transitions augmenté, 174–176
- résolution
 - de conflits, 8, 36, 56, 68, 77, 82, 428
 - de contraintes, 463
 - de problèmes, 28

- de problèmes, 7, 34, 448–465
- de problèmes distribués, 37
- distribuée de problèmes, 35
- par contraintes, 38
- par coordination, 38
- rétroaction positive, 394
- révision (des croyances), 247, 255
- rôle, 95, 268, 344
- règles
 - d'inférences, 245
 - déontiques, 277
 - de franchissement, 176
- raisonnement, 235
- React*, 166, 168, 194, 406
- REACTALK, 146
- recuit simulé, 463
- redondance, 119, 121
 - degré, 120, 127
- Reflexe*, 194, 206
- regroupement, 83
- relation
 - de satisfaction, 449
 - entre actions, 406
 - négative, 406, 425
 - organisationnelle, 115
 - positive, 406, 424
- requête (cogniton), 228
- ressources
 - accès aux, 69
 - limitation des, 70, 81, 283, 403, 433
 - relations aux, 67
 - synchronisation, 414–415
- robotique distribuée, 50
- robots
 - déménageurs, 49
 - explorateurs, 47, 76, 80, 83, 89, 123–127, 208, 264, 390, 443
 - transporteurs, 63, 80
- sélection d'action, 285
- sélection d'actions, 271, 298
- sémantique
 - des mondes possibles, 249–254
 - logique, 243
 - situationnelle, 232
- SAM, 379–388
- satisfaction, 449
 - comportement de, 456
 - critère de, 462
- section critique, 414
- signaux, 184, 233, 309, 314, 389, 391, 394
- signe, 310
- SIMDELTA, 41
- SIMPOP, 42
- simulation multi-agent, 38–43
- situé
 - action, 195–201
 - agent, 15, 16, 112
 - objet, 14
 - SMA, 14
- situation
 - analyse des, 72
 - collaboration coordonnée, 70, 403
 - collaboration simple, 70
 - compétition collective, 71
 - compétition individuelle, 71
 - conflit collectif, 71
 - conflit individuel, 71
 - d'antagonisme, 66
 - d'indépendance, 70, 403
 - d'indifférence, 66
 - d'interaction, 65, 69
 - de collaboration, 76
 - de coopération, 65, 66
 - de résolution de conflits, 77
 - engorgement, 70
 - macro/micro, 72
- SMA
 - communicant, 14, 212
 - définition, 14
 - situé, 14, 194, 218
 - tropique, 21, 194
- sociogénèse, 396
- spécialisation, 84, 119, 121, 293, 315, 391, 396
 - degré, 119, 127
- stimuli, 193, 231, 232, 271, 274, 275, 285, 294, 310, 314, 345, 392
- STRIPS, 157, 160–163, 165, 166, 205, 215, 416, 453, 462

- postulats, 162
- StripsSTRIPS, 153
- structure
 - égalitaire, 117
 - compétitive, 118
 - hiérarchique, 117, 125
 - organisationnelle, 92, 122, 126, 268
- succès et satisfaction (d'un acte de langage), 323–325, 330
- survie, 79
- symbole, 242
- symbolico-cognitivism, 240, 241
- synchronisation, 216, 413–415, 423, 432, 434
 - de ressources, 415
 - de mouvements, 413
 - de plans, 421
 - de ressources, 414–425
- systémique, 57
- système
 - conatif, 229, 230, 268–385
 - décisionnel, 273
 - interactionnel, 228, 231–233
 - motivationnel, 272, 352, 356, 398
 - organisationnel, 96, 228, 230, 352, 356, 358, 385, 401
 - ouvert, 29, 238
 - perceptif, 231
 - prévisionnel, 260
 - représentationnel, 229, 233–268
 - végétatif, 97, 229
- système ouvert, 447
- tâche (cogniton), 230, 276
- taquin, 458
- taxie, 286
- tendance, 13, 229, 271, 274
- TMS, 258–259
- tours de Hanoï, 456
- transition
 - d'automate, 171
 - de réseaux de Petri, 176
- tri collectif, 84
- viabilité, 32, 66, 270
- vie artificielle, 28, 31
- YAMS, 376